



UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA CIVIL Y MECÁNICA

CARRERA DE INGENIERÍA CIVIL

**TRABAJO EXPERIMENTAL PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO CIVIL**

TEMA:

**“DESARROLLO DE UN SOFTWARE PARA EL CÁLCULO DE CANALES
ABIERTOS DE FLUJO UNIFORME.”**

AUTOR: César Sebastián Naranjo Bustos

TUTOR: Ing. M. Sc. Rodrigo Acosta

Ambato – Ecuador

2016

CERTIFICACIÓN DEL TUTOR

Yo, Ing. M. Sc. Rodrigo Acosta, certifico que el presente trabajo bajo el tema: **DESARROLLO DE UN SOFTWARE PARA EL CÁLCULO DE CANALES ABIERTOS DE FLUJO UNIFORME**, es de autoría del Sr. César Sebastián Naranjo Bustos, el mismo que ha sido realizado bajo mi supervisión y tutoría.

Es todo cuanto puedo certificar en honor a la verdad.

Ambato, 17 de Noviembre del 2016

Ing. M. Sc. Rodrigo Acosta.

AUTORÍA

Yo, César Sebastián Naranjo Bustos con C.I: 1805066527-7, egresado de la Facultad de Ingeniería Civil y Mecánica de la Universidad Técnica de Ambato, certifico por medio de la presente que el trabajo con el tema: **DESARROLLO DE UN SOFTWARE PARA EL CÁLCULO DE CANALES ABIERTOS DE FLUJO UNIFORME**, es de mi completa autoría.

Ambato, 17 de Noviembre del 2016

César Sebastián Naranjo Bustos.

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga de este Trabajo Experimental o parte de él, un documento disponible para su lectura, consulta y procesos de investigación, según las normas de la Institución.

Cedo los Derechos en línea patrimoniales de mi Trabajo Experimental con fines de difusión pública, además apruebo la reproducción de éste Trabajo Experimental dentro de las regulaciones de la Universidad, siempre y cuando ésta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor.

Ambato, 17 de Noviembre del 2016

Autor

César Sebastián Naranjo Bustos.

APROBACIÓN DEL TRIBUNAL DE GRADO

Los miembros del tribunal examinador aprueban el informe de investigación, sobre el tema: “DESARROLLO DE UN SOFTWARE PARA EL CÁLCULO DE CANALES ABIERTOS DE FLUJO UNIFORME”, del egresado César Sebastián Naranjo Bustos, de la Facultad de Ingeniería Civil y Mecánica.

Ambato, 17 de Noviembre del 2016

Para constancia firman.

Ing. Mg. Galo Núñez

Ing. Mg. Fabián Morales.

DEDICATORIA

De manera especial dedico este trabajo a DIOS, por haberme dado la fuerza y voluntad de luchar para culminar una meta más en mi vida.

A mis padres Javier y Rocío quienes han hecho todo cuanto ha estado a su alcance para que yo pudiera alcanzar cada una de las metas que me he propuesto, alentándome y apoyándome en cada paso que he dado aún cuando se veía difícil. A ellos quienes me han brindado siempre su apoyo, amor, cariño y paciencia.

A mi hermano Mathías a quien amo y que ha sido mi inspiración para poder ser un ejemplo en su vida, quien comprendió varias veces mi falta a su lado para así cumplir con una meta más.

A mi familia y amigos quienes han estado junto a mí acompañándome y apoyándome en el transcurso de esta etapa de mi vida.

AGRADECIMIENTO

Agradezco principalmente a Dios por ser el guía espiritual en mi vida y darme salud y vida para superarme a diario.

A mi madre celestial la virgen del Perpetuo Socorro y al Divino niño Jesús por ser los protectores de mi vida.

A mis padres que me dieron la vida y su ejemplo de lucha y constancia para cumplir los objetivos que me he propuesto en mi vida.

A mi hermano por ser quien inspira cada uno de mis días y me da una razón más para superarme.

A toda mi familia y amigos que estuvieron siempre pendientes y brindándome su apoyo en esta etapa de mi vida.

A la Ing. Maritza Ureña y a los Ingenieros Fabián Morales y Juan Garcés por compartir sus conocimientos encaminándome en la realización de este trabajo.

Al Ing. Rodrigo Acosta, mi tutor académico, por su profesionalismo, compromiso y por guiarme con sus conocimientos para así culminar satisfactoriamente con este trabajo.

A todos mis maestros mis más profundos agradecimientos por haber impartido todos sus conocimientos para poder crecer profesionalmente.

Mi especial agradecimiento a mi querida Universidad Técnica de Ambato que me brindó el espacio para poder alcanzar un título profesional en mi vida.

ÍNDICE GENERAL DE CONTENIDOS

A. PÁGINAS PRELIMINARES

UNIVERSIDAD TÉCNICA DE AMBATO.....	I
CERTIFICACIÓN DEL TUTOR.....	II
AUTORÍA.....	III
DERECHOS DE AUTOR.....	IV
APROBACIÓN DEL TRIBUNAL DE GRADO.....	V
DEDICATORIA.....	VI
AGRADECIMIENTO.....	VII
ÍNDICE GENERAL DE CONTENIDOS.....	VIII
ÍNDICE DE TABLAS.....	XIII
ÍNDICE DE FIGURAS.....	XIV
RESUMEN EJECUTIVO.....	XV
SUMMARY.....	XVI

B. TEXTO

CAPÍTULO I.....	1
ANTECEDENTES.....	1
1.1. TEMA DEL TRABAJO EXPERIMENTAL.....	1
1.2. ANTECEDENTES.....	1
1.3. JUSTIFICACIÓN.....	3
1.4. OBJETIVOS.....	4
1.4.1. General.....	4
1.4.2. Específicos.....	4
CAPÍTULO II.....	5
FUNDAMENTACIÓN.....	5
2.1. FUNDAMENTACIÓN TEÓRICA.....	5
2.1.1. Canales abiertos.....	5
2.1.2. Geometría de los canales.....	5
2.1.2.1. Elementos geométricos de una sección de canal.....	8
2.1.2.1.1. Profundidad del flujo.....	8
2.1.2.1.2. Nivel.....	9
2.1.2.1.3. Ancho superficial.....	9

2.1.2.1.4.	Área mojada.	9
2.1.2.1.5.	Perímetro mojado	9
2.1.2.1.6.	Radio hidráulico	10
2.1.2.1.7.	Profundidad hidráulica.....	10
2.1.2.1.8.	Factor de sección para el cálculo del flujo crítico	10
2.1.2.1.9.	Factor de sección para el cálculo del flujo uniforme.....	10
2.1.3.	Estado de flujo en canales abiertos.....	11
2.1.3.1.	Efecto de la gravedad.....	11
2.1.3.1.1.	Número de Froude	11
2.1.3.1.1.1.	Flujo crítico	12
2.1.3.1.1.2.	Flujo subcrítico.....	12
2.1.3.1.1.3.	Flujo supercrítico.....	12
2.1.4.	Flujo uniforme en canales abiertos.....	12
2.1.5.	Pendiente del canal	13
2.1.5.1.	Formas para expresar la pendiente	14
2.1.6.	Ecuación de Manning	15
2.1.6.1.	Valores del coeficiente de rugosidad de Manning	16
2.1.6.1.1.	Factores que afectan el coeficiente de rugosidad de Manning.....	16
2.1.6.1.1.1.	Resumen de los valores de n que afectan el coeficiente de rugosidad de Manning.....	20
2.1.7.	Cálculo del flujo uniforme	21
2.1.7.1.	El factor de sección para el cálculo del flujo uniforme	22
2.1.7.2.	Cálculo de la pendiente normal y crítica.....	24
2.1.7.3.	Problemas de cálculo de flujo uniforme	24
2.1.8.	Diseño de canales con flujo uniforme.....	26
2.1.9.	Sección hidráulica óptima	26
2.1.9.1.	Notas de la sección hidráulica óptima	28
2.1.10.	Energía.	28
2.1.10.1.	Energía específica.....	31
2.1.10.2.	Curvas de energía específica	31
2.1.10.2.1.	Características de la curva de energía.....	33
2.1.11.	Flujo crítico	34
2.1.11.1.	Condiciones del flujo crítico	36

2.1.11.2.	Determinación del flujo crítico.....	37
2.1.11.3.	El factor de sección para el cálculo del flujo crítico.....	39
2.2.	HIPÓTESIS.....	39
2.3.	SEÑALAMIENTO DE LAS VARIABLES DE LA HIPÓTESIS.....	40
2.3.1.	Variable independiente	40
2.3.2.	Variable dependiente	40
CAPÍTULO III		41
METODOLOGÍA.....		41
3.1.	NIVEL O TIPO DE INVESTIGACIÓN	41
3.2.	POBLACIÓN Y MUESTRA.....	41
3.2.1.	Población.....	41
3.2.2.	Muestra	41
3.3.	OPERACIONALIZACIÓN DE VARIABLES.....	42
3.3.1.	Variable independiente	42
3.3.2.	Variable dependiente	43
3.4.	PLAN DE RECOLECCIÓN DE INFORMACIÓN	44
3.5.	PLAN DE PROCESAMIENTO Y ANÁLISIS.....	44
CAPÍTULO IV		45
ANÁLISIS E INTERPRETACIÓN DE RESULTADOS.....		45
4.1.	RECOLECCIÓN DE DATOS	45
4.1.1.	EJEMPLO PARA EL CÁLCULO DEL TIRANTE NORMAL DE UNA SECCIÓN TRAPEZOIDAL	45
4.1.1.1.	MÉTODO MANUAL.....	46
4.1.1.2.	MÉTODO CON UN SOFTWARE EXISTENTE.....	51
4.1.1.3.	MÉTODO CON EL SOFTWARE DESARROLLADO	51
4.1.2.	EJEMPLO PARA EL CÁLCULO DEL TIRANTE CRÍTICO DE UNA SECCIÓN TRAPEZOIDAL.	52
4.1.2.1.	MÉTODO MANUAL.....	52
4.1.2.2.	MÉTODO CON UN SOFTWARE EXISTENTE.....	56
4.1.2.3.	MÉTODO CON EL SOFTWARE DESARROLLADO	57
4.1.3.	EJEMPLO PARA EL CÁLCULO DEL CAUDAL DE UNA SECCIÓN CIRCULAR	57
4.1.3.1.	MÉTODO MANUAL.....	58

4.1.3.2.	MÉTODO CON UN SOFTWARE EXISTENTE.....	60
4.1.3.3.	MÉTODO CON EL SOFTWARE DESARROLLADO	61
4.1.4.	EJEMPLO PARA EL CÁLCULO DE LOS ELEMENTOS GEOMÉTRICOS DE UN CANAL TRIANGULAR CON FONDO REDONDEADO.....	61
4.1.4.1.	MÉTODO MANUAL.....	62
4.1.4.2.	MÉTODO CON EL SOFTWARE DESARROLLADO	64
4.1.5.	EJEMPLO PARA EL CÁLCULO DE LA SECCIÓN HIDRÁULICA ÓPTIMA.....	64
4.1.5.1.	MÉTODO MANUAL.....	65
4.1.5.2.	MÉTODO CON UN SOFTWARE EXISTENTE.....	68
4.1.5.3.	MÉTODO CON EL SOFTWARE DESARROLLADO	68
4.2.	ANÁLISIS DE RESULTADOS.....	69
4.3.	VERIFICACIÓN DE HIPÓTESIS.....	70
CAPÍTULO V		71
CONCLUSIONES Y RECOMENDACIONES		71
5.1.	CONCLUSIONES.....	71
5.2.	RECOMENDACIONES	72
C. MATERIAL DE REFERENCIA		
1.	BIBLIOGRAFÍA.....	73
2.	ANEXOS	74
2.1.	CODIFICACIÓN DEL SOFTWARE DESARROLLADO	74
2.1.1.	TIRANTE NORMAL.....	74
2.1.1.1.	SECCIÓN RECTANGULAR.....	74
2.1.1.2.	SECCIÓN TRAPEZOIDAL	81
2.1.1.3.	SECCIÓN TRIANGULAR.....	88
2.1.1.4.	SECCIÓN CIRCULAR	95
2.1.1.5.	SECCIÓN PARABÓLICA	102
2.1.1.6.	SECCIÓN PARABÓLICA CON “T” CONOCIDO	102
2.1.1.6.1.	SECCIÓN PARABÓLICA CON “T” NO CONOCIDO.....	109
2.1.1.6.2.	SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS.....	117
2.1.1.7.	SECCIÓN TRIANGULAR CON FONDO REDONDEADO.....	125

2.1.2.	TIRANTE CRÍTICO	133
2.1.2.1.	SECCIÓN RECTANGULAR	133
2.1.2.2.	SECCIÓN TRAPEZOIDAL	139
2.1.2.3.	SECCIÓN TRIANGULAR.....	146
2.1.2.4.	SECCIÓN CIRCULAR	152
2.1.2.5.	SECCIÓN PARABÓLICA	159
2.1.2.5.1.	SECCIÓN PARABÓLICA CON “T” CONOCIDO	159
2.1.2.5.2.	SECCIÓN PARABÓLICA CON “T” NO CONOCIDO.....	165
2.1.2.6.	SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS.....	173
2.1.2.7.	SECCIÓN TRIANGULAR CON FONDO REDONDEADO.....	180
2.1.3.	CAUDALES.....	187
2.1.3.1.	SECCIÓN RECTANGULAR	187
2.1.3.2.	SECCIÓN TRAPEZOIDAL	193
2.1.3.3.	SECCIÓN TRIANGULAR.....	200
2.1.3.4.	SECCIÓN CIRCULAR	206
2.1.3.5.	SECCIÓN PARABÓLICA	213
2.1.3.6.	SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS.....	220
2.1.3.7.	SECCIÓN TRIANGULAR CON FONDO REDONDEADO.....	227
2.1.4.	ELEMENTOS GEOMÉTRICOS.....	234
2.1.4.1.	SECCIÓN RECTANGULAR	234
2.1.4.2.	SECCIÓN TRAPEZOIDAL	237
2.1.4.3.	SECCIÓN TRIANGULAR.....	241
2.1.4.4.	SECCIÓN CIRCULAR	245
2.1.4.5.	SECCIÓN PARABÓLICA	248
2.1.4.6.	SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS.....	252
2.1.4.7.	SECCIÓN TRIANGULAR CON FONDO REDONDEADO.....	256
2.1.5.	SECCIÓN HIDRÁULICA ÓPTIMA	260
2.1.5.1.	SECCIÓN RECTANGULAR	260
2.1.5.2.	SECCIÓN TRAPEZOIDAL	267
2.1.5.2.1.	SECCIÓN TRAPEZODIAL CON “z” CONOCIDO.....	267

2.1.5.2.2.	SECCIÓN TRAPEZODIAL CON “z” NO CONOCIDO	274
2.1.5.3.	SECCIÓN TRIANGULAR.....	281
2.1.5.4.	SECCIÓN SEMICIRCULAR	288
2.1.6.	PANTALLA DE INCIO	295
2.1.7.	SELECCIÓN	297
2.1.8.	SELECCIÓN 1	298
2.1.9.	SELECCIÓN 2	299
2.1.10.	ESPERA.....	299
2.1.11.	ACERCA DE SN CANALES	300
2.2.	MANUAL DE USUARIO	301

ÍNDICE DE TABLAS

Tabla 1:	Elementos geométricos de secciones de canal.....	7
Tabla 2:	Ecuaciones para el cálculo del flujo uniforme en unidades métricas e inglesas.....	22
Tabla 3:	Operacionalización de la variable independiente.	42
Tabla 4:	Operacionalización de la variable dependiente.	43
Tabla 5:	Plan de recolección de información.	44
Tabla 6:	Comparación de resultados en el cálculo del tirante normal de una sección trapezoidal.	69
Tabla 7:	Comparación de resultados en el cálculo del tirante crítico de una sección trapezoidal.	69
Tabla 8:	Comparación de resultados en el cálculo del caudal de una sección circular.69	
Tabla 9:	Comparación de resultados en el cálculo los elementos geométricos de una sección triangular con fondo redondeado.	70
Tabla 10:	Comparación de resultados en el cálculo de la sección hidráulica óptima de una sección rectangular.	70

ÍNDICE DE FIGURAS

Figura 1: Flujo en conductos.....	5
Figura 2: Relación entre la profundidad (y) y la profundidad (d).....	8
Figura 3: Flujo uniforme	13
Figura 4: Pendiente de un canal	14
Figura 5: Sección hidráulica óptima	27
Figura 6: Energía total en una sección de un canal	29
Figura 7: Curva de energía específica	32
Figura 8: Profundidad crítica y profundidades alternas.....	33
Figura 9: Ilustración gráfica del ejemplo para el cálculo del tirante normal de una sección trapezoidal.....	46
Figura 10: Ilustración gráfica del ejemplo para el cálculo del tirante crítico de una sección trapezoidal.....	52
Figura 11: Ilustración gráfica del ejemplo para el cálculo del caudal de una sección circular.....	58
Figura 12: Ilustración gráfica del ejemplo para el cálculo de los elementos geométricos de una sección circular.	62
Figura 13: Ilustración gráfica del ejemplo para el cálculo de los elementos geométricos de una sección circular.	65

RESUMEN EJECUTIVO

TEMA: “DESARROLLO DE UN SOFTWARE PARA EL CÁLCULO DE CANALES ABIERTOS DE FLUJO UNIFORME.”

AUTOR: César Sebastián Naranjo Bustos

TUTOR: Ing. M. Sc. Rodrigo Acosta

FECHA: Noviembre 2016

En el presente proyecto de investigación se realizó la codificación de un software para el cálculo de canales abiertos de flujo uniforme, el cual permite obtener de forma rápida y segura el tirante normal, tirante crítico, elementos geométricos, caudal, sección hidráulica óptima, número de Froude, tipo de flujo y energía específica de las secciones planteadas por Ven Te Chow. Las secciones son: rectangular, trapezoidal, triangular, circular, parabólica, rectangular con esquinas redondeadas y triangular con fondo redondeado.

Para comprobar la validez del programa desarrollado los resultados fueron comparados con los obtenidos mediante un software existente el cual fue HCANALES y los obtenidos de manera manual; obteniendo un margen de error menor al 1% con lo que se verificó que los valores arrojados por el software son confiables.

SUMMARY

TOPIC: “DEVELOPMENT OF A SOFTWARE TO CALCULATE OPEN CHANNELS WITH UNIFORM FLOW.”

AUTHOR: César Sebastián Naranjo Bustos

TUTOR: Ing. M. Sc. Rodrigo Acosta

DATE: November 2016

The present research contains the codification of a software to calculate open channels with uniform flow, which allows to obtain of fast and safe way the normal depth, critical depth, geometric elements, flow, optimal hydraulic section, Froude number, flow type and specific energy of the sections given by Ven Te Chow. The sections are: rectangular, trapezoidal, triangular, circular, parabolic, rectangular with rounded corners and triangular with rounded bottom.

To test the validity of the developed program the results were compared with those obtained through an existing software which was HCANALES and those obtained manually; with what was obtained a margin of error less than 1% with which was verified that the values given by the software are trustworthy.

CAPÍTULO I

ANTECEDENTES

1.1. TEMA DEL TRABAJO EXPERIMENTAL

DESARROLLO DE UN SOFTWARE PARA EL CÁLCULO DE CANALES ABIERTOS DE FLUJO UNIFORME.

1.2. ANTECEDENTES

En sus inicios la Ingeniería no contaba con los recursos tecnológicos con los que cuenta actualmente, ya que anteriormente para el desarrollo de un cálculo matemático y diseño de alguna estructura se involucraba el gasto de varias horas para su ejecución, por lo que en aquellos días era común el uso de tablas y nomogramas en los cuales se resumía información relevante que ayudaba a realizar los cálculos [1].

El proceso de cálculo se efectuaba generalmente con reglas de cálculo y calculadoras de bolsillo, donde el uso continuo y prolongado de estos elementos probablemente conllevaba a errores que el diseñador obviaba no por omisión propia sino muy seguramente por cansancio o fatiga al efectuar procesos tan repetitivos y desgastantes [1].

En la actualidad, gracias a los avances de la Informática, el software se encuentra en diversos campos de la actividad humana, por lo que resulta sumamente necesario que reúna ciertos criterios de calidad para satisfacer en gran medida las necesidades de los usuarios. Debido a la expansión del uso de las computadoras a un gran número de la sociedad ha hecho que la usabilidad del software adquiera una gran importancia en el desarrollo de sistemas informáticos [2].

El rol de la programación en los últimos sesenta años ha sido crucial para comprender la evolución que ha tenido la sistematización de tareas y el manejo de la información que hoy en día damos como un hecho. En efecto, la misma tiene como principal función el hecho de conseguir que innumerables trabajos que antes ejercíamos de forma manual y con un alto costo sean ejecutados por un ordenador con un ahorro significativo de tiempo [3].

En 1992 en la Escuela de Ingeniería Agrícola del Instituto Tecnológico de Costa Rica, se desarrolló el software **HCANALES** por el Ing. Máximo Villón Béjar. El sistema permite resolver los problemas más frecuentes que se presentan en el diseño de canales y estructuras hidráulicas, las cuales son: calcular el tirante normal, tirante crítico, resalto hidráulico, la curva de remanso, caudales y cálculos variados, como pendiente, ancho de solera, coeficiente de rugosidad, diámetros de tuberías para las secciones transversales artificiales de uso común: triangular, rectangular, trapezoidal, parabólica y circular [4].

En febrero de 2007 en Bogotá – Colombia fue presentado en la Facultad de Ingeniería Civil de la Universidad de la Salle, el trabajo de grado **SOFTWARE PARA EL DISEÑO DE CANALES ABIERTOS** por Luis Acosta, Jairo Barragán y Andrés Reyes. Donde se buscó crear un software que permite el diseño de canales y la posibilidad de ser utilizado como herramienta pedagógica y didáctica para el aprendizaje de la hidráulica de canales, y así mejorar el nivel académico del estudiante, optimizando el tiempo del ingeniero y permitiendo al docente manejar una herramienta para el desarrollo de su cátedra. En el trabajo se desarrollan los temas básicos de la hidráulica de canales, con un enfoque teórico y práctico. En el enfoque teórico se hace mención de los temas que deben ser conocidos por cualquier ingeniero civil y en el enfoque práctico se desarrolló el software para el diseño y el aprendizaje de la hidráulica de canales abiertos, complementado por un escrito donde se hace énfasis en las principales metodologías de diseño de canales abiertos, con su respectiva descripción y análisis de cada método [1].

1.3. JUSTIFICACIÓN

En la época en la que nos encontramos es indispensable el uso de herramientas tecnológicas con el propósito de disminuir el tiempo involucrado por los diseñadores de canales al concebirlos y diseñarlos dentro de un marco referencial limitado por aspectos teóricos aceptados. Además, que en la etapa de diseño de canales abiertos existen procesos repetitivos e iterativos donde debido al factor humano se pueden producir errores, por lo que estos pueden ser optimizados a través del uso de un software donde se deje de lado los cálculos tediosos y se pueda profundizar más en los conceptos [1].

En el mundo el software que está marcando pauta para el diseño de canales es HCANALES, ya que se ha popularizado tanto en Costa Rica, Nicaragua, El Salvador, Guatemala, México, Perú, Chile, Cuba, Ecuador, Bolivia, Argentina, Venezuela y otros; ya sea por las divulgación que se ha realizado o porque puede ser descargado de internet, al punto que en varios países las instituciones encargadas de la revisión de los trabajos exigen que dentro de las memorias de cálculo se presenten capturas de pantallas de este programa [5].

En el país generalmente se realiza este tipo de cálculos mediante el uso de hojas de cálculo, por lo que surge la iniciativa de implementar el uso de un software especializado que permita agilizar estos procesos, donde además de saber utilizarlo se pueda dejar un precedente para la creación de nuestros propios programas que nos ayuden a optimizar el tiempo que influyen los cálculos.

En la ciudad de Ambato mediante el desarrollo de este software se busca introducir una nueva herramienta para el diseño de canales de manera que con su funcionamiento sea un producto de calidad, en el cual se implementarán secciones que no se encuentran en los programas u hojas de cálculo actualmente disponibles ya que estos solo realizan el cálculo de secciones típicas como son rectangulares, triangulares, trapezoidales, parabólicas y circulares [1], [6] y [7].

Es por estas razones que se pretende desarrollar un software que permita realizar el cálculo de canales abiertos de flujo uniforme en el que además de tener las secciones típicas se implementen la sección rectangular con esquinas redondeadas y triangular con fondo redondeado (planteados por Ven Te Chow), para así contribuir en la disminución del tiempo de cálculo y la confiabilidad de los resultados.

1.4. OBJETIVOS

1.4.1. General

Diseñar un software para el cálculo de canales abiertos de flujo uniforme.

1.4.2. Específicos

- Brindar a la Ingeniería Civil una herramienta que facilite los cálculos en el diseño de canales.
- Elaborar un documento guía para el estudio de la hidráulica de canales.
- Mejorar los programas existentes para el cálculo de canales.

CAPÍTULO II

FUNDAMENTACIÓN

2.1. FUNDAMENTACIÓN TEÓRICA

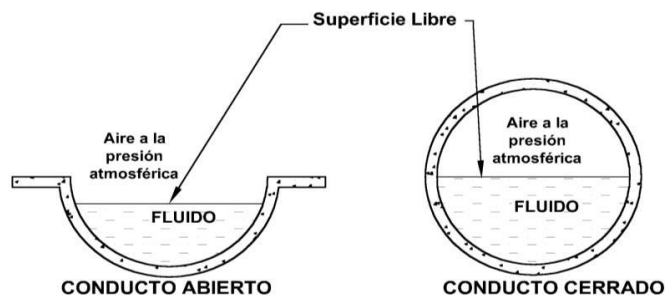
Después del aire que respiramos, el agua es el elemento más esencial para el hombre. Desde hace por lo menos 5000 años el hombre ha inventado y construido obras para el aprovechamiento del agua; entre las más antiguas están los CANALES, usados para llevar el agua de un lugar a otro [8].

2.1.1. Canales abiertos

Un canal abierto es un conducto en el que el agua fluye con una superficie libre [9].

Los canales pueden ser conductos abiertos o cerrados en los cuales el agua circula debido a la acción de la gravedad y sin ninguna presión, pues la superficie libre del líquido está en contacto con la atmósfera; esto quiere decir que el agua fluye impulsada por la presión atmosférica y su propio peso [8].

Figura 1: Flujo en conductos



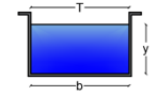
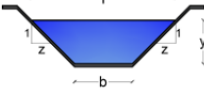



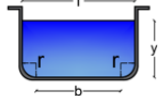
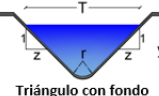
2.1.2. Geometría de los canales

Un canal construido con una sección transversal invariable y una pendiente de fondo constante, se conoce como canal prismático. De otra manera, el canal es no prismático [10].

El término sección de canal se refiere a la sección transversal de un canal tomada en forma perpendicular a la dirección del flujo. Una sección vertical del canal, sin embargo, es la sección vertical que pasa a través del punto más bajo o del fondo de la sección del canal. Para canales horizontales, por consiguiente, la sección del canal es siempre una sección vertical del canal [10].

Los canales artificiales a menudo se diseñan con secciones de figuras geométricas regulares. La tabla 1, relaciona siete formas geométricas utilizadas comúnmente. La forma trapezoidal es la más común para canales con terraplenes de tierra sin revestir, debido a que se proveen las pendientes necesarias para estabilidad. El rectángulo y el triángulo son casos especiales del trapecio. Debido a que el rectángulo tiene lados verticales, por lo general se utiliza para canales construidos con materiales estables, como mampostería, roca, metal o madera. La sección rectangular es utilizada solo para pequeñas acequias, cunetas a lo largo de carreteras y trabajos de laboratorio. El círculo es la sección más común para colectores y alcantarillas de tamaño pequeño y mediano. La parábola se utiliza como una aproximación a secciones de canales naturales de tamaño pequeño y mediano. El rectángulo con esquinas redondeadas es una modificación del rectángulo. El triángulo con fondo redondeado es una aproximación de la parábola; ésta es la forma creada a menudo con la utilización de excavadoras [10].

Tabla 1:Elementos geométricos de secciones de canal

Sección	Área A	Perímetro mojado P	Radio hidráulico R	Ancho superficial T	Profundidad hidráulica D	Factor de sección Z
 Rectángulo	by	$b + 2y$	$\frac{by}{b + 2y}$	b	y	$by^{1.5}$
 Trapezio	$(b + zy)y$	$b + 2y\sqrt{1 + z^2}$	$\frac{(b + zy)y}{b + 2y\sqrt{1 + z^2}}$	$b + 2zy$	$\frac{(b + zy)y}{b + 2zy}$	$\frac{[(b + zy)y]^{1.5}}{\sqrt{b + 2zy}}$
 Triángulo	zy^2	$2y\sqrt{1 + z^2}$	$\frac{zy}{2\sqrt{1 + z^2}}$	$2zy$	$\frac{1}{2}y$	$\frac{\sqrt{2}}{2}zy^{2.5}$
 Círculo	$\frac{1}{8}(\theta - \sin \theta)d^2$	$\frac{1}{2}\theta d$	$\frac{1}{4}\left(1 - \frac{\sin \theta}{\theta}\right)d$	$\frac{\left(\sin \frac{\theta}{2}\right) d}{2\sqrt{y(d - y)}}$	$\frac{1}{8}\left(\frac{\theta - \sin \theta}{\sin \frac{\theta}{2}}\right) d$	$\frac{\sqrt{2}}{32} \frac{(\theta - \sin \theta)^{1.5}}{(\sin \frac{\theta}{2})^{0.5}} d^{2.5}$
 Parábola	$\frac{2}{3}Ty$	$T + \frac{8y^{2*}}{3T}$	$\frac{2T^2y}{3T^2 + 8y^2}$	$\frac{3A}{2y}$	$\frac{2}{3}y$	$\frac{2}{9}\sqrt{6}Ty^{1.5}$
 Rectángulo con esquinas redondeadas ($y > r$)	$\left(\frac{\pi}{2} - 2\right)r^2 + (b + 2r)y$	$(\pi - 2)r + b + 2y$	$\frac{\left(\frac{\pi}{2} - 2\right)r^2 + (b + 2r)y}{(\pi - 2)r + b + 2y}$	$b + 2r$	$\frac{\left(\frac{\pi}{2} - 2\right)r^2}{b + 2r} + y$	$\frac{\left[\left(\frac{\pi}{2} - 2\right)r^2 + (b + 2r)y\right]^{1.5}}{\sqrt{b + 2r}}$
 Triángulo con fondo redondeado	$\frac{T^2}{4z} - \frac{r^2}{z}(1 - z \cot^{-1} z)$	$\frac{T}{z}\sqrt{1 + z^2} - \frac{2r}{z}(1 - z \cot^{-1} z)$	$\frac{A}{P}$	$2[z(y - r) + r\sqrt{1 + z^2}]$	$\frac{A}{T}$	$A\sqrt{\frac{A}{T}}$

*Aproximación satisfactoria para el intervalo $0 < x \leq 1$, donde $x=4y/T$. Cuando $x > 1$, utilice la expresión exacta $P = \left(\frac{T}{2}\right)[\sqrt{1 + x^2} + \frac{1}{x} \ln(x + \sqrt{1 + x^2})]$

Fuente: V. Chow. Hidráulica de Canales Abiertos. 1994.

2.1.2.1. Elementos geométricos de una sección de canal.

Los elementos geométricos son propiedades de una sección del canal que pueden ser definidos por completo por la geometría de la sección y la profundidad del flujo. Estos elementos son muy importantes y se utilizan con amplitud en el cálculo del flujo [10].

2.1.2.1.1. Profundidad del flujo.

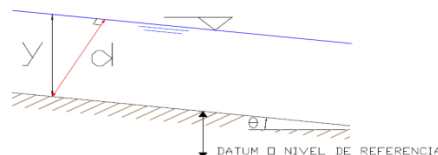
La profundidad del flujo (y) o tirante, es la distancia vertical desde el punto más bajo de una sección del canal hasta la superficie libre. Este término a menudo se intercambia con la profundidad de flujo de la sección (d). En efecto, la profundidad de flujo de la sección es la profundidad de flujo perpendicular a la dirección de éste, o la altura de la sección del canal que contiene el agua. Para un canal con un ángulo de pendiente longitudinal θ , puede verse que la profundidad del flujo es igual a la profundidad de la sección dividida por $\cos\theta$ (Ecuación 1). En el caso de canales con grandes pendientes, por consiguiente, los dos términos deben utilizarse de manera discriminada [9].

Según [1] tenemos la ecuación 1 y la figura 2.

Ecuación 1: Relación entre la profundidad del agua y el ángulo de la pendiente del fondo del canal.

$$y = \frac{d}{\cos \theta}$$

Figura 2: Relación entre la profundidad (y) y la profundidad (d)



Según la figura 2 se tiene que:

θ = Angulo de la pendiente del fondo del canal con una línea horizontal.

Si θ es pequeño $y \cong d$

En un canal se tiene una pendiente baja si la pendiente $S_o \leq 0.01$, y se tiene una pendiente alta si $S_o > 0.01$ [1].

2.1.2.1.2. Nivel.

El nivel es la elevación o distancia vertical desde un nivel de referencia hasta la superficie libre. Si el punto más bajo de la sección de canal se escoge como el nivel de referencia, el nivel es idéntico a la profundidad de flujo [10].

2.1.2.1.3. Ancho superficial.

El ancho superficial (T) es el ancho de la sección del canal en la superficie libre [10].

2.1.2.1.4. Área mojada.

El área mojada (A) es el área de la sección transversal del flujo perpendicular a la dirección del flujo [10].

2.1.2.1.5. Perímetro mojado

El perímetro mojado (P) es la longitud de la línea de intersección de la superficie de canal mojada y de un plano transversal perpendicular a la dirección de flujo [10].

Es decir, es el perímetro de la sección en contacto con una frontera rígida sin incluir la superficie libre [11].

2.1.2.1.6. Radio hidráulico

El radio hidráulico (R) es la relación entre el área mojada con respecto a su perímetro mojado [10].

Ecuación 2: Definición del radio hidráulico

$$R = \frac{A}{P}$$

2.1.2.1.7. Profundidad hidráulica

La profundidad hidráulica (D) es la relación entre el área mojada y el ancho en la superficie [10].

Ecuación 3: Definición de la profundidad hidráulica

$$D = \frac{A}{T}$$

2.1.2.1.8. Factor de sección para el cálculo del flujo crítico

El factor de sección para el cálculo de flujo crítico (Z), es el producto del área mojada y la raíz cuadrada de la profundidad hidráulica [10].

Ecuación 4: Definición del factor de sección para el cálculo del flujo crítico

$$Z = A\sqrt{D} = A\sqrt{\frac{A}{T}}$$

2.1.2.1.9. Factor de sección para el cálculo del flujo uniforme.

El factor de sección para el cálculo de flujo uniforme $AR^{2/3}$ es el producto del área mojada y el radio hidráulico elevado a la potencia 2/3 [10].

2.1.3. Estado de flujo en canales abiertos

El estado o comportamiento del flujo en canales abiertos está gobernado por la gravedad en relación con las fuerzas inerciales del flujo [11].

2.1.3.1. Efecto de la gravedad

El efecto de la gravedad sobre el estado de flujo se representa por la relación entre las fuerzas inerciales y las fuerzas gravitacionales. Esta relación está dada por el número de Froude [10].

Según sea la magnitud de las fuerzas de gravedad e inercia, un flujo es clasificado como subcrítico, crítico y supercrítico [1].

2.1.3.1.1. Número de Froude

Según [10] se tiene la ecuación 5 y la clasificación del flujo de acuerdo al número de Froude.

Ecuación 5: Número de Froude

$$NF = \frac{V}{\sqrt{gD}}$$

Donde:

NF= Número de Froude.

V= Velocidad del flujo

D= Profundidad hidráulica

g= Aceleración de la gravedad

De acuerdo con el número de Froude, el flujo puede ser:

- Supercrítico: Cuando el número de Froude es mayor a 1
- Crítico: Cuando el número de Froude es igual a 1
- Subcrítico: Cuando el número de Froude es menor a 1

2.1.3.1.1.1. Flujo crítico

El flujo crítico se presenta cuando la velocidad del flujo es igual que la velocidad de la onda de gravedad, el parámetro adimensional $NF=1$ [1].

2.1.3.1.1.2. Flujo subcrítico

El flujo subcrítico o flujo lento es aquel en el cual la velocidad del flujo es menor que la velocidad de la onda de gravedad, el parámetro adimensional $NF<1$ [1].

En este estado el papel jugado por las fuerzas gravitacionales es más pronunciado; por tanto, el flujo tiene una velocidad baja y a menudo se describe como tranquilo y de corriente lenta [10].

2.1.3.1.1.3. Flujo supercrítico

El flujo supercrítico o flujo rápido es aquel en el cual la velocidad del flujo es mayor que la velocidad de la onda de gravedad, el parámetro adimensional $NF>1$ [1].

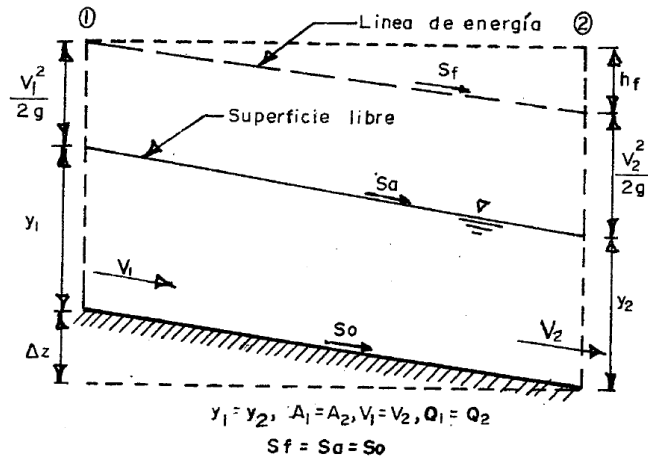
En este estado las fuerzas inerciales se vuelven dominantes; el flujo tiene una alta velocidad y se describe usualmente como rápido, ultrarrápido y torrencial [10].

2.1.4. Flujo uniforme en canales abiertos

El flujo uniforme se presenta cuando la velocidad media permanece constante en cualquier sección del canal. Con una superficie libre, esto implica que la sección transversal y el tirante permanecen constantes (figura 3). Como consecuencia de la definición, en flujo uniforme la pendiente de la línea de energía de fricción (S_f), la pendiente de la superficie libre del agua (S_a) y la pendiente geométrica del canal (S_o) son iguales. El hecho de que la velocidad media permanezca constante, se refiere estrictamente al hecho de que el flujo posea una velocidad constante en cada punto de la sección transversal a lo largo del canal; es decir, que la distribución de

velocidades de cada sección no se altera. El tirante correspondiente al flujo uniforme se conoce como tirante normal [11].

Figura 3: Flujo uniforme



El flujo uniforme es un estado ideal que difícilmente se logra. Sin embargo, en la mayoría de los casos (y sobre todo en canales rectos y largos de sección transversal y pendiente de fondo constante), se alcanza un flujo casi uniforme, de tal manera que la suposición es razonable especialmente porque simplifica el análisis [11].

El flujo uniforme es una condición de importancia básica que debe ser considerado en todos los problemas de diseño de canales. En un canal con cierta pendiente y rugosidad y que debe conducir cierto gasto, la condición de flujo uniforme es el criterio que gobierna el área de la sección transversal mínima requerida, o aun cuando exista otro criterio que determine las dimensiones de la sección, estas no podrán ser menores que dicha sección mínima [11].

2.1.5. Pendiente del canal

La pendiente longitudinal del fondo de un canal por lo general está dada por la topografía y por la altura de energía requerida para el flujo de agua. En muchos casos, la pendiente también depende del propósito del canal; por ejemplo, los canales utilizados para propósitos de distribución de agua, como los utilizado en irrigación abastecimientos de agua, minería hidráulica y proyectos hidroeléctricos requieren un

alto nivel en el punto de entrega. Por consiguiente, es conveniente una pendiente pequeña para mantener en el mínimo posible las pérdidas en elevación [10].

La pendiente de un canal se expresa de modos diferentes. En forma ideal, se define como la relación de la caída vertical (h) a la distancia horizontal en que ocurre dicha caída. Para pendientes pequeñas, comunes en el flujo en canales abiertos, es más práctico utilizar [12]:

Ecuación 6: Pendiente de un canal

$$S = \frac{h}{L}$$

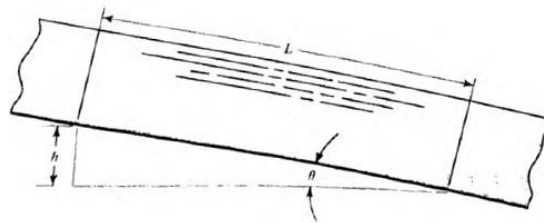
Donde:

S= Pendiente del canal

L= Longitud del canal

h= Distancia vertical en que ocurre la caída vertical

Figura 4: Pendiente de un canal



2.1.5.1. Formas para expresar la pendiente

Es posible expresar una pendiente de 0.001 [12]:

1. El canal desciende 1m por cada 1000m que avanza.
2. En porcentaje, donde $0.001=0.1\%$.
3. En forma de ángulo debido a que $\theta = h/L$. Por tanto, $\text{sen } \theta = 0.001$. Entonces,
 $\theta = \text{sen}^{-1}(0.001) = 0.057^\circ$

2.1.6. Ecuación de Manning

En el año 1889, el ingeniero irlandés Robert Manning presentó por primera vez la ecuación durante la lectura de un artículo en una reunión del Institute of Civil Engineers de Irlanda. El artículo fue publicado más adelante en Transactions, del Instituto. La ecuación en principio fue dada en una forma complicada y luego simplificada. Después, ésta fue modificada posteriormente por otros y expresada en unidades métricas [10].

Ecuación 7: Ecuación de Manning para flujo uniforme en unidades métricas

$$V = \frac{1}{n} * R^{\frac{2}{3}} * S^{\frac{1}{2}}$$

Donde:

V= Velocidad del flujo

n= Factor de resistencia o n de Manning

R= Radio hidráulico

S= Pendiente del canal.

En ésta, las unidades deben ser consistentes. La velocidad del flujo (V), estará en m/s si el radio hidráulico (R) se expresa en m. La pendiente del canal (S), es adimensional. El valor de n depende de la condición de la superficie del canal y, por tanto, es algo análogo a la rugosidad de la pared de un tubo [12].

Más tarde, fue reconvertida a unidades inglesas, dando como resultado [10].

Ecuación 8: Ecuación de Manning para flujo uniforme en unidades inglesas

$$V = \frac{1.486}{n} * R^{\frac{2}{3}} * S^{\frac{1}{2}}$$

Aquí, la velocidad se expresará en pies por segundo (pies/s), si R está en pies [12].

En esta conversión el valor numérico de n se mantuvo inmodificado. En consecuencia, el mismo valor de n es bastante utilizado en ambos sistemas de unidades [10].

Debido a la simplicidad de su forma y a los resultados satisfactorios que arroja en aplicaciones prácticas, la ecuación de Manning se ha convertido en la más utilizada de todas las ecuaciones de flujo uniforme para cálculos de flujos de canales abiertos [10].

2.1.6.1. Valores del coeficiente de rugosidad de Manning

2.1.6.1.1. Factores que afectan el coeficiente de rugosidad de Manning

No es raro que los ingenieros piensen que un canal tiene un valor único de n para todas las ocasiones. En realidad, el valor de n es muy variable y depende de un cierto número de factores. Para seleccionar el valor de n adecuado para diferentes condiciones de diseño, resulta muy útil tener un conocimiento básico de estos factores. Los factores que ejercen la mayor influencia sobre el coeficiente de rugosidad tanto en canales artificiales como en canales naturales son descritos a continuación. Debería hacerse notar que estos factores están hasta cierto punto interrelacionados; por tanto, la discusión de uno de los factores puede repetirse en conexión con otro [10].

- a) **Rugosidad superficial.** La rugosidad superficial se representa por el tamaño y la forma de los granos del material que forman el perímetro mojado y que producen un efecto retardador del flujo. Por lo general éste se considera como el único factor para la selección de un coeficiente de rugosidad, pero en realidad es sólo uno de varios factores principales. En general, granos finos dan como resultado un valor relativamente bajo de n , y granos gruesos, un valor alto de n [10].

- b) **Vegetación.** La vegetación puede considerarse como una clase de rugosidad superficial, pero también reduce de manera notable la capacidad del canal y retarda el flujo. Este efecto depende por completo de la altura, la densidad, la distribución y del tipo de vegetación, y es muy importante en el diseño de pequeños canales de drenaje [10].

En la universidad de Illinois se ha hecho una investigación para determinar el efecto de la vegetación en el coeficiente de rugosidad. En uno de los canales de drenaje bajo investigación en Illinois central, se midió un valor promedio de n de 0.033 en marzo de 1925, cuando el canal estaba en buenas condiciones. En abril de 1926, había materiales de sauces y hierbas secas en los taludes laterales, y se encontró que n era 0.055. Este incremento en n representa de un año de crecimiento de la vegetación. Durante los veranos de 1925 y 1926 ocurrió un gran crecimiento de plantas con hojas largas y planas en el fondo del canal. El valor de n de Manning correspondiente a niveles de mediados del verano fue aproximadamente de 0.115, y para la sección a banca llena fue de 0.099. Las plantas altas del fondo del canal fueron removidas por las aguas altas en septiembre de 1926; el valor promedio de n después de que ocurrió esto fue de 0.072. Las conclusiones arrojadas de esta investigación fueron, en parte, las siguientes [10]:

1. El valor mínimo de n que debe utilizarse para el diseño de canales de drenaje en Illinois central es 0.040. Este valor se obtiene en niveles altos durante los meses de verano en la mayor parte de los canales con mantenimiento cuidadoso, en los cuales el fondo del canal está libre de vegetación y las pendientes laterales están cubiertas con pasto o hierbas pequeñas, pero no con matorrales. No debe utilizarse este valor bajo de n a menos que el canal se limpie anualmente de todas sus hierbas y matorrales [10].
2. Debe utilizarse un valor de $n=0.050$ si el canal se limpia sólo en años alternados. Hierbas altas y matorrales de sauces de 3 a 4 pies de altura en los taludes laterales producirán este valor de n [10].
3. Si los canales no se limpian durante un determinado número de años, el crecimiento puede volverse tan abundante de tal modo que se encuentren valores de $n > 0.100$ [10].

4. Los árboles con diámetros de 6 a 8 pulg que crecen en los taludes laterales no impiden el flujo tanto como lo hacen los crecimientos de matorrales pequeños, siempre y cuando se corten las ramas colgantes [10].

El U.S. Soil Conservation Service ha hecho una serie de estudios del flujo de agua en canales pequeños poco profundos protegidos con recubrimiento vegetal. Se ha encontrado que para estos canales los valores de n varían con la forma y la sección transversal del canal, la pendiente del lecho del canal y la profundidad del flujo. Al comparar dos canales, y manteniendo los factores iguales, la menor profundidad promedio arroja un valor de n mayor, debido a la mayor proporción afectada por la vegetación. Así, un canal triangular tiene un valor de n mayor que un canal trapezoidal, y un canal ancho tiene un valor de n menor que un canal angosto. Un flujo con suficiente profundidad tiende a doblar y a sumergir la vegetación, con lo cual se producen valores bajos de n . Una pendiente alta genera altas velocidades, con lo cual se producen valores bajos de n [10].

- c) **Irregularidad del canal.** Las irregularidades del canal incluyen irregularidades en el perímetro mojado y variaciones en la sección transversal, tamaño, y forma de estas a lo largo del canal. En canales naturales, tales irregularidades por lo general son producidas por la presencia de barras de arena, ondas de arena, crestas y depresiones, hoyos y montículos en el lecho del canal. Estas irregularidades introducen rugosidad adicional a la causada por la rugosidad superficial y otros factores. En general, un cambio gradual y uniforme en la sección transversal, o en su tamaño y forma no produce efectos apreciables en el valor de n , pero cambios abruptos o alteraciones de secciones pequeñas y grandes requieren el uso de un valor grande de n . En este caso, el incremento de n puede ser 0.005 o mayor. Los cambios que hacen que el flujo cambie de manera sinuosa de un lado al otro del canal producirán el mismo efecto [10].
- d) **Alineamiento del canal.** Curvas suaves con radios grandes producirán valores de n relativamente bajos, en tanto que curvas bruscas con meandros severos incrementarán n . Con base en pruebas de laboratorio llevadas a cabo en

canaletas, Scobey sugirió que el valor n se incrementara en 0.001 por cada 20 grados de curvatura en 100 pies de canal. Aunque es dudoso que la curvatura llegue a aumentar el valor de n en más de 0.002 ó 0.003, su efecto no debería ignorarse, debido a que la curvatura puede inducir la acumulación de material flotante y, por consiguiente, incrementar indirectamente el valor de n . En general, el incremento de la rugosidad en canales no revestidos que conducen agua con baja velocidad es insignificante. Un aumento de 0.002 en el valor de n constituye una provisión adecuada para la pérdida en curvas en la mayor parte de las canaletas que contengan curvaturas pronunciadas sin importar que estén construidos en concreto o en otros materiales. La presencia de meandros en corrientes naturales, sin embargo, puede incrementar el valor de n tan alto como 30% [10].

- e) **Sedimentación y socavación.** En general, la sedimentación puede cambiar un canal muy irregular en un canal relativamente uniforme y disminuir el n , en tanto que la socavación puede hacer lo contrario e incrementar el n . Sin embargo, el efecto dominante de la sedimentación dependerá de la naturaleza del material depositado. Depósitos no uniformes, como barras de arena y ondulaciones de arena, constituyen irregularidades del canal e incrementarán la rugosidad. La cantidad y uniformidad de la socavación dependerán del material que conforma el perímetro mojado. Así, un lecho de arena o de gravas se erosionará más uniformemente que un lecho de arcillas. La sedimentación de las arcillas erosionadas en los terrenos aguas arriba tenderá a emparejar las irregularidades de un canal dragado a través de un suelo arcilloso. La energía utilizada para erosionar y mover el material en suspensión o por saltación a lo largo del lecho también incrementará el valor de n . El efecto de la socavación no es importante siempre y cuando la erosión en el lecho del canal causada por velocidades altas progrese igual y uniformemente [10].
- f) **Obstrucción.** La presencia de obstrucciones de troncos, pilas de puente y estructuras similares tienden a incrementar el n . La magnitud de este aumento depende de la naturaleza de las obstrucciones, de su tamaño, forma, número y distribución [10].

- g) **Tamaño y forma del canal.** No existe evidencia definitiva acerca del tamaño y la forma del canal como factores importantes que afecten el valor de n . Un incremento en el radio hidráulico puede aumentar o disminuir n , según la condición del canal [10].
- h) **Nivel y caudal.** En la mayor parte de las corrientes el valor de n disminuye con el aumento en el nivel y en el caudal. Cuando el agua es poco profunda, las irregularidades del fondo del canal quedan expuestas y sus efectos se vuelven pronunciados. Sin embargo, el valor de n puede ser grande en niveles altos si las bancas están cubiertas por pastos o son rugosas [10].

2.1.6.1.1.1. Resumen de los valores de n que afectan el coeficiente de rugosidad de Manning

RUGOSIDAD DE LA SUPERFICIE

- Granos finos = Valor bajo de n
- Granos gruesos = Valor alto de n

VEGETACIÓN

- Retarda el flujo
- Reduce la capacidad del canal, depende de (altura, densidad, distribución y tipo de vegetación)
 - ✓ Aumento de vegetación = Aumento de n
 - ✓ Menor vegetación = Menor valor de n

IRREGULARIDADES DEL CANAL

- Comprende irregularidades en el perímetro mojado y variación en la sección transversal, tamaño, y forma a lo largo de la longitud del canal
- Cambio gradual y uniforme NO AFECTA el valor de n
- Cambios abruptos o alteraciones justifican el uso de un valor grande de n

ALINEAMIENTO DEL CANAL

- Curvas suaves = Valor bajo de n

- Curvas agudas = Valor alto de n

SEDIMENTACIÓN Y SOCAVACIÓN

- Sedimentación, los depósitos pueden cambiar un canal muy irregular en uno comparativamente uniforme y disminuye n
- Socavación, en un canal uniforme, regular y aumenta n

OBSTRUCCIÓN

- Troncos, pilas de puentes, aumentan el valor de n

TAMAÑO Y FORMA DEL CANAL

- Radio hidráulico aumenta o disminuye
- No existe evidencia que el tamaño y forma afecte el valor de n

NIVEL Y CAUDAL

- Disminuye n, con el aumento en el nivel y en el caudal

2.1.7. Cálculo del flujo uniforme

El flujo volumétrico en el canal se calcula a partir de la ecuación de continuidad [12]:

Ecuación 9: Ecuación de continuidad

$$Q = A * V$$

En el análisis del flujo en canales abiertos, es común que Q reciba el nombre de caudal. Al sustituir la ecuación 7 en la 9, se obtiene una ecuación que relaciona en forma directa el caudal con los parámetros físicos del canal [12]:

Ecuación 10: Ecuación de Manning para caudal en unidades métricas

$$Q = \left(\frac{1}{n}\right) AR^{2/3} S^{1/2}$$

Éste es el único valor del caudal en el que habrá flujo uniforme para la profundidad del canal, y se denomina caudal normal. Las unidades de Q son m³/s, si el área se expresa en metros cuadrados (m²) y el radio en metros (m) [12].

Según [10] se puede obtener otra forma equivalente a la ecuación 10 para el sistema inglés.

Ecuación 11: Ecuación de Manning para caudal en unidades inglesas

$$Q = \left(\frac{1.486}{n} \right) AR^{\frac{2}{3}} S^{\frac{1}{2}}$$

En estas ecuaciones, Q es el caudal normal expresado en pies cúbicos por segundo (pies³/s), si A está en pies cuadrados (pies²) y R se expresa en pies [12].

A las ecuaciones 10 y 11 se las puede expresar en función de la velocidad. En la tabla 2 se hace un resumen de las ecuaciones para el sistema métrico e inglés.

Tabla 2: Ecuaciones para el cálculo del flujo uniforme en unidades métricas e inglesas.

SISTEMA MÉTRICO	SISTEMA INGLÉS
$V = \frac{1}{n} * R^{2/3} * S^{1/2}$ $Q = V * A$ $Q = \frac{1}{n} * A * R^{2/3} * S^{1/2}$	$V = \frac{1.486}{n} * R^{2/3} * S^{1/2}$ $Q = V * A$ $Q = \frac{1.486}{n} * A * R^{2/3} * S^{1/2}$
UNIDADES	
V = m/s S = m/m Q = m ³ /s R = m	V = ft/s S = ft/ft Q = ft ³ /s R = ft

Fuente: J. Evett, R. Giles y C. Liu. Mecánica de los fluidos e hidráulica, 1994.

2.1.7.1. El factor de sección para el cálculo del flujo uniforme

La expresión AR^{2/3} se conoce como factor de sección para el cálculo de flujo uniforme, y es un elemento importante en el cálculo de flujo uniforme. A partir de la ecuación 10, este factor puede expresarse como [10]:

Ecuación 12: Factor de sección para el cálculo de flujo uniforme en unidades métricas

$$AR^{2/3} = \left(\frac{nQ}{S^{1/2}} \right)$$

La ecuación 12 se aplica a una sección de canal cuando el flujo es uniforme. La parte derecha de la ecuación contiene los valores de n , Q y S ; pero su parte izquierda depende sólo de la geometría del área mojada. Por consiguiente, la ecuación muestra que para una determinada condición de n , Q y S , existe sólo una profundidad posible para mantener un flujo uniforme, siempre y cuando el valor de $AR^{2/3}$ aumente con incrementos en la profundidad, lo cual es cierto en la mayor parte de los casos. Esta profundidad es la profundidad normal (y_n). Cuando en una sección de canal se conocen n y S en la ecuación 12 puede verse que puede existir sólo un caudal para mantener un flujo uniforme a través de la sección, siempre y cuando $AR^{2/3}$ aumente siempre con un incremento en la profundidad. Este caudal es el caudal normal (Q_n) [10].

La ecuación 12 es una herramienta muy útil para el cálculo y el análisis del flujo uniforme. Cuando se conocen el caudal, la pendiente y la rugosidad, esta ecuación da el factor de sección $AR^{2/3}$ y, por consiguiente, la profundidad normal (y_n) también conocida como tirante normal. Por otra parte, cuando n , S y la profundidad y por consiguiente el factor de sección, se conocen, puede calcularse el caudal normal (Q_n) utilizando la ecuación 10 [10].

Según [10] se puede obtener otra forma equivalente a la ecuación 12 para el sistema inglés.

Ecuación 13: Factor de sección para el cálculo de flujo uniforme en unidades inglesas

$$AR^{2/3} = \left(\frac{nQ}{1.486 S^{1/2}} \right)$$

2.1.7.2. Cálculo de la pendiente normal y crítica

Cuando se conocen el caudal y la rugosidad, la ecuación de Manning puede utilizarse para determinar la pendiente en un canal prismático en el cual el flujo es uniforme a determinada profundidad normal de flujo (y_n). La pendiente determinada de esta manera algunas veces se llama específicamente pendiente normal (S_n) [10].

Al variar la pendiente del canal hasta cierto valor, es posible cambiar la profundidad normal y hacer que el flujo uniforme ocurra en un estado crítico para el caudal y la rugosidad determinados. La pendiente así obtenida es la pendiente crítica (S_c), y la profundidad normal correspondiente es igual a la profundidad crítica [10].

Según [8] se tienen las ecuaciones 14 y 15 para el cálculo de la pendiente correspondientes al sistema métrico e inglés respectivamente.

Ecuación 14: Cálculo de pendiente para el sistema métrico

$$S = \left(\frac{V n}{R^{2/3}} \right)^2$$

Ecuación 15: Cálculo de pendiente para el sistema inglés

$$S = \left(\frac{V n}{1.486 R^{2/3}} \right)^2$$

Donde:

S= Pendiente del canal

V= Velocidad del flujo

n= Factor de resistencia o n de Manning

R= Radio hidráulico

2.1.7.3. Problemas de cálculo de flujo uniforme

El cálculo de flujo uniforme puede llevarse a cabo a partir de dos ecuaciones: la ecuación de continuidad y una ecuación de flujo uniforme. Cuando se utiliza la

ecuación de Manning como ecuación de flujo uniforme, el cálculo involucrará las siguientes seis variables [10]:

- 1) El caudal normal Q
- 2) La velocidad media de flujo V
- 3) La profundidad normal y
- 4) El coeficiente de rugosidad n
- 5) La pendiente de canal S
- 6) Los elementos geométricos que dependen de la forma de la sección de canal, como A , R , etc.

Cuando se conocen cuatro de las anteriores seis variables, los dos restantes pueden determinarse a partir de las dos ecuaciones. Los siguientes son algunos de los tipos de problemas de cálculo de flujo uniforme [10]:

- a) **Calcular el caudal normal.** En aplicaciones prácticas, este cálculo se requiere para la determinación de la capacidad de un canal determinado [10].
- b) **Determinar la velocidad de flujo.** Este cálculo tiene muchas aplicaciones. Por ejemplo, a menudo se requiere para el estudio de efectos de socavación y sedimentación de un canal determinado [10].
- c) **Calcular la profundidad normal.** Este cálculo se requiere para la determinación del nivel de flujo en un canal determinado [10].
- d) **Determinar la rugosidad del canal.** Este cálculo se utiliza para averiguar el coeficiente de rugosidad en un canal determinado. [10].
- e) **Calcular la pendiente del canal.** Este cálculo se requiere para ajustar la pendiente de un canal determinado [10].
- f) **Determinar las dimensiones de la sección de canal.** Este cálculo se requiere principalmente para propósitos de diseño [10].

2.1.8. Diseño de canales con flujo uniforme

El diseño de un canal consiste en elegir la forma y las dimensiones de la sección transversal de tal manera que cumplan con requisitos hidráulicos y económicos [1].

Cuando se diseña un canal en flujo uniforme la profundidad de diseño será la profundidad normal, se busca que el flujo que se transportará en el canal se comporte sin fluctuaciones ni irregularidades, es decir evitando flujo crítico [1].

Durante el diseño de un canal abierto la sección transversal del canal, la rugosidad y la pendiente del fondo del canal son conocidas, el objetivo primordial del diseño es determinar la magnitud de la velocidad media del flujo, la profundidad y el caudal sea conocido cualquiera de ellos [1].

El diseñador calcula las dimensiones del canal artificial mediante una ecuación de flujo uniforme y luego decide acerca de las dimensiones finales con base en el eficiencia hidráulica o reglas empíricas de sección óptima, aspectos prácticos constructivos y economía. Los factores que se consideran en el diseño son: la clase del material que conforma el cuerpo del canal, la cual determina el coeficiente de rugosidad; la velocidad mínima permisible, para evitar la depositación si el agua mueve limos o basuras; la pendiente del fondo del canal y las pendientes laterales; y la sección más eficiente [10].

2.1.9. Sección hidráulica óptima

También conocida como sección recta de máximo rendimiento o sección de máxima eficiencia hidráulica.

En el diseño de canales se involucra la selección de la forma del canal y la pendiente del fondo para transportar un caudal dado con una profundidad de flujo dada [1].

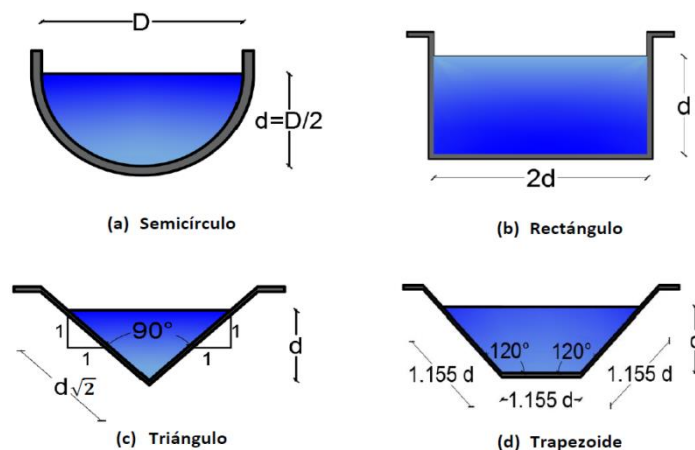
Si se tiene un caudal, una pendiente y una rugosidad, en el diseño se debe tratar de minimizar el área de la sección transversal (A) para así reducir los costos de construcción [1].

La conductividad o tránsito de agua en un canal, depende en gran medida de la relación existente entre el área mojada y el perímetro mojado en un canal [1].

Se sabe que la conductividad de una sección de canal se incrementa con el aumento en el radio hidráulico o la disminución en el perímetro mojado [10].

La sección recta de máximo rendimiento para un canal abierto se define como aquella sección que dé el máximo caudal cuando se dan la pendiente, el área y el coeficiente de rugosidad. Si estas magnitudes se mantienen constantes, la de velocidad (y, por tanto, el caudal) será máximo cuando el perímetro mojado sea mínimo. Basándose en esta premisa, se puede determinar la sección recta de mayor rendimiento (y, por tanto, la más económica) para las formas más comunes. De todas las secciones rectas, la de máximo rendimiento es el semicírculo, ya que tiene el perímetro mojado mínimo para un área dada. Para una sección rectangular, la de mayor rendimiento es la que tiene una profundidad igual a la mitad de su anchura. Para una sección triangular, la que tiene las pendientes de los lados igual a la unidad es la de máximo rendimiento. Y para una sección trapezoidal es la que es igual a la mitad de un hexágono regular (es decir los tres lados iguales con ángulos interiores de 120° cada uno). Todas estas secciones se muestran en la figura 5 [13].

Figura 5: Sección hidráulica óptima



2.1.9.1. Notas de la sección hidráulica óptima

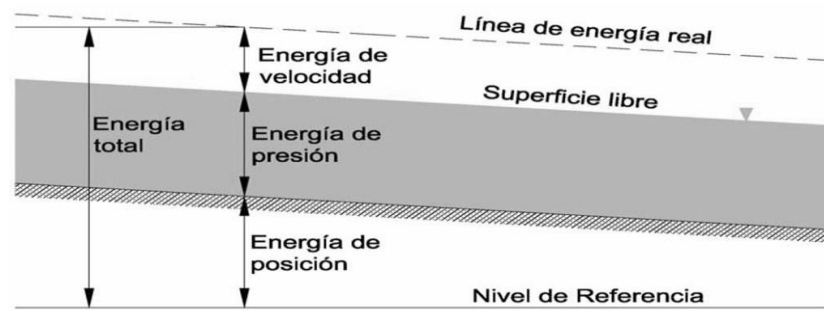
- 1) Para todos los canales trapezoidales, la sección hidráulica óptima se obtiene cuando el radio hidráulico es igual a la mitad del tirante. La sección simétrica será un semihexágono [13].
- 2) Para un canal rectangular la altura va a ser igual al ancho del canal para dos y se cumple también la condición de que el radio hidráulico sea igual a la mitad del tirante. Entonces la profundidad óptima es la mitad de la anchura con el radio hidráulico igual a la mitad de la profundidad [13].
- 3) El círculo tiene el menor perímetro para un área dada. Un canal abierto semicircular desaguará más agua que cualquier otro de distinta forma (para la misma área, pendiente y coeficiente n) [13].
- 4) Para un área de una sección recta y una pendiente dadas, el caudal a través de un canal con una rugosidad dada será máximo cuando el radio hidráulico sea máximo. El radio hidráulico será máximo cuando el perímetro mojado sea mínimo [13].

2.1.10. Energía.

La energía se define como la capacidad para realizar un trabajo. El trabajo resulta al aplicar una fuerza a lo largo de cierto recorrido y por lo general, se define como el producto de una fuerza por la longitud del recorrido en la dirección de la aplicación. La energía y el trabajo se expresan en las mismas unidades, es decir kpm o Nm. El Nm es el julio (J) [13].

Los fluidos en movimiento poseen energía, la cual aparece en tres formas distintas: potencial o de posición, cinética o de velocidad y energía de presión [13].

Figura 6: Energía total en una sección de un canal



Fuente: P. Rodríguez. Hidráulica II. 2008.

Energía potencial: Se refiere a la energía que posee el elemento de fluido debido a su elevación respecto a la cota de referencia. La energía potencial (PE) viene determinada cuantitativamente por el producto del peso (W) del elemento por la distancia vertical de éste a la cota de referencia (z) [13].

Ecuación 16: Energía potencial

$$PE = W z$$

PE, se representa con el tirante (d) o profundidad del agua en el canal [14].

Energía cinética: Se refiere a la energía que posee el elemento de fluido debido a su velocidad. La energía cinética (KE) viene determinada cuantitativamente por el producto de la masa (m) del elemento por el cuadrado de su velocidad (V) dividido por dos. La masa (m) puede ser sustituida por $\frac{W}{g}$ (donde W es el peso y g la aceleración de la gravedad) [13].

Ecuación 17: Energía cinética

$$KE = \frac{W V^2}{2 g}$$

Energía de presión: Llamada algunas veces energía del flujo, es la cantidad de trabajo que se requiere para forzar al fluido a moverse de cierta distancia contra la presión. La energía de presión (FE) se puede calcular determinando el trabajo necesario para mover el elemento de fluido una distancia igual a la longitud del

segmento recorrido (d). La fuerza que realiza el trabajo es igual al producto de la presión (p) por el área de la sección recta (A) del elemento. El término Ad es, de hecho, el volumen del elemento, que puede reemplazarse por $\frac{W}{\gamma}$, donde γ es el peso específico del fluido. [13].

Ecuación 18: Energía de presión

$$FE = \frac{p W}{\gamma}$$

La energía total (ET) es la suma de PE, KE Y FE [13].

Ecuación 19: Energía total

$$ET = W z + \frac{W V^2}{2 g} + \frac{p W}{\gamma}$$

Teóricamente es conveniente manejar la energía como carga, o lo que es lo mismo, como cantidad de energía por unidad de peso del fluido. La ecuación 19 se puede modificar al expresar la energía total como altura de carga (H), dividiendo todos los términos de la ecuación por W, peso del fluido [13].

Ecuación 20: Altura de carga

$$H = z + \frac{V^2}{2 g} + \frac{p}{\gamma}$$

El término z se llama cota topográfica; $\frac{V^2}{2 g}$ se conoce como altura de velocidad y $\frac{p}{\gamma}$ como altura de presión. Cada término de la ecuación 20 viene expresado en unidades de longitud [13].

2.1.10.1. Energía específica

La energía específica se define como la cantidad de energía por unidad de peso es decir por kilogramo de agua que fluye a través de la sección de canal, medida con respecto al fondo del canal [14].

Ecuación 21: Energía específica

$$E = y + \frac{V^2}{2g}$$

Fuente: J. Evett, R. Giles y C. Liu. Mecánica de los fluidos e hidráulica. 1994.

La energía específica es, pues la suma del tirante y la carga de velocidad. Como está referida al fondo del canal va a cambiar cada vez que éste ascienda o descienda, en pocas palabras la energía específica depende del tirante del agua [14].

La ecuación 21 puede también expresarse en función del caudal (Q) y el área (A) de la sección transversal, que es función del tirante ($V=Q/A$), y sustituyendo el valor de la velocidad en la ecuación de la energía específica, se tiene [14]:

Ecuación 22: Energía específica en función del caudal

$$E = y + \frac{Q^2}{2gA^2}$$

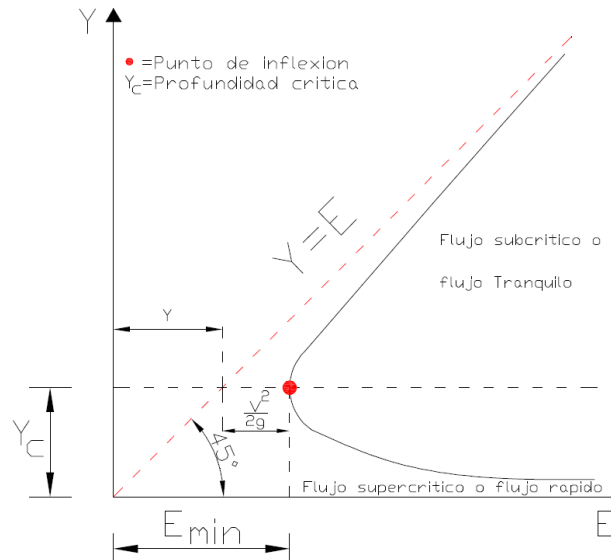
2.1.10.2. Curvas de energía específica

Para visualizar los regímenes posibles de flujo en un canal, es útil una gráfica de la profundidad versus la energía específica [12].

La curva de energía específica es la representación gráfica del régimen y estado de flujo en un canal, es decir es la visualización de cómo cambia la energía cada vez que se cambia la profundidad [1].

Para una sección y descarga particulares en un canal, la curva de la energía específica se asemeja a la que se ilustra en la figura 7[12].

Figura 7: Curva de energía específica



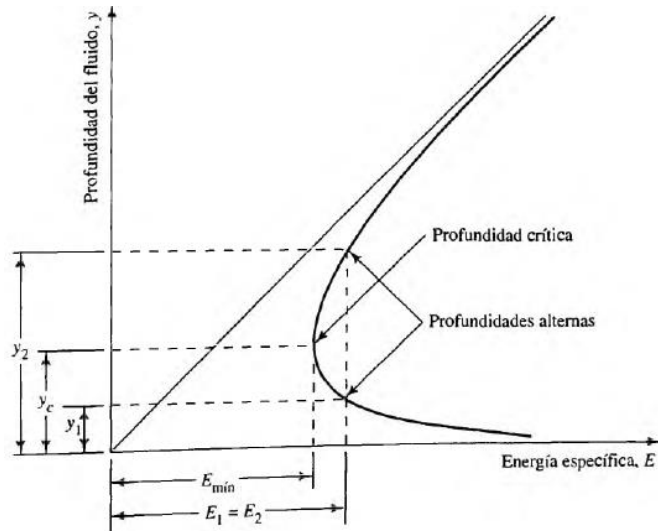
Fuente: L. Acosta, J. Barragan, y A. Reyes. Software para el diseño de canales abiertos. 2007.

La línea a 45° sobre la gráfica representa $E = y$. Entonces, para cualquier punto sobre la curva, la distancia horizontal entre esta línea y el eje y representa la energía potencial (y). La distancia restante a la curva de la energía específica es la energía cinética $\left(\frac{v^2}{2g}\right)$. Aparece un valor mínimo definido de E , y se demuestra que ocurre cuando el flujo se halla en el estado crítico, es decir cuando $NF = 1$ [12].

La profundidad correspondiente a la energía específica mínima, entonces, se denomina profundidad crítica (y_c). El flujo es subcrítico para cualquier profundidad mayor que y_c . A la inversa, para cualquier profundidad menor que y_c , el flujo es supercrítico. Observe que para cualquier nivel de energía mayor que el mínimo, pueden existir dos profundidades diferentes. En la figura 8, tanto y_1 por debajo de la profundidad crítica (y_c), como y_2 por arriba de y_c , tienen la misma energía. En el caso de y_1 , el flujo es supercrítico, y gran parte de la energía es cinética debido a su velocidad elevada. A la profundidad mayor y_2 , el flujo es más lento y sólo una parte

pequeña de la energía es cinética. Se denomina a las dos trayectorias, y_1 y y_2 , como trayectorias alternas de la energía específica (E) [12].

Figura 8: Profundidad crítica y profundidades alternas



Energía específica mínima ($E_{mín.}$): Se llama energía específica mínima la que puede tener la lámina de agua para ser capaz de transportar el caudal que dio origen a la curva [14].

2.1.10.2.1. Características de la curva de energía

- La parte superior de la curva representa el flujo subcrítico porque el número de Froude es menor que la unidad, $NF < 1$ y la profundidad normal es mayor que la profundidad crítica $y_n > y_c$ [1].
- El punto de inflexión en la curva corresponde al estado crítico del flujo entendiendo que el estado crítico del flujo sucede cuando el número de Froude sea igual a la unidad $NF = 1$ [1].
- La parte inferior de la curva corresponde al estado supercrítico del flujo, es decir que el número de Froude es mayor que la unidad, $NF > 1$ y la profundidad normal es menor que la profundidad crítica $y_n < y_c$ [1].

- Cada caudal tiene su propia curva de energía específica, es decir que la curva se desplaza [1].
- Cuando existe proporcionalidad entre la profundidad del flujo y la energía específica, se formará una recta a 45° y el estado de flujo subcrítico tiende a ser asintótico a esta recta [1].
- En general la recta posee dos ramales, uno asintótico al eje de las abscisas y otro, asintótico a una línea que forma un ángulo de 45° con relación a la horizontal [1].
- Cuando se presente flujo subcrítico, siempre que se aumente la profundidad del flujo, la energía específica aumentara y viceversa [1].
- Cuando se presente flujo supercrítico, siempre que se aumente la profundidad del flujo, la energía específica disminuirá y viceversa [1].

Cuando se presentan los flujos subcríticos y supercríticos, las velocidades son menores y mayores que la velocidad crítica, respectivamente, entonces en el flujo subcrítico aparecen pequeñas ondas superficiales avanzando aguas arriba, mientras que en el flujo supercrítico dichas ondas serán barridas aguas abajo, formando un ángulo β ; estas ondas son llamadas comúnmente como ondas diamante [1].

Se puede concluir, que para una energía específica dada, es posible tener dos profundidades, una con el flujo subcrítico y otra con el flujo supercrítico; estas dos profundidades se conocen con el nombre de profundidades secuentes o alternas [1].

2.1.11. Flujo crítico

Los análisis sobre el estado crítico de flujo se han referido principalmente a una sección particular de canal, conocida como sección crítica. Si el estado crítico del flujo existe a través de toda la longitud del canal o a lo largo de un tramo de éste, el flujo en el canal es un flujo crítico. La pendiente del canal que mantiene un

determinado caudal con una profundidad uniforme y crítica se conoce como pendiente crítica (S_c). Una pendiente de canal menor que la pendiente crítica producirá un flujo más lento de naturaleza subcrítica para el caudal determinado y por consiguiente, se conoce como pendiente suave o subcrítica. Una pendiente mayor que la pendiente crítica producirá un flujo más rápido de naturaleza supercrítica y se conoce como pendiente empinada o supercrítica [10].

Un flujo en estado crítico o cerca de él es inestable. Esto se debe a que un pequeño cambio de energía específica en estado crítico, o cerca de él, producirá un cambio grande en la profundidad. Este hecho también puede identificarse en la curva de energía específica. Como la curva es casi vertical cerca de la profundidad crítica, un ligero cambio en la energía cambiaría la profundidad a profundidades alternas mucho más pequeñas o más grandes, correspondientes a la energía específica después del cambio. Cuando el flujo está cerca del estado crítico, la superficie del agua parece inestable y ondulada. Por lo general, tales fenómenos son causados por los pequeños cambios en energía debido a las variaciones en la rugosidad del canal, la sección transversal, la pendiente o algunos depósitos de sedimentos o basuras. Si en el diseño de un canal se encuentra que la profundidad es igual o muy cercana a la profundidad crítica a lo largo de una gran longitud de canal, la forma o la pendiente del canal deben modificarse, si es posible, para asegurar una mayor estabilidad [10].

El criterio para un estado crítico de flujo es la base para el cálculo de flujo crítico [10].

El flujo crítico se puede conseguir en forma práctica: reduciendo la sección, provocando una sobre elevación del fondo del cauce o utilizando los dos criterios anteriores [14]

De lo anterior los términos del régimen crítico pueden definirse como sigue:

- **Caudal crítico.**

Es el caudal máximo para una energía específica determinada, o el gasto que se producirá con la energía específica mínima [14].

- **Tirante crítico.**

Es el tirante hidráulico que existe cuando el gasto es el máximo para una energía específica determinada, o el tirante al que ocurre un caudal determinado con la energía específica mínima [14].

- **Velocidad crítica.**

La velocidad media cuando el caudal es el crítico [14].

- **Pendiente crítica.**

Es el valor particular de la pendiente del fondo del canal para la cual este conduce un caudal (Q) en régimen uniforme y con energía específica mínima, o sea, que en todas secciones se tiene el tirante crítico [14].

2.1.11.1. Condiciones del flujo crítico

a) La energía específica es mínima para un caudal determinado [10].

b) El caudal es máximo para una determinada energía específica [10].

c) La fuerza específica es mínima para un caudal determinado [10].

d) La altura de velocidad es igual a la mitad de la profundidad hidráulica en un canal de baja pendiente [10].

e) El número de Froude es igual a la unidad [10].

f) La velocidad de flujo en un canal de baja pendiente con distribución uniforme de velocidades es igual a la celeridad de pequeñas ondas gravitacionales en aguas poco profundas causadas por perturbaciones locales [10].

2.1.11.2. Determinación del flujo crítico

Para que se presente un estado de flujo crítico, se ha de cumplir la condición en la cual el número de Froude sea igual a uno, bajo este estado de flujo se tiene que la energía específica es mínima para un caudal determinado además de esto se presenta que la corriente es inestable y está sujeta a fluctuaciones en la profundidad del líquido [1].

Es por este motivo que no se debe diseñar canales con flujo crítico sino con flujo subcrítico o supercrítico, esto en función de la pendiente que se tenga en el canal, en el diseño se deben buscar profundidades en un rango de $1.1y_c < y < 0.9y_c$ con $y \neq y_c$ [1].

En la curva de energía específica se observa que la profundidad crítica se presenta cuando la energía específica es mínima, es decir que la ecuación general de flujo crítico se obtiene al derivar la ecuación de energía específica con respecto a la profundidad del flujo e igualarla a cero [1].

Ecuación 23: Ecuación diferencial del flujo crítico

$$\frac{dE}{dy} = 0$$

Al derivar la ecuación de la energía específica con respecto a y y al notar que Q es constante [10],

$$\frac{dE}{dy} = 1 - \frac{Q^2}{gA^3} \frac{dA}{dy}$$

El diferencial del área mojada dA cerca de la superficie libre es igual Tdy . Ahora $\frac{dA}{dy} = T$, luego la anterior ecuación se convierte en [1]:

Ecuación 24: Ecuación general del flujo crítico

$$\frac{Q^2}{gA^3} T = 1$$

Con la ecuación 24, dejando el caudal en términos de la velocidad se puede determinar el estado crítico de flujo [1].

Ecuación 25: Estado crítico de flujo

$$\frac{V}{\sqrt{gD}} = 1$$

Lo cual significa que $NF = 1$; ésta es la definición de flujo crítico dada anteriormente [10].

Si en la ecuación 24 se separan las variables geométricas de la que no son geométricas, se llega a la condición general del flujo crítico [1].

Ecuación 26: Condición general del flujo crítico

$$\frac{Q^2}{g} = \frac{A^3}{T}$$

En el criterio para flujo crítico se establece que en el estado crítico del flujo la altura de velocidad es igual a la mitad de la profundidad hidráulica [10].

Ecuación 27: Criterio para flujo crítico

$$\frac{V^2}{2g} = \frac{D}{2}$$

2.1.11.3. El factor de sección para el cálculo del flujo crítico

Al sustituir $V = Q/A$ en la ecuación 27 y simplificando se tiene [10]:

Ecuación 28: Factor de sección Z para una sección de canal en estado crítico de flujo

$$Z = \frac{Q}{\sqrt{g}}$$

Donde $Z = A\sqrt{D}$, es el factor de sección para el cálculo del flujo crítico [10].

La ecuación 28 establece que el factor de sección Z para una sección de canal en estado crítico de flujo es igual al caudal dividido por la raíz cuadrada de g . Debido a que el factor de sección Z por lo general es una función de valor único de la profundidad, la ecuación indica que existe sólo una profundidad crítica posible para mantener determinado caudal en un canal y, de manera similar, cuando se fija la profundidad, que puede existir solo un caudal que mantenga un flujo crítico y que haga crítica la profundidad en una determinada sección [10].

La ecuación 28 es una herramienta muy útil para el cálculo y el análisis del flujo crítico en un canal abierto. Cuando se conoce el caudal, la ecuación da el factor de sección crítico (Z_c) y, por consiguiente, la profundidad crítica (y_c). Por otra parte, cuando la profundidad y, por tanto, el factor de sección son conocidos, el caudal crítico (Q_c) puede calcularse mediante la ecuación 29 de la siguiente manera[10]:

Ecuación 29: Cálculo del caudal para una sección de canal en estado crítico de flujo

$$Q = Z\sqrt{g}$$

2.2. HIPÓTESIS

El Desarrollo de Software influye significativamente en el tiempo de cálculo de canales abiertos de flujo uniforme.

2.3. SEÑALAMIENTO DE LAS VARIABLES DE LA HIPÓTESIS

2.3.1. Variable independiente

Desarrollo de Software

2.3.2. Variable dependiente

Cálculo de canales abiertos de flujo uniforme.

CAPÍTULO III

METODOLOGÍA

3.1. NIVEL O TIPO DE INVESTIGACIÓN

Los niveles de investigación que se utilizarán en este proyecto serán: explicativa y aplicada.

Será explicativa debido a que se busca la relación causa-efecto que se genera en el cálculo de canales abiertos de flujo uniforme al utilizar un proceso manual y el uso de un software.

Será aplicada ya que se empleará los conocimientos de hidráulica de canales para desarrollar un software que permita facilitar y disminuir el tiempo de cálculo de canales abiertos de flujo uniforme proporcionando así una herramienta útil para ingenieros y estudiantes.

3.2. POBLACIÓN Y MUESTRA

3.2.1. Población

Canales abiertos

3.2.2. Muestra

Canales abiertos de flujo uniforme

3.3. OPERACIONALIZACIÓN DE VARIABLES

3.3.1. Variable independiente

Desarrollo de Software

Tabla 3: Operacionalización de la variable independiente.

Conceptualización	Dimensiones	Indicadores	Ítems	Técnicas e Instrumentos
El software es una secuencia de instrucciones y procesos que permiten la realización de determinadas tareas al computador.	Lenguaje de programación	<ul style="list-style-type: none"> - Código de programación - Método de cálculo 	¿Cómo aplicar el lenguaje de programación para codificar correctamente el método de cálculo?	Investigación bibliográfica.
	Agilizar el proceso de cálculo	<ul style="list-style-type: none"> - Rapidez de los resultados - Precisión de los resultados 	¿Cómo el software permite agilizar y optimizar el procesos de cálculo?	Investigación bibliográfica.

Fuente: Sebastián Naranjo.

3.3.2. Variable dependiente

Cálculo de canales abiertos de flujo uniforme.

Tabla 4: Operacionalización de la variable dependiente.

Conceptualización	Dimensiones	Indicadores	Ítems	Técnicas e Instrumentos
Un canal abierto es un conducto en el que el agua fluye con una superficie libre impulsada por la presión atmosférica y su propio peso. El flujo uniforme se presenta cuando la velocidad media permanece constante en cualquier sección del canal, esto implica que la sección transversal y el tirante permanecen constantes.	Elementos geométricos	<ul style="list-style-type: none"> - Área hidráulica - Perímetro mojado - Radio hidráulico - Ancho superficial - Profundidad hidráulica - Factor de sección 	¿Cuáles son las ecuaciones para calcular los elementos geométricos?	Investigación bibliográfica.
	Profundidad del flujo o tirante	<ul style="list-style-type: none"> - Tirante normal - Tirante crítico 	¿Cómo se calcula la profundidad del flujo? ¿Cuándo se presenta el tirante normal y el crítico?	Investigación bibliográfica.
	Diseño de canales	<ul style="list-style-type: none"> - Sección hidráulica óptima 	¿Cuáles son las secciones hidráulicas óptimas?	Investigación bibliográfica.
	Propiedades del flujo	<ul style="list-style-type: none"> - Energía específica - Número de Froude - Tipo de flujo 	¿Cómo se determinan las propiedades del flujo?	Investigación bibliográfica

Fuente: Sebastián Naranjo.

3.4. PLAN DE RECOLECCIÓN DE INFORMACIÓN

Tabla 5: Plan de recolección de información.

Preguntas Básicas	Explicación
1. ¿Para qué?	Disminuir el tiempo y facilitar el proceso de cálculo de canales abiertos de flujo uniforme, de tal manera que se puedan obtener resultados óptimos y precisos de una manera rápida.
2. ¿De qué personas u objetos?	De canales abiertos de flujo uniforme.
3. ¿Sobre qué aspectos?	Elementos geométricos, tirante normal, tirante crítico, energía específica, número de Froude, sección óptima y caudal.
4. ¿Quién?	César Sebastián Naranjo Bustos
5. ¿Dónde?	Biblioteca de la Facultad de Ingeniería Civil y Mecánica de la Universidad técnica de Ambato.
6. ¿Cómo?	Investigación Bibliográfica.

Fuente: Sebastián Naranjo.

3.5. PLAN DE PROCESAMIENTO Y ANÁLISIS

Se seguirá un plan estructurado de la siguiente manera:

- 1) Revisión de la bibliografía
- 2) Identificación del lenguaje de programación
- 3) Desarrollo del software
- 4) Resolución de ejercicios de manera manual
- 5) Resolución de ejercicios con un software existente
- 6) Resolución de ejercicios con el software desarrollado
- 7) Análisis y comparación entre resultados
- 8) Verificación de hipótesis

CAPÍTULO IV

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1. RECOLECCIÓN DE DATOS

Los ejercicios a resolver serán desarrollados mediante tres métodos: de forma manual, con un software existente y con el software desarrollado para posteriormente realizar un análisis con los resultados obtenidos.

Para desarrollar los ejercicios de manera manual se utilizarán las definiciones y ecuaciones que se trataron en el capítulo II del presente escrito.

Para desarrollar los ejercicios mediante un software existente se utilizará HCANALES; teniendo presente que el mismo no contiene todas las herramientas para resolver todos los ejercicios planteados.

4.1.1. EJEMPLO PARA EL CÁLCULO DEL TIRANTE NORMAL DE UNA SECCIÓN TRAPEZOIDAL

Calcular el tirante normal, para un canal trapezoidal, considerando que su caudal de diseño es de $0.3 \text{ m}^3/\text{s}$, el coeficiente de rugosidad es $n=0.012$, el ancho de la solera es $b=40 \text{ cm}$, el talud es 1:1.5, la pendiente del canal es del 1%. Determine también la velocidad normal, los elementos geométricos, el tipo de flujo y la energía específica.

Datos:

$$b = 40 \text{ cm} = 0.4 \text{ m}$$

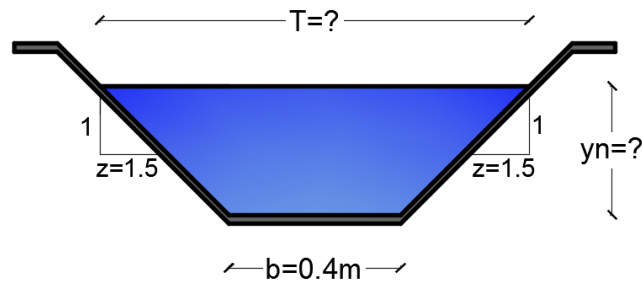
$$Q = 0.3 \text{ m}^3/\text{s}$$

$$n = 0.012$$

$$z = 1.5$$

$$S = 1\% = 0.01$$

Figura 9: Ilustración gráfica del ejemplo para el cálculo del tirante normal de una sección trapezoidal.



Fuente: Sebastián Naranjo.

4.1.1.1. MÉTODO MANUAL

El tirante normal es obtenido mediante un método de iteraciones utilizando la siguiente ecuación:

$$AR^{2/3} = \left(\frac{nQ}{S^{1/2}} \right)$$

Con la cual se debe seguir los siguientes pasos:

- 1) Calcular la relación $\frac{nQ}{S^{1/2}}$
- 2) Proponer un tirante (y)
- 3) Calcular el área hidráulica (A) y el perímetro mojado (P) con las ecuaciones de la tabla 2 dependiendo de la geometría del canal
- 4) Calcular el radio hidráulico con la ecuación 2
- 5) Calcular $AR^{2/3}$
- 6) Verificar si el valor calculado en el punto 5 es igual al valor calculado en el punto 1 y si no es así proponer otro valor del tirante hasta que la igualdad se cumpla.

$$\frac{nQ}{S^{1/2}} = \frac{0.3 * 0.012}{0.01^{1/2}} = 0.036$$

- Iteración 1

$$y = 30\text{cm} = 0.3\text{m}$$

Cálculo del área hidráulica para una sección trapezoidal mediante la siguiente ecuación:

$$A = (b + zy)y$$

$$A = (0.4 + (1.5 * 0.3)) * 0.3$$

$$A = 0.255 \text{ m}^2$$

Cálculo del perímetro mojado para una sección trapezoidal mediante la siguiente ecuación:

$$P = b + 2y\sqrt{1 + z^2}$$

$$P = 0.4 + 2 * (0.3)\sqrt{1 + 1.5^2}$$

$$P = 1.482\text{m}$$

Calculo del radio hidráulico mediante la siguiente ecuación:

$$R = \frac{A}{P}$$

$$R = \frac{0.255}{1.482}$$

$$R = 0.172 \text{ m}$$

Verificación si con los valores obtenidos se cumple la igualdad.

$$R^{\frac{2}{3}} * A = 0.036$$

$$0.172^{\frac{2}{3}} * 0.255 = 0.036$$

$$0.044 \neq 0.036 \therefore \text{NO CUMPLE}$$

- Iteración 2

$$y = 20\text{cm} = 0.2\text{m}$$

$$A = (b + zy)y$$

$$A = (0.4 + (1.5 * 0.2)) * 0.2$$

$$A = 0.140\text{m}^2$$

$$P = b + 2y\sqrt{1 + z^2}$$

$$P = 0.4 + 2 * (0.2) * \sqrt{1 + 1.5^2}$$

$$P = 1.120\text{m}$$

$$R = \frac{A}{P}$$

$$R = \frac{0.140}{1.120}$$

$$R = 0.125\text{m}$$

$$R^{\frac{2}{3}} * A = 0.036$$

$$0.125^{\frac{2}{3}} * 0.140 = 0.036$$

$$0.035 \neq 0.036 \therefore \text{NO CUMPLE}$$

- Iteración 3

$$y = 20.3\text{cm} = 0.203\text{m}$$

$$A = (b + zy)y$$

$$A = (0.4 + (1.5 * 0.203)) * 0.203$$

$$A = 0.143\text{m}^2$$

$$P = b + 2y\sqrt{1 + z^2}$$

$$P = 0.4 + 2 * (0.203) * \sqrt{1 + 1.5^2}$$

$$P = 1.132 \text{ m}$$

$$R = \frac{A}{P}$$

$$R = \frac{0.143}{1.132}$$

$$R = 0.126 \text{ m}$$

$$R^{\frac{2}{3}} * A = 0.036$$

$$0.126^{\frac{2}{3}} * 0.143 = 0.036$$

$$0.036 = 0.036 \therefore \text{CUMPLE}$$

Como la condición cumple, el tirante normal es $y_n = 0.203 \text{ m}$

Cálculo de la velocidad normal mediante la siguiente ecuación:

$$V = \frac{1}{n} * R^{\frac{2}{3}} * S^{\frac{1}{2}}$$

$$V = \frac{1}{0.012} * 0.126^{2/3} * 0.01^{1/2}$$

$$V = 2.09 \text{ m/s}$$

Cálculo del ancho superficial para una sección trapezoidal mediante la siguiente ecuación:

$$T = b + 2zy$$

$$T = 0.4 + 2 * 1.5 * 0.203$$

$$T = 1.009 \text{ m}$$

Cálculo de la profundidad hidráulica mediante la siguiente ecuación:

$$D = \frac{A}{T}$$
$$D = \frac{0.143}{1.009}$$
$$D = 0.1417 \text{ m}$$

Cálculo del número de Froude mediante la siguiente ecuación:

$$NF = \frac{V}{\sqrt{g D}}$$
$$NF = \frac{2.09}{\sqrt{9.81 * 0.1417}}$$
$$NF = 1.7727$$

Como el número de Froude es mayor a 1 el flujo es supercrítico.

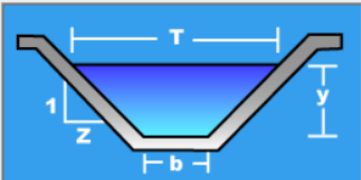
Cálculo de la energía específica mediante la siguiente ecuación:

$$E = y + \frac{Q^2}{2gA^2}$$
$$E = 0.203 + \frac{0.3^2}{2 * 9.81 * 0.143^2}$$
$$E = 0.4273 \text{ m} - Kg/Kg$$

4.1.1.2. MÉTODO CON UN SOFTWARE EXISTENTE

Cálculo del tirante normal de una sección trapezoidal mediante el programa HCANALES.

Datos:	
Caudal (Q):	<input type="text" value="0.3"/> m ³ /s
Ancho de solera (b):	<input type="text" value="0.4"/> m
Talud (Z):	<input type="text" value="1.5"/>
Rugosidad (n):	<input type="text" value="0.012"/>
Pendiente (S):	<input type="text" value="0.01"/> m/m



Resultados:			
Tirante normal (y):	<input type="text" value="0.2030"/> m	Perímetro (p):	<input type="text" value="1.1318"/> m
Área hidráulica (A):	<input type="text" value="0.1430"/> m ²	Radio hidráulico (R):	<input type="text" value="0.1263"/> m
Espejo de agua (T):	<input type="text" value="1.0089"/> m	Velocidad (v):	<input type="text" value="2.0981"/> m/s
Número de Froude (F):	<input type="text" value="1.7794"/>	Energía específica (E):	<input type="text" value="0.4273"/> m-Kg/Kg
Tipo de flujo:	<input type="text" value="Supercrítico"/>		

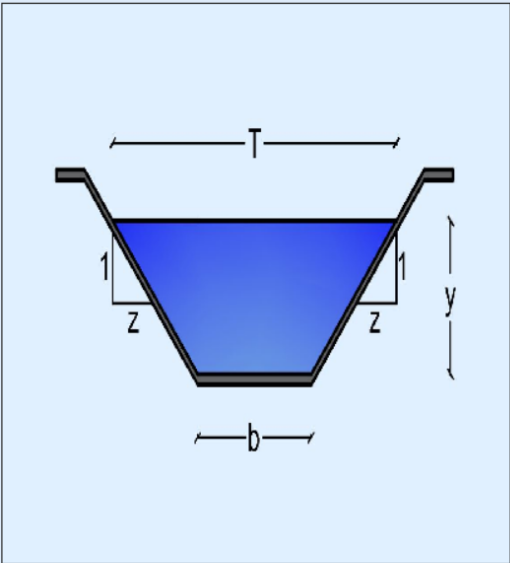
4.1.1.3. MÉTODO CON EL SOFTWARE DESARROLLADO

Cálculo del tirante normal de una sección trapezoidal mediante el software desarrollado.

CÁLCULO DEL TIRANTE NORMAL DE UNA SECCIÓN TRAPEZOIDAL

Datos de entrada:	
Caudal (Q)=	<input type="text" value="0.3"/> m ³ /s
Ancho de la solera (b)=	<input type="text" value="0.4"/> m
Talud (z)=	<input type="text" value="1.5"/>
Pendiente (S)=	<input type="text" value="0.01"/> m/m
Coficiente de Manning (n)=	<input type="text" value="0.012"/>

Datos de salida:	
Tirante normal (y)=	<input type="text" value="0.203"/> m
Área mojada (A)=	<input type="text" value="0.143"/> m ²
Perímetro mojado (P)=	<input type="text" value="1.1318"/> m
Radio hidráulico (R)=	<input type="text" value="0.1263"/> m
Ancho superficial (T)=	<input type="text" value="1.0089"/> m
Profundidad hidráulica (D)=	<input type="text" value="0.1417"/> m
Velocidad (V)=	<input type="text" value="2.0981"/> m/s
Número de Froude (NF)=	<input type="text" value="1.7794"/>
Tipo de flujo=	<input type="text" value="Supercrítico"/>
Energía específica (E)=	<input type="text" value="0.4273"/> m-Kg/Kg



Sistema de unidades:
 Sistema métrico
 Sistema inglés

4.1.2. EJEMPLO PARA EL CÁLCULO DEL TIRANTE CRÍTICO DE UNA SECCIÓN TRAPEZOIDAL.

Calcular el tirante crítico para un canal trapezoidal, considerando que su caudal de diseño es de $0.3\text{m}^3/\text{s}$, el ancho de la solera es $b=40\text{cm}$, el talud es 1:1.5. Determine también la velocidad crítica, los elementos geométricos, el tipo de flujo y la energía específica.

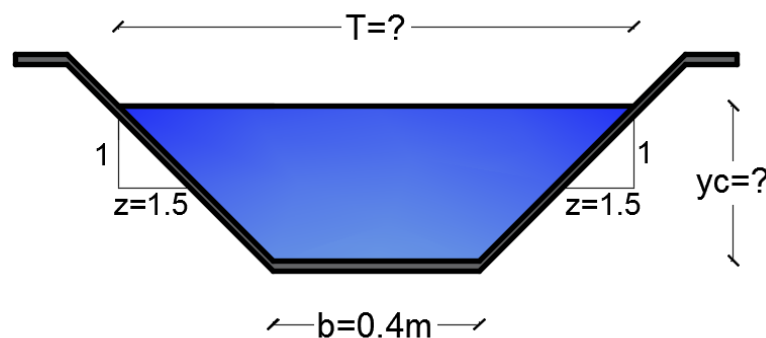
Datos:

$$b = 40 \text{ cm} = 0.4\text{m}$$

$$Q = 0.3\text{m}^3/\text{s}$$

$$z = 1.5$$

Figura 10: Ilustración gráfica del ejemplo para el cálculo del tirante crítico de una sección trapezoidal.



Fuente: Sebastián Naranjo.

4.1.2.1. MÉTODO MANUAL

El tirante crítico es obtenido mediante un método de iteraciones utilizando la siguiente ecuación:

$$\frac{Q^2}{g} = \frac{A^3}{T}$$

Con la cual se debe seguir los siguientes pasos:

- 1) Calcular la relación $\frac{Q^2}{g}$
- 2) Proponer un tirante (y)
- 3) Calcular el área hidráulica (A) y el ancho superficial (T) con las ecuaciones de la tabla 2 dependiendo de la geometría del canal
- 4) Calcular la relación $\frac{A^3}{T}$
- 5) Verificar si el valor calculado en el punto 4 es igual al valor calculado en el punto 1 y si no es así proponer otro valor del tirante hasta que la igualdad se cumpla.

$$\frac{Q^2}{g} = \frac{0.3^2}{9.81} = 0.009174$$

- Iteración 1

$$y = 25\text{cm} = 0.25\text{m}$$

Cálculo del área hidráulica para una sección trapezoidal mediante la siguiente ecuación:

$$\begin{aligned} A &= (b + zy)y \\ A &= (0.4 + (1.5 * 0.25)) * 0.25 \\ A &= 1.15 \text{ m}^2 \end{aligned}$$

Cálculo del ancho superficial para una sección trapezoidal mediante la siguiente ecuación:

$$\begin{aligned} T &= b + 2zy \\ T &= 0.40 + 2(1.5 * 0.25) \\ T &= 1.15 \text{ m} \end{aligned}$$

Verificación si con los valores obtenidos se cumple la igualdad.

$$\frac{A^3}{T} = 0.0091$$

$$\frac{0.193^3}{1.15} = 0.0092$$

$$0.0063 \neq 0.0092 \therefore \text{NO CUMPLE}$$

- Iteración 2

$$y = 27.6 \text{ cm} = 0.276 \text{ m}$$

$$A = (b + zy)y$$

$$A = (0.4 + (1.5 * 0.276)) * 0.276$$

$$A = 0.2247 \text{ m}^2$$

$$T = b + 2zy$$

$$T = 0.40 \text{ m} + 2(1.5 * 0.276 \text{ m})$$

$$T = 1.228 \text{ m}$$

$$\frac{A^3}{T} = 0.0091$$

$$\frac{0.2247^3}{1.228} = 0.0092$$

$$0.0092 \neq 0.0092 \therefore \text{CUMPLE}$$

Como la condición cumple, el tirante crítico es $y_c = 0.276 \text{ m}$

Cálculode la velocidad crítica mediante la siguiente ecuación:

$$V = Q/A$$

$$V = \frac{0.3}{0.2247}$$

$$V = 1.34 \text{ m/s}$$

Cálculo de la profundidad hidráulica mediante la siguiente ecuación:

$$D = \frac{A}{T}$$
$$D = \frac{0.2247}{1.228}$$
$$D = 0.183 \text{ m}$$

Cálculo del perímetro mojado para una sección trapezoidal mediante la siguiente ecuación:

$$P = b + 2y\sqrt{1 + z^2}$$
$$P = 0.4 + 2 * (0.276)\sqrt{1 + 1.5^2}$$
$$P = 1.3951 \text{ m}$$

Calculo del radio hidráulico mediante la siguiente ecuación:

$$R = \frac{A}{P}$$
$$R = \frac{0.2247}{1.3951}$$
$$R = 0.1611 \text{ m}$$

Cálculo del número de Froude mediante la siguiente ecuación:

$$NF = \frac{V}{\sqrt{g D}}$$
$$NF = \frac{1.34}{\sqrt{9.81 * 0.183}}$$
$$NF = 1$$

Como el número de Froude es igual a 1 el flujo es Crítico.

Cálculo de la energía específica mediante la siguiente ecuación:

$$E = y + \frac{Q^2}{2gA^2}$$

$$E = 0.276 + \frac{0.3^2}{2 * 9.81 * 0.2247^2}$$

$$E = 0.3669 \text{ m} - \text{Kg/Kg}$$

4.1.2.2. MÉTODO CON UN SOFTWARE EXISTENTE

Cálculo del tirante crítico de una sección trapezoidal mediante el programa HCANALES.

Datos:					
Caudal (Q):	<input type="text" value="0.3"/>	m3/s			
Ancho de solera (b):	<input type="text" value="0.4"/>	m			
Talud (Z):	<input type="text" value="1.5"/>				
Resultados:					
Tirante crítico (y):	<input type="text" value="0.2756"/>	m	Perímetro (p):	<input type="text" value="1.3938"/>	m
Área hidráulica (A):	<input type="text" value="0.2242"/>	m ²	Radio hidráulico (R):	<input type="text" value="0.1609"/>	m
Espejo de agua (T):	<input type="text" value="1.2269"/>	m	Velocidad (v):	<input type="text" value="1.3381"/>	m/s
Número de Froude (F):	<input type="text" value="1.0000"/>		Energía específica (E):	<input type="text" value="0.3669"/>	m-Kg/Kg

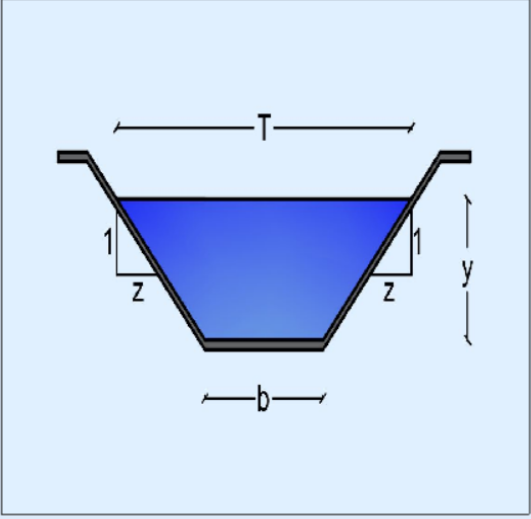
4.1.2.3. MÉTODO CON EL SOFTWARE DESARROLLADO

Cálculo del tirante crítico de una sección trapezoidal mediante el software desarrollado.

CÁLCULO DEL TIRANTE CRÍTICO DE UNA SECCIÓN TRAPEZOIDAL

Datos de entrada:		
Caudal (Q)=	<input type="text" value="0.3"/>	m³/s
Ancho de la solera (b)=	<input type="text" value="0.4"/>	m
Talud (z)=	<input type="text" value="1.5"/>	

Datos de salida:		
Tirante crítico (y)=	<input type="text" value="0.2755"/>	m
Área mojada (A)=	<input type="text" value="0.2241"/>	m²
Perímetro mojado (P)=	<input type="text" value="1.3935"/>	m
Radio hidráulico (R)=	<input type="text" value="0.1608"/>	m
Ancho superficial (T)=	<input type="text" value="1.2266"/>	m
Profundidad hidráulica (D)=	<input type="text" value="0.1827"/>	m
Velocidad (V)=	<input type="text" value="1.3387"/>	m/s
Número de Froude (NF)=	<input type="text" value="1"/>	
Tipo de flujo=	<input type="text" value="Crítico"/>	
Energía específica (E)=	<input type="text" value="0.3669"/>	m-Kg/Kg



Sistema de unidades:
 Sistema métrico
 Sistema inglés

4.1.3. EJEMPLO PARA EL CÁLCULO DEL CAUDAL DE UNA SECCIÓN CIRCULAR

¿Cuál es el caudal de agua en una tubería de alcantarillado con un coeficiente de rugosidad $n=0.013$, 61 cm de diámetro, estando la tubería semi llena y teniendo una pendiente de 0.0025 ? Determine también los elementos geométricos, la velocidad del flujo, el tipo de flujo y la energía específica.

Datos:

$$d = 61 \text{ cm} = 0.61 \text{ m}$$

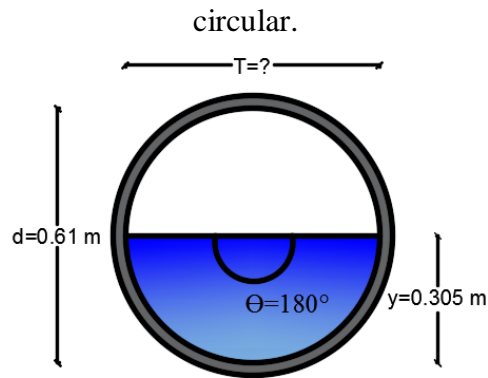
$$y = 30.5 \text{ cm} = 0.305 \text{ m}$$

$$\theta = 180^\circ = \pi \text{ rad}$$

$$S = 0.0025$$

$$n = 0.013$$

Figura 11: Ilustración gráfica del ejemplo para el cálculo del caudal de una sección circular.



Fuente: Sebastián Naranjo.

4.1.3.1. MÉTODO MANUAL

Cálculo del área hidráulica para una sección circular mediante la siguiente ecuación:

$$A = \frac{1}{8}(\theta - \text{sen}\theta)d^2$$

$$A = \frac{1}{8}(\pi - \text{sen } 180) * 0.61^2$$

$$A = 0.146\text{ m}^2$$

Cálculo del perímetro mojado para una sección circular mediante la siguiente ecuación:

$$P = \frac{1}{2}\theta d$$

$$P = \frac{1}{2} * \pi * 0.61$$

$$P = 0.958$$

Cálculo del radio hidráulico mediante la siguiente ecuación:

$$R = \frac{A}{P}$$

$$R = \frac{0.146}{0.958}$$

$$R = 0.152\text{ m}$$

Cálculo del caudal mediante la siguiente ecuación:

$$Q = \frac{1}{n} AR^{2/3} S^{1/2}$$
$$Q = \frac{1}{0.013} * 0.146 * 0.152^{2/3} * 0.0025^{1/2}$$
$$Q = 0.16 \text{ m}^3/\text{s}$$

Cálculo del ancho superficial para una sección circular mediante la siguiente ecuación:

$$T = (\text{sen } \theta/2)d$$
$$T = (\text{sen } \pi/2) * 0.61$$
$$T = 0.61 \text{ m}$$

Cálculo de la profundidad hidráulica mediante la siguiente ecuación:

$$D = \frac{A}{T}$$
$$D = \frac{0.146}{0.61}$$
$$D = 0.2393 \text{ m}$$

Cálculo de la velocidad mediante la siguiente ecuación:

$$V = \frac{Q}{A}$$
$$V = \frac{0.16}{0.146}$$
$$V = 1.0959 \text{ m/s}$$

Cálculo del número de Froude mediante la siguiente ecuación:

$$NF = \frac{V}{\sqrt{gD}}$$

$$NF = \frac{1.0959}{\sqrt{9.81 * 0.2393}}$$

$$NF = 0.7153$$

Como el número de Froude es menor a 1 el flujo es subcrítico.

Cálculo de la energía específica mediante la siguiente ecuación:

$$E = y + \frac{Q^2}{2gA^2}$$

$$E = 0.305 + \frac{0.16^2}{2 * 9.81 * 0.146^2}$$

$$E = 0.3662 \text{ m} - \text{Kg/Kg}$$

4.1.3.2. MÉTODO CON UN SOFTWARE EXISTENTE

Cálculo del caudal de una sección circular mediante el programa HCANALES.

Datos:			
Tirante (y):	<input type="text" value="0.305"/>	m	
Diámetro (d):	<input type="text" value="0.61"/>	m	
Rugosidad (n):	<input type="text" value="0.013"/>		
Pendiente (S):	<input type="text" value="0.0025"/>	m/m	



Resultados:			
Caudal (Q):	<input type="text" value="0.1604"/>	m3/s	Velocidad (v): <input type="text" value="1.0978"/> m/s
Area hidráulica (A):	<input type="text" value="0.1461"/>	m2	Perímetro mojado (p): <input type="text" value="0.9582"/> m
Radio hidráulico (R):	<input type="text" value="0.1525"/>	m	Espejo de agua (T): <input type="text" value="0.6100"/> m
Número de Froude (F):	<input type="text" value="0.7162"/>		Energía específica (E): <input type="text" value="0.3664"/> m-Kg/Kg
Tipo de flujo:	<input type="text" value="Subcrítico"/>		

4.1.3.3. MÉTODO CON EL SOFTWARE DESARROLLADO

Cálculo del caudal de una sección circular mediante el software desarrollado.

CÁLCULO DEL CAUDAL DE UNA SECCIÓN CIRCULAR

Datos de entrada:		
Tirante (y)=	<input type="text" value="0.305"/>	m
Diámetro (d)=	<input type="text" value="0.61"/>	m
Pendiente (S)=	<input type="text" value="0.0025"/>	m/m
Coficiente de Manning (n)=	<input type="text" value="0.013"/>	

Datos de salida:		
Caudal (Q)=	<input type="text" value="0.1604"/>	m ³ /s
Área mojada (A)=	<input type="text" value="0.1461"/>	m ²
Perímetro mojado (P)=	<input type="text" value="0.9582"/>	m
Radio hidráulico (R)=	<input type="text" value="0.1525"/>	m
Ancho superficial (T)=	<input type="text" value="0.61"/>	m
Profundidad hidráulica (D)=	<input type="text" value="0.2395"/>	m
Velocidad (V)=	<input type="text" value="1.0978"/>	m/s
Número de Froude (NF)=	<input type="text" value="0.7162"/>	
Tipo de flujo=	<input type="text" value="Subcrítico"/>	
Energía específica (E)=	<input type="text" value="0.3664"/>	m-Kg/Kg

Sistema de unidades:
 Sistema métrico
 Sistema inglés

4.1.4. EJEMPLO PARA EL CÁLCULO DE LOS ELEMENTOS GEOMÉTRICOS DE UN CANAL TRIANGULAR CON FONDO REDONDEADO

Calcular los elementos del canal triangular con fondo redondeado considerando un tirante de 7m, un radio de 4m y la relación del talud 1:1.

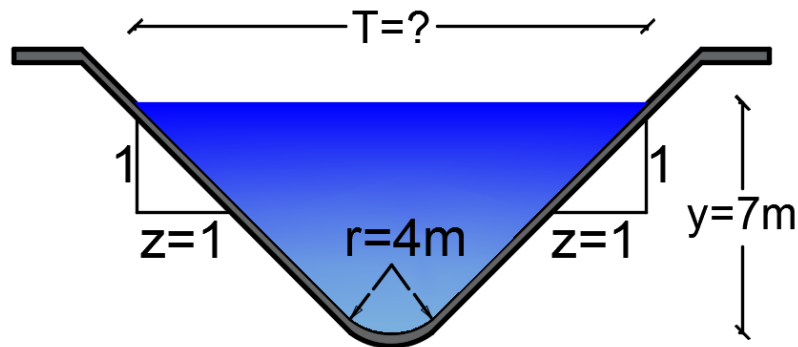
Datos:

$$y = 7\text{m}$$

$$r = 4\text{m}$$

$$z = 1$$

Figura 12: Ilustración gráfica del ejemplo para el cálculo de los elementos geométricos de una sección circular.



Fuente: Sebastián Naranjo.

4.1.4.1. MÉTODO MANUAL

Cálculo del ancho superficial para una sección triangular con fondo redondeado mediante la siguiente ecuación:

$$T = 2 * (z * (y - r) + r * \sqrt{1 + z^2})$$

$$T = 2 * (1 * (7 - r) + 4 * \sqrt{1 + 1^2})$$

$$T = 17,314 \text{ m}$$

Cálculo del área hidráulica para una sección triangular con fondo redondeado mediante la siguiente ecuación:

$$A = \frac{T^2}{4 * z} - \frac{r^2}{z} * (1 - \cot^{-1} z)$$

$$A = \frac{17,314^2}{4 * 1} - \frac{4^2}{1} * (1 - \cot^{-1} 1)$$

$$A = 71,504 \text{ m}^2$$

Cálculo del perímetro mojado para una sección triangular con fondo redondeado mediante la siguiente ecuación:

$$P = \frac{T}{z} * \sqrt{1 + z^2} - \frac{2 * r}{z} * (1 - \cot^{-1}z)$$

$$P = \frac{17,314}{1} * \sqrt{1 + 1^2} - \frac{2 * 4}{1} * (1 - \cot^{-1}1)$$

$$P = 22,763 \text{ m}$$

Cálculo del radio hidráulico mediante la siguiente ecuación:

$$R = \frac{A}{P}$$

$$R = \frac{71,504}{22,763}$$

$$R = 3,141 \text{ m}$$

Cálculo de la profundidad hidráulica mediante la siguiente ecuación:

$$D = \frac{A}{T}$$

$$D = \frac{71,504}{17,314}$$

$$D = 4,129 \text{ m}$$

Cálculo del factor de sección mediante la siguiente ecuación:

$$Z = \sqrt{\frac{A}{T}}$$

$$Z = \sqrt{\frac{71,504}{17,314}}$$

$$Z = 2.032 \text{ m}$$

4.1.4.2. MÉTODO CON EL SOFTWARE DESARROLLADO

Cálculo de los elementos geométricos de una sección triangular con fondo redondeado mediante el software desarrollado.

ELEMENTOS GEOMÉTRICOS DE UNA SECCIÓN TRIANGULAR CON FONDO REDONDEADO

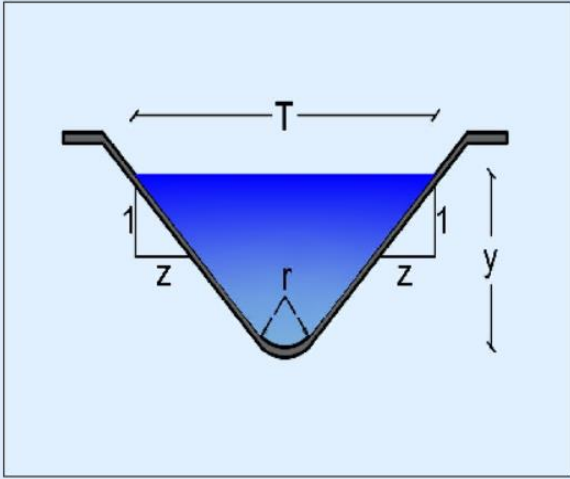
Datos de entrada:

Tirante (y)=	<input type="text" value="7"/>	m
Radio (r)=	<input type="text" value="4"/>	m
Talud (z)=	<input type="text" value="1"/>	

CALCULAR LIMPIAR

Datos de salida:

Área mojada (A)=	<input type="text" value="71.5075"/>	m ²
Perímetro mojado (P)=	<input type="text" value="22.7685"/>	m
Radio hidráulico (R)=	<input type="text" value="3.1406"/>	m
Ancho superficial (T)=	<input type="text" value="17.3137"/>	m
Profundidad hidráulica (D)=	<input type="text" value="4.1301"/>	m
Factor de sección (Z)=	<input type="text" value="2.0323"/>	m



MENÚ PRINCIPAL

4.1.5. EJEMPLO PARA EL CÁLCULO DE LA SECCIÓN HIDRÁULICA ÓPTIMA

Una acequia desagua $1.19 \text{ m}^3/\text{s}$ con una pendiente de 0.50 m sobre 1000 m . La sección es rectangular y el coeficiente de rugosidad $n = 0.012$. Determinar las dimensiones óptimas, es decir, las dimensiones que dan el menor perímetro mojado, además la velocidad del flujo, los elementos geométricos, el tipo de flujo y la energía específica.

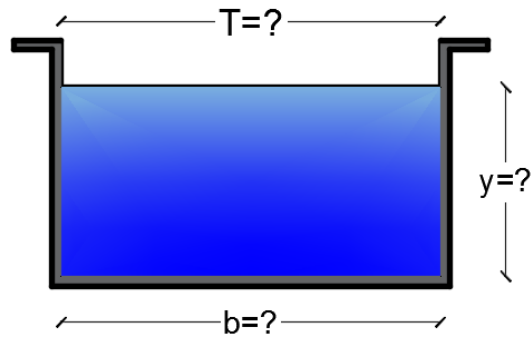
Datos:

$$Q = 1.19 \text{ m}^3/\text{s}$$

$$S = \frac{0.5 \text{ m}}{1000 \text{ m}} = 0.0005$$

$$n = 0.012$$

Figura 13: Ilustración gráfica del ejemplo para el cálculo de los elementos geométricos de una sección circular.



Fuente: Sebastián Naranjo.

4.1.5.1. MÉTODO MANUAL

Para una sección rectangular sea óptima la base debe ser igual a dos veces el tirante.

$$b=2y$$

Cálculo del tirante mediante la siguiente ecuación:

$$Q = \frac{1}{n} * A * R^{\frac{2}{3}} * S^{\frac{1}{2}}$$

$$Q = \frac{1}{n} * (b * y) * \left(\frac{b * y}{b + 2y} \right)^{\frac{2}{3}} * S^{\frac{1}{2}}$$

$$1.19 = \frac{1}{0.012} (2y * y) \left(\frac{2y * y}{2y + 2y} \right)^{\frac{2}{3}} (0.0005)^{\frac{1}{2}}$$

$$0.319 = y^2 \left(\frac{2y^2}{4y} \right)^{\frac{2}{3}}$$

$$0.319 = y^2 \left(\frac{y}{2} \right)^{\frac{2}{3}}$$

$$0.319 * 2^{\frac{2}{3}} = y^2 * y^{\frac{2}{3}}$$

$$(0.506)^{3/8} = (y^{8/3})^{3/8}$$

$$y = 0.7746 \text{ m}$$

Cálculo del ancho de la solera mediante la siguiente ecuación.

$$b=2y$$

$$b=2*0.7746$$

$$b=1.5492 \text{ m}$$

Cálculo del área hidráulica para una sección rectangular mediante la siguiente ecuación:

$$A = by$$

$$A = 1.5492 * 0.7746$$

$$A = 1.20 \text{ m}^2$$

Cálculo del perímetro mojado para una sección rectangular mediante la siguiente ecuación:

$$P = b + 2y$$

$$P = 1.5492 + 2 * 0.7746$$

$$P = 3.0984 \text{ m}$$

Cálculo del radio hidráulico mediante la siguiente ecuación:

$$R = \frac{A}{P}$$

$$R = \frac{1.2}{3.0984}$$

$$R = 0.3873 \text{ m}$$

Cálculo del ancho superficial para una sección rectangular mediante la siguiente ecuación:

$$T = b$$

$$T = 1.5492 \text{ m}$$

Cálculo de la profundidad hidráulica mediante la siguiente ecuación:

$$D = \frac{A}{T}$$
$$D = \frac{1.2}{1.5492}$$
$$D = 0.7746 \text{ m}$$

Cálculo de la velocidad mediante la siguiente ecuación:

$$V = \frac{Q}{A}$$
$$V = \frac{1.19}{1.20}$$
$$V = 0.9917 \text{ m/s}$$

Cálculo del número de Froude mediante la siguiente ecuación:

$$NF = \frac{V}{\sqrt{gD}}$$
$$NF = \frac{0.9917}{\sqrt{9.81 * 0.7746}}$$
$$NF = 0.3598$$

Como el número de Froude es menor a 1 el flujo es subcrítico.

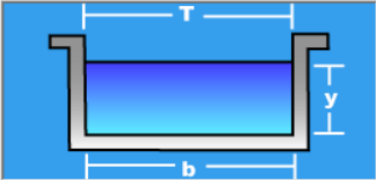
Cálculo de la energía específica mediante la siguiente ecuación:

$$E = y + \frac{Q^2}{2gA^2}$$
$$E = 0.7746 + \frac{1.19^2}{2 * 9.81 * 1.2^2}$$
$$E = 0.8247 \text{ m} - \text{Kg/Kg}$$

4.1.5.2. MÉTODO CON UN SOFTWARE EXISTENTE

Cálculo de la sección hidráulica óptima de una sección rectangular mediante el programa HCANALES.

Datos:	
Caudal (Q):	1.19 m ³ /s
Talud (Z):	0
Rugosidad (n):	0.012
Pendiente (S):	0.0005 m/m



Resultados:			
Tirante (y):	0.7751 m	Ancho de solera (b):	1.5501 m
Perímetro (p):	3.1003 m	Área hidráulica (A):	1.2014 m ²
Radio hidráulico (R):	0.3875 m	Espejo de agua (T):	1.5501 m
Velocidad (v):	0.9905 m/s	Número de Froude (F):	0.3592
Energía específica (E):	0.8251 m-Kg/Kg	Tipo de flujo:	Subcrítico

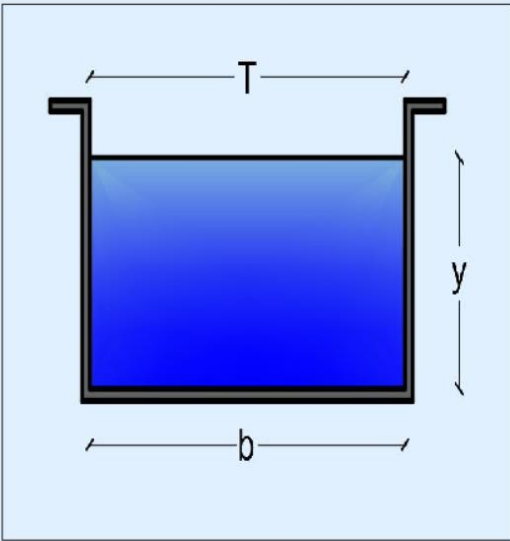
4.1.5.3. MÉTODO CON EL SOFTWARE DESARROLLADO

Cálculo de la sección hidráulica óptima de una sección rectangular mediante el software desarrollado.

SECCIÓN HIDRÁULICA ÓPTIMA DE UNA SECCIÓN RECTANGULAR

Datos de entrada:	
Caudal (Q)=	1.19 m ³ /s
Pendiente (S)=	0.0005 m/m
Coefficiente de Manning (n)=	0.012

Datos de salida:	
Ancho de la solera (b)=	1.5501 m
Tirante normal (y)=	0.7751 m
Área mojada (A)=	1.2014 m ²
Perímetro mojado (P)=	3.1003 m
Radio hidráulico (R)=	0.3875 m
Ancho superficial (T)=	1.5501 m
Profundidad hidráulica (D)=	0.7751 m
Velocidad (V)=	0.9905 m/s
Número de Froude (NF)=	0.3592
Tipo de flujo=	Subcrítico
Energía específica (E)=	0.8251 m-Kg/Kg



Sistema de unidades:
 Sistema métrico
 Sistema inglés

4.2. ANÁLISIS DE RESULTADOS

Tabla 6: Comparación de resultados en el cálculo del tirante normal de una sección trapezoidal.

	MÉTODO			UNIDADES	DIFERENCIA CON SOFTWARE DESARROLLADO			
	MANUAL	H CANALES	SOFTWARE DESARROLLADO		CANTIDAD		%	
					MANUAL	HCANALES	MANUAL	HCANALES
Tirante normal	0.203	0.203	0.203	m	0	0	0.00%	0.00%
Área hidráulica	0.143	0.143	0.143	m ²	0	0	0.00%	0.00%
Perímetro mojada	1.132	1.1318	1.1318	m	0.0002	0	0.02%	0.00%
Radio hidráulico	0.126	0.1263	0.1263	m	0.0003	0	0.24%	0.00%
Ancho superficial	1.009	1.0089	1.0089	m	0.0001	0	0.01%	0.00%
Profundidad hidráulica	0.1417	-	0.1417	m	0	-	0.00%	-
Velocidad normal	2.09	2.0981	2.0981	m/s	0.0081	0	0.39%	0.00%
Número de Froude	1.7727	1.7794	1.7794		0.0067	0	0.38%	0.00%
Tipo de flujo	Supercrítico	Supercrítico	Supercrítico		0	0	0.00%	0.00%
Energía específica	0.4273	0.4273	0.4273	m-Kg/Kg	0	0	0.00%	0.00%

Fuente: Sebastián Naranjo

Tabla 7: Comparación de resultados en el cálculo del tirante crítico de una sección trapezoidal.

	MÉTODO			UNIDADES	DIFERENCIA CON SOFTWARE DESARROLLADO			
	MANUAL	H CANALES	SOFTWARE DESARROLLADO		CANTIDAD		%	
					MANUAL	HCANALES	MANUAL	HCANALES
Tirante crítico	0.276	0.2756	0.2755	m	0.0005	0.0001	0.18%	0.04%
Área hidráulica	0.2247	0.2242	0.2241	m ²	0.0006	0.0001	0.27%	0.04%
Perímetro mojada	1.3951	1.3938	1.3935	m	0.0016	0.0003	0.11%	0.02%
Radio hidráulico	0.1611	0.1609	0.1608	m	0.0003	0.0001	0.19%	0.06%
Ancho superficial	1.228	1.2269	1.2266	m	0.0014	0.0003	0.11%	0.02%
Profundidad hidráulica	0.183	-	0.1827	m	0.0003	-	0.16%	-
Velocidad crítica	1.34	1.3381	1.3387	m/s	0.0013	0.0006	0.10%	0.04%
Número de Froude	1	1	1		0	0	0.00%	0.00%
Tipo de flujo	Crítico	-	Crítico		0	0	0.00%	0.00%
Energía específica	0.3669	0.3669	0.3669	m-Kg/Kg	0	0	0.00%	0.00%

Fuente: Sebastián Naranjo

Tabla 8: Comparación de resultados en el cálculo del caudal de una sección circular.

	MÉTODO			UNIDADES	DIFERENCIA CON SOFTWARE DESARROLLADO			
	MANUAL	H CANALES	SOFTWARE DESARROLLADO		CANTIDAD		%	
					MANUAL	HCANALES	MANUAL	HCANALES
Caudal	0.16	0.1604	0.1604	m	0.0004	0	0.25%	0.00%
Área hidráulica	0.146	0.1461	0.1461	m ²	0.0001	0	0.07%	0.00%
Perímetro mojada	0.958	0.9582	0.9582	m	0.0002	0	0.02%	0.00%
Radio hidráulico	0.152	0.1525	0.1525	m	0.0005	0	0.33%	0.00%
Ancho superficial	0.61	0.61	0.61	m	0	0	0.00%	0.00%
Profundidad hidráulica	0.2393	-	0.2395	m	0.0002	-	0.08%	-
Velocidad	1.0959	1.0978	1.0978	m/s	0.0019	0	0.17%	0.00%
Número de Froude	0.7153	0.7162	0.7162		0.0009	0	0.13%	0.00%
Tipo de flujo	Subcrítico	Subcrítico	Subcrítico		0	0	0.00%	0.00%
Energía específica	0.3662	0.3664	0.3664	m-Kg/Kg	0.0002	0	0.05%	0.00%

Fuente: Sebastián Naranjo

Tabla 9: Comparación de resultados en el cálculo los elementos geométricos de una sección triangular con fondo redondeado.

	MÉTODO		UNIDADES	DIFERENCIA CON SOFTWARE DESARROLLADO	
	MANUAL	SOFTWARE DESARROLLADO		CANTIDAD	
					%
Área hidráulica	71.504	71.5075	m ²	0.0035	0.00%
Perímetro mojada	22.763	22.7685	m	0.0055	0.02%
Radio hidráulico	3.141	3.1406	m	0.0004	0.01%
Ancho superficial	17.314	17.3137	m	0.0003	0.00%
Profundidad hidráulica	4.129	4.1301	m	0.0011	0.03%
Factor de sección	2.032	2.0323	m	0.0003	0.01%

Fuente: Sebastián Naranjo

Tabla 10: Comparación de resultados en el cálculo de la sección hidráulica óptima de una sección rectangular.

	MÉTODO			UNIDADES	DIFERENCIA CON SOFTWARE DESARROLLADO			
	MANUAL	H CANALES	SOFTWARE DESARROLLADO		CANTIDAD		%	
					MANUAL	HCANALES	MANUAL	HCANALES
Ancho de la solera	1.5492	1.5501	1.5501	m	0.0009	0	0.06%	0.00%
Tirante	0.7746	0.7751	0.7751	m	0.0005	0	0.06%	0.00%
Área hidráulica	1.2	1.2014	1.2014	m ²	0.0014	0	0.12%	0.00%
Perímetro mojada	3.0984	3.1003	3.1003	m	0.0019	0	0.06%	0.00%
Radio hidráulico	0.3873	0.3875	0.3875	m	0.0002	0	0.05%	0.00%
Ancho superficial	1.5492	1.5501	1.5501	m	0.0009	0	0.06%	0.00%
Profundidad hidráulica	0.7746	-	0.7751	m	0.0005	-	0.06%	-
Velocidad	0.9917	0.9905	0.9905	m/s	0.0012	0	0.12%	0.00%
Número de Froude	0.3598	0.3592	0.3592		0.0006	0	0.17%	0.00%
Tipo de flujo	Subcrítico	Subcrítico	Subcrítico		0	0	0.00%	0.00%
Energía específica	0.8247	0.8251	0.8251	m-Kg/Kg	0.0004	0	0.05%	0.00%

Fuente: Sebastián Naranjo

La diferencia de los resultados entregados por el software elaborado al ser comparado por los cálculos manuales y los valores entregados por un software existente que en este caso fue HCANALES, se presentan debido al número de decimales con los cuales se trabaja en cada caso, lo cual provoca un error aparente.

4.3. VERIFICACIÓN DE HIPÓTESIS

De acuerdo a la hipótesis planteada anteriormente en esta investigación “El Desarrollo de Software influye significativamente en el tiempo de cálculo de canales abiertos de flujo uniforme”, se determina que efectivamente al disponer de un software se disminuye el tiempo que toman los cálculos de los mismos, a la vez que se tienen valores más exactos que al realizar un cálculo manual, sin tomar en cuenta que al realizar el cálculo de esta forma existe la posibilidad de error.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Los resultados del software desarrollado tienen en error menor al 0.1% comparado con los valores calculados por HCANALES, por lo cual es válida la utilización de éste programa para los diferentes cálculos a realizarse.
- Los resultados del software desarrollado tienen en error menor al 1% comparado con los valores calculados manualmente, por lo cual es válida la utilización de éste programa para los diferentes cálculos a realizarse.
- El error que se presenta entre los cálculos mediante un método manual y un software se deben al número de decimales con los que se trabaja.
- El desarrollo de paquetes informáticos, es indispensables en la actualidad para la resolución de ejercicios como el cálculo del tirante normal, tirante crítico, caudal, seccion hidráulica óptima, elementos geométricos, número de Froude, tipo de flujo y energía específica ya que con estos programas se optimiza de mejor manera el tiempo empleado en los cálculos.
- Mediante la utilización de programas para la resolución de ejercicios se reduce el riesgo de cometer errores en el cálculo que al resolverlos de manera manual.
- Mediante la utilización del software desarrollado se puede realizar el cálculo de la sección transversal del canal de una manera más rápida y eficiente; para posteriormente realizar el diseño respectivo del canal a construirse.

5.2. RECOMENDACIONES

- Se debe comprender el lenguaje de programación antes de comenzar la codificación del programa, debido a que es primordial entender las funciones y los comandos para ordenar adecuadamente la secuencia que debe llevar el proceso de cálculo.
- Se recomienda que en lo posible se utilicen paquetes informáticos al realizar cálculos debido a que se reduce la posibilidad de error, además de que reducen el tiempo que toman los cálculos.
- Para la utilización adecuada del software se requiere un conocimiento básico de la hidráulica de canales, por lo que su utilización debe estar enfocada con criterio ingenieril.
- Se recomienda que se utilice el programa desarrollado llamado SN CANALES para el cálculo y diseño de los canales en las obras que se requiera su construcción.
- Se recomienda que en lo posible al desarrollar paquetes informáticos se utilicen las versiones más recientes de los softwares que permiten su realización ya que contienen una mayor cantidad de comandos.
- Se debe utilizar el manual de usuario descrito en el presente proyecto.
- Se recomienda verificar que se encuentre instalado MATLAB Runtime 9.0.1 o superior antes de ejecutar el software desarrollado.

C. MATERIALES DE REFERENCIA

1. BIBLIOGRAFÍA

- [1] L. Acosta, J. Barragan, y A. Reyes. “Software para el diseño de canales abiertos”. Pregrado, Universidad de la Salle, 2007.
- [2] “Importancia de la programación de computadoras (Desarrollo de Software)”. Internet: <https://pingtonyzamora.wordpress.com/2014/11/24/importancia-de-la-programacion-de-computadoras-desarrollo-de-software/>, [Agos. 26, 2016]
- [3] “Importancia de la Programación (Informática)”. Internet: <http://www.importancia.org/programacion-informatica.php> [Agos. 26, 2016]
- [4] “HCANALES”. Internet: <http://www.tec.ac.cr/sitios/Docencia/agricola/Documents/Hcanales.pdf>, [Agos. 26, 2016]
- [5] M. Choque. “HCANALES”. Internet: http://transversalppd.blogspot.com/2015/12/choque-ramos-m-cesar_57.html [Agos. 26, 2016]
- [6] G. Pérez. “Aplicación del software HCANLES”. Internet: http://biblioteca.uns.edu.pe/saladocentes/archivoz/publicacionez/002_conferencia_dicitada_sobre_apliacion_sobre_hcanales.pdf, Nov. 13, 2012 [Agos. 26, 2016]
- [7] J. Olivares. “5 aplicaciones software de hidráulica para PC”. Internet: <http://www.hidrojing.com/software-de-hidraulica/>, Abr. 08, 2013 [Agos. 26, 2016]
- [8] P. Rodríguez. Hidráulica II. 2008.
- [9] S. Krochin. Diseño hidráulico. Ecuador: Escuela Politécnica Nacional, 1986.
- [10] V. Chow. Hidráulica de Canales Abiertos. Colombia: McGraw-Hill, 1994.
- [11] G. Sotelo. Apuntes de Hidráulica II. México: Universidad Nacional Autónoma de México, 1997.
- [12] R. Mott. Mecánica de fluidos. México: Pearson, 2006.
- [13] J. Evett, R. Giles y C. Liu. Mecánica de los fluidos e hidráulica. España: McGraw-Hill, 1994
- [14] P. Rodríguez. Hidráulica II. 2008.

2. ANEXOS

2.1. CODIFICACIÓN DEL SOFTWARE DESARROLLADO

2.1.1. TIRANTE NORMAL

2.1.1.1. SECCIÓN RECTANGULAR

```
function varargout = TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR_OpeningFcn, ...
                  'gui_OutputFcn',  @TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR is made visible.
function TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR_OpeningFcn(hObject, eventdata,
handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('RECTÁNGULO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
```

```

set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%Obtener el dato del ancho de la solera
VAL2=get(handles.edit2,'String');
b=str2double(VAL2)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
b1=isnan(b);
S1=isnan(S);
n1=isnan(n);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif b<=0
errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif S<=0
errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif b1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
ESPERA
%Cálculo del tirante normal
M=100000
err=0.000001
y=100
for i=1:M
FY=((1/n)*(b*y)*((b*y)/(b+(2*y)))^(2/3))*S^(1/2))-Q
if FY==0
break
else
FDY=(S^(1/2)*b*((b*y)/(b+(2*y)))^(2/3))/n+(2*S^(1/2)*b*y*(b/(b+(2*y))-
(2*b*y)/(b+(2*y)^2))/(3*n*((b*y)/(b+(2*y)))^(1/3))
ER=FY/FDY
abs(ER)
y=y-ER
if abs(ER)<=err
break
end
end
end
end
%Valor de la gravedad en m/s2
g=9.81
%Cálculo del área mojada
A=b*y
%Cálculo del perímetro mojado
P=b+2*y
%Cálculo del ancho superficial

```

```

T=b
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del radio hidráulico
R=A/P
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
y=round(y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif b<=0
    errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif b1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
%Cálculo del tirante normal
M=100000
err=0.000001
y=100
for i=1:M

```



```

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.

```

```

function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados

```

```

set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Mostrar unidades métricas;
set(handles.text6,'Visible','on');
set(handles.text7,'Visible','on');
set(handles.text9,'Visible','on');
set(handles.text19,'Visible','on');
set(handles.text22,'Visible','on');
set(handles.text25,'Visible','on');
set(handles.text26,'Visible','on');
set(handles.text33,'Visible','on');
set(handles.text24,'Visible','on');
set(handles.text23,'Visible','on');
set(handles.text30,'Visible','on');
%Ocultar unidades inglesas;
set(handles.text35,'Visible','off');
set(handles.text36,'Visible','off');
set(handles.text37,'Visible','off');
set(handles.text38,'Visible','off');
set(handles.text39,'Visible','off');
set(handles.text42,'Visible','off');
set(handles.text43,'Visible','off');
set(handles.text45,'Visible','off');
set(handles.text41,'Visible','off');
set(handles.text40,'Visible','off');
set(handles.text44,'Visible','off');
case 'Sistema inglés'
SM1='2'
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Ocultar unidades métricas;
set(handles.text6,'Visible','off');
set(handles.text7,'Visible','off');
set(handles.text9,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text26,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
%Mostrar unidades inglesas;
set(handles.text35,'Visible','on');
set(handles.text36,'Visible','on');
set(handles.text37,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

% --- Executes during object creation, after setting all properties.

```

```
function pushbutton1_CreateFcn(hObject, eventdata, handles)
```

2.1.1.2. SECCIÓN TRAPEZOIDAL

```
function varargout = TIRANTE_NORMAL_DE_UNA_SECCION_TRAPEZOIDAL(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @TIRANTE_NORMAL_DE_UNA_SECCION_TRAPEZOIDAL_OpeningFcn, ...
                  'gui_OutputFcn',   @TIRANTE_NORMAL_DE_UNA_SECCION_TRAPEZOIDAL_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before TIRANTE_NORMAL_DE_UNA_SECCION_TRAPEZOIDAL is made visible.
function TIRANTE_NORMAL_DE_UNA_SECCION_TRAPEZOIDAL_OpeningFcn(hObject, eventdata,
handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('TRAPECIO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = TIRANTE_NORMAL_DE_UNA_SECCION_TRAPEZOIDAL_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%Obtener el dato del ancho de la solera
VAL2=get(handles.edit2,'String');
b=str2double(VAL2)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
```

```

S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%Obtener el dato del talud
VAL6=get(handles.edit16,'String');
z=str2double(VAL6)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
b1=isnan(b);
S1=isnan(S);
n1=isnan(n);
z1=isnan(z);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif b<=0
    errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif b1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Cálculo del tirante normal
    M=100000
    err=0.000001
    y=100
    for i=1:M
        FY=((1/n)*((b+z*y)*y)*((b+z*y)*y)/(b+2*y*sqrt(1+(z^2))))^(2/3))*(S^(1/2))-Q
        if FY==0
            break
        else
            FDY=(S^(1/2)*(b+y*z)*((y*(b+y*z))/(b+2*y*(z^2+1)^(1/2))))^(2/3)/n
            + (S^(1/2)*y*z*((y*(b+y*z))/(b+2*y*(z^2+1)^(1/2))))^(2/3)/n + (2*S^(1/2)*y*(b+y*z)*((b+y*z)/(b+2*y*(z^2+1)^(1/2)))+(y*z)/(b+2*y*(z^2+1)^(1/2))- (2*y*(b+y*z)*(z^2+1)^(1/2))/(b+2*y*(z^2+1)^(1/2)^2))/(3*n*((y*(b+y*z))/(b+2*y*(z^2+1)^(1/2))))^(1/3)
            ER=FY/FDY
            abs(ER)
            y=y-ER
            if abs(ER)<=err
                break
            end
        end
    end
    %Valor de la gravedad en m/s2
    g=9.81
    %Cálculo del área mojada
    A=(b+z*y)*y

```

```

%Cálculo del perímetro mojado
P=b+2*y*sqrt(1+(z^2))
%Cálculo del ancho superficial
T=b+2*z*y
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del radio hidráulico
R=A/P
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
y=round(y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif b<=0
    errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif b1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z1==1

```



```

        set(handles.edit14,'String',E);
    end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

```

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit16,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_NORMAL_DE_UNA_SECCION_TRAPEZOIDAL

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Mostrar unidades métricas;
        set(handles.text6,'Visible','on');
        set(handles.text7,'Visible','on');
        set(handles.text9,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text26,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');
        set(handles.text23,'Visible','on');
        set(handles.text30,'Visible','on');
        %Ocultar unidades inglesas;
        set(handles.text35,'Visible','off');
        set(handles.text36,'Visible','off');
        set(handles.text37,'Visible','off');
        set(handles.text38,'Visible','off');
        set(handles.text39,'Visible','off');
        set(handles.text42,'Visible','off');
        set(handles.text43,'Visible','off');
        set(handles.text45,'Visible','off');
        set(handles.text41,'Visible','off');
        set(handles.text40,'Visible','off');
        set(handles.text44,'Visible','off');
    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Ocultar unidades métricas;
        set(handles.text6,'Visible','off');
        set(handles.text7,'Visible','off');
        set(handles.text9,'Visible','off');
        set(handles.text19,'Visible','off');
        set(handles.text22,'Visible','off');
        set(handles.text25,'Visible','off');
        set(handles.text26,'Visible','off');
        set(handles.text33,'Visible','off');
        set(handles.text24,'Visible','off');
        set(handles.text23,'Visible','off');
        set(handles.text30,'Visible','off');
        %Mostrar unidades inglesas;

```

```

        set(handles.text35,'Visible','on');
        set(handles.text36,'Visible','on');
        set(handles.text37,'Visible','on');
        set(handles.text38,'Visible','on');
        set(handles.text39,'Visible','on');
        set(handles.text42,'Visible','on');
        set(handles.text43,'Visible','on');
        set(handles.text45,'Visible','on');
        set(handles.text41,'Visible','on');
        set(handles.text40,'Visible','on');
        set(handles.text44,'Visible','on');
    end
    set(handles.uibuttongroup1,'Tag',SM1)

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

2.1.1.3. SECCIÓN TRIANGULAR

```

function varargout = TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_OpeningFcn, ...
                  'gui_OutputFcn',  @TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR is made visible.
function TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_OpeningFcn(hObject, eventdata,
handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('TRIÁNGULO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

```

```

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%Obtener el dato del talud
VAL2=get(handles.edit2,'String');
z=str2double(VAL2)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE;
Q1=isnan(Q);
z1=isnan(z);
S1=isnan(S);
n1=isnan(n);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Cálculo del tirante normal
    M=100000
    err=0.000001
    y=100
    for i=1:M
        FY=((1/n)*(z*(y^2))*(((z*(y^2))/(2*y*sqrt(1+(z^2))))^(2/3))*(S^(1/2)))-Q
        if FY==0
            break
        else
            FDY=(2*S^(1/2)*y*z*((y*z)/(2*(z^2 + 1)^(1/2)))^(2/3))/n +
(S^(1/2)*y^2*z^2)/(3*n*(z^2 + 1)^(1/2))*((y*z)/(2*(z^2 + 1)^(1/2)))^(1/3))
            ER=FY/FDY
            abs(ER)
            y=y-ER
            if abs(ER)<=err

```

```

        break
    end
end
end
end
%Valor de la gravedad en m/s2
g=9.81
%Cálculo del área mojada
A=z*(y^2)
%Cálculo del perímetro mojado
P=2*y*sqrt(1+(z^2))
%Cálculo del ancho superficial
T=2*z*y
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del radio hidráulico
R=A/P
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
y=round(y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1

```



```

end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```



```

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');

```

```

end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject, 'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6, 'String', '');
        set(handles.edit9, 'String', '');
        set(handles.edit7, 'String', '');
        set(handles.edit8, 'String', '');
        set(handles.edit15, 'String', '');
        set(handles.edit10, 'String', '');
        set(handles.edit13, 'String', '');
        set(handles.edit11, 'String', '');
        set(handles.edit12, 'String', '');
        set(handles.edit14, 'String', '');
        %Mostrar unidades métricas;
        set(handles.text6, 'Visible', 'on');
        set(handles.text9, 'Visible', 'on');
        set(handles.text19, 'Visible', 'on');
        set(handles.text22, 'Visible', 'on');
        set(handles.text25, 'Visible', 'on');
        set(handles.text26, 'Visible', 'on');
        set(handles.text33, 'Visible', 'on');
        set(handles.text24, 'Visible', 'on');
        set(handles.text23, 'Visible', 'on');
        set(handles.text30, 'Visible', 'on');
        %Ocultar unidades inglesas;
        set(handles.text35, 'Visible', 'off');
        set(handles.text37, 'Visible', 'off');
        set(handles.text38, 'Visible', 'off');
        set(handles.text39, 'Visible', 'off');
        set(handles.text42, 'Visible', 'off');
        set(handles.text43, 'Visible', 'off');
        set(handles.text45, 'Visible', 'off');
        set(handles.text41, 'Visible', 'off');
        set(handles.text40, 'Visible', 'off');
        set(handles.text44, 'Visible', 'off');
    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit6, 'String', '');
        set(handles.edit9, 'String', '');
        set(handles.edit7, 'String', '');
        set(handles.edit8, 'String', '');
        set(handles.edit15, 'String', '');
        set(handles.edit10, 'String', '');
        set(handles.edit13, 'String', '');
        set(handles.edit11, 'String', '');
        set(handles.edit12, 'String', '');
        set(handles.edit14, 'String', '');
        %Ocultar unidades métricas;
        set(handles.text6, 'Visible', 'off');
        set(handles.text9, 'Visible', 'off');
        set(handles.text19, 'Visible', 'off');
        set(handles.text22, 'Visible', 'off');
        set(handles.text25, 'Visible', 'off');
        set(handles.text26, 'Visible', 'off');
        set(handles.text33, 'Visible', 'off');
        set(handles.text24, 'Visible', 'off');
        set(handles.text23, 'Visible', 'off');
        set(handles.text30, 'Visible', 'off');
        %Mostrar unidades inglesas;
        set(handles.text35, 'Visible', 'on');
        set(handles.text37, 'Visible', 'on');
        set(handles.text38, 'Visible', 'on');
        set(handles.text39, 'Visible', 'on');
        set(handles.text42, 'Visible', 'on');
        set(handles.text43, 'Visible', 'on');

```

```

        set(handles.text45,'Visible','on');
        set(handles.text41,'Visible','on');
        set(handles.text40,'Visible','on');
        set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

```

2.1.1.4. SECCIÓN CIRCULAR

```

function varargout = TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_OpeningFcn, ...
                  'gui_OutputFcn',  @TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before TIRANTE NORMAL DE UNA SECCION TRIANGULAR is made visible.
function TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_OpeningFcn(hObject, eventdata,
handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('TRIANGULO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)

```

```

%Obtener el dato del talud
VAL2=get(handles.edit2,'String');
z=str2double(VAL2)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE;
Q1=isnan(Q);
z1=isnan(z);
S1=isnan(S);
n1=isnan(n);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Cálculo del tirante normal
    M=100000
    err=0.000001
    y=100
    for i=1:M
        FY=((1/n)*(z*(y^2))*(((z*(y^2))/(2*y*sqrt(1+(z^2))))^(2/3))*(S^(1/2)))-Q
        if FY==0
            break
        else
            FDY=(2*S^(1/2)*y*z*((y*z)/(2*(z^2 + 1)^(1/2)))^(2/3))/n +
(S^(1/2)*y^2*z^2)/(3*n*(z^2 + 1)^(1/2))*((y*z)/(2*(z^2 + 1)^(1/2)))^(1/3)
            ER=FY/FDY
            abs(ER)
            y=y-ER
            if abs(ER)<=err
                break
            end
        end
    end
    end
    %Valor de la gravedad en m/s2
    g=9.81
    %Cálculo del área mojada
    A=z*(y^2)
    %Cálculo del perimetro mojado
    P=2*y*sqrt(1+(z^2))
    %Cálculo del ancho superficial
    T=2*z*y
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo del radio hidráulico
    R=A/P

```

```

%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
y=round(y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Cálculo del tirante normal
    M=100000
    err=0.000001
    y=100
    for i=1:M
        FY=((1.486/n)*(z*(y^2))*(((z*(y^2))/(2*y*sqrt(1+(z^2))))^(2/3))*(S^(1/2)))-Q
        if FY==0
            break
        else

```



```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');

```



```

set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Mostrar unidades métricas;
set(handles.text6,'Visible','on');
set(handles.text9,'Visible','on');
set(handles.text19,'Visible','on');
set(handles.text22,'Visible','on');
set(handles.text25,'Visible','on');
set(handles.text26,'Visible','on');
set(handles.text33,'Visible','on');
set(handles.text24,'Visible','on');
set(handles.text23,'Visible','on');
set(handles.text30,'Visible','on');
%Ocultar unidades inglesas;
set(handles.text35,'Visible','off');
set(handles.text37,'Visible','off');
set(handles.text38,'Visible','off');
set(handles.text39,'Visible','off');
set(handles.text42,'Visible','off');
set(handles.text43,'Visible','off');
set(handles.text45,'Visible','off');
set(handles.text41,'Visible','off');
set(handles.text40,'Visible','off');
set(handles.text44,'Visible','off');
case 'Sistema inglés'
SM1='2'
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Ocultar unidades métricas;
set(handles.text6,'Visible','off');
set(handles.text9,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text26,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
%Mostrar unidades inglesas;
set(handles.text35,'Visible','on');
set(handles.text37,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

```

2.1.1.5. SECCIÓN PARABÓLICA

2.1.1.6. SECCIÓN PARABÓLICA CON “T” CONOCIDO

```
function varargout = TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA_OpeningFcn, ...
                  'gui_OutputFcn',  @TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA is made visible.
function TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA_OpeningFcn(hObject, eventdata,
handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('PARÁBOLA.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%Obtener el dato del ancho superficial
VAL2=get(handles.edit2,'String');
T=str2double(VAL2)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
```

```

S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
T1=isnan(T);
S1=isnan(S);
n1=isnan(n);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif T<=0
    errordlg('Ingrese el valor del ancho superficial (T) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif T1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    C=(n*Q)/(sqrt(S))
    C1=round(C,4)
    y=2
    %Cálculo del área mojada
    A=(2/3)*T*y
    %Cálculo del perimetro mojado
    x=(4*y)/T
    if ((x<=1) && (x>0))
        P=T+((8/3)*((y^2)/T))
    else
        P=(T/2)*(sqrt(1+x^2)+(1/x)*log(x+sqrt(1+x^2)))
    end
    R=A/P
    F=A*R^(2/3)
    F1=round(F,4)
    while F1~C1
        if F1>C1
            y=(y-y/4)
        else
            y=(y+y/4)
        end
        %Cálculo del área mojada
        A=(2/3)*T*y
        %Cálculo del perimetro mojado
        x=(4*y)/T
        if ((x<=1) && (x>0))
            P=T+((8/3)*((y^2)/T))
        else
            P=(T/2)*(sqrt(1+x^2)+(1/x)*log(x+sqrt(1+x^2)))
        end
        R=A/P
        F=A*R^(2/3)
        F1=round(F,4)
    end
    %Valor de la gravedad en m/s2
    g=9.81

```



```

else
    P=(T/2)*(sqrt(1+x^2)+(1/x)*log(x+sqrt(1+x^2)))
end
R=A/P
F=A*R^(2/3)
F1=round(F,4)
while F1~=C1
    if F1>C1
        y=(y-y/4)
    else
        y=(y+y/4)
    end
    %Cálculo del área mojada
    A=(2/3)*T*y
    %Cálculo del perimetro mojado
    x=(4*y)/T
    if ((x<=1) && (x>0))
        P=T+((8/3)*((y^2)/T))
    else
        P=(T/2)*(sqrt(1+x^2)+(1/x)*log(x+sqrt(1+x^2)))
    end
    R=A/P
    F=A*R^(2/3)
    F1=round(F,4)
end
%Valor de la gravedad en ft/s2
g=32.18504
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
y=round(y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y)
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit15,'String',R);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

```

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

```

```

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados

```

```

set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Mostrar unidades métricas;
set(handles.text6,'Visible','on');
set(handles.text7,'Visible','on');
set(handles.text9,'Visible','on');
set(handles.text19,'Visible','on');
set(handles.text22,'Visible','on');
set(handles.text25,'Visible','on');
set(handles.text33,'Visible','on');
set(handles.text24,'Visible','on');
set(handles.text23,'Visible','on');
set(handles.text30,'Visible','on');
%Ocultar unidades inglesas;
set(handles.text35,'Visible','off');
set(handles.text36,'Visible','off');
set(handles.text37,'Visible','off');
set(handles.text38,'Visible','off');
set(handles.text39,'Visible','off');
set(handles.text42,'Visible','off');
set(handles.text45,'Visible','off');
set(handles.text41,'Visible','off');
set(handles.text40,'Visible','off');
set(handles.text44,'Visible','off');
case 'Sistema inglés'
SM1='2'
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Ocultar unidades métricas;
set(handles.text6,'Visible','off');
set(handles.text7,'Visible','off');
set(handles.text9,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
%Mostrar unidades inglesas;
set(handles.text35,'Visible','on');
set(handles.text36,'Visible','on');
set(handles.text37,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)

```



```

function edit20_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit20_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit21_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit21_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit22_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit22_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit17_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit18_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit19_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit19_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

2.1.1.6.1. SECCIÓN PARABÓLICA CON “T” NO CONOCIDO

```

function varargout = TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA_2(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn',
@TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA_2_OpeningFcn, ...

```

```

        'gui_OutputFcn',
@TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA_2_OutputFcn, ...
        'gui_LayoutFcn', [], ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if narginout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA_2 is made
visible.
function TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA_2_OpeningFcn(hObject, eventdata,
handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('PARÁBOLA.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA_2_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
set(handles.text103,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
S1=isnan(S);
n1=isnan(n);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif S<=0

```

```

    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.', 'Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.', 'Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
else
    ESPERA
    C=(n*Q)/(sqrt(S))
    C1=round(C,4)
    y=2
    T=2*sqrt(y)
    %Cálculo del área mojada
    A=(2/3)*T*y
    %Cálculo del perímetro mojado
    x=(4*y)/T
    if ((x<=1) && (x>0))
        P=T+((8/3)*((y^2)/T))
    else
        P=(T/2)*(sqrt(1+x^2)+(1/x)*log(x+sqrt(1+x^2)))
    end
    R=A/P
    F=A*R^(2/3)
    F1=round(F,4)
    while F1~=C1
        if F1>C1
            y=(y-y/4)
        else
            y=(y+y/4)
        end
        T=2*sqrt(y)
        %Cálculo del área mojada
        A=(2/3)*T*y
        %Cálculo del perímetro mojado
        x=(4*y)/T
        if ((x<=1) && (x>0))
            P=T+((8/3)*((y^2)/T))
        else
            P=(T/2)*(sqrt(1+x^2)+(1/x)*log(x+sqrt(1+x^2)))
        end
        R=A/P
        F=A*R^(2/3)
        F1=round(F,4)
    end
    %Valor de la gravedad en m/s2
    g=9.81
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))
    %Aproximación de resultados a 4 dígitos
    y=round(y,4);
    A=round(A,4);
    P=round(P,4);

```

```

T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit15,'String',R);
set(handles.text103,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    C=(n*Q)/(1.486*sqrt(S))
    C1=round(C,4)
    y=2
    T=2*sqrt(y)
    %Cálculo del área mojada
    A=(2/3)*T*y
    %Cálculo del perimetro mojado
    x=(4*y)/T
    if ((x<=1) && (x>0))
        P=T+((8/3)*((y^2)/T))
    else
        P=(T/2)*(sqrt(1+x^2)+(1/x)*log(x+sqrt(1+x^2)))
    end
    R=A/P
    F=A*R^(2/3)
    F1=round(F,4)
    while F1~C1
        if F1>C1
            y=(y-y/4)
        else
            y=(y+y/4)
        end
        T=2*sqrt(y)
        %Cálculo del área mojada
        A=(2/3)*T*y
        %Cálculo del perimetro mojado
        x=(4*y)/T
        if ((x<=1) && (x>0))
            P=T+((8/3)*((y^2)/T))
        else
            P=(T/2)*(sqrt(1+x^2)+(1/x)*log(x+sqrt(1+x^2)))
        end
        R=A/P
        F=A*R^(2/3)
        F1=round(F,4)
    end
end
end

```

```

end
%Valor de la gravedad en ft/s2
g=32.18504
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
y=round(y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit15,'String',R);
set(handles.text103,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
set(handles.text103,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA_2

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        set(handles.text103,'String','');
        %Mostrar unidades métricas;
        set(handles.text6,'Visible','on');
        set(handles.text9,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');
        set(handles.text23,'Visible','on');
        set(handles.text30,'Visible','on');
        set(handles.text104,'Visible','on');
        %Ocultar unidades inglesas;
        set(handles.text35,'Visible','off');

```

```

set(handles.text37,'Visible','off');
set(handles.text38,'Visible','off');
set(handles.text39,'Visible','off');
set(handles.text42,'Visible','off');
set(handles.text45,'Visible','off');
set(handles.text41,'Visible','off');
set(handles.text40,'Visible','off');
set(handles.text44,'Visible','off');
set(handles.text105,'Visible','off');

case 'Sistema inglés'
SM1='2'
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
set(handles.text103,'String','');
%Ocultar unidades métricas;
set(handles.text6,'Visible','off');
set(handles.text9,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
set(handles.text104,'Visible','off');
%Mostrar unidades inglesas;
set(handles.text35,'Visible','on');
set(handles.text37,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
set(handles.text105,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)

function edit20_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit20_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit21_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit21_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit22_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.

```



```

function edit22_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit17_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit18_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit19_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit19_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

2.1.1.6.2. SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS

```

function varargout = TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA_OpeningFcn, ...
                  'gui_OutputFcn',  @TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA is
made visible.
function TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA_OpeningFcn(hObject,
eventdata, handles, varargin)
%Código para centrar la ventana

```

```

scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('RECTANGULAR_REDONDEADAS.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout =
TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%Obtener el dato del ancho de la solera
VAL2=get(handles.edit2,'String');
b=str2double(VAL2)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%Obtener el dato del radio
VAL6=get(handles.edit16,'String');
r=str2double(VAL6)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
b1=isnan(b);
S1=isnan(S);
n1=isnan(n);
r1=isnan(r);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif b<=0
errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif S<=0
errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1

```

```

        errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif bl==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif r<=0
    errordlg('Ingrese el valor del radio (r) teniendo en cuenta que debe ser mayor
que 0 y menor que el tirante (y).','Mensaje');
elseif r1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Cálculo del tirante normal
    M=100000
    err=0.000001
    y=100
    for i=1:M
        FY=(1/n)*(((pi/2)-2)*(r^2)+((b+2*r)*y))*((((pi/2)-
2)*(r^2)+((b+2*r)*y))/(((pi-2)*r)+b+(2*y)))^(2/3))*(S^(1/2))-Q
        if FY==0
            break
        else
            FDY=(S^(1/2)*(b+2*r)*((r^2*(pi/2-2)+y*(b+2*r))/(b+2*y+r*(pi-
2)))^(2/3))/n+(2*S^(1/2)*(r^2*(pi/2-2)+y*(b+2*r))*((b+2*r)/(b+2*y+r*(pi-
2))-(2*(r^2*(pi/2-2)+y*(b+2*r)))/(b+2*y+r*(pi-
2))^2))/(3*n*((r^2*(pi/2-2)+y*(b+2*r))/(b+2*y+r*(pi-2)))^(1/3))
            ER=FY/FDY
            abs(ER)
            y=y-ER
            if abs(ER)<=err
                break
            end
        end
    end
end
if y>r
    %Valor de la gravedad en m/s2
    g=9.81
    %Cálculo del área mojada
    A=(((pi/2)-2)*(r^2)+((b+2*r)*y))
    %Cálculo del perímetro mojado
    P=(((pi-2)*r)+b+(2*y))
    %Cálculo del ancho superficial
    T=b+2*r
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g));
    %Aproximación de resultados a 4 dígitos
    y=round(y,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);

```

```

R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
else
close ESPERA
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
errordlg('El tirante no puede ser mayor que el valor de r. Disminuya el ancho
de la solera (b), la pendiente (S) o el radio (r).','Mensaje');
end
end
else
%Para unidades del sistema inglés
if Q<=0
errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif b<=0
errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif S<=0
errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif b1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif r<=0
errordlg('Ingrese el valor del radio (r) teniendo en cuenta que debe ser mayor
que 0 y menor que el tirante (y).','Mensaje');
elseif r1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
ESPERA
%Cálculo del tirante normal
M=100000
err=0.000001
y=100
for i=1:M
FY=(1.486/n)*(((pi/2)-2)*(r^2)+((b+2*r)*y))*((((pi/2)-
2)*(r^2)+((b+2*r)*y))/(((pi-2)*r)+b+(2*y))^(2/3))* (S^(1/2))-Q
if FY==0
break
else

```

```

        FDY=(1.486*S^(1/2)*(b + 2*r)*((r^2*(pi/2 - 2) + y*(b + 2*r))/(b + 2*y +
r*(pi - 2)))^(2/3))/n + (0.990666666666666666666666666666666666666666666667*S^(1/2)*(r^2*(pi/2 - 2) +
y*(b + 2*r))*((b + 2*r)/(b + 2*y + r*(pi - 2)) - (2*(r^2*(pi/2 - 2) + y*(b +
2*r)))/(b + 2*y + r*(pi - 2))^2))/(n*((r^2*(pi/2 - 2) + y*(b + 2*r))/(b + 2*y + r*(pi
- 2)))^(1/3))
        ER=FY/FDY
        abs(ER)
        y=y-ER
        if abs(ER)<=err
            break
        end
    end
end
if y>r
    %Valor de la gravedad en ft/s2
    g=32.18504
    %Cálculo del área mojada
    A=((pi/2)-2)*(r^2)+((b+2*r)*y)
    %Cálculo del perímetro mojado
    P=((pi-2)*r)+b+(2*y)
    %Cálculo del ancho superficial
    T=b+2*r
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))
    %Aproximación de resultados a 4 dígitos
    y=round(y,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);
    E=round(E,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit6,'String',y)
    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',V);
    set(handles.edit11,'String',NF);
    set(handles.edit12,'String',TF);
    set(handles.edit14,'String',E);
else
    close ESPERA
    set(handles.edit6,'String','');
    set(handles.edit9,'String','');
    set(handles.edit7,'String','');
    set(handles.edit8,'String','');
    set(handles.edit15,'String','');
    set(handles.edit10,'String','');
    set(handles.edit13,'String','');
    set(handles.edit11,'String','');
    set(handles.edit12,'String','');
    set(handles.edit14,'String','');

```

```

        errordlg('El tirante no puede ser mayor que el radio. Disminuya el ancho de
la solera (b), la pendiente (S) o el radio (r).','Mensaje');
    end
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```

        set(hObject,'BackgroundColor','white');
    end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit16,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.

```

```

function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Mostrar unidades métricas;
        set(handles.text6,'Visible','on');
        set(handles.text7,'Visible','on');
        set(handles.text9,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text26,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');
        set(handles.text23,'Visible','on');
        set(handles.text30,'Visible','on');
        set(handles.text48,'Visible','on');
        %Ocultar unidades inglesas;
        set(handles.text35,'Visible','off');
        set(handles.text36,'Visible','off');
        set(handles.text37,'Visible','off');
        set(handles.text38,'Visible','off');
        set(handles.text39,'Visible','off');
        set(handles.text42,'Visible','off');
        set(handles.text43,'Visible','off');
        set(handles.text45,'Visible','off');
        set(handles.text41,'Visible','off');
        set(handles.text40,'Visible','off');
        set(handles.text44,'Visible','off');
        set(handles.text47,'Visible','off');
    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Ocultar unidades métricas;
        set(handles.text6,'Visible','off');
        set(handles.text7,'Visible','off');
        set(handles.text9,'Visible','off');
        set(handles.text19,'Visible','off');
        set(handles.text22,'Visible','off');
        set(handles.text25,'Visible','off');
        set(handles.text26,'Visible','off');
        set(handles.text33,'Visible','off');
        set(handles.text24,'Visible','off');

```



```

set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
set(handles.text48,'Visible','off');
%Mostrar unidades inglesas;
set(handles.text35,'Visible','on');
set(handles.text36,'Visible','on');
set(handles.text37,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
set(handles.text47,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

2.1.1.7. SECCIÓN TRIANGULAR CON FONDO REDONDEADO

```

function varargout = TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OpeningFcn, ...
                  'gui_OutputFcn',  @TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO is
made visible.
function TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OpeningFcn(hObject,
eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('TRIANGULAR_REDONDEADO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

```

```

% --- Outputs from this function are returned to the command line.
function varargout =
TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%Obtener el dato del radio
VAL2=get(handles.edit2,'String');
r=str2double(VAL2)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%Obtener el dato del talud
VAL6=get(handles.edit16,'String');
z=str2double(VAL6)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
r1=isnan(r);
S1=isnan(S);
n1=isnan(n);
z1=isnan(z);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif r<=0
errordlg('Ingrese el valor del radio (r) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif S<=0
errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif r1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z<=0
errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');

```

```

else
    ESPERA
    %Cálculo del del ancho superficial
    M=100000
    err=0.000001
    y=100
    for i=1:M
        FY=(1/n)*(((y^2)/(4*z))-((r^2)/z)*(1-(z*acot(z))))*(((y^2)/(4*z))-
        (((r^2)/z)*(1-(z*acot(z)))))/(((y/z)*sqrt(1+(z^2)))-((2*r)/z)*(1-
        z*acot(z)))^(2/3))*(S^(1/2))-Q
        if FY==0
            break
        else
            FDY=(S^(1/2)*y*(((z*acot(z) - 1)*r^2)/z + y^2/(4*z))/((y*(z^2 +
            1)^(1/2))/z + (2*r*(z*acot(z) - 1))/z)^(2/3))/(2*n*z) + (2*S^(1/2)*y/(2*z*(y*(z^2
            + 1)^(1/2))/z + (2*r*(z*acot(z) - 1))/z) - ((z^2 + 1)^(1/2)*((z*acot(z) - 1)*r^2)/z
            + y^2/(4*z)))/(z*((y*(z^2 + 1)^(1/2))/z + (2*r*(z*acot(z) - 1))/z)^2)*(((z*acot(z) -
            1)*r^2)/z + y^2/(4*z))/(3*n*((y^2)/(4*z) + (r^2*(z*acot(z) - 1))/z)/((y*(z^2 +
            1)^(1/2))/z + (2*r*(z*acot(z) - 1))/z)^(1/3))
            ER=FY/FDY
            abs(ER)
            y=y-ER
            if abs(ER)<=err
                break
            end
        end
    end
    T=y
    %Valor de la gravedad en m/s2
    g=9.81
    %Cálculo del tirante normal
    w=atan(1/z)
    w1=pi-(2*w)
    B=((2*pi)-(2*w1))/2
    Y2=r*(1-cos(B/2))
    M=2*r*sin(B/2)
    C=(T-M)/2
    Y1=tan(w)*C
    Y=Y1+Y2
    %Cálculo del area mojada
    A=(((T^2)/(4*z))-((r^2)/z)*(1-(z*acot(z))))
    %Cálculo del perímetro mojado
    P=(((T/z)*sqrt(1+(z^2)))-((2*r)/z)*(1-z*acot(z)))
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=Y+((V^2)/(2*g))
    %Aproximación de resultados a 4 dígitos
    Y=round(Y,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);
    E=round(E,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit6,'String',Y);
    set(handles.edit9,'String',A);

```



```

B=((2*pi)-(2*w1))/2
Y2=r*(1-cos(B/2))
M=2*r*sin(B/2)
C=(T-M)/2
Y1=tan(w)*C
Y=Y1+Y2
%Cálculo del area mojada
A=((T^2)/(4*z))-((r^2)/z)*(1-(z*acot(z))))
%Cálculo del perímetro mojado
P=((T/z)*sqrt(1+(z^2)))-((2*r)/z)*(1-z*acot(z)))
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del radio hidráulico
R=A/P
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=Y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
Y=round(Y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',Y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit16,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Mostrar unidades métricas;
        set(handles.text6,'Visible','on');

```

```

set(handles.text7,'Visible','on');
set(handles.text9,'Visible','on');
set(handles.text19,'Visible','on');
set(handles.text22,'Visible','on');
set(handles.text25,'Visible','on');
set(handles.text26,'Visible','on');
set(handles.text33,'Visible','on');
set(handles.text24,'Visible','on');
set(handles.text23,'Visible','on');
set(handles.text30,'Visible','on');
%Ocultar unidades inglesas;
set(handles.text35,'Visible','off');
set(handles.text36,'Visible','off');
set(handles.text37,'Visible','off');
set(handles.text38,'Visible','off');
set(handles.text39,'Visible','off');
set(handles.text42,'Visible','off');
set(handles.text43,'Visible','off');
set(handles.text45,'Visible','off');
set(handles.text41,'Visible','off');
set(handles.text40,'Visible','off');
set(handles.text44,'Visible','off');
case 'Sistema inglés'
SM1='2'
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Ocultar unidades métricas;
set(handles.text6,'Visible','off');
set(handles.text7,'Visible','off');
set(handles.text9,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text26,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
%Mostrar unidades inglesas;
set(handles.text35,'Visible','on');
set(handles.text36,'Visible','on');
set(handles.text37,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```


2.1.2. TIRANTE CRÍTICO

2.1.2.1. SECCIÓN RECTANGULAR

```
function varargout = TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR_OpeningFcn, ...
                  'gui_OutputFcn',  @TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR is made
% visible.
function TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR_OpeningFcn(hObject, eventdata,
handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('RECTANGULO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%Obtener el dato del ancho de la solera
VAL2=get(handles.edit2,'String');
b=str2double(VAL2)
```

```

%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
b1=isnan(b);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif b<=0
    errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif b1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Valor de la gravedad en m/s2
    g=9.81
    %Cálculo del tirante crítico
    M=100000
    err=0.000001
    y=100
    for i=1:M
        FY=((b*y)^3/b)-(Q^2/g)
        if FY==0
            break
        else
            FDY=3*b^2*y^2
            ER=FY/FDY
            abs(ER)
            y=y-ER
            if abs(ER)<=err
                break
            end
        end
    end
    %Cálculo del área mojada
    A=b*y
    %Cálculo del perímetro mojado
    P=b+2*y
    %Cálculo del ancho superficial
    T=b
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))
    %Aproximación de resultados a 4 dígitos
    y=round(y,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);
    E=round(E,4);

```

```

%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif b<=0
    errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif b1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Valor de la gravedad en ft/s2
    g=32.18504
    %Cálculo del tirante crítico;
    M=100000
    err=0.000001
    y=100
    for i=1:M
        FY=((b*y)^3/b)-(Q^2/g)
        if FY==0
            break
        else
            FDY=3*b^2*y^2
            ER=FY/FDY
            abs(ER)
            y=y-ER
            if abs(ER)<=err
                break
            end
        end
    end
    %Cálculo del área mojada
    A=b*y
    %Cálculo del perímetro mojado
    P=b+2*y
    %Cálculo del ancho superficial
    T=b
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercritico'
    elseif NF<1
        TF='Subcritico'
    else NF==1
        TF='Critico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))
    %Aproximación de resultados a 4 dígitos

```

```

y=round(y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');

```

```

set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Mostrar unidades métricas;
        set(handles.text6,'Visible','on');
        set(handles.text7,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text26,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');
        set(handles.text23,'Visible','on');
        set(handles.text30,'Visible','on');
        %Ocultar unidades inglesas;
        set(handles.text35,'Visible','off');
        set(handles.text36,'Visible','off');
        set(handles.text38,'Visible','off');
        set(handles.text39,'Visible','off');
        set(handles.text42,'Visible','off');
        set(handles.text43,'Visible','off');
        set(handles.text45,'Visible','off');
        set(handles.text41,'Visible','off');
        set(handles.text40,'Visible','off');
        set(handles.text44,'Visible','off');
    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');

```

```

%Ocultar unidades métricas;
set(handles.text6,'Visible','off');
set(handles.text7,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text26,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
%Mostrar unidades inglesas;
set(handles.text35,'Visible','on');
set(handles.text36,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

% --- Executes during object creation, after setting all properties.
function pushbutton1_CreateFcn(hObject, eventdata, handles)

```

2.1.2.2. SECCIÓN TRAPEZOIDAL

```

function varargout = TIRANTE_CRITICO_DE_UNA_SECCION_TRAPEZOIDAL(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @TIRANTE_CRITICO_DE_UNA_SECCION_TRAPEZOIDAL_OpeningFcn, ...
                  'gui_OutputFcn',  @TIRANTE_CRITICO_DE_UNA_SECCION_TRAPEZOIDAL_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before TIRANTE_CRITICO_DE_UNA_SECCION_TRAPEZOIDAL is made
visible.
function TIRANTE_CRITICO_DE_UNA_SECCION_TRAPEZOIDAL_OpeningFcn(hObject, eventdata,
handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('TRAPECIO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;

```

```

guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = TIRANTE_CRITICO_DE_UNA_SECCION_TRAPEZOIDAL_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6, 'String', '');
set(handles.edit9, 'String', '');
set(handles.edit7, 'String', '');
set(handles.edit8, 'String', '');
set(handles.edit15, 'String', '');
set(handles.edit10, 'String', '');
set(handles.edit13, 'String', '');
set(handles.edit11, 'String', '');
set(handles.edit12, 'String', '');
set(handles.edit14, 'String', '');
%Obtener el dato del caudal
VAL1=get(handles.edit1, 'String');
Q=str2double(VAL1)
%Obtener el dato del ancho de la solera
VAL2=get(handles.edit2, 'String');
b=str2double(VAL2)
%Obtener el dato del talud z
VAL6=get(handles.edit16, 'String');
z=str2double(VAL6)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
b1=isnan(b);
z1=isnan(z);
VAL5=get(handles.uibuttongroup1, 'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.', 'Mensaje');
elseif b<=0
errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.', 'Mensaje');
elseif Q1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif b1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif z<=0
errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.', 'Mensaje');
elseif z1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
else
ESPERA
%Valor de la gravedad en m/s2
g=9.81
%Cálculo del tirante crítico
M=100000
err=0.000001
y=100
for i=1:M
FY=(( (b+z*y) *y) ^3/(b+2*z*y)) - (Q^2/g)
if FY==0
break
else
FDY=(3*y^2*(b + y*z)^3)/(b + 2*y*z) + (3*y^3*z*(b + y*z)^2)/(b + 2*y*z) -
(2*y^3*z*(b + y*z)^3)/(b + 2*y*z)^2
ER=FY/FDY
abs(ER)
y=y-ER
if abs(ER)<=err
break

```



```

        end
    end
end
%Cálculo del área mojada
A=(b+z*y)*y
%Cálculo del perímetro mojado
P=b+2*y*sqrt(1+(z^2))
%Cálculo del ancho superficial
T=b+2*z*y
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del radio hidráulico
R=A/P
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
y=round(y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor que 0.','Mensaje');
elseif b<=0
    errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de calcular.','Mensaje');
elseif b1==1
    errordlg('Complete los campos de entrada adecuadamente antes de calcular.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor que 0.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de calcular.','Mensaje');
else
    ESPERA
    %Valor de la gravedad en ft/s2
    g=32.18504
    %Cálculo del tirante crítico

```

```

M=100000
err=0.000001
y=100
for i=1:M
    FY=((b+z*y)*y)^3/(b+2*z*y)-(Q^2/g)
    if FY==0
        break
    else
        FDY=(3*y^2*(b + y*z)^3)/(b + 2*y*z) + (3*y^3*z*(b + y*z)^2)/(b + 2*y*z) -
(2*y^3*z*(b + y*z)^3)/(b + 2*y*z)^2
        ER=FY/FDY
        abs(ER)
        y=y-ER
        if abs(ER)<=err
            break
        end
    end
end
end
%Cálculo del área mojada
A=(b+z*y)*y
%Cálculo del perímetro mojado
P=b+2*y*sqrt(1+(z^2))
%Cálculo del ancho superficial
T=b+2*z*y
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del radio hidráulico
R=A/P
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
y=round(y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y)
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

```

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

```

```

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit16,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_CRITICO_DE_UNA_SECCION_TRAPEZOIDAL

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados

```

```

set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Mostrar unidades métricas;
set(handles.text6,'Visible','on');
set(handles.text7,'Visible','on');
set(handles.text19,'Visible','on');
set(handles.text22,'Visible','on');
set(handles.text25,'Visible','on');
set(handles.text26,'Visible','on');
set(handles.text33,'Visible','on');
set(handles.text24,'Visible','on');
set(handles.text23,'Visible','on');
set(handles.text30,'Visible','on');
%Ocultar unidades inglesas;
set(handles.text35,'Visible','off');
set(handles.text36,'Visible','off');
set(handles.text38,'Visible','off');
set(handles.text39,'Visible','off');
set(handles.text42,'Visible','off');
set(handles.text43,'Visible','off');
set(handles.text45,'Visible','off');
set(handles.text41,'Visible','off');
set(handles.text40,'Visible','off');
set(handles.text44,'Visible','off');
case 'Sistema inglés'
SM1='2'
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Ocultar unidades métricas;
set(handles.text6,'Visible','off');
set(handles.text7,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text26,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
%Mostrar unidades inglesas;
set(handles.text35,'Visible','on');
set(handles.text36,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultuicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

2.1.2.3. SECCIÓN TRIANGULAR

```

function varargout = TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR_OpeningFcn, ...
                  'gui_OutputFcn',   @TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR is made visible.
function TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR_OpeningFcn(hObject, eventdata,
handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('TRIANGULO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%Obtener el dato del talud
VAL2=get(handles.edit2,'String');

```

```

z=str2double(VAL2)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGRESADOS CORRECTAMENTE
Q1=isnan(Q);
z1=isnan(z);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Valor de la gravedad en m/s2
    g=9.81
    %Cálculo del tirante crítico
    M=100000
    err=0.000001
    y=100
    for i=1:M
        FY=((z*(y^2))^3/(2*z*y))-(Q^2/g)
        if FY==0
            break
        else
            FDY=(5*y^4*z^2)/2
            ER=FY/FDY
            abs(ER)
            y=y-ER
            if abs(ER)<=err
                break
            end
        end
    end
    end
    %Cálculo del área mojada
    A=z*(y^2)
    %Cálculo del perímetro mojado
    P=2*y*sqrt(1+(z^2))
    %Cálculo del ancho superficial
    T=2*z*y
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))
    %Aproximación de resultados a 4 dígitos
    y=round(y,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);

```

```

E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif Ql==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Valor de la gravedad en ft/s2
    g=32.18504
    %Cálculo del tirante crítico
    M=100000
    err=0.000001
    y=100
    for i=1:M
        FY=((z*(y^2))^3/(2*z*y))-(Q^2/g)
        if FY==0
            break
        else
            FDY=(5*y^4*z^2)/2
            ER=FY/FDY
            abs(ER)
            y=y-ER
            if abs(ER)<=err
                break
            end
        end
    end
    %Cálculo del área mojada
    A=z*(y^2)
    %Cálculo del perímetro mojado
    P=2*y*sqrt(1+(z^2))
    %Cálculo del ancho superficial
    T=2*z*y
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))

```



```

    %Aproximación de resultados a 4 dígitos
    y=round(y,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);
    E=round(E,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit6,'String',y);
    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',V);
    set(handles.edit11,'String',NF);
    set(handles.edit12,'String',TF);
    set(handles.edit14,'String',E);
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```

        set(hObject,'BackgroundColor','white');
    end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');

```

```

set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Mostrar unidades métricas;
        set(handles.text6,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text26,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');
        set(handles.text23,'Visible','on');
        set(handles.text30,'Visible','on');
        %Ocultar unidades inglesas;
        set(handles.text35,'Visible','off');
        set(handles.text38,'Visible','off');
        set(handles.text39,'Visible','off');
        set(handles.text42,'Visible','off');
        set(handles.text43,'Visible','off');
        set(handles.text45,'Visible','off');
        set(handles.text41,'Visible','off');
        set(handles.text40,'Visible','off');
        set(handles.text44,'Visible','off');
    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Ocultar unidades métricas;

```

```

set(handles.text6,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text26,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
%Mostrar unidades inglesas;
set(handles.text35,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

```

2.1.2.4. SECCIÓN CIRCULAR

```

function varargout = TIRANTE_CRITICO_DE_UNA_SECCION_CIRCULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @TIRANTE_CRITICO_DE_UNA_SECCION_CIRCULAR_OpeningFcn, ...
                  'gui_OutputFcn',  @TIRANTE_CRITICO_DE_UNA_SECCION_CIRCULAR_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before TIRANTE_CRITICO_DE_UNA_SECCION_CIRCULAR is made visible.
function TIRANTE_CRITICO_DE_UNA_SECCION_CIRCULAR_OpeningFcn(hObject, eventdata,
handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('CIRCULAR.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = TIRANTE_CRITICO_DE_UNA_SECCION_CIRCULAR_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.

```

```

function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%Obtener el dato del diámetro
VAL2=get(handles.edit2,'String');
d=str2double(VAL2)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
d1=isnan(d);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif d<=0
errordlg('Ingrese el valor del diámetro (d) teniendo en cuenta que debe ser mayor
que 0 y mayor que el tirante (y).','Mensaje');
elseif Q1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif d1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
ESPERA
%Valor de la gravedad en m/s2
g=9.81
%Cálculo del ángulo
M=100000
err=0.000001
y=pi
for i=1:M
FY=((1/8)*(y-sin(y))*(d^2))^3/(d*sin(y/2))- (Q^2/g)
if FY==0
break
else
FDY=- (d^5*cos(y/2)*(y - sin(y))^3)/(1024*sin(y/2)^2) - (3*d^5*(y -
sin(y))^2*(cos(y) - 1))/(512*sin(y/2))
ER=FY/FDY
abs(ER)
y=y-ER
if abs(ER)<=err
break
end
end
end
end
%Cálculo del tirante crítico
Y=(d/2)*(1-cos(y/2))
%Cálculo del área mojada
A=(1/8)*(y-sin(y))*(d^2)
%Cálculo del perimetro mojado
P=(1/2)*y*d
%Cálculo del ancho superficial
T=d*sin(y/2)
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del radio hidráulico
R=A/P
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude

```



```

        if abs(ER) <= err
            break
        end
    end
end
end
%Cálculo del tirante crítico
Y=(d/2)*(1-cos(y/2))
%Cálculo del área mojada
A=(1/8)*(y-sin(y))*(d^2)
%Cálculo del perímetro mojado
P=(1/2)*y*d
%Cálculo del ancho superficial
T=d*sin(y/2)
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del radio hidráulico
R=A/P
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=Y+((V^2)/(2*g))
close ESPERA
A1=(pi*(d^2))/4
if A>=A1
    errordlg('El tirante es mayor que el diámetro. Incremente el
diámetro.','Mensaje');
elseif A<=0
    errordlg('El tirante es mayor que el diámetro. Incremente el
diámetro.','Mensaje');
else
    %Aproximación de resultados a 4 dígitos
    Y=round(Y,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);
    E=round(E,4);
    %Impresión de resultados
    set(handles.edit6,'String',Y);
    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',V);
    set(handles.edit11,'String',NF);
    set(handles.edit12,'String',TF);
    set(handles.edit14,'String',E);
end
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

```

```

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)

```



```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_CRITICO_DE_UNA_SECCION_CIRCULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');

```

```

set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Mostrar unidades métricas;
set(handles.text6,'Visible','on');
set(handles.text7,'Visible','on');
set(handles.text19,'Visible','on');
set(handles.text22,'Visible','on');
set(handles.text25,'Visible','on');
set(handles.text26,'Visible','on');
set(handles.text33,'Visible','on');
set(handles.text24,'Visible','on');
set(handles.text23,'Visible','on');
set(handles.text30,'Visible','on');
%Ocultar unidades inglesas;
set(handles.text35,'Visible','off');
set(handles.text36,'Visible','off');
set(handles.text38,'Visible','off');
set(handles.text39,'Visible','off');
set(handles.text42,'Visible','off');
set(handles.text43,'Visible','off');
set(handles.text45,'Visible','off');
set(handles.text41,'Visible','off');
set(handles.text40,'Visible','off');
set(handles.text44,'Visible','off');
case 'Sistema inglés'
SM1='2'
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Ocultar unidades métricas;
set(handles.text6,'Visible','off');
set(handles.text7,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text26,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
%Mostrar unidades inglesas;
set(handles.text35,'Visible','on');
set(handles.text36,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

```

2.1.2.5. SECCIÓN PARABÓLICA

2.1.2.5.1. SECCIÓN PARABÓLICA CON “T” CONOCIDO

```
function varargout = TIRANTE_CRITICO_DE_UNA_SECCION_CIRCULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @TIRANTE_CRITICO_DE_UNA_SECCION_CIRCULAR_OpeningFcn, ...
                  'gui_OutputFcn',  @TIRANTE_CRITICO_DE_UNA_SECCION_CIRCULAR_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before TIRANTE_CRITICO_DE_UNA_SECCION_CIRCULAR is made visible.
function TIRANTE_CRITICO_DE_UNA_SECCION_CIRCULAR_OpeningFcn(hObject, eventdata,
handles, varargin)
%Código para centrar la ventana
scrsz=get(0, 'ScreenSize');
pos_act=get(gcf, 'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf, 'Position', [xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('CIRCULAR.JPG');
%Código para mostrar la imagen
image(imagen, 'Parent', handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1, 'XTick', [], 'YTick', []);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = TIRANTE_CRITICO_DE_UNA_SECCION_CIRCULAR_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6, 'String', '');
set(handles.edit9, 'String', '');
set(handles.edit7, 'String', '');
set(handles.edit8, 'String', '');
set(handles.edit15, 'String', '');
set(handles.edit10, 'String', '');
set(handles.edit13, 'String', '');
set(handles.edit11, 'String', '');
set(handles.edit12, 'String', '');
set(handles.edit14, 'String', '');
%Obtener el dato del caudal
VAL1=get(handles.edit1, 'String');
Q=str2double(VAL1)
%Obtener el dato del diámetro
VAL2=get(handles.edit2, 'String');
d=str2double(VAL2)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
```

```

Q1=isnan(Q);
d1=isnan(d);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif d<=0
    errordlg('Ingrese el valor del diámetro (d) teniendo en cuenta que debe ser mayor
que 0 y mayor que el tirante (y)','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif d1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Valor de la gravedad en m/s2
    g=9.81
    %Cálculo del ángulo
    M=100000
    err=0.000001
    y=pi
    for i=1:M
        FY=((1/8)*(y-sin(y))*(d^2))^3/(d*sin(y/2))- (Q^2/g)
        if FY==0
            break
        else
            FDY=- (d^5*cos(y/2)*(y - sin(y))^3)/(1024*sin(y/2)^2) - (3*d^5*(y -
sin(y))^2*(cos(y) - 1))/(512*sin(y/2))
            ER=FY/FDY
            abs(ER)
            y=y-ER
            if abs(ER)<=err
                break
            end
        end
    end
    %Cálculo del tirante crítico
    Y=(d/2)*(1-cos(y/2))
    %Cálculo del área mojada
    A=(1/8)*(y-sin(y))*(d^2)
    %Cálculo del perímetro mojado
    P=(1/2)*y*d
    %Cálculo del ancho superficial
    T=d*sin(y/2)
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=Y+((V^2)/(2*g))
    close ESPERA
    A1=(pi*(d^2))/4
    if (A>=A1)
        errordlg('El tirante es mayor que el diámetro. Incremente el
diámetro.','Mensaje');
    elseif A<=0

```

```

        errordlg('El tirante es mayor que el diámetro. Incremente el
diámetro.', 'Mensaje');
    else
        %Aproximación de resultados a 4 dígitos
        Y=round(Y,4);
        A=round(A,4);
        P=round(P,4);
        T=round(T,4);
        D=round(D,4);
        R=round(R,4);
        V=round(V,4);
        E=round(E,4);
        %Impresión de resultados
        set(handles.edit6, 'String', Y);
        set(handles.edit9, 'String', A);
        set(handles.edit7, 'String', P);
        set(handles.edit8, 'String', R);
        set(handles.edit15, 'String', T);
        set(handles.edit10, 'String', D);
        set(handles.edit13, 'String', V);
        set(handles.edit11, 'String', NF);
        set(handles.edit12, 'String', TF);
        set(handles.edit14, 'String', E);
    end
end
else
%Para unidades del sistema inglés
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.', 'Mensaje');
elseif d<=0
    errordlg('Ingrese el valor del diámetro (d) teniendo en cuenta que debe ser mayor
que 0 y mayor que el tirante (y).', 'Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif d1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
else
    ESPERA
    %Valor de la gravedad en ft/s2
    g=32.18504
    %Cálculo del ángulo
    M=100000
    err=0.000001
    y=pi
    for i=1:M
        FY=((1/8)*(y-sin(y))*(d^2))^3/(d*sin(y/2))- (Q^2/g)
        if FY==0
            break
        else
            FDY=- (d^5*cos(y/2)*(y - sin(y))^3)/(1024*sin(y/2)^2) - (3*d^5*(y -
sin(y))^2*(cos(y) - 1))/(512*sin(y/2))
            ER=FY/FDY
            abs(ER)
            y=y-ER
            if abs(ER)<=err
                break
            end
        end
    end
end
end
%Cálculo del tirante crítico
Y=(d/2)*(1-cos(y/2))
%Cálculo del área mojada
A=(1/8)*(y-sin(y))*(d^2)
%Cálculo del perímetro mojado
P=(1/2)*y*d
%Cálculo del ancho superficial
T=d*sin(y/2)
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del radio hidráulico
R=A/P
%Cálculo de la velocidad

```

```

V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=Y+((V^2)/(2*g))
close ESPERA
A1=(pi*(d^2))/4
if A>=A1
    errordlg('El tirante es mayor que el diámetro. Incremente el
diámetro.','Mensaje');
elseif A<=0
    errordlg('El tirante es mayor que el diámetro. Incremente el
diámetro.','Mensaje');
else
    %Aproximación de resultados a 4 dígitos
    Y=round(Y,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);
    E=round(E,4);
    %Impresión de resultados
    set(handles.edit6,'String',Y);
    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',V);
    set(handles.edit11,'String',NF);
    set(handles.edit12,'String',TF);
    set(handles.edit14,'String',E);
end
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

```

```

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_CRITICO_DE_UNA_SECCION_CIRCULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Mostrar unidades métricas;
        set(handles.text6,'Visible','on');
        set(handles.text7,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text26,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');
        set(handles.text23,'Visible','on');
        set(handles.text30,'Visible','on');

```



```

%Ocultar unidades inglesas;
set(handles.text35,'Visible','off');
set(handles.text36,'Visible','off');
set(handles.text38,'Visible','off');
set(handles.text39,'Visible','off');
set(handles.text42,'Visible','off');
set(handles.text43,'Visible','off');
set(handles.text45,'Visible','off');
set(handles.text41,'Visible','off');
set(handles.text40,'Visible','off');
set(handles.text44,'Visible','off');
case 'Sistema inglés'
SM1='2'
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Ocultar unidades métricas;
set(handles.text6,'Visible','off');
set(handles.text7,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text26,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
%Mostrar unidades inglesas;
set(handles.text35,'Visible','on');
set(handles.text36,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

```

2.1.2.5.2. SECCIÓN PARABÓLICA CON “T” NO CONOCIDO

```

function varargout = TIRANTE_CRITICO_DE_UNA_SECCION_PARABOLICA_2(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @TIRANTE_CRITICO_DE_UNA_SECCION_PARABOLICA_2_OpeningFcn, ...
                  'gui_OutputFcn',  @TIRANTE_CRITICO_DE_UNA_SECCION_PARABOLICA_2_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

```

```

% --- Executes just before TIRANTE_CRITICO_DE_UNA_SECCION_PARABOLICA_2 is made
visible.
function TIRANTE_CRITICO_DE_UNA_SECCION_PARABOLICA_2_OpeningFcn(hObject, eventdata,
handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('PARÁBOLA.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = TIRANTE_CRITICO_DE_UNA_SECCION_PARABOLICA_2_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
set(handles.text103,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif Q1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
ESPERA
%Valor de la gravedad en m/s2
g=9.81
%Cálculo del tirante crítico
C=(Q^2)/g
C1=round(C,4)
y=2
T=2*sqrt(y)
%Cálculo del área mojada
A=(2/3)*T*y
%Cálculo del perímetro mojado
x=(4*y)/T
if ((x>0) && (x<=1))
P=T+((8/3)*((y^2)/T))
else
P=(T/2)*(sqrt(1+x^2)+(1/x)*log(x+sqrt(1+x^2)))
end
R=A/P
F=(A^3)/T

```

```

F1=round(F,4)
while F1~=C1
    if F1>C1
        y=(y-y/4)
    else
        y=(y+y/4)
    end
    T=2*sqrt(y)
    %Cálculo del área mojada
    A=(2/3)*T*y
    %Cálculo del perímetro mojado
    x=(4*y)/T
    if ((x>0) && (x<=1))
        P=T+((8/3)*((y^2)/T))
    else
        P=(T/2)*(sqrt(1+x^2)+(1/x)*log(x+sqrt(1+x^2)))
    end
    R=A/P
    F=(A^3)/T
    F1=round(F,4)
end
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
y=round(y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit15,'String',R);
set(handles.text103,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Valor de la gravedad en ft/s2
    g=32.18504
    %Cálculo del tirante crítico
    C=(Q^2)/g
    C1=round(C,4)

```

```

y=2
T=2*sqrt(y)
%Cálculo del área mojada
A=(2/3)*T*y
%Cálculo del perímetro mojado
x=(4*y)/T
if ((x>0) && (x<=1))
    P=T+((8/3)*((y^2)/T))
else
    P=(T/2)*(sqrt(1+x^2)+(1/x)*log(x+sqrt(1+x^2)))
end
R=A/P
F=(A^3)/T
F1=round(F,4)
while F1~C1
    if F1>C1
        y=(y-y/4)
    else
        y=(y+y/4)
    end
    T=2*sqrt(y)
    %Cálculo del área mojada
    A=(2/3)*T*y
    %Cálculo del perímetro mojado
    x=(4*y)/T
    if ((x>0) && (x<=1))
        P=T+((8/3)*((y^2)/T))
    else
        P=(T/2)*(sqrt(1+x^2)+(1/x)*log(x+sqrt(1+x^2)))
    end
    R=A/P
    F=(A^3)/T
    F1=round(F,4)
end
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
y=round(y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit15,'String',R);
set(handles.text103,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
end

```

```

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.

```

```

function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
set(handles.text103,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_CRITICO_DE_UNA_SECCION_PARABOLICA_2

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.

```

```

function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        set(handles.text103,'String','');
        %Mostrar unidades métricas;
        set(handles.text6,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');
        set(handles.text23,'Visible','on');
        set(handles.text30,'Visible','on');
        set(handles.text104,'Visible','on');
        %Ocultar unidades inglesas;
        set(handles.text35,'Visible','off');
        set(handles.text38,'Visible','off');
        set(handles.text39,'Visible','off');
        set(handles.text42,'Visible','off');
        set(handles.text45,'Visible','off');
        set(handles.text41,'Visible','off');
        set(handles.text40,'Visible','off');
        set(handles.text44,'Visible','off');
        set(handles.text105,'Visible','off');

    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        set(handles.text103,'String','');
        %Ocultar unidades métricas;
        set(handles.text6,'Visible','off');
        set(handles.text19,'Visible','off');
        set(handles.text22,'Visible','off');
        set(handles.text25,'Visible','off');
        set(handles.text33,'Visible','off');
        set(handles.text24,'Visible','off');
        set(handles.text23,'Visible','off');
        set(handles.text30,'Visible','off');
        set(handles.text104,'Visible','off');
        %Mostrar unidades inglesas;
        set(handles.text35,'Visible','on');
        set(handles.text38,'Visible','on');
        set(handles.text39,'Visible','on');
        set(handles.text42,'Visible','on');
        set(handles.text45,'Visible','on');
        set(handles.text41,'Visible','on');
        set(handles.text40,'Visible','on');
        set(handles.text44,'Visible','on');
        set(handles.text105,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)

```

```

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)

function edit20_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit20_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit21_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit21_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit22_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit22_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit17_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit18_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit19_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit19_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```


2.1.2.6. SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS

```

function varargout = TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR_REDONDEADA(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR_REDONDEADA_OpeningFcn, ...
                  'gui_OutputFcn',  @TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR_REDONDEADA_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR_REDONDEADA is
% made visible.
function TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR_REDONDEADA_OpeningFcn(hObject,
eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('RECTANGULAR_REDONDEADAS.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout =
TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR_REDONDEADA_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%Obtener el dato del ancho de la solera
VAL2=get(handles.edit2,'String');
b=str2double(VAL2)
%Obtener el dato del radio

```

```

VAL6=get(handles.edit16,'String');
r=str2double(VAL6)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGRESADOS CORRECTAMENTE
Q1=isnan(Q);
b1=isnan(b);
r1=isnan(r);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif b<=0
    errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif b1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif r<=0
    errordlg('Ingrese el valor del radio (r) teniendo en cuenta que debe ser mayor
que 0 y menor que el tirante (y).','Mensaje');
elseif r1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Valor de la gravedad en m/s2
    g=9.81
    %Cálculo del tirante crítico;
    M=100000
    err=0.000001
    y=100
    for i=1:M
        FY=(((pi/2)-2)*(r^2)+((b+2*r)*y))^3/(b+2*r)-(Q^2/g)
        if FY==0
            break
        else
            FDY=3*(r^2*(pi/2 - 2) + y*(b + 2*r))^2
            ER=FY/FDY
            abs(ER)
            y=y-ER
            if abs(ER)<=err
                break
            end
        end
    end
    if y>r
        %Cálculo del área mojada
        A=(((pi/2)-2)*(r^2)+((b+2*r)*y))
        %Cálculo del perímetro mojado
        P=(((pi-2)*r)+b+(2*y))
        %Cálculo del ancho superficial
        T=b+2*r
        %Cálculo de la profundidad hidráulica
        D=A/T
        %Cálculo del radio hidráulico
        R=A/P
        %Cálculo de la velocidad
        V=Q/A
        %Cálculo del número de froude
        NF=V/sqrt(g*D);
        NF=round(NF,4)
        %Identificación del tipo de flujo
        if NF>1
            TF='Supercrítico'
        elseif NF<1
            TF='Subcrítico'
        else
            TF='Crítico'
        end
        %Cálculo de la energía específica

```

```

E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
y=round(y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
else
close ESPERA
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
errordlg('El tirante no puede ser mayor que el valor de r. Disminuya el ancho
de la solera (b) o el radio (r).','Mensaje');
end
end
else
%Para unidades del sistema inglés
if Q<=0
errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif b<=0
errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif Q1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif b1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif r<=0
errordlg('Ingrese el valor del radio (r) teniendo en cuenta que debe ser mayor
que 0 y menor que el tirante (y).','Mensaje');
elseif r1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
ESPERA
%Valor de la gravedad en ft/s2
g=32.18504
%Cálculo del tirante crítico
M=100000
err=0.000001
y=100
for i=1:M
FY=(((pi/2)-2)*(r^2)+((b+2*r)*y)^3/(b+2*r))-(Q^2/g)
if FY==0
break
else
FDY=3*(r^2*(pi/2 - 2) + y*(b + 2*r))^2
ER=FY/FDY
abs(ER)
y=y-ER

```

```

        if abs(ER) <= err
            break
        end
    end
end
end
if y > r
    %Cálculo del área mojada
    A = ((pi/2) - 2) * (r^2) + ((b + 2*r) * y)
    %Cálculo del perímetro mojado
    P = ((pi - 2) * r) + b + (2 * y)
    %Cálculo del ancho superficial
    T = b + 2 * r
    %Cálculo de la profundidad hidráulica
    D = A / T
    %Cálculo del radio hidráulico
    R = A / P
    %Cálculo de la velocidad
    V = Q / A
    %Cálculo del número de froude
    NF = V / sqrt(g * D);
    NF = round(NF, 4)
    %Identificación del tipo de flujo
    if NF > 1
        TF = 'Supercrítico'
    elseif NF < 1
        TF = 'Subcrítico'
    else
        TF = 'Crítico'
    end
    %Cálculo de la energía específica
    E = y + ((V^2) / (2 * g))
    %Aproximación de resultados a 4 dígitos
    y = round(y, 4);
    A = round(A, 4);
    P = round(P, 4);
    T = round(T, 4);
    D = round(D, 4);
    R = round(R, 4);
    V = round(V, 4);
    E = round(E, 4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit6, 'String', y);
    set(handles.edit9, 'String', A);
    set(handles.edit7, 'String', P);
    set(handles.edit8, 'String', R);
    set(handles.edit15, 'String', T);
    set(handles.edit10, 'String', D);
    set(handles.edit13, 'String', V);
    set(handles.edit11, 'String', NF);
    set(handles.edit12, 'String', TF);
    set(handles.edit14, 'String', E);
else
    close ESPERA
    set(handles.edit6, 'String', '');
    set(handles.edit9, 'String', '');
    set(handles.edit7, 'String', '');
    set(handles.edit8, 'String', '');
    set(handles.edit15, 'String', '');
    set(handles.edit10, 'String', '');
    set(handles.edit13, 'String', '');
    set(handles.edit11, 'String', '');
    set(handles.edit12, 'String', '');
    set(handles.edit14, 'String', '');
    errordlg('El tirante no puede ser mayor que el valor de r. Disminuya el ancho
de la solera (b) o el radio (r).', 'Mensaje');
end
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit16,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR_REDONDEADA

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');

```

```

switch SM
case 'Sistema métrico'
SM1='1'
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Mostrar unidades métricas;
set(handles.text6,'Visible','on');
set(handles.text7,'Visible','on');
set(handles.text19,'Visible','on');
set(handles.text22,'Visible','on');
set(handles.text25,'Visible','on');
set(handles.text26,'Visible','on');
set(handles.text33,'Visible','on');
set(handles.text24,'Visible','on');
set(handles.text23,'Visible','on');
set(handles.text30,'Visible','on');
set(handles.text48,'Visible','on');
%Ocultar unidades inglesas;
set(handles.text35,'Visible','off');
set(handles.text36,'Visible','off');
set(handles.text38,'Visible','off');
set(handles.text39,'Visible','off');
set(handles.text42,'Visible','off');
set(handles.text43,'Visible','off');
set(handles.text45,'Visible','off');
set(handles.text41,'Visible','off');
set(handles.text40,'Visible','off');
set(handles.text44,'Visible','off');
set(handles.text47,'Visible','off');
case 'Sistema inglés'
SM1='2'
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Ocultar unidades métricas;
set(handles.text6,'Visible','off');
set(handles.text7,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text26,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
set(handles.text48,'Visible','off');
%Mostrar unidades inglesas;
set(handles.text35,'Visible','on');
set(handles.text36,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
set(handles.text47,'Visible','on');

```

```

end
set(handles.uibuttongroup1,'Tag',SM1)

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

2.1.2.7. SECCIÓN TRIANGULAR CON FONDO REDONDEADO

```

function varargout = TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR_REDONDEADO(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OpeningFcn, ...
                  'gui_OutputFcn',  @TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR_REDONDEADO is
% made visible.
function TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OpeningFcn(hObject,
eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('TRIANGULAR_REDONDEADO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout =
TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');

```



```

set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%Obtener el dato del radio
VAL2=get(handles.edit2,'String');
r=str2double(VAL2)
%Obtener el dato del talud
VAL6=get(handles.edit16,'String');
z=str2double(VAL6)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
r1=isnan(r);
z1=isnan(z);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif r<=0
errordlg('Ingrese el valor del radio (r) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif Q1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif r1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z<=0
errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
ESPERA
%Valor de la gravedad en m/s2
g=9.81
%Cálculo del del ancho superficial
M=100000
err=0.000001
y=100
for i=1:M
FY=(((y^2)/(4*z))-((r^2)/z)*(1-(z*acot(z))))^3/y)-(Q^2/g)
if FY==0
break
else
FDY=(3*((z*acot(z) - 1)*r^2)/z + y^2/(4*z))^2/(2*z) - ((z*acot(z) -
1)*r^2)/z + y^2/(4*z)^3/y^2
ER=FY/FDY
abs(ER)
y=y-ER
if abs(ER)<=err
break
end
end
end
T=y
%Cálculo del tirante crítico
w=atan(1/z)
w1=pi-(2*w)
B=((2*pi)-(2*w1))/2
Y2=r*(1-cos(B/2))
M=2*r*sin(B/2)
C=(T-M)/2
Y1=tan(w)*C
Y=Y1+Y2
%Cálculo del área mojada
A=(((T^2)/(4*z))-((r^2)/z)*(1-(z*acot(z))))
%Cálculo del perímetro mojado

```

```

P=((T/z)*sqrt(1+(z^2)))-(((2*r)/z)*(1-z*acot(z)))
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del radio hidráulico
R=A/P
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=Y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
Y=round(Y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',Y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif r<=0
    errordlg('Ingrese el valor del radio (r) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif r1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Valor de la gravedad en ft/s2
    g=32.18504
    %Cálculo del del ancho superficial
    M=100000
    err=0.000001
    y=100
    for i=1:M
        FY((((y^2)/(4*z))-(((r^2)/z)*(1-(z*acot(z))))^3/y)-(Q^2/g)
        if FY==0
            break
        else

```

```

1)*r^2)/z + y^2/(4*z))^3/y^2
    FDY=(3*((z*acot(z) - 1)*r^2)/z + y^2/(4*z))^2/(2*z) - (((z*acot(z) -
    ER=FY/FDY
    abs(ER)
    y=y-ER
    if abs(ER)<=err
        break
    end
end
end
end
T=y
%Cálculo del tirante crítico
w=atan(1/z)
w1=pi-(2*w)
B=((2*pi)-(2*w1))/2
Y2=r*(1-cos(B/2))
M=2*r*sin(B/2)
C=(T-M)/2
Y1=tan(w)*C
Y=Y1+Y2
%Cálculo del área mojada
A=((T^2)/(4*z))-((r^2)/z)*(1-(z*acot(z))))
%Cálculo del perímetro mojado
P=((T/z)*sqrt(1+(z^2)))-((2*r)/z)*(1-z*acot(z)))
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del radio hidráulico
R=A/P
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
end
%Cálculo de la energía específica
E=Y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
Y=round(Y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',Y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

```

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

```

```

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit16,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR_REDONDEADO

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados

```

```

set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Mostrar unidades métricas;
set(handles.text6,'Visible','on');
set(handles.text7,'Visible','on');
set(handles.text19,'Visible','on');
set(handles.text22,'Visible','on');
set(handles.text25,'Visible','on');
set(handles.text26,'Visible','on');
set(handles.text33,'Visible','on');
set(handles.text24,'Visible','on');
set(handles.text23,'Visible','on');
set(handles.text30,'Visible','on');
%Ocultar unidades inglesas;
set(handles.text35,'Visible','off');
set(handles.text36,'Visible','off');
set(handles.text38,'Visible','off');
set(handles.text39,'Visible','off');
set(handles.text42,'Visible','off');
set(handles.text43,'Visible','off');
set(handles.text45,'Visible','off');
set(handles.text41,'Visible','off');
set(handles.text40,'Visible','off');
set(handles.text44,'Visible','off');
case 'Sistema inglés'
SM1='2'
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Ocultar unidades métricas;
set(handles.text6,'Visible','off');
set(handles.text7,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text26,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
%Mostrar unidades inglesas;
set(handles.text35,'Visible','on');
set(handles.text36,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultuicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

2.1.3. CAUDALES

2.1.3.1. SECCIÓN RECTANGULAR

```

function varargout = CAUDAL_DE_UNA_SECCION_RECTANGULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @CAUDAL_DE_UNA_SECCION_RECTANGULAR_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @CAUDAL_DE_UNA_SECCION_RECTANGULAR_OutputFcn,
                  ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before CAUDAL_DE_UNA_SECCION_RECTANGULAR is made visible.
function CAUDAL_DE_UNA_SECCION_RECTANGULAR_OpeningFcn(hObject, eventdata, handles,
varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('RECTANGULO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = CAUDAL_DE_UNA_SECCION_RECTANGULAR_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del tirante

```

```

VAL1=get(handles.edit1,'String');
y=str2double(VAL1)
%Obtener el dato del ancho de la solera
VAL2=get(handles.edit2,'String');
b=str2double(VAL2)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
y1=isnan(y);
b1=isnan(b);
S1=isnan(S);
n1=isnan(n);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if y<=0
    errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif b<=0
    errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif y1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif b1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Valor de la gravedad en m/s2
    g=9.81
    %Cálculo del área mojada
    A=b*y
    %Cálculo del perímetro mojado
    P=b+2*y
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo del caudal
    Q=(1/n)*A*R^(2/3)*S^(1/2)
    %Cálculo del ancho superficial
    T=b
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))

```



```

%Aproximación de resultados a 4 dígitos
Q=round(Q,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',Q);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if y<=0
    errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif b<=0
    errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif y1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif b1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Valor de la gravedad en ft/s2
    g=32.18504
    %Cálculo del área mojada
    A=b*y
    %Cálculo del perímetro mojado
    P=b+2*y
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo del caudal
    Q=(1.486/n)*A*R^(2/3)*S^(1/2)
    %Cálculo del ancho superficial
    T=b
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de Froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else

```

```

        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))
    %Aproximación de resultados a 4 dígitos
    Q=round(Q,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);
    E=round(E,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit6,'String',Q);
    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',V);
    set(handles.edit11,'String',NF);
    set(handles.edit12,'String',TF);
    set(handles.edit14,'String',E);
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

```

```

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');

```

```

set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close CAUDAL_DE_UNA_SECCION_RECTANGULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Mostrar unidades métricas
        set(handles.text6,'Visible','on');
        set(handles.text7,'Visible','on');
        set(handles.text9,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text26,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');
        set(handles.text23,'Visible','on');
        set(handles.text30,'Visible','on');
        %Ocultar unidades inglesas
        set(handles.text35,'Visible','off');
        set(handles.text36,'Visible','off');
        set(handles.text37,'Visible','off');
        set(handles.text38,'Visible','off');
        set(handles.text39,'Visible','off');
        set(handles.text42,'Visible','off');
        set(handles.text43,'Visible','off');
        set(handles.text45,'Visible','off');
        set(handles.text41,'Visible','off');
        set(handles.text40,'Visible','off');
        set(handles.text44,'Visible','off');
    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');

```

```

set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Ocultar unidades métricas
set(handles.text6,'Visible','off');
set(handles.text7,'Visible','off');
set(handles.text9,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text26,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
%Mostrar unidades inglesas
set(handles.text35,'Visible','on');
set(handles.text36,'Visible','on');
set(handles.text37,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

% --- Executes during object creation, after setting all properties.
function pushbutton1_CreateFcn(hObject, eventdata, handles)

```

2.1.3.2. SECCIÓN TRAPEZOIDAL

```

function varargout = CAUDAL_DE_UNA_SECCION_TRAPEZOIDAL(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @CAUDAL_DE_UNA_SECCION_TRAPEZOIDAL_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @CAUDAL_DE_UNA_SECCION_TRAPEZOIDAL_OutputFcn,
                  ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before CAUDAL_DE_UNA_SECCION_TRAPEZOIDAL is made visible.
function CAUDAL_DE_UNA_SECCION_TRAPEZOIDAL_OpeningFcn(hObject, eventdata, handles,
varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);

```

```

set(gcf, 'Position', [xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('TRAPECIO.JPG');
%Código para mostrar la imagen
image(imagen, 'Parent', handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1, 'XTick', [], 'YTick', []);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = CAUDAL_DE_UNA_SECCION_TRAPEZOIDAL_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6, 'String', '');
set(handles.edit9, 'String', '');
set(handles.edit7, 'String', '');
set(handles.edit8, 'String', '');
set(handles.edit15, 'String', '');
set(handles.edit10, 'String', '');
set(handles.edit13, 'String', '');
set(handles.edit11, 'String', '');
set(handles.edit12, 'String', '');
set(handles.edit14, 'String', '');
%Obtener el dato del tirante
VAL1=get(handles.edit1, 'String');
y=str2double(VAL1)
%Obtener el dato del ancho de la solera
VAL2=get(handles.edit2, 'String');
b=str2double(VAL2)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3, 'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4, 'String');
n=str2double(VAL4)
%Obtener el dato del talud
VAL6=get(handles.edit16, 'String');
z=str2double(VAL6)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
y1=isnan(y);
b1=isnan(b);
S1=isnan(S);
n1=isnan(n);
z1=isnan(z);
VAL5=get(handles.uibuttongroup1, 'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if y<=0
errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.', 'Mensaje');
elseif b<=0
errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.', 'Mensaje');
elseif S<=0
errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.', 'Mensaje');
elseif n<=0
errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.', 'Mensaje');
elseif y1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif b1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif S1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');

```

```

elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Valor de la gravedad en m/s2
    g=9.81
    %Cálculo del área mojada
    A=(b+z*y)*y
    %Cálculo del perímetro mojado
    P=b+2*y*sqrt(1+z^2)
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo del caudal
    Q=(1/n)*A*R^(2/3)*S^(1/2)
    %Cálculo del ancho superficial
    T=b+2*z*y
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))
    %Aproximación de resultados a 4 dígitos
    Q=round(Q,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);
    E=round(E,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit6,'String',Q);
    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',V);
    set(handles.edit11,'String',NF);
    set(handles.edit12,'String',TF);
    set(handles.edit14,'String',E);
end
else
    %Para unidades del sistema inglés
    if y<=0
        errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
    elseif b<=0
        errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
    elseif S<=0
        errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
    elseif n<=0

```

```

        errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif y1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif b1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Valor de la gravedad en ft/s2
    g=32.18504
    %Cálculo del área mojada
    A=(b+z*y)*y
    %Cálculo del perímetro mojado
    P=b+2*y*sqrt(1+z^2)
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo del caudal
    Q=(1.486/n)*A*R^(2/3)*S^(1/2)
    %Cálculo del ancho superficial
    T=b+2*z*y
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))
    %Aproximación de resultados a 4 dígitos
    Q=round(Q,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);
    E=round(E,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit6,'String',Q);
    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',V);
    set(handles.edit11,'String',NF);
    set(handles.edit12,'String',TF);
    set(handles.edit14,'String',E);
end
end

```



```

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.

```

```

function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit16,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close CAUDAL_DE_UNA_SECCION_TRAPEZOIDAL

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject, 'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6, 'String', '');
        set(handles.edit9, 'String', '');
        set(handles.edit7, 'String', '');
        set(handles.edit8, 'String', '');
        set(handles.edit15, 'String', '');
        set(handles.edit10, 'String', '');
        set(handles.edit13, 'String', '');
        set(handles.edit11, 'String', '');
        set(handles.edit12, 'String', '');
        set(handles.edit14, 'String', '');
        %Mostrar unidades métricas
        set(handles.text6, 'Visible', 'on');
        set(handles.text7, 'Visible', 'on');
        set(handles.text9, 'Visible', 'on');
        set(handles.text19, 'Visible', 'on');
        set(handles.text22, 'Visible', 'on');
        set(handles.text25, 'Visible', 'on');
        set(handles.text26, 'Visible', 'on');
        set(handles.text33, 'Visible', 'on');
        set(handles.text24, 'Visible', 'on');
        set(handles.text23, 'Visible', 'on');
        set(handles.text30, 'Visible', 'on');
        %Ocultar unidades inglesas
        set(handles.text35, 'Visible', 'off');
        set(handles.text36, 'Visible', 'off');
        set(handles.text37, 'Visible', 'off');
        set(handles.text38, 'Visible', 'off');
        set(handles.text39, 'Visible', 'off');
        set(handles.text42, 'Visible', 'off');
        set(handles.text43, 'Visible', 'off');
        set(handles.text45, 'Visible', 'off');
        set(handles.text41, 'Visible', 'off');
        set(handles.text40, 'Visible', 'off');
        set(handles.text44, 'Visible', 'off');
    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit6, 'String', '');
        set(handles.edit9, 'String', '');
        set(handles.edit7, 'String', '');
        set(handles.edit8, 'String', '');
        set(handles.edit15, 'String', '');
        set(handles.edit10, 'String', '');
        set(handles.edit13, 'String', '');
        set(handles.edit11, 'String', '');
        set(handles.edit12, 'String', '');
        set(handles.edit14, 'String', '');
        %Ocultar unidades métricas
        set(handles.text6, 'Visible', 'off');
        set(handles.text7, 'Visible', 'off');
        set(handles.text9, 'Visible', 'off');
        set(handles.text19, 'Visible', 'off');
        set(handles.text22, 'Visible', 'off');
        set(handles.text25, 'Visible', 'off');
        set(handles.text26, 'Visible', 'off');
        set(handles.text33, 'Visible', 'off');
        set(handles.text24, 'Visible', 'off');
        set(handles.text23, 'Visible', 'off');
        set(handles.text30, 'Visible', 'off');
        %Mostrar unidades inglesas
        set(handles.text35, 'Visible', 'on');
        set(handles.text36, 'Visible', 'on');
        set(handles.text37, 'Visible', 'on');
        set(handles.text38, 'Visible', 'on');
        set(handles.text39, 'Visible', 'on');

```

```

        set(handles.text42,'Visible','on');
        set(handles.text43,'Visible','on');
        set(handles.text45,'Visible','on');
        set(handles.text41,'Visible','on');
        set(handles.text40,'Visible','on');
        set(handles.text44,'Visible','on');
    end
    set(handles.uibuttongroup1,'Tag',SM1)

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

2.1.3.3. SECCIÓN TRIANGULAR

```

function varargout = CAUDAL_DE_UNA_SECCION_TRIANGULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @CAUDAL_DE_UNA_SECCION_TRIANGULAR_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @CAUDAL_DE_UNA_SECCION_TRIANGULAR_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before CAUDAL_DE_UNA_SECCION_TRIANGULAR is made visible.
function CAUDAL_DE_UNA_SECCION_TRIANGULAR_OpeningFcn(hObject, eventdata, handles,
varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('TRIANGULO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = CAUDAL_DE_UNA_SECCION_TRIANGULAR_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');

```

```

set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del tirante
VAL1=get(handles.edit1,'String');
y=str2double(VAL1)
%Obtener el dato del talud
VAL2=get(handles.edit2,'String');
z=str2double(VAL2)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
y1=isnan(y);
z1=isnan(z);
S1=isnan(S);
n1=isnan(n);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if y<=0
errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z<=0
errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif S<=0
errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif y1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
ESPERA
%Valor de la gravedad en m/s2
g=9.81
%Cálculo del área mojada
A=z*y^2
%Cálculo del perímetro mojado
P=2*y*sqrt(1+z^2)
%Cálculo del radio hidráulico
R=A/P
%Cálculo del caudal
Q=(1/n)*A*R^(2/3)*S^(1/2)
%Cálculo del ancho superficial
T=2*z*y
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1

```

```

        TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
Q=round(Q,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',Q);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if y<=0
    errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif y1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Valor de la gravedad en ft/s2
    g=32.18504
    %Cálculo del área mojada
    A=z*y^2
    %Cálculo del perímetro mojado
    P=2*y*sqrt(1+z^2)
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo del caudal
    Q=(1.486/n)*A*R^(2/3)*S^(1/2)
    %Cálculo del ancho superficial
    T=2*z*y
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude

```

```

NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
Q=round(Q,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',Q);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.

```

```

function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```



```

        set(hObject,'BackgroundColor','white');
    end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close CAUDAL_DE_UNA_SECCION_TRIANGULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Mostrar unidades métricas
        set(handles.text6,'Visible','on');
        set(handles.text9,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text26,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');
        set(handles.text23,'Visible','on');
        set(handles.text30,'Visible','on');
        %Ocultar unidades inglesas
        set(handles.text35,'Visible','off');
        set(handles.text37,'Visible','off');
        set(handles.text38,'Visible','off');
        set(handles.text39,'Visible','off');
        set(handles.text42,'Visible','off');
        set(handles.text43,'Visible','off');
        set(handles.text45,'Visible','off');
        set(handles.text41,'Visible','off');

```

```

        set(handles.text40,'Visible','off');
        set(handles.text44,'Visible','off');
    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Ocultar unidades métricas
        set(handles.text6,'Visible','off');
        set(handles.text9,'Visible','off');
        set(handles.text19,'Visible','off');
        set(handles.text22,'Visible','off');
        set(handles.text25,'Visible','off');
        set(handles.text26,'Visible','off');
        set(handles.text33,'Visible','off');
        set(handles.text24,'Visible','off');
        set(handles.text23,'Visible','off');
        set(handles.text30,'Visible','off');
        %Mostrar unidades inglesas
        set(handles.text35,'Visible','on');
        set(handles.text37,'Visible','on');
        set(handles.text38,'Visible','on');
        set(handles.text39,'Visible','on');
        set(handles.text42,'Visible','on');
        set(handles.text43,'Visible','on');
        set(handles.text45,'Visible','on');
        set(handles.text41,'Visible','on');
        set(handles.text40,'Visible','on');
        set(handles.text44,'Visible','on');
    end
    set(handles.uibuttongroup1,'Tag',SM1)

```

2.1.3.4. SECCIÓN CIRCULAR

```

function varargout = CAUDAL_DE_UNA_SECCION_CIRCULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @CAUDAL_DE_UNA_SECCION_CIRCULAR_OpeningFcn, ...
                  'gui_OutputFcn',  @CAUDAL_DE_UNA_SECCION_CIRCULAR_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before CAUDAL_DE_UNA_SECCION_CIRCULAR is made visible.
function CAUDAL_DE_UNA_SECCION_CIRCULAR_OpeningFcn(hObject, eventdata, handles,
varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);

```

```

%Código para leer la imagen;
imagen=imread('CIRCULAR.JPG');
%Código para mostrar la imagen;
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes;
set(handles.axes1,'XTick',[],'YTick',[]);
handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = CAUDAL_DE_UNA_SECCION_CIRCULAR_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del tirante
VAL1=get(handles.edit1,'String');
y=str2double(VAL1)
%Obtener el dato del diametro
VAL2=get(handles.edit2,'String');
d=str2double(VAL2)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGRESADOS CORRECTAMENTE
y1=isnan(y);
d1=isnan(d);
S1=isnan(S);
n1=isnan(n);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if y<=0
errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif d<=0
errordlg('Ingrese el valor del diámetro (d) teniendo en cuenta que debe ser mayor
que 0 y mayor que el tirante (y).','Mensaje');
elseif S<=0
errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif y1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif d1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif y>=d
errordlg('El tirante (y) debe ser menor que el diámetro','Mensaje');
else

```

```

ESPERA
%Cálculo del ángulo
if y==d/2
    x=pi
elseif y>d/2
    y1=y-(d/2)
    t=acos(y1/(d/2))
    t1=2*t
    x=(2*pi)-t1
else
    y1=(d/2)-y
    t=acos(y1/(d/2))
    x=2*t
end
%Valor de la gravedad en m/s2
g=9.81
%Cálculo del área mojada
A=(1/8)*(x-sin(x))*(d^2)
%Cálculo del perímetro mojado
P=(1/2)*x*d
%Cálculo del radio hidráulico
R=A/P
%Cálculo del caudal
Q=(1/n)*A*R^(2/3)*S^(1/2)
%Cálculo del ancho superficial
T=d*sin(x/2)
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
Q=round(Q,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',Q);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if y<=0
    errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif d<=0
    errordlg('Ingrese el valor del diámetro (d) teniendo en cuenta que debe ser mayor
que 0 y mayor que el tirante (y).','Mensaje');
elseif S<=0

```

```

        errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.', 'Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.', 'Mensaje');
elseif y1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif d1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif y>=d
    errordlg('El tirante (y) debe ser menor que el diámetro', 'Mensaje');
else
    ESPERA
    %Cálculo del ángulo
    if y==d/2
        x=pi
    elseif y>d/2
        y1=y-(d/2)
        t=acos(y1/(d/2))
        t1=2*t
        x=(2*pi)-t1
    else
        y1=(d/2)-y
        t=acos(y1/(d/2))
        x=2*t
    end
    %Valor de la gravedad en ft/s2
    g=32.18504
    %Cálculo del área mojada
    A=(1/8)*(x-sin(x))*(d^2)
    %Cálculo del perimetro mojado
    P=(1/2)*x*d
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo del caudal
    Q=(1.486/n)*A*R^(2/3)*S^(1/2)
    %Cálculo del ancho superficial
    T=d*sin(x/2)
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))
    %Aproximación de resultados a 4 dígitos
    Q=round(Q,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);
    E=round(E,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit6, 'String', Q);

```

```

    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',V);
    set(handles.edit11,'String',NF);
    set(handles.edit12,'String',TF);
    set(handles.edit14,'String',E);
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO

```

```

close CAUDAL_DE_UNA_SECCION_CIRCULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Mostrar unidades métricas
        set(handles.text6,'Visible','on');
        set(handles.text7,'Visible','on');
        set(handles.text9,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text26,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');
        set(handles.text23,'Visible','on');
        set(handles.text30,'Visible','on');
        %Ocultar unidades inglesas
        set(handles.text35,'Visible','off');
        set(handles.text36,'Visible','off');
        set(handles.text37,'Visible','off');
        set(handles.text38,'Visible','off');
        set(handles.text39,'Visible','off');
        set(handles.text42,'Visible','off');
        set(handles.text43,'Visible','off');
        set(handles.text45,'Visible','off');
        set(handles.text41,'Visible','off');
        set(handles.text40,'Visible','off');
        set(handles.text44,'Visible','off');
    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Ocultar unidades métricas
        set(handles.text6,'Visible','off');
        set(handles.text7,'Visible','off');
        set(handles.text9,'Visible','off');
        set(handles.text19,'Visible','off');
        set(handles.text22,'Visible','off');
        set(handles.text25,'Visible','off');

```



```

set(handles.text26,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
%Mostrar unidades inglesas
set(handles.text35,'Visible','on');
set(handles.text36,'Visible','on');
set(handles.text37,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

```

2.1.3.5. SECCIÓN PARABÓLICA

```

function varargout = CAUDAL_DE_UNA_SECCION_PARABOLICA(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @CAUDAL_DE_UNA_SECCION_PARABOLICA_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @CAUDAL_DE_UNA_SECCION_PARABOLICA_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before CAUDAL_DE_UNA_SECCION_PARABOLICA is made visible.
function CAUDAL_DE_UNA_SECCION_PARABOLICA_OpeningFcn(hObject, eventdata, handles,
varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('PARÁBOLA.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = CAUDAL_DE_UNA_SECCION_PARABOLICA_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');

```

```

set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
y=str2double(VAL1)
%Obtener el dato del ancho superficial
VAL2=get(handles.edit2,'String');
T=str2double(VAL2)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
y1=isnan(y);
T1=isnan(T);
S1=isnan(S);
n1=isnan(n);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if y<=0
errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif T<=0
errordlg('Ingrese el valor del ancho superficial (T) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif S<=0
errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif y1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif T1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
ESPERA
%Valor de la gravedad en m/s2
g=9.81
%Cálculo del área mojada
A=(2/3)*T*y
%Cálculo del perímetro mojado
x=(4*y)/T
if ((x<=1)&&(x>0))
P=T+((8/3)*((y^2)/T))
else
P=(T/2)*(sqrt(1+x^2)+(1/x)*log(x+sqrt(1+x^2)))
end
%Cálculo del radio hidráulico
R=A/P
%Cálculo del caudal
Q=(1/n)*A*R^(2/3)*S^(1/2)
%Cálculo del ancho superficial
T=(3*A)/(2*y)
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del factor de sección

```



```

end
%Cálculo del radio hidráulico
R=A/P
%Cálculo del caudal
Q=(1.486/n)*A*R^(2/3)*S^(1/2)
%Cálculo del ancho superficial
T=(3*A)/(2*y)
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del factor de sección
Z=A*sqrt(D)
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
Q=round(Q,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit6,'String',Q);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit15,'String',R);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

```

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

```

```

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close CAUDAL_DE_UNA_SECCION_PARABOLICA

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Mostrar unidades métricas
        set(handles.text6,'Visible','on');
        set(handles.text7,'Visible','on');
        set(handles.text9,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');

```

```

set(handles.text23,'Visible','on');
set(handles.text30,'Visible','on');
%Ocultar unidades inglesas
set(handles.text35,'Visible','off');
set(handles.text36,'Visible','off');
set(handles.text37,'Visible','off');
set(handles.text38,'Visible','off');
set(handles.text39,'Visible','off');
set(handles.text42,'Visible','off');
set(handles.text45,'Visible','off');
set(handles.text41,'Visible','off');
set(handles.text40,'Visible','off');
set(handles.text44,'Visible','off');
case 'Sistema inglés'
SM1='2'
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Ocultar unidades métricas
set(handles.text6,'Visible','off');
set(handles.text7,'Visible','off');
set(handles.text9,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
%Mostrar unidades inglesas
set(handles.text35,'Visible','on');
set(handles.text36,'Visible','on');
set(handles.text37,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)

function edit20_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit20_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit21_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit21_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit22_Callback(hObject, eventdata, handles)

```

```

% --- Executes during object creation, after setting all properties.
function edit22_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit17_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit18_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit19_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit19_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

2.1.3.6. SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS

```

function varargout = CAUDAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @CAUDAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA_OpeningFcn, ...
                  'gui_OutputFcn',   @CAUDAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before CAUDAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA is made
visible.

```



```

function CAUDAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA_OpeningFcn(hObject, eventdata,
handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('RECTANGULAR_REDONDEADAS.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = CAUDAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del tirante
VAL1=get(handles.edit1,'String');
y=str2double(VAL1)
%Obtener el dato del ancho de la solera
VAL2=get(handles.edit2,'String');
b=str2double(VAL2)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%Obtener el dato del radio
VAL6=get(handles.edit16,'String');
r=str2double(VAL6)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
y1=isnan(y);
b1=isnan(b);
S1=isnan(S);
n1=isnan(n);
r1=isnan(r);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if y<=0
errorldg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif b<=0
errorldg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif S<=0
errorldg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
errorldg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');

```

```

elseif y1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif b1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif r<=0
    errordlg('Ingrese el valor del radio (r) teniendo en cuenta que debe ser mayor
que 0 y menor que el tirante (y).','Mensaje');
elseif r1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif y<=r
    errordlg('El tirante no puede ser menor o igual que el radio (r).','Mensaje');
else
    ESPERA
    %Valor de la gravedad en m/s2
    g=9.81
    %Cálculo del área mojada
    A=((pi/2)-2)*(r^2)+((b+2*r)*y)
    %Cálculo del perímetro mojado
    P=((pi-2)*r)+b+(2*y)
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo del caudal
    Q=(1/n)*A*R^(2/3)*S^(1/2)
    %Cálculo del ancho superficial
    T=b+2*r
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))
    %Aproximación de resultados a 4 dígitos
    Q=round(Q,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);
    E=round(E,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit6,'String',Q);
    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',V);
    set(handles.edit11,'String',NF);
    set(handles.edit12,'String',TF);
    set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés

```

```

if y<=0
    errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.', 'Mensaje');
elseif b<=0
    errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.', 'Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.', 'Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.', 'Mensaje');
elseif y1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif b1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif r<=0
    errordlg('Ingrese el valor del radio (r) teniendo en cuenta que debe ser mayor
que 0 y menor que el tirante (y).', 'Mensaje');
elseif r1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif y<=r
    errordlg('El tirante no puede ser menor o igual que el radio (r).', 'Mensaje');
else
    ESPERA
    %Valor de la gravedad en ft/s2
    g=32.18504
    %Cálculo del área mojada
    A=((pi/2)-2)*(r^2)+((b+2*r)*y)
    %Cálculo del perímetro mojado
    P=((pi-2)*r)+b+(2*y)
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo del caudal
    Q=(1.486/n)*A*R^(2/3)*S^(1/2)
    %Cálculo del ancho superficial
    T=b+2*r
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo de la velocidad
    V=Q/A;
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))
    %Aproximación de resultados a 4 dígitos
    Q=round(Q,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);
    E=round(E,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit6,'String',Q);

```

```

    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',V);
    set(handles.edit11,'String',NF);
    set(handles.edit12,'String',TF);
    set(handles.edit14,'String',E);
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

```

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit16,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)

```

```

INICIO
close CAUDAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Mostrar unidades métricas
        set(handles.text6,'Visible','on');
        set(handles.text7,'Visible','on');
        set(handles.text9,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text26,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');
        set(handles.text23,'Visible','on');
        set(handles.text30,'Visible','on');
        set(handles.text48,'Visible','on');
        %Ocultar unidades inglesas
        set(handles.text35,'Visible','off');
        set(handles.text36,'Visible','off');
        set(handles.text37,'Visible','off');
        set(handles.text38,'Visible','off');
        set(handles.text39,'Visible','off');
        set(handles.text42,'Visible','off');
        set(handles.text43,'Visible','off');
        set(handles.text45,'Visible','off');
        set(handles.text41,'Visible','off');
        set(handles.text40,'Visible','off');
        set(handles.text44,'Visible','off');
        set(handles.text47,'Visible','off');
    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Ocultar unidades métricas
        set(handles.text6,'Visible','off');
        set(handles.text7,'Visible','off');
        set(handles.text9,'Visible','off');

```

```

set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text26,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
set(handles.text48,'Visible','off');
%Mostrar unidades inglesas
set(handles.text35,'Visible','on');
set(handles.text36,'Visible','on');
set(handles.text37,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
set(handles.text47,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

2.1.3.7. SECCIÓN TRIANGULAR CON FONDO REDONDEADO

```

function varargout = CAUDAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @CAUDAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OpeningFcn, ...
                  'gui_OutputFcn',  @CAUDAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before CAUDAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO is made
visible.
function CAUDAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OpeningFcn(hObject, eventdata,
handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('TRIANGULAR_REDONDEADO.JPG');
%Código para mostrar la imagen

```

```

image(imagen, 'Parent', handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1, 'XTick', [], 'YTick', []);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = CAUDAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6, 'String', '');
set(handles.edit9, 'String', '');
set(handles.edit7, 'String', '');
set(handles.edit8, 'String', '');
set(handles.edit15, 'String', '');
set(handles.edit10, 'String', '');
set(handles.edit13, 'String', '');
set(handles.edit11, 'String', '');
set(handles.edit12, 'String', '');
set(handles.edit14, 'String', '');
%Obtener el dato del caudal
VAL1=get(handles.edit1, 'String');
y=str2double(VAL1)
%Obtener el dato del radio
VAL2=get(handles.edit2, 'String');
r=str2double(VAL2)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3, 'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4, 'String');
n=str2double(VAL4)
%Obtener el dato del talud
VAL6=get(handles.edit16, 'String');
z=str2double(VAL6)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
y1=isnan(y);
r1=isnan(r);
S1=isnan(S);
n1=isnan(n);
z1=isnan(z);
VAL5=get(handles.uibuttongroup1, 'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if y<=0
errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.', 'Mensaje');
elseif r<=0
errordlg('Ingrese el valor del radio (r) teniendo en cuenta que debe ser mayor
que 0.', 'Mensaje');
elseif S<=0
errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.', 'Mensaje');
elseif n<=0
errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.', 'Mensaje');
elseif y1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif r1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif S1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif n1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif z<=0

```



```

        errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.', 'Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
else
    ESPERA
    %Valor de la gravedad en m/s2
    g=9.81
    %Cálculo del ancho superficial
    T=2*(z*(y-r)+r*sqrt(1+z^2))
    %Cálculo del área mojada
    A=((T^2)/(4*z))-((r^2)/z)*(1-(z*acot(z))))
    %Cálculo del perímetro mojado
    P=((T/z)*sqrt(1+(z^2)))-((2*r)/z)*(1-z*acot(z)))
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo del caudal
    Q=(1/n)*A*R^(2/3)*S^(1/2)
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))
    %Aproximación de resultados a 4 dígitos
    Q=round(Q,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);
    E=round(E,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit6, 'String', Q);
    set(handles.edit9, 'String', A);
    set(handles.edit7, 'String', P);
    set(handles.edit8, 'String', R);
    set(handles.edit15, 'String', T);
    set(handles.edit10, 'String', D);
    set(handles.edit13, 'String', V);
    set(handles.edit11, 'String', NF);
    set(handles.edit12, 'String', TF);
    set(handles.edit14, 'String', E);
end
else
    %Para unidades del sistema inglés
    if y<=0
        errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.', 'Mensaje');
    elseif r<=0
        errordlg('Ingrese el valor del radio (r) teniendo en cuenta que debe ser mayor
que 0.', 'Mensaje');
    elseif S<=0
        errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.', 'Mensaje');
    elseif n<=0
        errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.', 'Mensaje');
    elseif y1==1
        errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');

```

```

elseif r1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Valor de la gravedad en ft/s2
    g=32.18504
    %Cálculo del ancho superficial
    T=2*(z*(y-r)+r*sqrt(1+z^2))
    %Cálculo del área mojada
    A=((T^2)/(4*z))-((r^2)/z)*(1-(z*acot(z))))
    %Cálculo del perímetro mojado
    P=((T/z)*sqrt(1+(z^2)))-((2*r)/z)*(1-z*acot(z)))
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo del caudal
    Q=(1.486/n)*A*R^(2/3)*S^(1/2)
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))
    %Aproximación de resultados a 4 dígitos
    Q=round(Q,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);
    E=round(E,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit6,'String',Q);
    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',V);
    set(handles.edit11,'String',NF);
    set(handles.edit12,'String',TF);
    set(handles.edit14,'String',E);
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit16,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close CAUDAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.

```

```

function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Mostrar unidades métricas
        set(handles.text6,'Visible','on');
        set(handles.text7,'Visible','on');
        set(handles.text9,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text26,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');
        set(handles.text23,'Visible','on');
        set(handles.text30,'Visible','on');
        %Ocultar unidades inglesas
        set(handles.text35,'Visible','off');
        set(handles.text36,'Visible','off');
        set(handles.text37,'Visible','off');
        set(handles.text38,'Visible','off');
        set(handles.text39,'Visible','off');
        set(handles.text42,'Visible','off');
        set(handles.text43,'Visible','off');
        set(handles.text45,'Visible','off');
        set(handles.text41,'Visible','off');
        set(handles.text40,'Visible','off');
        set(handles.text44,'Visible','off');
    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Ocultar unidades métricas
        set(handles.text6,'Visible','off');
        set(handles.text7,'Visible','off');
        set(handles.text9,'Visible','off');
        set(handles.text19,'Visible','off');
        set(handles.text22,'Visible','off');
        set(handles.text25,'Visible','off');
        set(handles.text26,'Visible','off');
        set(handles.text33,'Visible','off');
        set(handles.text24,'Visible','off');
        set(handles.text23,'Visible','off');
        set(handles.text30,'Visible','off');
        %Mostrar unidades inglesas
        set(handles.text35,'Visible','on');
        set(handles.text36,'Visible','on');
        set(handles.text37,'Visible','on');
        set(handles.text38,'Visible','on');
        set(handles.text39,'Visible','on');
        set(handles.text42,'Visible','on');
        set(handles.text43,'Visible','on');
        set(handles.text45,'Visible','on');
        set(handles.text41,'Visible','on');

```

```

        set(handles.text40,'Visible','on');
        set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

2.1.4. ELEMENTOS GEOMÉTRICOS

2.1.4.1. SECCIÓN RECTANGULAR

```

function varargout = ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR_OpeningFcn, ...
                  'gui_OutputFcn',  @ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR is made
visible.
function ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR_OpeningFcn(hObject,
eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('RECTÁNGULO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout =
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)

```

```

%Borrar celdas de resultados
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
%Obtener el dato del ancho de la solera
VAL1=get(handles.edit1,'String');
b=str2double(VAL1)
%Obtener el dato del tirante
VAL2=get(handles.edit2,'String');
y=str2double(VAL2)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE;
bl=isnan(b);
yl=isnan(y);
if b<=0
    errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif y<=0
    errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif bl==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif yl==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Cálculo del área mojada
    A=b*y
    %Cálculo del perímetro mojado
    P=b+2*y
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo del ancho superficial
    T=b
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo del factor de sección
    Z=sqrt(D)
    %Aproximación de resultados a 4 dígitos
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    Z=round(Z,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',Z);
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

```



```

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

```

2.1.4.2. SECCIÓN TRAPEZOIDAL

```

function varargout = ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRAPEZOIDAL(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRAPEZOIDAL_OpeningFcn, ...
                  'gui_OutputFcn',  @ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRAPEZOIDAL_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

```

```

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRAPEZOIDAL is made
visible.
function ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRAPEZOIDAL_OpeningFcn(hObject,
eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('TRAPECIO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout =
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRAPEZOIDAL_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
%Obtener el dato del tirante
VAL1=get(handles.edit1,'String');
y=str2double(VAL1)
%Obtener el dato del ancho de la solera
VAL2=get(handles.edit2,'String');
b=str2double(VAL2)
%Obtener el dato del talud
VAL3=get(handles.edit16,'String');
z=str2double(VAL3)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
y1=isnan(y);
b1=isnan(b);
z1=isnan(z);
if y<=0
    errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif b<=0
    errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif y1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif b1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else

```

```

ESPERA
%Cálculo del área mojada
A=(b+z*y)*y
%Cálculo del perímetro mojado
P=b+2*y*sqrt(1+z^2)
%Cálculo del radio hidráulico
R=A/P
%Cálculo del ancho superficial
T=b+2*z*y
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del factor de sección
Z=sqrt(D)
%Aproximación de resultados a 4 dígitos
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
Z=round(Z,4);
%Impresión de resultados
close ESPERA
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',Z);
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

```

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas

```

```

set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit16,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRAPEZOIDAL

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

2.1.4.3. SECCIÓN TRIANGULAR

```

function varargout = ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR_OpeningFcn, ...
                  'gui_OutputFcn',  @ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR is made
visible.
function ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR_OpeningFcn(hObject,
eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen

```

```

imagen=imread('TRIÁNGULO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout =
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
%Obtener el dato del tirante
VAL1=get(handles.edit1,'String');
y=str2double(VAL1)
%Obtener el dato del talud
VAL2=get(handles.edit2,'String');
z=str2double(VAL2)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
y1=isnan(y);
z1=isnan(z);
if y<=0
    errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif y1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Cálculo del área mojada
    A=z*y^2
    %Cálculo del perímetro mojado
    P=2*y*sqrt(1+z^2)
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo del ancho superficial
    T=2*z*y
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo del factor de sección
    Z=sqrt(D)
    %Aproximación de resultados a 4 dígitos
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    Z=round(Z,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',Z);
end

```

```

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

```

```

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

```


2.1.4.4. SECCIÓN CIRCULAR

```
function varargout = ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_CIRCULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_CIRCULAR_OpeningFcn, ...
                  'gui_OutputFcn',  @ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_CIRCULAR_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_CIRCULAR is made
% visible.
function ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_CIRCULAR_OpeningFcn(hObject, eventdata,
handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('CIRCULAR.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_CIRCULAR_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
%Obtener el dato del diámetro
VAL1=get(handles.edit1,'String');
d=str2double(VAL1)
%Obtener el dato del tirante
VAL2=get(handles.edit2,'String');
y=str2double(VAL2)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
d1=isnan(d);
y1=isnan(y);
if d<=0
    errordlg('Ingrese el valor del diámetro (d) teniendo en cuenta que tiene que ser
mayor que 0 y mayor que el tirante (y).','Mensaje');
elseif y<=0
```

```

        errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que tiene que ser
mayor que 0 y menor que el diámetro (d).','Mensaje');
elseif dl==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif y1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif y>=d
    errordlg('El tirante (y) debe ser menor que el diámetro','Mensaje');
else
    ESPERA
    %Cálculo del ángulo
    if y==d/2
        x=pi
    elseif y>d/2
        y1=y-(d/2)
        t=acos(y1/(d/2))
        t1=2*t
        x=(2*pi)-t1
    else
        y1=(d/2)-y
        t=acos(y1/(d/2))
        x=2*t
    end
    %Cálculo del área mojada
    A=(1/8)*(x-sin(x))*(d^2)
    %Cálculo del perímetro mojado
    P=(1/2)*x*d
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo del ancho superficial
    T=d*sin(x/2)
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo del factor de sección
    Z=sqrt(D)
    %Aproximación de resultados a 4 dígitos
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    Z=round(Z,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',Z);
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.

```

```

function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```

        set(hObject,'BackgroundColor','white');
    end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_CIRCULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

```

2.1.4.5. SECCIÓN PARABÓLICA

```

function varargout = ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_PARABOLICA(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_PARABOLICA_OpeningFcn, ...
                  'gui_OutputFcn',  @ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_PARABOLICA_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

```

```

% --- Executes just before ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_PARABOLICA is made
visible.
function ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_PARABOLICA_OpeningFcn(hObject,
eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('PARÁBOLA.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout =
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_PARABOLICA_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
%Obtener el dato del tirante
VAL1=get(handles.edit1,'String');
y=str2double(VAL1)
%Obtener el dato del ancho superficial
VAL2=get(handles.edit2,'String');
T=str2double(VAL2)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
y1=isnan(y);
T1=isnan(T);
if y<=0
    errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif T<=0
    errordlg('Ingrese el valor del ancho superficial (T) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif y1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif T1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Cálculo del área mojada
    A=(2/3)*T*y
    %Cálculo del perímetro mojado
    x=(4*y)/T
    if ((x<=1)&&(x>0))
        P=T+((8/3)*((y^2)/T))
    else
        P=(T/2)*(sqrt(1+x^2)+(1/x)*log(x+sqrt(1+x^2)))
    end
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo del ancho superficial
    T=(3*A)/(2*y)
    %Cálculo de la profundidad hidráulica

```

```

D=A/T
%Cálculo del factor de sección
Z=sqrt(D)
%Aproximación de resultados a 4 dígitos
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
Z=round(Z,4);
%Impresión de resultados
close ESPERA
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',Z);
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

```

```

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');

% --- Executes on button press in pushbutton5.

```

```

function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_PARABOLICA

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

```

2.1.4.6. SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS

```

function varargout =
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR_REDONDEADO(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR_REDONDEADO_OpeningFcn, ...
                  'gui_OutputFcn',  @ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR_REDONDEADO_OutputFcn, ...
                  'gui_LayerFcn',    [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR_REDONDEADO is made visible.
function
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR_REDONDEADO_OpeningFcn(hObject,
eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('RECTANGULAR_REDONDEADAS.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout =
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR_REDONDEADO_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.

```



```

function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
%Obtener el dato del tirante
VAL1=get(handles.edit1,'String');
y=str2double(VAL1)
%Obtener el dato del ancho de la solera
VAL2=get(handles.edit2,'String');
b=str2double(VAL2)
%Obtener el dato del radio
VAL3=get(handles.edit16,'String');
r=str2double(VAL3)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE;
y1=isnan(y);
b1=isnan(b);
r1=isnan(r);
if y<=0
    errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0 y que el radio (r).','Mensaje');
elseif b<=0
    errordlg('Ingrese el valor del ancho de la solera (b) teniendo en cuenta que debe
ser mayor que 0.','Mensaje');
elseif y1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif b1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif r<=0
    errordlg('Ingrese el valor del radio (r) teniendo en cuenta que debe ser mayor
que 0 y menor que el tirante (y).','Mensaje');
elseif r1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif y<=r
    errordlg('El tirante no puede ser menor o igual que el radio (r).','Mensaje');
else
    ESPERA
    %Cálculo del área mojada
    A=((pi/2)-2)*(r^2)+((b+2*r)*y)
    %Cálculo del perímetro mojado
    P=((pi-2)*r)+b+(2*y)
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo del ancho superficial
    T=b+2*r
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo del factor de sección
    Z=sqrt(D)
    %Aproximación de resultados a 4 dígitos
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    Z=round(Z,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',Z);
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.

```

```

function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```

        set(hObject,'BackgroundColor','white');
    end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit16,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR_REDONDEADO

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))

```

```

        set(hObject,'BackgroundColor','white');
    end

```

2.1.4.7. SECCIÓN TRIANGULAR CON FONDO REDONDEADO

```

function varargout =
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR_REDONDEADO(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OpeningFcn, ...
                  'gui_OutputFcn',  @ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR_REDONDEADO
% is made visible.
function
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OpeningFcn(hObject,
eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('TRIANGULAR_REDONDEADO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout =
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR_REDONDEADO_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
%Obtener el dato del tirante
VAL1=get(handles.edit1,'String');
y=str2double(VAL1)
%Obtener el dato del radio
VAL2=get(handles.edit2,'String');
r=str2double(VAL2)
%Obtener el dato del talud

```

```

VAL3=get(handles.edit16,'String');
z=str2double(VAL3)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGRESADOS CORRECTAMENTE
y1=isnan(y);
r1=isnan(r);
z1=isnan(z);
if y<=0
    errordlg('Ingrese el valor del tirante (y) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif r<=0
    errordlg('Ingrese el valor del radio (r) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif y1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif r1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Cálculo del ancho superficial
    T=2*(z*(y-r)+r*sqrt(1+z^2))
    %Cálculo del área mojada
    A=((T^2)/(4*z))-((r^2)/z)*(1-(z*acot(z))))
    %Cálculo del perímetro mojado
    P=((T/z)*sqrt(1+(z^2)))-((2*r)/z)*(1-z*acot(z)))
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo del factor de sección
    Z=sqrt(D)
    %Aproximación de resultados a 4 dígitos
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    Z=round(Z,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',Z);
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

```

```

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit16,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR_REDONDEADO

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

2.1.5. SECCIÓN HIDRÁULICA ÓPTIMA

2.1.5.1. SECCIÓN RECTANGULAR

```
function varargout = SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_RECTANGULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_RECTANGULAR_OpeningFcn, ...
                  'gui_OutputFcn',  @SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_RECTANGULAR_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_RECTANGULAR is
% made visible.
function SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_RECTANGULAR_OpeningFcn(hObject,
eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('RECTÁNGULO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout =
SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_RECTANGULAR_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit2,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%Obtener el dato de la pendiente
```



```

VAL3=get(handles.edit3,'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente de manning
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
S1=isnan(S);
n1=isnan(n);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif S<=0
errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
ESPERA
%Cálculo del tirante
M=100000
err=0.000001
y=100
for i=1:M
FY=((1/n)*(2*y*y)*(((2*y*y)/(2*y+(2*y)))^(2/3))*(S^(1/2))))-Q
if FY==0
break
else
FDY=(4*S^(1/2)*y*(y/2)^(2/3))/n + (2*S^(1/2)*y^2)/(3*n*(y/2)^(1/3))
ER=FY/FDY
abs(ER)
y=y-ER
if abs(ER)<=err
break
end
end
end
%Valor de la gravedad en m/s2
g=9.81
%Cálculo del ancho de la solera
b=2*y
%Cálculo del area mojada
A=b*y
%Cálculo del perímetro mojado
P=b+2*y
%Cálculo del ancho superficial
T=b
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del radio hidráulico
R=A/P
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
TF='Supercrítico'
elseif NF<1
TF='Subcrítico'

```



```

A=b*y
%Cálculo del perímetro mojado
P=b+2*y
%Cálculo del ancho superficial
T=b
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del radio hidráulico
R=A/P
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
b=round(b,4);
y=round(y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit2,'String',b);
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');

```

```

end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_RECTANGULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit2,'String','');
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Mostrar unidades métricas;
        set(handles.text6,'Visible','on');
        set(handles.text7,'Visible','on');
        set(handles.text9,'Visible','on');

```

```

set(handles.text19,'Visible','on');
set(handles.text22,'Visible','on');
set(handles.text25,'Visible','on');
set(handles.text26,'Visible','on');
set(handles.text33,'Visible','on');
set(handles.text24,'Visible','on');
set(handles.text23,'Visible','on');
set(handles.text30,'Visible','on');
%Ocultar unidades inglesas;
set(handles.text35,'Visible','off');
set(handles.text36,'Visible','off');
set(handles.text37,'Visible','off');
set(handles.text38,'Visible','off');
set(handles.text39,'Visible','off');
set(handles.text42,'Visible','off');
set(handles.text43,'Visible','off');
set(handles.text45,'Visible','off');
set(handles.text41,'Visible','off');
set(handles.text40,'Visible','off');
set(handles.text44,'Visible','off');
case 'Sistema inglés'
SM1='2'
%Borrar celdas de resultados
set(handles.edit2,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Ocultar unidades métricas;
set(handles.text6,'Visible','off');
set(handles.text7,'Visible','off');
set(handles.text9,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text26,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
%Mostrar unidades inglesas;
set(handles.text35,'Visible','on');
set(handles.text36,'Visible','on');
set(handles.text37,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

% --- Executes during object creation, after setting all properties.
function pushbutton1_CreateFcn(hObject, eventdata, handles)

```

2.1.5.2. SECCIÓN TRAPEZOIDAL

2.1.5.2.1. SECCIÓN TRAPEZODIAL CON “z” CONOCIDO

```
function varargout = SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL_2(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL_2_OpeningFcn, ...
                  'gui_OutputFcn',   @SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL_2_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL_2 is
% made visible.
function SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL_2_OpeningFcn(hObject,
eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('TRAPECIO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout =
SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL_2_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
set(handles.text51,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String')
Q=str2double(VAL1);
%Obtener el dato del talud
```

```

VAL2=get(handles.edit16,'String');
z=str2double(VAL2)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
z1=isnan(z);
S1=isnan(S);
n1=isnan(n);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif z<=0
    errordlg('Ingrese el valor del talud (z) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif z1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Cálculo del tirante
    M=100000
    err=0.000001
    y=100
    for i=1:M
        FY=(1/n)*(((2*y*(sqrt(1+z^2)-z))+z*y)*y)*((y/2)^(2/3))*(S^(1/2))-Q
        if FY==0
            break
        else
            FDY=(S^(1/2)*(y*z - 2*y*(z - (z^2 + 1)^(1/2)))*(y/2)^(2/3))/n -
(S^(1/2)*y*(z - 2*(z^2 + 1)^(1/2))*(y/2)^(2/3))/n + (S^(1/2)*y*(y*z - 2*y*(z - (z^2 +
1)^(1/2))))/(3*n*(y/2)^(1/3))
            ER=FY/FDY
            abs(ER)
            y=y-ER
            if abs(ER)<=err
                break
            end
        end
    end
    end
    %Valor de la gravedad en m/s2
    g=9.81
    %Cálculo del ancho de la solera
    b=2*y*(sqrt(1+z^2)-z)
    %Cálculo del área mojada
    A=(b+z*y)*y
    %Cálculo del perímetro mojado
    P=b+2*y*sqrt(1+(z^2))
    %Cálculo del ancho superficial
    T=b+2*z*y
    %Cálculo de la profundidad hidráulica
    D=A/T

```



```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit16,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
set(handles.text51,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL_2

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.

```

```

function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        set(handles.text51,'String','');
        %Mostrar unidades métricas;
        set(handles.text6,'Visible','on');
        set(handles.text9,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text26,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');
        set(handles.text23,'Visible','on');
        set(handles.text30,'Visible','on');
        set(handles.text52,'Visible','on');
        %Ocultar unidades inglesas;
        set(handles.text35,'Visible','off');
        set(handles.text37,'Visible','off');
        set(handles.text38,'Visible','off');
        set(handles.text39,'Visible','off');
        set(handles.text42,'Visible','off');
        set(handles.text43,'Visible','off');
        set(handles.text45,'Visible','off');
        set(handles.text41,'Visible','off');
        set(handles.text40,'Visible','off');
        set(handles.text44,'Visible','off');
        set(handles.text53,'Visible','off');
    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        set(handles.text51,'String','');
        %Ocultar unidades métricas;
        set(handles.text6,'Visible','off');
        set(handles.text9,'Visible','off');
        set(handles.text19,'Visible','off');
        set(handles.text22,'Visible','off');
        set(handles.text25,'Visible','off');
        set(handles.text26,'Visible','off');
        set(handles.text33,'Visible','off');
        set(handles.text24,'Visible','off');
        set(handles.text23,'Visible','off');
        set(handles.text30,'Visible','off');
        set(handles.text52,'Visible','off');
        %Mostrar unidades inglesas;
        set(handles.text35,'Visible','on');
        set(handles.text37,'Visible','on');
        set(handles.text38,'Visible','on');
        set(handles.text39,'Visible','on');
        set(handles.text42,'Visible','on');
        set(handles.text43,'Visible','on');
        set(handles.text45,'Visible','on');

```

```

        set(handles.text41,'Visible','on');
        set(handles.text40,'Visible','on');
        set(handles.text44,'Visible','on');
        set(handles.text53,'Visible','on');
    end
    set(handles.uibuttongroup1,'Tag',SM1)

% --- Executes during object creation, after setting all properties.
function pushbutton1_CreateFcn(hObject, eventdata, handles)

function edit16_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

2.1.5.2.2. SECCIÓN TRAPEZODIAL CON “z” NO CONOCIDO

```

function varargout = SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL_OpeningFcn, ...
                  'gui_OutputFcn',   @SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL is
made visible.
function SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL_OpeningFcn(hObject,
eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('TRAPECIO_ÓPTIMO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout =
SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)

```

```

%Borrar celdas de resultados
set(handles.edit2,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
set(handles.text51,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
S1=isnan(S);
n1=isnan(n);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Cálculo del talud
    z=1/tand(60)
    %Cálculo del tirante
    M=100000
    err=0.000001
    y=100
    for i=1:M
FY=((1/n)*(((1.155*y)+z*y)*y)*(((1.155*y)+z*y)*y)/((1.155*y)+2*y*sqrt(1+(z^2))))^(2/
3))* (S^(1/2))-Q
        if FY==0
            break
        else
            FDY=(S^(1/2)*(1.155*y + y*z)*((y*(1.155*y + y*z))/(1.155*y + 2*y*(z^2 +
1)^(1/2))))^(2/3)/n + (S^(1/2)*y*(z + 1.155)*((y*(1.155*y + y*z))/(1.155*y + 2*y*(z^2
+ 1)^(1/2))))^(2/3)/n + (2*S^(1/2)*y*(1.155*y + y*z)*((1.155*y + y*z)/(1.155*y +
2*y*(z^2 + 1)^(1/2)) + (y*(z + 1.155))/(1.155*y + 2*y*(z^2 + 1)^(1/2)) - (y*(2*(z^2 +
1)^(1/2) + 1.155)*(1.155*y + y*z))/(1.155*y + 2*y*(z^2 +
1)^(1/2))^2)/(3*n*(y*(1.155*y + y*z))/(1.155*y + 2*y*(z^2 + 1)^(1/2))))^(1/3))
            ER=FY/FDY
            abs(ER)
            y=y-ER
            if abs(ER)<=err
                break
            end
        end
    end
end

```

```

        end
    end
    %Valor de la gravedad en m/s2
    g=9.81
    %Cálculo del ancho de la solera
    b=1.155*y
    %Cálculo del área mojada
    A=(b+z*y)*y
    %Cálculo del perímetro mojado
    P=b+2*y*sqrt(1+(z^2))
    %Cálculo del ancho superficial
    T=b+2*z*y
    %Cálculo de la profundidad hidráulica
    D=A/T
    %Cálculo del radio hidráulico
    R=A/P
    %Cálculo de la velocidad
    V=Q/A
    %Cálculo del número de froude
    NF=V/sqrt(g*D);
    NF=round(NF,4)
    %Identificación del tipo de flujo
    if NF>1
        TF='Supercrítico'
    elseif NF<1
        TF='Subcrítico'
    else
        TF='Crítico'
    end
    %Cálculo de la energía específica
    E=y+((V^2)/(2*g))
    %Aproximación de resultados a 4 dígitos
    z=round(z,4);
    y=round(y,4);
    b=round(b,4);
    A=round(A,4);
    P=round(P,4);
    T=round(T,4);
    D=round(D,4);
    R=round(R,4);
    V=round(V,4);
    E=round(E,4);
    %Impresión de resultados
    close ESPERA
    set(handles.edit2,'String',z);
    set(handles.edit6,'String',y);
    set(handles.text51,'String',b);
    set(handles.edit9,'String',A);
    set(handles.edit7,'String',P);
    set(handles.edit8,'String',R);
    set(handles.edit15,'String',T);
    set(handles.edit10,'String',D);
    set(handles.edit13,'String',V);
    set(handles.edit11,'String',NF);
    set(handles.edit12,'String',TF);
    set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1

```



```

        set(handles.edit6,'String',y);
        set(handles.text51,'String',b);
        set(handles.edit9,'String',A);
        set(handles.edit7,'String',P);
        set(handles.edit8,'String',R);
        set(handles.edit15,'String',T);
        set(handles.edit10,'String',D);
        set(handles.edit13,'String',V);
        set(handles.edit11,'String',NF);
        set(handles.edit12,'String',TF);
        set(handles.edit14,'String',E);
end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
set(handles.text51,'String','');

```

```

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit2,'String','');
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        set(handles.text51,'String','');
        %Mostrar unidades métricas;
        set(handles.text6,'Visible','on');
        set(handles.text9,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text26,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');
        set(handles.text23,'Visible','on');
        set(handles.text30,'Visible','on');
        set(handles.text52,'Visible','on');
        %Ocultar unidades inglesas;
        set(handles.text35,'Visible','off');
        set(handles.text37,'Visible','off');
        set(handles.text38,'Visible','off');
        set(handles.text39,'Visible','off');
        set(handles.text42,'Visible','off');
        set(handles.text43,'Visible','off');
        set(handles.text45,'Visible','off');
        set(handles.text41,'Visible','off');
        set(handles.text40,'Visible','off');
        set(handles.text44,'Visible','off');
        set(handles.text53,'Visible','off');
    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit2,'String','');
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        set(handles.text51,'String','');

```

```

%Ocultar unidades métricas;
set(handles.text6,'Visible','off');
set(handles.text9,'Visible','off');
set(handles.text19,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text25,'Visible','off');
set(handles.text26,'Visible','off');
set(handles.text33,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text30,'Visible','off');
set(handles.text52,'Visible','off');
%Mostrar unidades inglesas;
set(handles.text35,'Visible','on');
set(handles.text37,'Visible','on');
set(handles.text38,'Visible','on');
set(handles.text39,'Visible','on');
set(handles.text42,'Visible','on');
set(handles.text43,'Visible','on');
set(handles.text45,'Visible','on');
set(handles.text41,'Visible','on');
set(handles.text40,'Visible','on');
set(handles.text44,'Visible','on');
set(handles.text53,'Visible','on');
end
set(handles.uibuttongroup1,'Tag',SM1)

% --- Executes during object creation, after setting all properties.
function pushbutton1_CreateFcn(hObject, eventdata, handles)

```

2.1.5.3. SECCIÓN TRIANGULAR

```

function varargout = SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRIANGULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',           mfilename, ...
                  'gui_Singleton',      gui_Singleton, ...
                  'gui_OpeningFcn',     @SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRIANGULAR_OpeningFcn, ...
                  'gui_OutputFcn',      @SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRIANGULAR_OutputFcn, ...
                  'gui_LayoutFcn',      [], ...
                  'gui_Callback',       []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRIANGULAR is
made visible.
function SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRIANGULAR_OpeningFcn(hObject,
eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('TRIÁNGULO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

```

```

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout =
SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRIANGULAR_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit2, 'String', '');
set(handles.edit6, 'String', '');
set(handles.edit9, 'String', '');
set(handles.edit8, 'String', '');
set(handles.edit15, 'String', '');
set(handles.edit10, 'String', '');
set(handles.edit13, 'String', '');
set(handles.edit11, 'String', '');
set(handles.edit12, 'String', '');
set(handles.edit14, 'String', '');
%Obtener el dato del caudal
VAL1=get(handles.edit1, 'String');
Q=str2double(VAL1)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3, 'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4, 'String');
n=str2double(VAL4)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
S1=isnan(S);
n1=isnan(n);
VAL5=get(handles.uibuttongroup1, 'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.', 'Mensaje');
elseif S<=0
errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.', 'Mensaje');
elseif n<=0
errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.', 'Mensaje');
elseif Q1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif S1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
elseif n1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.', 'Mensaje');
else
ESPERA
%Cálculo del talud
z=1
%Cálculo del tirante
M=100000
err=0.000001
y=100
for i=1:M
FY=((1/n)*(z*(y^2))*(((z*(y^2))/(2*y*sqrt(1+(z^2))))^(2/3))*(S^(1/2)))-Q
if FY==0
break
else
FDY=(2*S^(1/2)*y*z*((y*z)/(2*(z^2 + 1)^(1/2)))^(2/3))/n +
(S^(1/2)*y^2*z^2)/(3*n*(z^2 + 1)^(1/2))*((y*z)/(2*(z^2 + 1)^(1/2)))^(1/3))
ER=FY/FDY
abs(ER)
y=y-ER

```

```

        if abs(ER) <= err
            break
        end
    end
end
end
%Valor de la gravedad en m/s2
g=9.81
%Cálculo del área mojada
A=z*(y^2)
%Cálculo del perímetro mojado
P=2*y*sqrt(1+(z^2))
%Cálculo del ancho superficial
T=2*z*y
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del radio hidráulico
R=A/P
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
z=round(z,4);
y=round(y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit2,'String',z);
set(handles.edit6,'String',y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que debe ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de calcular.','Mensaje');
elseif n1==1

```



```

        set(handles.edit14,'String',E);
    end
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

```

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRIANGULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```

        set(hObject,'BackgroundColor','white');
    end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject,'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit2,'String','');
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Mostrar unidades métricas;
        set(handles.text6,'Visible','on');
        set(handles.text9,'Visible','on');
        set(handles.text19,'Visible','on');
        set(handles.text22,'Visible','on');
        set(handles.text25,'Visible','on');
        set(handles.text26,'Visible','on');
        set(handles.text33,'Visible','on');
        set(handles.text24,'Visible','on');
        set(handles.text23,'Visible','on');
        set(handles.text30,'Visible','on');
        %Ocultar unidades inglesas;
        set(handles.text35,'Visible','off');
        set(handles.text37,'Visible','off');
        set(handles.text38,'Visible','off');
        set(handles.text39,'Visible','off');
        set(handles.text42,'Visible','off');
        set(handles.text43,'Visible','off');
        set(handles.text45,'Visible','off');
        set(handles.text41,'Visible','off');
        set(handles.text40,'Visible','off');
        set(handles.text44,'Visible','off');
    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit2,'String','');
        set(handles.edit6,'String','');
        set(handles.edit9,'String','');
        set(handles.edit7,'String','');
        set(handles.edit8,'String','');
        set(handles.edit15,'String','');
        set(handles.edit10,'String','');
        set(handles.edit13,'String','');
        set(handles.edit11,'String','');
        set(handles.edit12,'String','');
        set(handles.edit14,'String','');
        %Ocultar unidades métricas;
        set(handles.text6,'Visible','off');
        set(handles.text9,'Visible','off');
        set(handles.text19,'Visible','off');
        set(handles.text22,'Visible','off');
        set(handles.text25,'Visible','off');
        set(handles.text26,'Visible','off');
        set(handles.text33,'Visible','off');
        set(handles.text24,'Visible','off');
        set(handles.text23,'Visible','off');
        set(handles.text30,'Visible','off');
        %Mostrar unidades inglesas;
        set(handles.text35,'Visible','on');
        set(handles.text37,'Visible','on');
        set(handles.text38,'Visible','on');

```

```

        set(handles.text39,'Visible','on');
        set(handles.text42,'Visible','on');
        set(handles.text43,'Visible','on');
        set(handles.text45,'Visible','on');
        set(handles.text41,'Visible','on');
        set(handles.text40,'Visible','on');
        set(handles.text44,'Visible','on');
    end
    set(handles.uibuttongroup1,'Tag',SM1)

% --- Executes during object creation, after setting all properties.
function pushbutton1_CreateFcn(hObject, eventdata, handles)

```

2.1.5.4. SECCIÓN SEMICIRCULAR

```

function varargout = SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_SEMICIRCULAR(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_SEMICIRCULAR_OpeningFcn, ...
                  'gui_OutputFcn',   @SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_SEMICIRCULAR_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_SEMICIRCULAR is
made visible.
function SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_SEMICIRCULAR_OpeningFcn(hObject,
eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('SEMICIRCULO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout =
SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_SEMICIRCULAR_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%Borrar celdas de resultados
set(handles.edit2,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');

```

```

set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');
%Obtener el dato del caudal
VAL1=get(handles.edit1,'String');
Q=str2double(VAL1)
%Obtener el dato de la pendiente
VAL3=get(handles.edit3,'String');
S=str2double(VAL3)
%Obtener el dato del coeficiente n
VAL4=get(handles.edit4,'String');
n=str2double(VAL4)
%VARIABLES PARA COMPROBAR SI LOS VALORES SON INGREADOS CORRECTAMENTE
Q1=isnan(Q);
S1=isnan(S);
n1=isnan(n);
VAL5=get(handles.uibuttongroup1,'Tag');
SM1=str2double(VAL5)
if SM1==1
%Para unidades del sistema métrico
if Q<=0
errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif S<=0
errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
ESPERA
%Cálculo del ángulo
a=pi
%Cálculo del diámetro
M=100000
err=0.000001
y=100
for i=1:M
FY=(1/n)*((1/8)*(a-sin(a))*(y^2))*(((1/8)*(a-
sin(a))*(y^2))/((1/2)*a*y)^(2/3))*(S^(1/2))-Q
if FY==0
break
else
FDY=(S^(1/2)*y*(a - sin(a))*((y*(a - sin(a)))/(4*a))^(2/3))/(4*n) +
(S^(1/2)*y^2*(a - sin(a))^2)/(48*a*n*((y*(a - sin(a)))/(4*a))^(1/3))
ER=FY/FDY
abs(ER)
y=y-ER
if abs(ER)<=err
break
end
end
end
%Cálculo del diámetro
d=y
%Valor de la gravedad en m/s2
g=9.81
%Cálculo del tirante
Y=(d/2)
%Cálculo del area mojada
A=(1/8)*(a-sin(a))*(d^2)
%Cálculo del perimetro mojado
P=(1/2)*a*d

```

```

%Cálculo del ancho superficial
T=d*sin(a/2)
%Cálculo de la profundidad hidráulica
D=A/T
%Cálculo del radio hidráulico
R=A/P
%Cálculo de la velocidad
V=Q/A
%Cálculo del número de froude
NF=V/sqrt(g*D);
NF=round(NF,4)
%Identificación del tipo de flujo
if NF>1
    TF='Supercrítico'
elseif NF<1
    TF='Subcrítico'
else
    TF='Crítico'
end
%Cálculo de la energía específica
E=Y+((V^2)/(2*g))
%Aproximación de resultados a 4 dígitos
d=round(d,4);
Y=round(Y,4);
A=round(A,4);
P=round(P,4);
T=round(T,4);
D=round(D,4);
R=round(R,4);
V=round(V,4);
E=round(E,4);
%Impresión de resultados
close ESPERA
set(handles.edit2,'String',d);
set(handles.edit6,'String',Y);
set(handles.edit9,'String',A);
set(handles.edit7,'String',P);
set(handles.edit8,'String',R);
set(handles.edit15,'String',T);
set(handles.edit10,'String',D);
set(handles.edit13,'String',V);
set(handles.edit11,'String',NF);
set(handles.edit12,'String',TF);
set(handles.edit14,'String',E);
end
else
%Para unidades del sistema inglés
if Q<=0
    errordlg('Ingrese el valor del caudal (Q) teniendo en cuenta que debe ser mayor
que 0.','Mensaje');
elseif S<=0
    errordlg('Ingrese el valor de la pendiente (S) teniendo en cuenta que debe ser
mayor que 0.','Mensaje');
elseif n<=0
    errordlg('Ingrese el valor del coeficiente de manning (n) teniendo en cuenta que
debe ser mayor que 0.','Mensaje');
elseif Q1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif S1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
elseif n1==1
    errordlg('Complete los campos de entrada adecuadamente antes de
calcular.','Mensaje');
else
    ESPERA
    %Cálculo del ángulo
    a=pi
    %Cálculo del diámetro
    M=100000
    err=0.000001
    y=100
    for i=1:M

```



```

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)

```



```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%Limpiar celdas
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit6,'String','');
set(handles.edit9,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit15,'String','');
set(handles.edit10,'String','');
set(handles.edit13,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit14,'String','');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
INICIO
close SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_SEMICIRCULAR

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on edit6 and none of its controls.
function edit6_KeyPressFcn(hObject, eventdata, handles)

```

```

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
SM=get(hObject, 'String');
switch SM
    case 'Sistema métrico'
        SM1='1'
        %Borrar celdas de resultados
        set(handles.edit2, 'String', '');
        set(handles.edit6, 'String', '');
        set(handles.edit9, 'String', '');
        set(handles.edit7, 'String', '');
        set(handles.edit8, 'String', '');
        set(handles.edit15, 'String', '');
        set(handles.edit10, 'String', '');
        set(handles.edit13, 'String', '');
        set(handles.edit11, 'String', '');
        set(handles.edit12, 'String', '');
        set(handles.edit14, 'String', '');
        %Mostrar unidades métricas;
        set(handles.text6, 'Visible', 'on');
        set(handles.text7, 'Visible', 'on');
        set(handles.text9, 'Visible', 'on');
        set(handles.text19, 'Visible', 'on');
        set(handles.text22, 'Visible', 'on');
        set(handles.text25, 'Visible', 'on');
        set(handles.text26, 'Visible', 'on');
        set(handles.text33, 'Visible', 'on');
        set(handles.text24, 'Visible', 'on');
        set(handles.text23, 'Visible', 'on');
        set(handles.text30, 'Visible', 'on');
        %Ocultar unidades inglesas;
        set(handles.text35, 'Visible', 'off');
        set(handles.text36, 'Visible', 'off');
        set(handles.text37, 'Visible', 'off');
        set(handles.text38, 'Visible', 'off');
        set(handles.text39, 'Visible', 'off');
        set(handles.text42, 'Visible', 'off');
        set(handles.text43, 'Visible', 'off');
        set(handles.text45, 'Visible', 'off');
        set(handles.text41, 'Visible', 'off');
        set(handles.text40, 'Visible', 'off');
        set(handles.text44, 'Visible', 'off');
    case 'Sistema inglés'
        SM1='2'
        %Borrar celdas de resultados
        set(handles.edit2, 'String', '');
        set(handles.edit6, 'String', '');
        set(handles.edit9, 'String', '');
        set(handles.edit7, 'String', '');
        set(handles.edit8, 'String', '');
        set(handles.edit15, 'String', '');
        set(handles.edit10, 'String', '');
        set(handles.edit13, 'String', '');
        set(handles.edit11, 'String', '');
        set(handles.edit12, 'String', '');
        set(handles.edit14, 'String', '');
        %Ocultar unidades métricas;
        set(handles.text6, 'Visible', 'off');
        set(handles.text7, 'Visible', 'off');
        set(handles.text9, 'Visible', 'off');
        set(handles.text19, 'Visible', 'off');
        set(handles.text22, 'Visible', 'off');
        set(handles.text25, 'Visible', 'off');
        set(handles.text26, 'Visible', 'off');
        set(handles.text33, 'Visible', 'off');
        set(handles.text24, 'Visible', 'off');
        set(handles.text23, 'Visible', 'off');
        set(handles.text30, 'Visible', 'off');
        %Mostrar unidades inglesas;
        set(handles.text35, 'Visible', 'on');
        set(handles.text36, 'Visible', 'on');
        set(handles.text37, 'Visible', 'on');
        set(handles.text38, 'Visible', 'on');
        set(handles.text39, 'Visible', 'on');

```

```

        set(handles.text42,'Visible','on');
        set(handles.text43,'Visible','on');
        set(handles.text45,'Visible','on');
        set(handles.text41,'Visible','on');
        set(handles.text40,'Visible','on');
        set(handles.text44,'Visible','on');
    end
    set(handles.uibuttongroup1,'Tag',SM1)

% --- Executes during object creation, after setting all properties.
function pushbutton1_CreateFcn(hObject, eventdata, handles)

```

2.1.6. PANTALLA DE INCIO

```

function varargout = INICIO(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @INICIO_OpeningFcn, ...
                  'gui_OutputFcn',  @INICIO_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before INICIO is made visible.
function INICIO_OpeningFcn(hObject, eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('INICIO.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = INICIO_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function TIRANTE_NORMAL_Callback(hObject, eventdata, handles)

function TIRANTE_CRITICO_Callback(hObject, eventdata, handles)

function CAUDALES_Callback(hObject, eventdata, handles)

function ELEMENTOS_GEOMETRICOS_Callback(hObject, eventdata, handles)

function Untitled_6_Callback(hObject, eventdata, handles)

function Untitled_18_Callback(hObject, eventdata, handles)
SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_RECTANGULAR
close INICIO

function Untitled_19_Callback(hObject, eventdata, handles)

```

```

SELECCION_2
close INICIO

function Untitled_20_Callback(hObject, eventdata, handles)
SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRIANGULAR
close INICIO

function Untitled_21_Callback(hObject, eventdata, handles)
SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_SEMICIRCULAR
close INICIO

function Untitled_7_Callback(hObject, eventdata, handles)
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR
close INICIO

function Untitled_8_Callback(hObject, eventdata, handles)
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRAPEZOIDAL
close INICIO

function Untitled_9_Callback(hObject, eventdata, handles)
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR
close INICIO

function Untitled_10_Callback(hObject, eventdata, handles)
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_CIRCULAR
close INICIO

function Untitled_11_Callback(hObject, eventdata, handles)
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_PARABOLICA
close INICIO

function Untitled_12_Callback(hObject, eventdata, handles)
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_RECTANGULAR_REDONDEADO
close INICIO

function Untitled_15_Callback(hObject, eventdata, handles)
ELEMENTOS_GEOMETRICOS_DE_UNA_SECCION_TRIANGULAR_REDONDEADO
close INICIO

function Untitled_37_Callback(hObject, eventdata, handles)
CAUDAL_DE_UNA_SECCION_RECTANGULAR
close INICIO

function Untitled_38_Callback(hObject, eventdata, handles)
CAUDAL_DE_UNA_SECCION_TRAPEZOIDAL
close INICIO

function Untitled_39_Callback(hObject, eventdata, handles)
CAUDAL_DE_UNA_SECCION_TRIANGULAR
close INICIO

function Untitled_40_Callback(hObject, eventdata, handles)
CAUDAL_DE_UNA_SECCION_CIRCULAR
close INICIO

function Untitled_41_Callback(hObject, eventdata, handles)
CAUDAL_DE_UNA_SECCION_PARABOLICA
close INICIO

function Untitled_42_Callback(hObject, eventdata, handles)
CAUDAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA
close INICIO

function Untitled_43_Callback(hObject, eventdata, handles)
CAUDAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO
close INICIO

function Untitled_30_Callback(hObject, eventdata, handles)
TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR
close INICIO

function Untitled_31_Callback(hObject, eventdata, handles)
TIRANTE_CRITICO_DE_UNA_SECCION_TRAPEZOIDAL
close INICIO

```

```

function Untitled_32_Callback(hObject, eventdata, handles)
TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR
close INICIO

function Untitled_33_Callback(hObject, eventdata, handles)
TIRANTE_CRITICO_DE_UNA_SECCION_CIRCULAR
close INICIO

function Untitled_34_Callback(hObject, eventdata, handles)
SELECCION_1
close INICIO

function Untitled_35_Callback(hObject, eventdata, handles)
TIRANTE_CRITICO_DE_UNA_SECCION_RECTANGULAR_REDONDEADA
close INICIO

function Untitled_36_Callback(hObject, eventdata, handles)
TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR_REDONDEADO
close INICIO

function Untitled_22_Callback(hObject, eventdata, handles)
TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR
close INICIO

function Untitled_23_Callback(hObject, eventdata, handles)
TIRANTE_NORMAL_DE_UNA_SECCION_TRAPEZOIDAL
close INICIO

function Untitled_24_Callback(hObject, eventdata, handles)
TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR
close INICIO

function Untitled_25_Callback(hObject, eventdata, handles)
TIRANTE_NORMAL_DE_UNA_SECCION_CIRCULAR
close INICIO

function Untitled_26_Callback(hObject, eventdata, handles)
SELECCION
close INICIO

function Untitled_27_Callback(hObject, eventdata, handles)
TIRANTE_NORMAL_DE_UNA_SECCION_RECTANGULAR_REDONDEADA
close INICIO

function Untitled_28_Callback(hObject, eventdata, handles)
TIRANTE_NORMAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO
close INICIO

function pushbutton4_Callback(hObject, eventdata, handles)
close all

function AYUDA_Callback(hObject, eventdata, handles)

function Untitled_44_Callback(hObject, eventdata, handles)
ACERCA_DE_SN_CANALES
close INICIO

function Untitled_45_Callback(hObject, eventdata, handles)
open('MANUAL_DE_USUARIO.PDF')

```

2.1.7. SELECCIÓN

```

function varargout = SELECCION(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @SELECCION_OpeningFcn, ...
                  'gui_OutputFcn',  @SELECCION_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})

```

```

        gui_State.gui_Callback = str2func(varargin{1});
    end

    if nargin
        [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
    else
        gui_mainfcn(gui_State, varargin{:});
    end

    % --- Executes just before SELECCION is made visible.
    function SELECCION_OpeningFcn(hObject, eventdata, handles, varargin)
    %Código para centrar la ventana
    scrsz=get(0,'ScreenSize');
    pos_act=get(gcf,'Position');
    xr=scrsz(3)-pos_act(3);
    xp=round(xr/2);
    yr=scrsz(4)-pos_act(4);
    yp=round(yr/2);
    set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
    handles.output = hObject;
    guidata(hObject, handles);

    % --- Outputs from this function are returned to the command line.
    function varargout = SELECCION_OutputFcn(hObject, eventdata, handles)
    varargout{1} = handles.output;

    % --- Executes on button press in pushbutton1.
    function pushbutton1_Callback(hObject, eventdata, handles)
    TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA
    close SELECCION

    % --- Executes on button press in pushbutton2.
    function pushbutton2_Callback(hObject, eventdata, handles)
    TIRANTE_NORMAL_DE_UNA_SECCION_PARABOLICA_2
    close SELECCION

```

2.1.8. SELECCIÓN 1

```

function varargout = SELECCION_1(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @SELECCION_1_OpeningFcn, ...
                  'gui_OutputFcn',  @SELECCION_1_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before SELECCION_1 is made visible.
function SELECCION_1_OpeningFcn(hObject, eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = SELECCION_1_OutputFcn(hObject, eventdata, handles)

```

```

varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
TIRANTE_CRITICO_DE_UNA_SECCION_PARABOLICA
close SELECCION_1

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
TIRANTE_CRITICO_DE_UNA_SECCION_PARABOLICA_2
close SELECCION_1

```

2.1.9. SELECCIÓN 2

```

function varargout = SELECCION_2(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @SELECCION_2_OpeningFcn, ...
                  'gui_OutputFcn',   @SELECCION_2_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before SELECCION_2 is made visible.
function SELECCION_2_OpeningFcn(hObject, eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = SELECCION_2_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL_2
close SELECCION_2

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_TRAPEZOIDAL
close SELECCION_2

```

2.1.10. ESPERA

```

function varargout = ESPERA(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @ESPERA_OpeningFcn, ...
                  'gui_OutputFcn',   @ESPERA_OutputFcn, ...

```

```

        'gui_LayoutFcn', [] , ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before ESPERA is made visible.
function ESPERA_OpeningFcn(hObject, eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);

handles.output = hObject;
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = ESPERA_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

```

2.1.11. ACERCA DE SN CANALES

```

function varargout = ACERCA_DE_SN_CANALES(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @ACERCA_DE_SN_CANALES_OpeningFcn, ...
                  'gui_OutputFcn',  @ACERCA_DE_SN_CANALES_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before ACERCA_DE_SN_CANALES is made visible.
function ACERCA_DE_SN_CANALES_OpeningFcn(hObject, eventdata, handles, varargin)
%Código para centrar la ventana
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
%Código para leer la imagen
imagen=imread('ACERCA.JPG');
%Código para mostrar la imagen
image(imagen,'Parent',handles.axes1);
%Código para quitar ejes a axes
set(handles.axes1,'XTick',[],'YTick',[]);
% Choose default command line output for ACERCA_DE_SN_CANALES
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);

```



```
% --- Outputs from this function are returned to the command line.
function varargout = ACERCA_DE_SN_CANALES_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
INICIO
close ACERCA_DE_SN_CANALES
```

2.2. MANUAL DE USUARIO

MANUAL DE USUARIO SN CANALES

INTRODUCCIÓN

El programa fue creado para el cálculo de canales abiertos de flujo uniforme de las secciones planteadas por Ven Te Chow como son rectangular, trapezoidal, triangular, circular, parabólica, rectangular con esquinas redondeadas y triangular con fondo redondeado; para así contribuir en la disminución del tiempo de cálculo y la confiabilidad de los resultados.

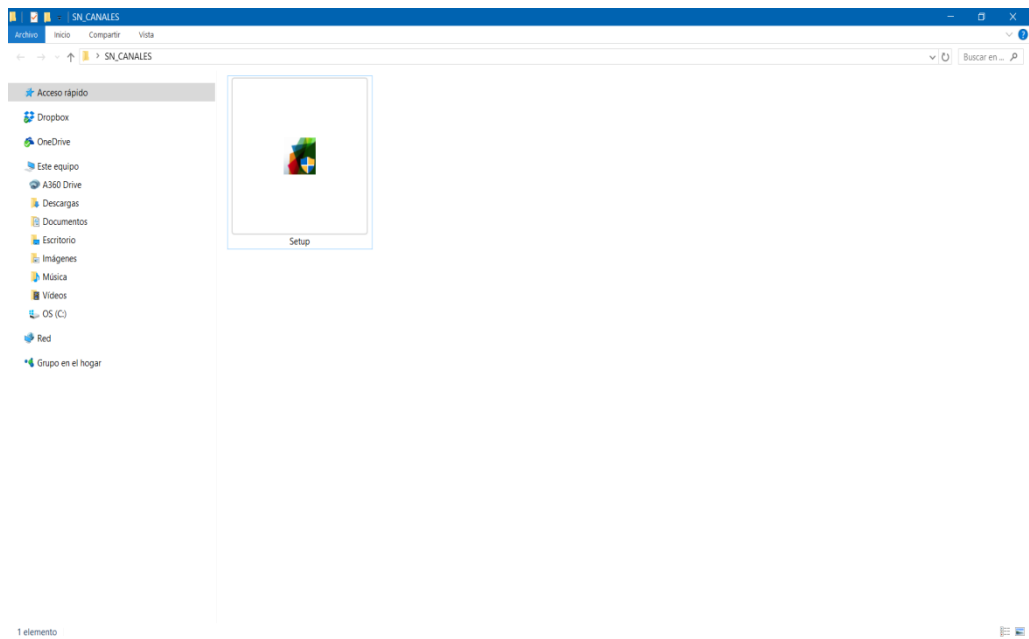
Con el programa se puede calcular el tirante normal, tirante crítico, elementos geométricos, caudal, sección hidráulica óptima, número de Froude, tipo de flujo y energía específica.

En el manual encontraremos una guía sobre la instalación del programa, además de ejercicios resueltos mediante el software.

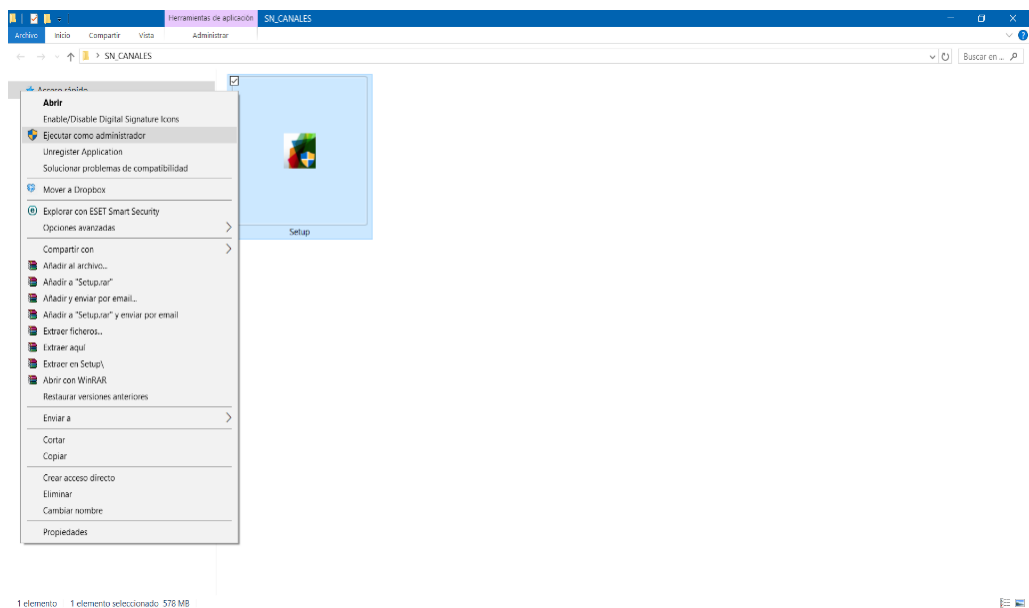
1. INSTALACION DE SN CANALES

Para la instalación de SN CANALES en su computador, seguir el proceso que se indica:

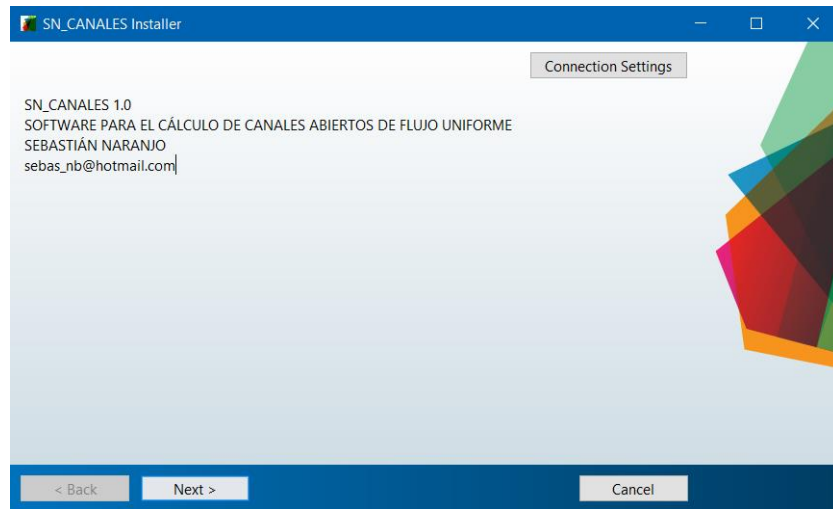
- 1) Abri la carpeta que contiene el instalador.



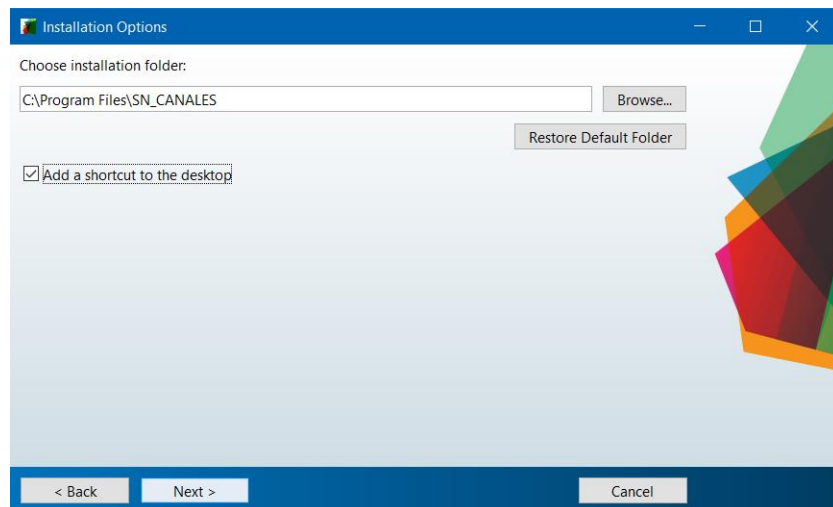
- 2) Ubicar el archivo *Setup* y ejecutar como administrador.



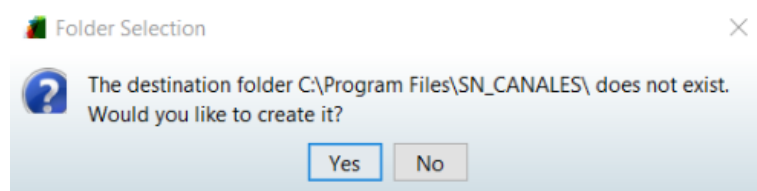
3) Cuando aparezca la pantalla que se muestra a continuación, dar clic *Next*.



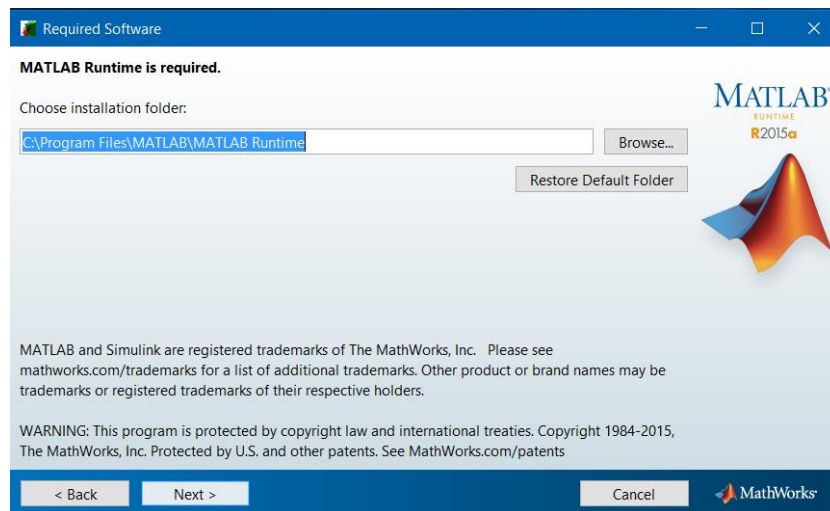
4) En la siguiente ventana marcar la casilla *Add a shortcut to the desktop*, para crear un acceso directo del programa en el escritorio y dar clic en *Next*.



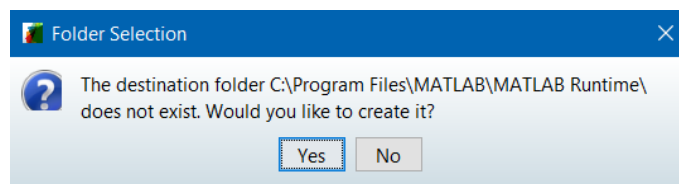
5) Dar clic en *Yes* para que se cree la carpeta donde se instalara el programa SN CANALES.



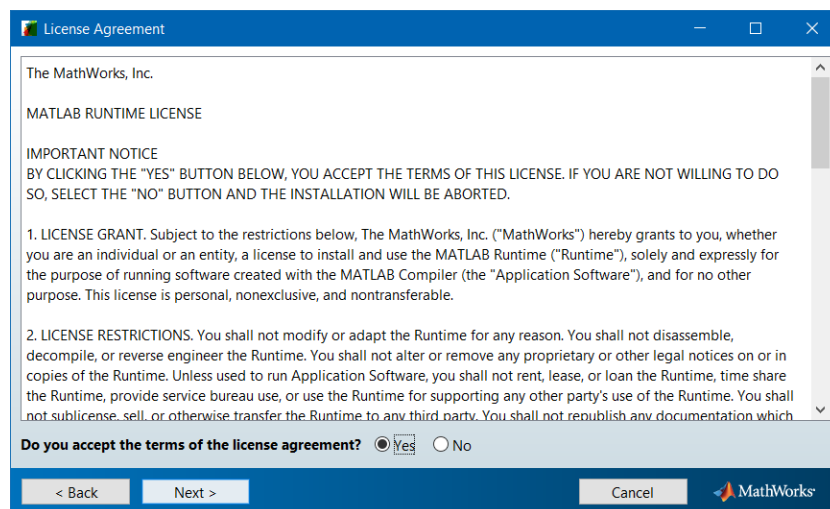
6) En la siguiente ventana dar clic en *Next*.



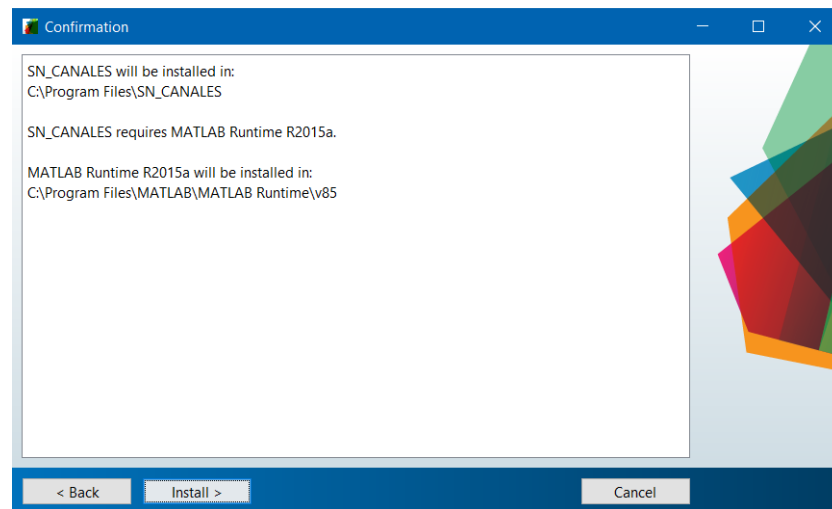
7) Dar clic en *Yes* para que se cree la carpeta donde se instalara MATLAB Runtime.



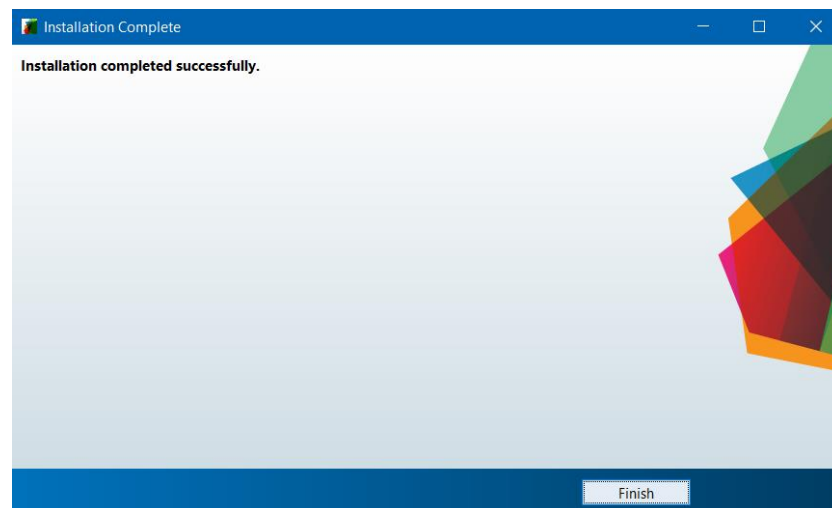
8) Aceptar los términos de licencia de Matlab Runtime marcando en *Yes* y dar clic en *Next*.



9) En la siguiente ventana que aparece dar clic en **Install**.



10) Después de este proceso, se inicia con la instalación y al culminar se muestra la siguiente pantalla indicando la culminación satisfactoria de SN CANALES en la cual se debe dar clic en **Finish**.



11) Con este proceso se tiene instalado y listo para ejecutar SN CANALES. El programa de instalación, crea el acceso directo dentro de **INICIO / TODAS LAS APLICACIONES**.

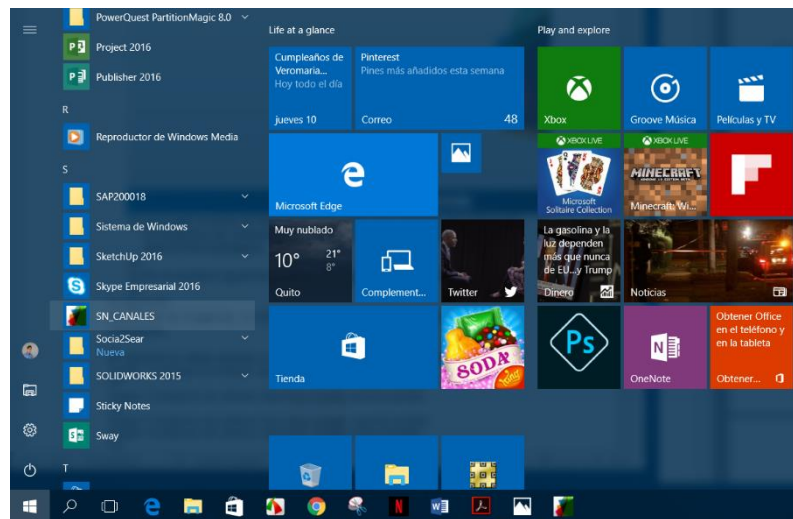
2. SOLUCIÓN A PROBLEMAS EN LA INSTALACIÓN

Cualquier consulta, sobre problemas que se puedan suscitar en la instalación puede hacerlo al e-mail: sebas_nb@hotmail.com

3. EJECUTAR SN CANALES

1) Para ejecutar SN CANLES, realizar las órdenes:

INICIO / TODAS LAS APLICACIONES / SNCANALES, como se muestra en la figura.



2) Después de ejecutar estas órdenes, se obtiene la imagen de presentación que nos indica que el programa a empezado a ejecutarse.



3) Una vez que se ha cargado se obtiene la pantalla principal de SN CANALES.



4) En este momento, SN CANALES está a su disposición para brindarle ayuda en sus cálculos necesarios para el diseño de canales abiertos de flujo uniforme.

4. CÁLCULOS CON SN CANALES.

En la pantalla principal se encuentran las funciones principales del software, accesibles mediante menús desplegables.

Desde los menús desplegables se tiene acceso a las diferentes pantallas que permiten el cálculo del:

- Tirante normal
- Tirante crítico
- Caudal
- Elementos geométricos
- Sección hidráulica óptima

En las diferentes pantallas de cálculo se ingresan los valores numéricos correspondientes a los datos necesarios para efectuar los cálculos correspondientes.

4.1. TIRANTE NORMAL

Para el cálculo del tirante normal, el software dispone de siete opciones, que corresponden a secciones transversales de canales como son: rectangular, trapezoidal, triangular, circular, parabólica, rectangular con esquinas redondeadas y triangular con fondo redondeado.

Para acceder a las ventanas de cálculo se debe seleccionar del menú principal la opción tirante normal en la cual se presenta un submenú como el mostrado en la siguiente imagen.



Para el cálculo en cualquiera de las secciones disponibles se debe de dar clic sobre la sección elegida, donde a continuación aparecerá la pantalla correspondiente para su cálculo.

Los datos de entrada dependerán de la sección transversal seleccionada como son:

- Sección rectangular: Caudal (Q), ancho de la solera (b), pendiente (S) y coeficiente de manning (n).
- Sección trapezoidal: Caudal (Q), ancho de la solera (b), talud (z), pendiente (S) y coeficiente de manning (n).

- Sección triangular: Caudal (Q), talud (z), pendiente (S) y coeficiente de manning (n).
- Sección circular: Caudal (Q), diámetro (d), pendiente (S) y coeficiente de manning (n).
- Sección parabólica: Caudal (Q), ancho superficial (T), pendiente (S) y coeficiente de manning (n).
- Sección rectangular con esquinas redondeadas: Caudal (Q), ancho de la solera (b), radio (r), pendiente (S) y coeficiente de manning (n).
- Sección triangular con fondo redondeado: Caudal (Q), radio (r), talud (z), pendiente (S) y coeficiente de manning (n).

En cada pantalla de cálculo se incorpora una imagen de referencia de acuerdo a la sección transversal.

Los datos son ingresados tecleando sobre cada casilla el valor correspondiente, además se puede utilizar la tecla Tab para usuarios que omitan utilizar el mouse.

Las unidades de los datos de entrada y de salida dependerán del sistema de unidades que se escoja para el cálculo, para lo cual tenemos el sistema métrico e inglés, que deberá ser seleccionado antes de calcular.

Los datos de salida que son calculados son:

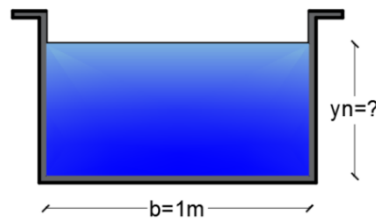
- Tirante normal (y)
- Área mojada (A)
- Perímetro mojado (P)
- Radio hidráulico (R)
- Ancho superficial (T)
- Profundidad hidráulica (D)
- Velocidad (V)
- Número de Froude (NF)
- Tipo de flujo
- Energía específica (E)

Para iniciar el cálculo se debe presionar el botón “CALCULAR”, además se dispone del botón “LIMPIAR” que permite borrar los datos de entrada y salida, y del botón “MENÚ PRINCIPAL” el cual permite al usuario salir de la ventana actual y volver a la pantalla principal.

4.1.1. EJEMPLO DE CÁLCULO DE UNA SECCIÓN RECTAGULAR.

Se desea construir un canal revestido con concreto ($n=0.014$) de sección rectangular con un ancho de solera de 1 m. Calcular el tirante normal, para que pueda transportar un caudal de $2\text{ m}^3/\text{s}$, si es trazado con una pendiente del 2%.

Datos del problema:



$$Q=2\text{ m}^3/\text{s}; b=1.5\text{m}; S=0.02; n=0.014$$

Uso de SN CANALES:

CÁLCULO DEL TIRANTE NORMAL DE UNA SECCIÓN RECTAGULAR

Datos de entrada:

Caudal (Q)=	2	m ³ /s
Ancho de la solera (b)=	1.5	m
Pendiente (S)=	0.02	m/m
Coefficiente de Manning (n)=	0.014	

CALCULAR **LIMPIAR**

Datos de salida:

Tirante normal (y)=	0.3452	m
Área mojada (A)=	0.5178	m ²
Perímetro mojado (P)=	2.1905	m
Radio hidráulico (R)=	0.2364	m
Ancho superficial (T)=	1.5	m
Profundidad hidráulica (D)=	0.3452	m
Velocidad (V)=	3.8622	m/s
Número de Froude (NF)=	2.0987	
Tipo de flujo=	Supercrítico	
Energía específica (E)=	1.1055	m-Kg/Kg

Sistema de unidades:

Sistema métrico

Sistema inglés

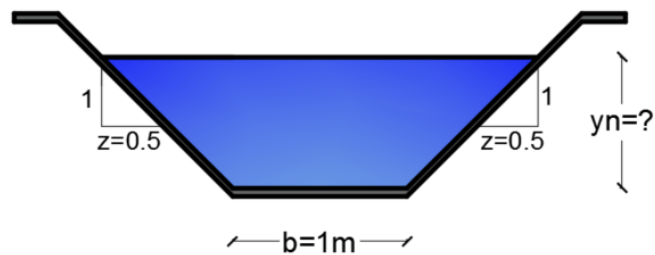
MENÚ PRINCIPAL

Diagrama de un canal rectangular con un ancho de solera b , una altura de agua y , y un ancho superficial T .

4.1.2. EJEMPLO DE CÁLCULO DE UNA SECCIÓN TRAPEZOIDAL.

Un canal de sección trapezoidal, con un ancho de solera 1 m y un talud $z=0.5$, se traza con una pendiente del 1%, y se construirá revestido de concreto ($n=0.014$). Calcular el tirante normal, para que pueda transportar un caudal de $5\text{ m}^3/\text{s}$.

Datos del problema:



$$Q=5\text{ m}^3/\text{s}; b=1\text{ m}; z=0.5; S=0.01; n=0.014$$

Uso de SN CANALES:

CÁLCULO DEL TIRANTE NORMAL DE UNA SECCIÓN TRAPEZOIDAL

Datos de entrada:

Caudal (Q)=	5	m ³ /s
Ancho de la solera (b)=	1	m
Talud (z)=	0.5	
Pendiente (S)=	0.01	m/m
Coficiente de Manning (n)=	0.014	

CALCULAR LIMPIAR

Datos de salida:

Tirante normal (y)=	0.8668	m
Área mojada (A)=	1.2425	m ²
Perímetro mojado (P)=	2.9383	m
Radio hidráulico (R)=	0.4229	m
Ancho superficial (T)=	1.8668	m
Profundidad hidráulica (D)=	0.6656	m
Velocidad (V)=	4.0242	m/s
Número de Froude (NF)=	1.5749	
Tipo de flujo=	Supercrítico	
Energía específica (E)=	1.6922	m-Kg/Kg

Sistema de unidades:

Sistema métrico
 Sistema inglés

MENÚ PRINCIPAL

Diagrama de una sección trapezoidal de un canal con el tirante normal (y) y el ancho superficial (T) indicados.

4.1.3. EJEMPLO DE CÁLCULO DE UNA SECCIÓN TRIANGULAR.

Se desea calcular el tirante normal, de un canal de sección triangular con un talud $z=1$, sabiendo que el coeficiente de rugosidad es $n=0.0149$, está trazado con una pendiente del 1.5%, y conduce un caudal de $3\text{ m}^3/\text{s}$.

Datos del problema:



$$Q=3\text{ m}^3/\text{s}; z=1; S=0.015; n=0.0149$$

Uso de SN CANALES:

CÁLCULO DEL TIRANTE NORMAL DE UNA SECCIÓN TRIANGULAR

Datos de entrada:

Caudal (Q)=	3	m ³ /s
Talud (z)=	1	
Pendiente (S)=	0.015	m/m
Coficiente de Manning (n)=	0.0149	

Datos de salida:

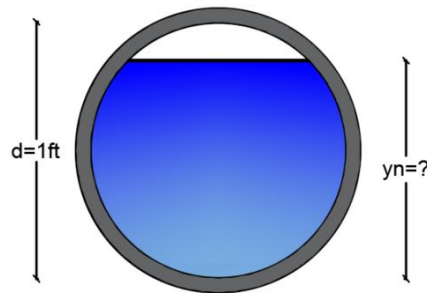
Tirante normal (y)=	0.8887	m
Área mojada (A)=	0.7897	m ²
Perímetro mojado (P)=	2.5135	m
Radio hidráulico (R)=	0.3142	m
Ancho superficial (T)=	1.7773	m
Profundidad hidráulica (D)=	0.4443	m
Velocidad (V)=	3.7989	m/s
Número de Froude (NF)=	1.8196	
Tipo de flujo=	Supercrítico	
Energía específica (E)=	1.6242	m-Kg/Kg

Sistema de unidades:
 Sistema métrico
 Sistema inglés

4.1.4. EJEMPLO DE CÁLCULO DE UNA SECCIÓN CIRCULAR.

Se desea calcular el tirante normal, de un canal de sección circular con diámetro de 1ft, sabiendo que el coeficiente de rugosidad de la tubería es $n=0.0149$, y está trazado con una pendiente del 2%, y conduce un caudal de $1\text{ft}^3/\text{s}$.

Datos del problema:



$$Q=1\text{ft}^3/\text{s}; d=1\text{ft}; S=0.02; n=0.0149$$

Uso de SN CANALES:

CÁLCULO DEL TIRANTE NORMAL DE UNA SECCIÓN CIRCULAR

Datos de entrada:

Caudal (Q)=	1	ft ³ /s
Diámetro (d)=	1	ft
Pendiente (S)=	0.02	ft/ft
Coficiente de Manning (n)=	0.0149	

CALCULAR LIMPIAR

Datos de salida:

Tirante normal (y)=	0.3243	ft
Área mojada (A)=	0.2207	ft ²
Perímetro mojado (P)=	1.2117	ft
Radio hidráulico (R)=	0.1821	ft
Ancho superficial (T)=	0.9362	ft
Profundidad hidráulica (D)=	0.2357	ft
Velocidad (V)=	4.5316	ft/s
Número de Froude (NF)=	1.6453	
Tipo de flujo=	Supercrítico	
Energía específica (E)=	0.6433	ft-lb/lb

Diagrama de una sección circular de un canal. El diámetro total está etiquetado como d . El nivel del agua está etiquetado como y . El ancho superficial del agua está etiquetado como T . El ángulo subtendido por el agua está etiquetado como θ . El interior del canal está sombreado de azul.

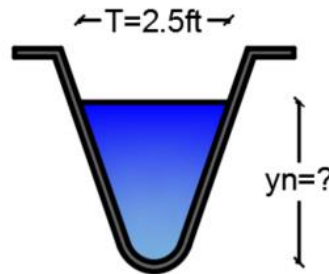
Sistema de unidades:
 Sistema métrico
 Sistema inglés

MENÚ PRINCIPAL

4.1.5. EJEMPLO DE CÁLCULO DE UNA SECCIÓN PARABÓLICA.

Se desea diseñar un canal con sección parabólico para conducir un caudal de $2 \text{ ft}^3/\text{s}$, la misma que se construirá con un coeficiente de rugosidad $n=0.025$, trazado con una pendiente del 1.5%, y que tenga un espejo de agua de 2.5ft. Indicar el tirante normal que debe tener, su velocidad, número de Froude, tipo de flujo y su energía específica para las condiciones señaladas

Datos del problema:



$$Q=2 \text{ ft}^3/\text{s}; T=2.5 \text{ ft}; S=0.015; n=0.025$$

Uso de SN CANALES:

CÁLCULO DEL TIRANTE NORMAL DE UNA SECCIÓN PARABÓLICA

Datos de entrada:		
Caudal (Q)=	2	ft ³ /s
Ancho Superficial (T)=	2.5	ft
Pendiente (S)=	0.015	ft/ft
Coefficiente de Manning (n)=	0.025	

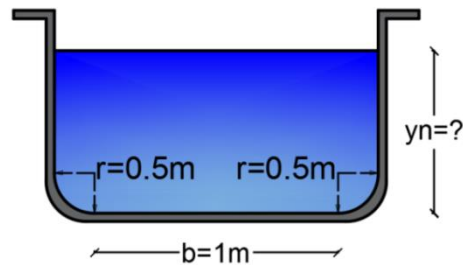
Datos de salida:		
Tirante normal (y)=	0.4099	ft
Área mojada (A)=	0.6831	ft ²
Perímetro mojado (P)=	2.6792	ft
Radio hidráulico (R)=	0.255	ft
Profundidad hidráulica (D)=	0.2732	ft
Velocidad (V)=	2.9279	ft/s
Número de Froude (NF)=	0.9873	
Tipo de flujo=	Subcrítico	
Energía específica (E)=	0.543	ft-lb/lb

Sistema de unidades:
 Sistema métrico
 Sistema inglés

4.1.6. EJEMPLO DE CÁLCULO DE UNA SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS.

Calcular el tirante normal si se desea construir un canal con un coeficiente de rugosidad $n=0.02$ de sección rectangular con esquinas redondeadas con un radio $r=0.5$ m y ancho de solera de 1m. El caudal de diseño es de $3\text{ m}^3/\text{s}$ y está trazado con una pendiente del 0.003.

Datos del problema:



$$Q=3\text{ m}^3/\text{s}; b=1\text{m}; r=0.5\text{m}; S=0.003; n=0.02$$

Uso de SN CANALES:

CÁLCULO DEL TIRANTE NORMAL DE UNA SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS

Datos de entrada:

Caudal (Q)=	3	m ³ /s
Ancho de la solera (b)=	1	m
Radio (r)=	0.5	m
Pendiente (S)=	0.003	m/m
Coficiente de Manning (n)=	0.02	

CALCULAR **LIMPIAR**

Datos de salida:

Tirante normal (y)=	0.9154	m
Área mojada (A)=	1.7236	m ²
Perímetro mojado (P)=	3.4017	m
Radio hidráulico (R)=	0.5067	m
Ancho superficial (T)=	2	m
Profundidad hidráulica (D)=	0.8618	m
Velocidad (V)=	1.7406	m/s
Número de Froude (NF)=	0.5986	
Tipo de flujo=	Subcrítico	
Energía específica (E)=	1.0699	m-Kg/Kg

Sistema de unidades:

Sistema métrico

Sistema inglés

MENÚ PRINCIPAL

Diagrama de una sección de canal rectangular con esquinas redondeadas, similar al anterior, pero con el tirante normal y y el ancho superficial T indicados.

4.1.7. EJEMPLO DE CÁLCULO DE UNA SECCIÓN TRIANGULAR CON FONDO REDONDEADO.

Se desea calcular el tirante normal, de un canal de sección triangular con fondo redondeado con un radio $r=2$ ft, un talud $z=1$ sabiendo que el coeficiente de rugosidad de la tubería es $n=0.03$, y está trazado con una pendiente del 0.005 , y conduce un caudal de $9 \text{ ft}^3/\text{s}$.

Datos del problema:



$$Q=9 \text{ ft}^3/\text{s}; z=1; r=2 \text{ ft}; S=0.005; n=0.03$$

Uso de SN CANALES:

CÁLCULO DEL TIRANTE NORMAL DE UNA SECCIÓN TRIANGULAR CON FONDO REDONDEADO

Datos de entrada:

Caudal (Q)=	9	ft ³ /s
Radio (r)=	2	ft
Talud (z)=	1	
Pendiente (S)=	0.005	ft/ft
Coficiente de Manning (n)=	0.03	

Datos de salida:

Tirante normal (y)=	1.2196	ft
Área mojada (A)=	3.3358	ft ²
Perímetro mojado (P)=	4.9341	ft
Radio hidráulico (R)=	0.6761	ft
Ancho superficial (T)=	4.096	ft
Profundidad hidráulica (D)=	0.8144	ft
Velocidad (V)=	2.698	ft/s
Número de Froude (NF)=	0.527	
Tipo de flujo=	Subcrítico	
Energía específica (E)=	1.3326	ft-lb/lb

Sistema de unidades:

Sistema métrico
 Sistema inglés

4.2. TIRANTE CRÍTICO

Para el cálculo del tirante crítico, el software dispone de siete opciones, que corresponden a secciones transversales de canales como son: rectangular, trapezoidal, triangular, circular, parabólica, rectangular con esquinas redondeadas y triangular con fondo redondeado.

Para acceder a las ventanas de cálculo se debe seleccionar del menú principal la opción tirante normal en la cual se presenta un submenú como el mostrado en la siguiente imagen.



Para el cálculo en cualquiera de las secciones disponibles se debe de dar clic sobre la sección elegida, donde a continuación aparecerá la pantalla correspondiente para su cálculo.

Los datos de entrada dependerán de la sección transversal seleccionada como son:

- Sección rectangular: Caudal (Q), ancho de la solera (b).
- Sección trapezoidal: Caudal (Q), ancho de la solera (b), talud (z).
- Sección triangular: Caudal (Q), talud (z).

- Sección circular: Caudal (Q), diámetro (d).
- Sección parabólica: Caudal (Q), ancho superficial (T).
- Sección rectangular con esquinas redondeadas: Caudal (Q), ancho de la solera (b), radio (r).
- Sección triangular con fondo redondeado: Caudal (Q), radio (r), talud (z).

En cada pantalla de cálculo se incorpora una imagen de referencia de acuerdo a la sección transversal.

Los datos son ingresados tecleando sobre cada casilla el valor correspondiente, además se puede utilizar la tecla Tab para usuarios que omitan utilizar el mouse.

Las unidades de los datos de entrada y de salida dependerán del sistema de unidades que se escoja para el cálculo, para lo cual tenemos el sistema métrico e inglés, que deberá ser seleccionado antes de calcular.

Los datos de salida que son calculados son:

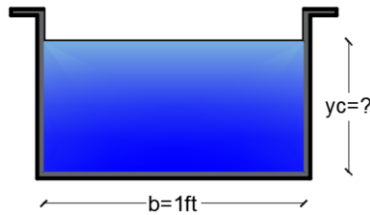
- Tirante crítico (y)
- Área mojada (A)
- Perímetro mojado (P)
- Radio hidráulico (R)
- Ancho superficial (T)
- Profundidad hidráulica (D)
- Velocidad (V)
- Número de Froude (NF)
- Tipo de flujo
- Energía específica (E)

Para iniciar el cálculo se debe presionar el botón “CALCULAR”, además se dispone del botón “LIMPIAR” que permite borrar los datos de entrada y salida, y del botón “MENÚ PRINCIPAL” el cual permite al usuario salir de la ventana actual y volver a la pantalla principal.

4.2.1. EJEMPLO DE CÁLCULO DE UNA SECCIÓN RECTANGULAR.

Calcular el tirante crítico, de un canal de sección rectangular con un ancho de solera de 1ft y que transporta un caudal de $2\text{ft}^3/\text{s}$.

Datos del problema:



$$Q=2\text{ft}^3/\text{s}; b=1\text{ft}$$

Uso de SN CANALES:

CÁLCULO DEL TIRANTE CRÍTICO DE UNA SECCIÓN RECTANGULAR

Datos de entrada:

Caudal (Q)=	2	ft ³ /s
Ancho de la solera (b)=	1	ft

CALCULAR LIMPIAR

Datos de salida:

Tirante crítico (y)=	0.499	ft
Área mojada (A)=	0.499	ft ²
Perímetro mojado (P)=	1.9981	ft
Radio hidráulico (R)=	0.2498	ft
Ancho superficial (T)=	1	ft
Profundidad hidráulica (D)=	0.499	ft
Velocidad (V)=	4.0077	ft/s
Número de Froude (NF)=	1	
Tipo de flujo=	Crítico	
Energía específica (E)=	0.7486	ft-lb/lb

Diagrama de un canal rectangular con un ancho de solera b , una altura de agua y , y un ancho superficial T .

Sistema de unidades:

Sistema métrico

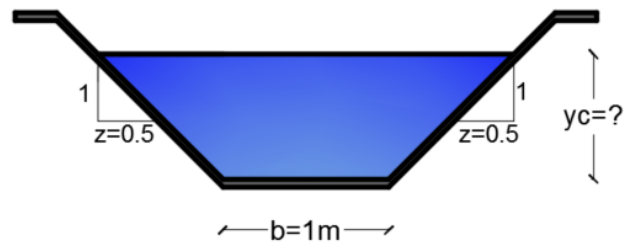
Sistema inglés

MENÚ PRINCIPAL

4.2.2. EJEMPLO DE CÁLCULO DE UNA SECCIÓN TRAPEZOIDAL.

Determinar el tirante crítico de un canal trapezoidal, con talud 0.5, ancho de solera de 1m y que conduce un caudal de $3\text{ m}^3/\text{s}$.

Datos del problema:



$$Q=3\text{ m}^3/\text{s}; b=1\text{ m}; z=0.5$$

Uso de SN CANALES:

CÁLCULO DEL TIRANTE CRÍTICO DE UNA SECCIÓN TRAPEZOIDAL

Datos de entrada:

Caudal (Q)= m³/s

Ancho de la solera (b)= m

Talud (z)=

Datos de salida:

Tirante crítico (y)= m

Área mojada (A)= m²

Perímetro mojado (P)= m

Radio hidráulico (R)= m

Ancho superficial (T)= m

Profundidad hidráulica (D)= m

Velocidad (V)= m/s

Número de Froude (NF)=

Tipo de flujo=

Energía específica (E)= m-Kg/Kg

Sistema de unidades:

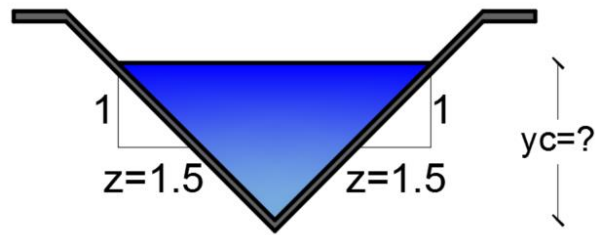
Sistema métrico

Sistema inglés

4.2.3. EJEMPLO DE CÁLCULO DE UNA SECCIÓN TRIANGULAR.

Calcular el tirante crítico de un canal de sección triangular con un talud $z=1.5$ y que conduce un caudal de $2\text{ m}^3/\text{s}$.

Datos del problema:



$$Q=2\text{ m}^3/\text{s}; z=1.5$$

Uso de SN CANALES:

CÁLCULO DEL TIRANTE CRÍTICO DE UNA SECCIÓN TRIANGULAR

Datos de entrada:

Caudal (Q)=	<input type="text" value="2"/>	m ³ /s
Talud (z)=	<input type="text" value="1.5"/>	

Datos de salida:

Tirante crítico (y)=	<input type="text" value="0.8163"/>	m
Área mojada (A)=	<input type="text" value="0.9995"/>	m ²
Perímetro mojado (P)=	<input type="text" value="2.9432"/>	m
Radio hidráulico (R)=	<input type="text" value="0.3396"/>	m
Ancho superficial (T)=	<input type="text" value="2.4489"/>	m
Profundidad hidráulica (D)=	<input type="text" value="0.4081"/>	m
Velocidad (V)=	<input type="text" value="2.001"/>	m/s
Número de Froude (NF)=	<input type="text" value="1"/>	
Tipo de flujo=	<input type="text" value="Crítico"/>	
Energía específica (E)=	<input type="text" value="1.0204"/>	m-Kg/Kg

Diagrama de un canal de sección triangular invertida similar al anterior, pero con un tirante crítico y y un ancho superficial T etiquetados. El talud z también está etiquetado.

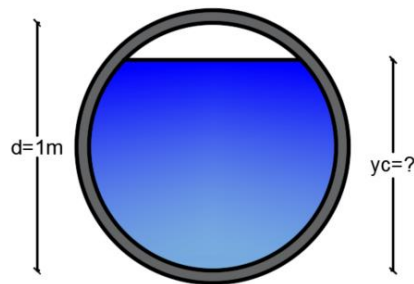
Sistema de unidades:

Sistema métrico
 Sistema inglés

4.2.4. EJEMPLO DE CÁLCULO DE UNA SECCIÓN CIRCULAR.

Se desea determinar el tirante crítico en una tubería de 1m dediámetro, cuando se conduce un caudal de $2\text{m}^3/\text{s}$.

Datos del problema:



$$Q=2\text{m}^3/\text{s}; d=1\text{m}$$

Uso de SN CANALES:

CÁLCULO DEL TIRANTE CRÍTICO DE UNA SECCIÓN CIRCULAR

Datos de entrada:	
Caudal (Q)=	2 m ³ /s
Diámetro (d)=	1 m

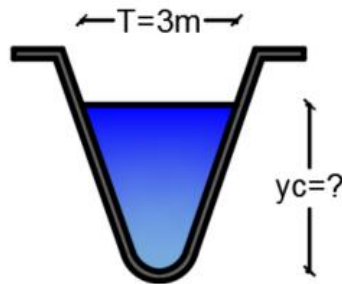
Datos de salida:	
Tirante crítico (y)=	0.812 m
Área mojada (A)=	0.683 m ²
Perímetro mojado (P)=	2.2445 m
Radio hidráulico (R)=	0.3043 m
Ancho superficial (T)=	0.7815 m
Profundidad hidráulica (D)=	0.874 m
Velocidad (V)=	2.9281 m/s
Número de Froude (NF)=	1
Tipo de flujo=	Crítico
Energía específica (E)=	1.249 m-Kg/Kg

Sistema de unidades:
 Sistema métrico
 Sistema inglés

4.2.5. EJEMPLO DE CÁLCULO DE UNA SECCIÓN PARABÓLICA.

Determinar el tirante crítico de un canal de sección parabólica, sabiendo que conduce un caudal de $2\text{ m}^3/\text{s}$ y el espejo de agua es 3m.

Datos del problema:



$$Q=2\text{ m}^3/\text{s}; T=3\text{m}$$

Uso de SN CANALES:

CÁLCULO DEL TIRANTE CRÍTICO DE UNA SECCIÓN PARABÓLICA

Datos de entrada:

Caudal (Q)=	<input type="text" value="2"/>	m ³ /s
Ancho Superficial (T)=	<input type="text" value="3"/>	m

Datos de salida:

Tirante crítico (y)=	<input type="text" value="0.5347"/>	m
Área mojada (A)=	<input type="text" value="1.0695"/>	m ²
Perímetro mojado (P)=	<input type="text" value="3.2542"/>	m
Radio hidráulico (R)=	<input type="text" value="0.3286"/>	m
Profundidad hidráulica (D)=	<input type="text" value="0.3565"/>	m
Velocidad (V)=	<input type="text" value="1.8701"/>	m/s
Número de Froude (NF)=	<input type="text" value="1"/>	
Tipo de flujo=	<input type="text" value="Crítico"/>	
Energía específica (E)=	<input type="text" value="0.713"/>	m-Kg/Kg

Sistema de unidades:

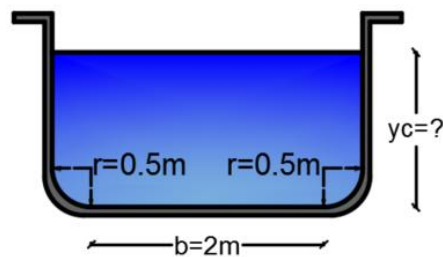
Sistema métrico
 Sistema inglés

Diagrama de un canal de sección parabólica con el ancho superficial T y la profundidad y .

4.2.6. EJEMPLO DE CÁLCULO DE UNA SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS.

Calcular el tirante crítico de un canal de sección rectangular con esquinas redondeadas que transporta un caudal de $5\text{ m}^3/\text{s}$, con un radio de 0.5m y un ancho de solera de 2m.

Datos del problema:



$$Q=5\text{ m}^3/\text{s}; b=2\text{m}; r=0.5\text{m}$$

Uso de SN CANALES:

CÁLCULO DEL TIRANTE CRÍTICO DE UNA SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS

Datos de entrada:

Caudal (Q)=	5	m ³ /s
Ancho de la solera (b)=	2	m
Radio (r)=	0.5	m

CALCULAR **LIMPIAR**

Datos de salida:

Tirante crítico (y)=	0.6924	m
Área mojada (A)=	1.97	m ²
Perímetro mojado (P)=	3.9557	m
Radio hidráulico (R)=	0.498	m
Ancho superficial (T)=	3	m
Profundidad hidráulica (D)=	0.6567	m
Velocidad (V)=	2.5381	m/s
Número de Froude (NF)=	1	
Tipo de flujo=	Critico	
Energía específica (E)=	1.0208	m-Kg/Kg

Sistema de unidades:

Sistema métrico

Sistema inglés

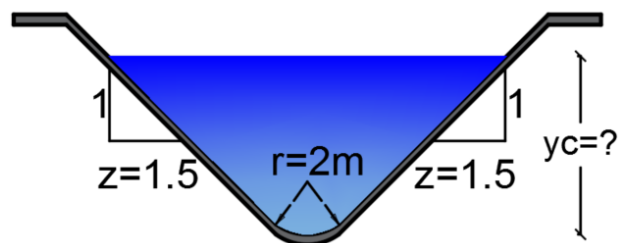
MENÚ PRINCIPAL

Diagrama de un canal con esquinas redondeadas que muestra el tirante crítico y y el ancho superficial T . El ancho de la solera es b y el radio de las esquinas es r .

4.2.7. EJEMPLO DE CÁLCULO DE UNA SECCIÓN TRIANGULAR CON FONDO REDONDEADO.

Se desea calcular el tirante crítico de un canal de sección triangular con fondo redondeado con un radio $r=2\text{m}$, un talud $z=1.5$ y que conduce un caudal de $3\text{m}^3/\text{s}$.

Datos del problema:



$$Q=3\text{m}^3/\text{s}; z=1.5; r=2\text{m}$$

Uso de SN CANALES:

CÁLCULO DEL TIRANTE CRÍTICO DE UNA SECCIÓN TRIANGULAR CON FONDO REDONDEADO

Datos de entrada:

Caudal (Q)= m³/s

Radio (r)= m

Talud (z)=

Datos de salida:

Tirante crítico (y)= m

Área mojada (A)= m²

Perímetro mojado (P)= m

Radio hidráulico (R)= m

Ancho superficial (T)= m

Profundidad hidráulica (D)= m

Velocidad (V)= m/s

Número de Froude (NF)=

Tipo de flujo=

Energía específica (E)= m-Kg/Kg

Sistema de unidades:

Sistema métrico

Sistema inglés

Diagrama de un canal de sección triangular con fondo redondeado, similar al anterior, pero con el tirante crítico denotado como y .

4.3. CAUDAL

Para el cálculo del caudal, el software dispone de siete opciones, que corresponden a secciones transversales de canales como son: rectangular, trapezoidal, triangular, circular, parabólica, rectangular con esquinas redondeadas y triangular con fondo redondeado.

Para acceder a las ventanas de cálculo se debe seleccionar del menú principal la opción tirante normal en la cual se presenta un submenú como el mostrado en la siguiente imagen.



Para el cálculo en cualquiera de las secciones disponibles se debe de dar clic sobre la sección elegida, donde a continuación aparecerá la pantalla correspondiente para su cálculo.

Los datos de entrada dependerán de la sección transversal seleccionada como son:

- Sección rectangular: Tirante (y), ancho de la solera (b), pendiente (S) y coeficiente de manning (n).
- Sección trapezoidal: Tirante (y), ancho de la solera (b), talud (z), pendiente (S) y coeficiente de manning (n).

- Sección triangular: Tirante (y), talud (z), pendiente (S) y coeficiente de manning (n).
- Sección circular: Tirante (y), diámetro (d), pendiente (S) y coeficiente de manning (n).
- Sección parabólica: Tirante (y), ancho superficial (T), pendiente (S) y coeficiente de manning (n).
- Sección rectangular con esquinas redondeadas: Tirante (y), ancho de la solera (b), radio (r), pendiente (S) y coeficiente de manning (n).
- Sección triangular con fondo redondeado: Tirante (y), radio (r), talud (z), pendiente (S) y coeficiente de manning (n).

En cada pantalla de cálculo se incorpora una imagen de referencia de acuerdo a la sección transversal.

Los datos son ingresados tecleando sobre cada casilla el valor correspondiente, además se puede utilizar la tecla Tab para usuarios que omitan utilizar el mouse.

Las unidades de los datos de entrada y de salida dependerán del sistema de unidades que se escoja para el cálculo, para lo cual tenemos el sistema métrico e inglés, que deberá ser seleccionado antes de calcular.

Los datos de salida que son calculados son:

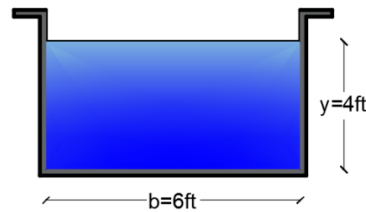
- Caudal (Q),
- Área mojada (A)
- Perímetro mojado (P)
- Radio hidráulico (R)
- Ancho superficial (T)
- Profundidad hidráulica (D)
- Velocidad (V)
- Número de Froude (NF)
- Tipo de flujo
- Energía específica (E)

Para iniciar el cálculo se debe presionar el botón “CALCULAR”, además se dispone del botón “LIMPIAR” que permite borrar los datos de entrada y salida, y del botón “MENÚ PRINCIPAL” el cual permite al usuario salir de la ventana actual y volver a la pantalla principal.

4.3.1. EJEMPLO DE CÁLCULO DE UNA SECCIÓN RECTAGULAR.

Calcule el caudal en un canal de sección rectangular con $b=6\text{ft}$, $y=4\text{ft}$, $n=0.015$ y $S=0.002$.

Datos del problema:



$y=4\text{ft}$; $b=6\text{ft}$; $S=0.002$; $n=0.015$

Uso de SN CANALES:

CÁLCULO DEL CAUDAL DE UNA SECCIÓN RECTANGULAR

Datos de entrada:		
Tirante (y)=	<input type="text" value="4"/>	ft
Ancho de la solera (b)=	<input type="text" value="6"/>	ft
Pendiente (S)=	<input type="text" value="0.002"/>	ft/ft
Coefficiente de Manning (n)=	<input type="text" value="0.015"/>	

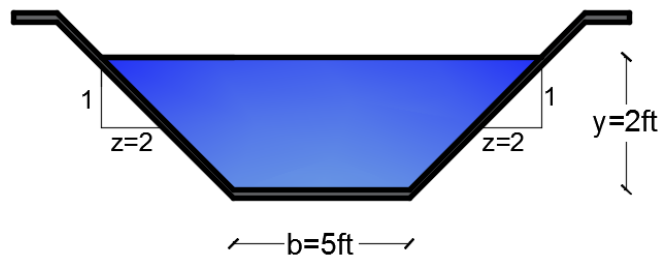
Datos de salida:		
Caudal (Q)=	<input type="text" value="152.303"/>	ft ³ /s
Área mojada (A)=	<input type="text" value="24"/>	ft ²
Perímetro mojado (P)=	<input type="text" value="14"/>	ft
Radio hidráulico (R)=	<input type="text" value="1.7143"/>	ft
Ancho superficial (T)=	<input type="text" value="6"/>	ft
Profundidad hidráulica (D)=	<input type="text" value="4"/>	ft
Velocidad (V)=	<input type="text" value="6.346"/>	ft/s
Número de Froude (NF)=	<input type="text" value="0.5593"/>	
Tipo de flujo=	<input type="text" value="Subcrítico"/>	
Energía específica (E)=	<input type="text" value="4.6256"/>	ft-lb/lb

Sistema métrico
 Sistema inglés

4.3.2. EJEMPLO DE CÁLCULO DE UNA SECCIÓN TRAPEZOIDAL.

Calcule el caudal en un canal de sección trapezoidal con un ancho de base de 5ft, pendientes laterales de 1 a 2, $y=2\text{ft}$, $n=0.0149$ y $S=0.003$.

Datos del problema:



$y=2\text{ft}$; $b=5\text{ft}$; $z=2$; $S=0.003$; $n=0.0149$

Uso de SN CANALES:

CAUDAL DE UNA SECCIÓN TRAPEZOIDAL

Datos de entrada:

Tirante (y)= ft

Ancho de la solera (b)= ft

Talud (z)=

Pendiente (S)= ft/ft

Coefficiente de Manning (n)=

Datos de salida:

Caudal (Q)= ft³/s

Área mojada (A)= ft²

Perímetro mojado (P)= ft

Radio hidráulico (R)= ft

Ancho superficial (T)= ft

Profundidad hidráulica (D)= ft

Velocidad (V)= ft/s

Número de Froude (NF)=

Tipo de flujo=

Energía específica (E)= ft-lb/lb

Sistema de unidades:

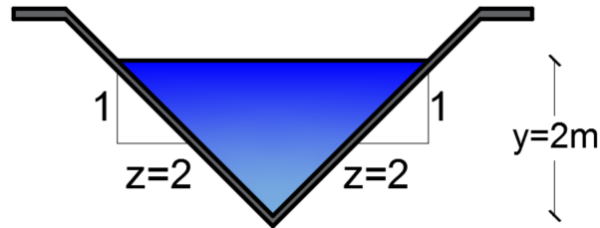
Sistema métrico

Sistema inglés

4.3.3. EJEMPLO DE CÁLCULO DE UNA SECCIÓN TRIANGULAR.

Calcule el caudal en una cuneta triangular de carretera cuando $z=2$, $n=0.017$, $y=2\text{m}$ y $S=0.015$.

Datos del problema:



$y=2\text{m}$; $z=2$; $S=0.015$; $n=0.017$

Uso de SN CANALES:

CÁLCULO DEL CAUDAL DE UNA SECCIÓN TRIANGULAR

Datos de entrada:

Tirante (y)=	<input type="text" value="2"/>	m
Talud (z)=	<input type="text" value="2"/>	
Pendiente (S)=	<input type="text" value="0.015"/>	m/m
Coefficiente de Manning (n)=	<input type="text" value="0.017"/>	

Datos de salida:

Caudal (Q)=	<input type="text" value="53.5036"/>	m ³ /s
Área mojada (A)=	<input type="text" value="8"/>	m ²
Perímetro mojado (P)=	<input type="text" value="8.9443"/>	m
Radio hidráulico (R)=	<input type="text" value="0.8944"/>	m
Ancho superficial (T)=	<input type="text" value="8"/>	m
Profundidad hidráulica (D)=	<input type="text" value="1"/>	m
Velocidad (V)=	<input type="text" value="6.688"/>	m/s
Número de Froude (NF)=	<input type="text" value="2.1353"/>	
Tipo de flujo=	<input type="text" value="Supercrítico"/>	
Energía específica (E)=	<input type="text" value="4.2798"/>	m-Kg/Kg

Sistema de unidades:

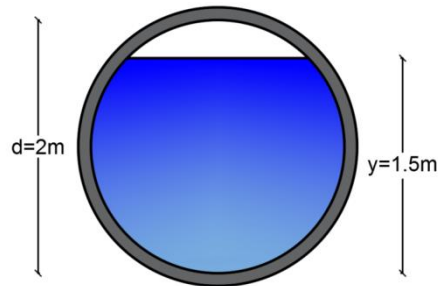
Sistema métrico

Sistema inglés

4.3.4. EJEMPLO DE CÁLCULO DE UNA SECCIÓN CIRCULAR.

Calcule el caudal en un canal de sección circular con 2m de diámetro, $y=1.5\text{m}$, $n=0.010$ y $S=0.025$.

Datos del problema:



$d=2\text{ m}$; $y=1.5\text{m}$; $S=0.025$; $n=0.010$

Uso de SN CANALES:

CÁLCULO DEL CAUDAL DE UNA SECCIÓN CIRCULAR

Datos de entrada:

Tirante (y)=	1.5	m
Diámetro (d)=	2	m
Pendiente (S)=	0.025	m/m
Coefficiente de Manning (n)=	0.01	

CALCULAR LIMPIAR

Datos de salida:

Caudal (Q)=	28.5345	m ³ /s
Área mojada (A)=	2.5274	m ²
Perímetro mojado (P)=	4.1888	m
Radio hidráulico (R)=	0.6034	m
Ancho superficial (T)=	1.7321	m
Profundidad hidráulica (D)=	1.4592	m
Velocidad (V)=	11.29	m/s
Número de Froude (NF)=	2.984	
Tipo de flujo=	Supercrítico	
Energía específica (E)=	7.9967	m-Kg/Kg

Sistema de unidades:

Sistema métrico
 Sistema inglés

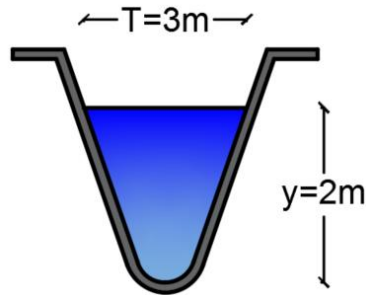
MENÚ PRINCIPAL

Diagrama de una sección circular de un canal. El diámetro total es d . La altura del agua desde el fondo del canal hasta la superficie libre es y . El ancho superficial del agua es T . El ángulo de contacto del agua con el fondo del canal es θ . El área de agua está sombreada en azul.

4.3.5. EJEMPLO DE CÁLCULO DE UNA SECCIÓN PARABÓLICA.

Calcule el caudal en un canal de sección parabólica con un ancho superficial de 3m, $y=2\text{m}$, $n=0.015$ y $S=0.012$.

Datos del problema:



$T=3\text{m}$; $y=2\text{m}$; $S=0.012$; $n=0.015$

Uso de SN CANALES:

CÁLCULO DEL CAUDAL DE UNA SECCIÓN PARABÓLICA

Datos de entrada:

Tirante (y)=	<input type="text" value="2"/>	m
Ancho Superficial (T)=	<input type="text" value="3"/>	m
Pendiente (S)=	<input type="text" value="0.012"/>	m/m
Coficiente de Manning (n)=	<input type="text" value="0.015"/>	

Datos de salida:

Caudal (Q)=	<input type="text" value="24.4229"/>	m ³ /s
Área mojada (A)=	<input type="text" value="4"/>	m ²
Perímetro mojado (P)=	<input type="text" value="5.2324"/>	m
Radio hidráulico (R)=	<input type="text" value="0.7645"/>	m
Profundidad hidráulica (D)=	<input type="text" value="1.3333"/>	m
Velocidad (V)=	<input type="text" value="6.1057"/>	m/s
Número de Froude (NF)=	<input type="text" value="1.6882"/>	
Tipo de flujo=	<input type="text" value="Supercritico"/>	
Energía específica (E)=	<input type="text" value="3.9001"/>	m-Kg/Kg

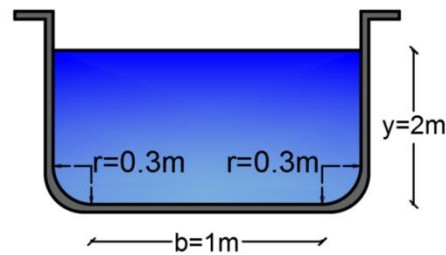
Sistema de unidades:

Sistema métrico
 Sistema inglés

4.3.6. EJEMPLO DE CÁLCULO DE UNA SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS.

Calcule el caudal en un canal de sección rectangular con esquinas redondeadas con $b=1\text{m}$, $y=2\text{m}$, $r=0.3\text{m}$, $n=0.03$ y $S=0.002$.

Datos del problema:



$y=2\text{m}$; $b=1\text{m}$; $r=0.3\text{m}$; $S=0.002$; $n=0.03$

Uso de SN CANALES:

CÁLCULO DEL CAUDAL DE UNA SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS

Datos de entrada:

Tirante (y)=	<input type="text" value="2"/>	m
Ancho de la solera (b)=	<input type="text" value="1"/>	m
Radio (r)=	<input type="text" value="0.3"/>	m
Pendiente (S)=	<input type="text" value="0.002"/>	m/m
Coficiente de Manning (n)=	<input type="text" value="0.03"/>	

Datos de salida:

Caudal (Q)=	<input type="text" value="3.3217"/>	m ³ /s
Área mojada (A)=	<input type="text" value="3.1614"/>	m ²
Perímetro mojado (P)=	<input type="text" value="5.3425"/>	m
Radio hidráulico (R)=	<input type="text" value="0.5917"/>	m
Ancho superficial (T)=	<input type="text" value="1.6"/>	m
Profundidad hidráulica (D)=	<input type="text" value="1.9759"/>	m
Velocidad (V)=	<input type="text" value="1.0507"/>	m/s
Número de Froude (NF)=	<input type="text" value="0.2387"/>	
Tipo de flujo=	<input type="text" value="Subcrítico"/>	
Energía específica (E)=	<input type="text" value="2.0563"/>	m-Kg/Kg

Sistema de unidades:

Sistema métrico

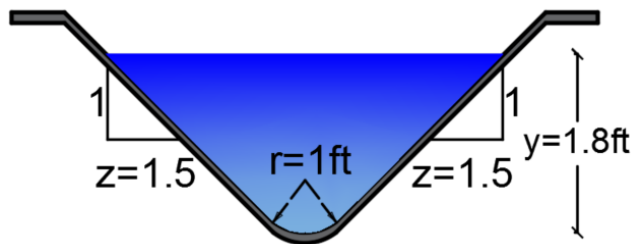
Sistema inglés

Diagrama de una sección de canal rectangular con esquinas redondeadas. El ancho superficial es T . El ancho de la solera es b . La profundidad del agua es y .

4.3.7. EJEMPLO DE CÁLCULO DE UNA SECCIÓN TRIANGULAR CON FONDO REDONDEADO.

Calcule el caudal en un canal de sección triangular con fondo redondeado con $y=1.8\text{ft}$, $r=1\text{ft}$, $z=1.5$, $n=0.02$ y $S=0.0015$.

Datos del problema:



$y=1.8\text{ft}$; $z=1.5$; $r=1\text{ft}$; $S=0.0015$; $n=0.02$

Uso de SN CANALES:

CÁLCULO DEL CAUDAL DE UNA SECCIÓN TRIANGULAR CON FONDO REDONDEADO

Datos de entrada:

Tirante (y)=	<input type="text" value="1.8"/>	ft
Radio (r)=	<input type="text" value="1"/>	ft
Talud (z)=	<input type="text" value="1.5"/>	
Pendiente (S)=	<input type="text" value="0.0015"/>	ft/ft
Coefficiente de Manning (n)=	<input type="text" value="0.02"/>	

Datos de salida:

Caudal (Q)=	<input type="text" value="15.2009"/>	ft ³ /s
Área mojada (A)=	<input type="text" value="5.9324"/>	ft ²
Perímetro mojado (P)=	<input type="text" value="7.0604"/>	ft
Radio hidráulico (R)=	<input type="text" value="0.8402"/>	ft
Ancho superficial (T)=	<input type="text" value="6.0056"/>	ft
Profundidad hidráulica (D)=	<input type="text" value="0.9878"/>	ft
Velocidad (V)=	<input type="text" value="2.5623"/>	ft/s
Número de Froude (NF)=	<input type="text" value="0.4544"/>	
Tipo de flujo=	<input type="text" value="Subcrítico"/>	
Energía específica (E)=	<input type="text" value="1.902"/>	ft-lb/lb

Sistema de unidades:

Sistema métrico

Sistema inglés

4.4. ELEMENTOS GEOMÉTRICOS

Para el cálculo de los elementos geométricos, el software dispone de siete opciones, que corresponden a secciones transversales de canales como son: rectangular, trapezoidal, triangular, circular, parabólica, rectangular con esquinas redondeadas y triangular con fondo redondeado.

Para acceder a las ventanas de cálculo se debe seleccionar del menú principal la opción tirante normal en la cual se presenta un submenú como el mostrado en la siguiente imagen.



Para el cálculo en cualquiera de las secciones disponibles se debe de dar clic sobre la sección elegida, donde a continuación aparecerá la pantalla correspondiente para su cálculo.

Los datos de entrada dependerán de la sección transversal seleccionada como son:

- Sección rectangular: Tirante (y), ancho de la solera (b).
- Sección trapezoidal: Tirante (y), ancho de la solera (b), talud (z).
- Sección triangular: Tirante (y), talud (z).
- Sección circular: Tirante (y), diámetro (d).

- Sección parabólica: Tirante (y), ancho superficial (T).
- Sección rectangular con esquinas redondeadas: Tirante (y), ancho de la solera (b), radio (r).
- Sección triangular con fondo redondeado: Tirante (y), radio (r), talud (z).

En cada pantalla de cálculo se incorpora una imagen de referencia de acuerdo a la sección transversal.

En cada pantalla de cálculo se incorpora una imagen de referencia de acuerdo a la sección transversal.

Los datos son ingresados tecleando sobre cada casilla el valor correspondiente, además se puede utilizar la tecla Tab para usuarios que omitan utilizar el mouse.

Las unidades de los datos de entrada y de salida se encuentran en el sistema métrico.

Los datos de salida que son calculados son:

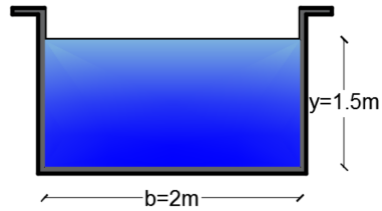
- Área mojada (A)
- Perímetro mojado (P)
- Radio hidráulico (R)
- Ancho superficial (T)
- Profundidad hidráulica (D)
- Factor de sección (Z)

Para iniciar el cálculo se debe presionar el botón “CALCULAR”, además se dispone del botón “LIMPIAR” que permite borrar los datos de entrada y salida, y del botón “MENÚ PRINCIPAL” el cual permite al usuario salir de la ventana actual y volver a la pantalla principal.

4.4.1. Ejemplo de cálculo de una sección rectangular.

Calcule el radio hidráulico y el factor de sección de un canal rectangular con un ancho de 2m y un tirante de 1.5m.

Datos del problema:



$y=1.5\text{m}$; $b=2\text{m}$

Uso de SN CANALES:

ELEMENTOS GEOMÉTRICOS DE UNA SECCIÓN RECTANGULAR

Datos de entrada:

Acho de la solera (b)=	<input type="text" value="2"/>	m
Tirante (y)=	<input type="text" value="1.5"/>	m

Datos de salida:

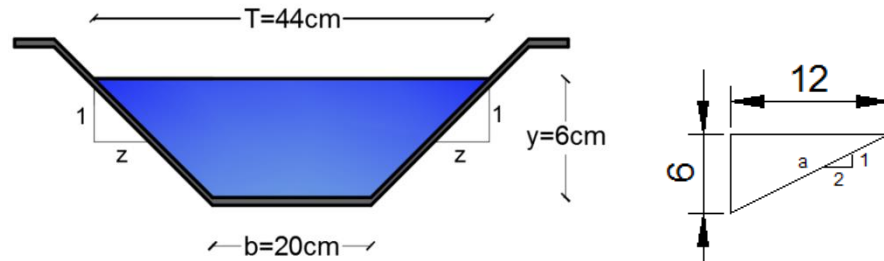
Área mojada (A)=	<input type="text" value="3"/>	m ²
Perímetro mojado (P)=	<input type="text" value="5"/>	m
Radio hidráulico (R)=	<input type="text" value="0.6"/>	m
Ancho superficial (T)=	<input type="text" value="2"/>	m
Profundidad hidráulica (D)=	<input type="text" value="1.5"/>	m
Factor de sección (Z)=	<input type="text" value="1.2247"/>	m

Diagrama de un canal rectangular con un ancho de b y un tirante de y . El canal está representado por un rectángulo azul con un contorno negro, y las dimensiones están indicadas con líneas de medida y flechas.

4.4.2. Ejemplo de cálculo de una sección trapezoidal.

Calcular el radio hidráulico, la profundidad hidráulica, y el factor de la sección de un canal trapezoidal como se indica en la figura.

Datos del problema:



$$y=0.06\text{m} ; b=0.2\text{m} ; z=2$$

Uso de SN CANALES:

ELEMENTOS GEOMÉTRICOS DE UNA SECCIÓN TRAPEZOIDAL

Datos de entrada:

Tirante (y)=	<input type="text" value="0.2"/>	m
Ancho de la solera (b)=	<input type="text" value="0.06"/>	m
Talud (z)=	<input type="text" value="2"/>	

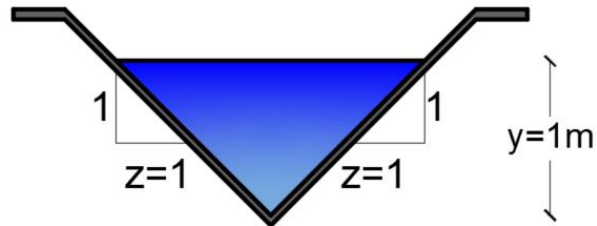
Datos de salida:

Área mojada (A)=	<input type="text" value="0.092"/>	m ²
Perímetro mojado (P)=	<input type="text" value="0.9544"/>	m
Radio hidráulico (R)=	<input type="text" value="0.0964"/>	m
Ancho superficial (T)=	<input type="text" value="0.86"/>	m
Profundidad hidráulica (D)=	<input type="text" value="0.107"/>	m
Factor de sección (Z)=	<input type="text" value="0.3271"/>	m

4.4.3. Ejemplo de cálculo de una sección triangular.

Determinar para una sección triangular los elementos geométricos si la profundidad del flujo es $y=1\text{m}$ y $z=1$.

Datos del problema:



$y=1\text{m}$; $z=1$

Uso de SN CANALES:

ELEMENTOS GEOMÉTRICOS DE UNA SECCIÓN TRIANGULAR

Datos de entrada:

Tirante (y)=	<input type="text" value="1"/>	m
Talud (z)=	<input type="text" value="1"/>	

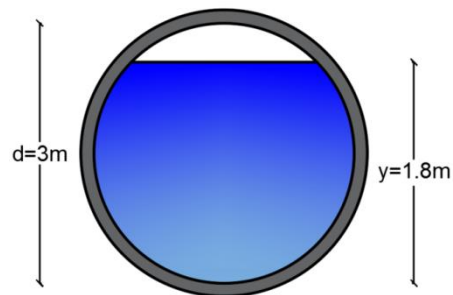
Datos de salida:

Área mojada (A)=	<input type="text" value="1"/>	m^2
Perímetro mojado (P)=	<input type="text" value="2.8284"/>	m
Radio hidráulico (R)=	<input type="text" value="0.3536"/>	m
Ancho superficial (T)=	<input type="text" value="2"/>	m
Profundidad hidráulica (D)=	<input type="text" value="0.5"/>	m
Factor de sección (Z)=	<input type="text" value="0.7071"/>	m

4.4.4. Ejemplo de cálculo de una sección circular.

En un canal circular de 3m de diámetro, circula agua con un tirante de 1.8m, determinar el área mojada, el radio hidráulico, el perímetro mojado, el espejo de agua y la profundidad hidráulica.

Datos del problema:



$d=3\text{m}$; $y=1.8\text{m}$

Uso de SN CANALES:

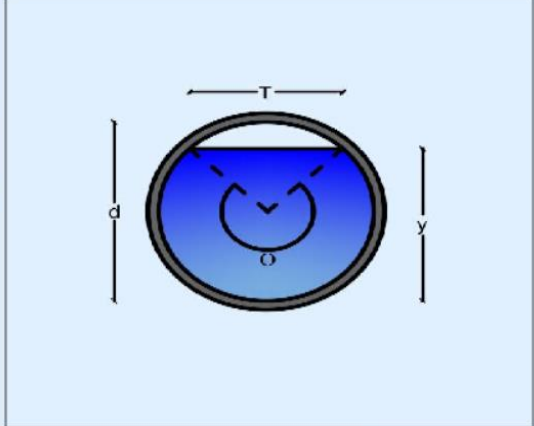
ELEMENTOS GEOMÉTRICOS DE UNA SECCIÓN CIRCULAR

Datos de entrada:

Diámetro (d)=	<input type="text" value="3"/>	m
Tirante (y)=	<input type="text" value="1.8"/>	m

Datos de salida:

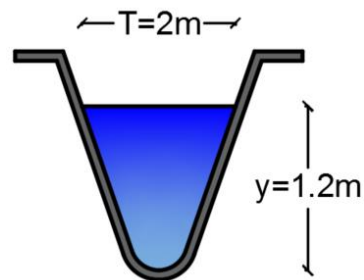
Área mojada (A)=	<input type="text" value="4.4283"/>	m ²
Perímetro mojado (P)=	<input type="text" value="5.3165"/>	m
Radio hidráulico (R)=	<input type="text" value="0.8329"/>	m
Ancho superficial (T)=	<input type="text" value="2.9394"/>	m
Profundidad hidráulica (D)=	<input type="text" value="1.5065"/>	m
Factor de sección (Z)=	<input type="text" value="1.2274"/>	m



4.4.5. Ejemplo de cálculo de una sección parabólica.

En un canal parabólico con un tirante de 1.2m y un ancho superficial de 2m determinar el área mojada, el radio hidráulico, el perímetro mojado, el espejo de agua y la profundidad hidráulica.

Datos del problema:



$y=1.2\text{m}$; $T=2\text{m}$

Uso de SN CANALES:

ELEMENTOS GEOMÉTRICOS DE UNA SECCIÓN PARABÓLICA

Datos de entrada:

Tirante (y)=	<input type="text" value="1.2"/>	m
Ancho superficial (T)=	<input type="text" value="2"/>	m

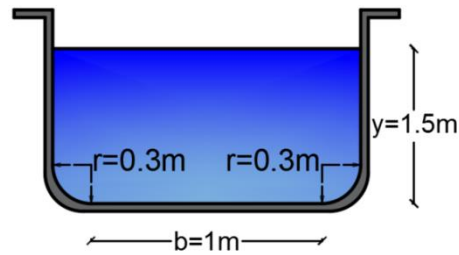
Datos de salida:

Área mojada (A)=	<input type="text" value="1.6"/>	m ²
Perímetro mojado (P)=	<input type="text" value="3.2706"/>	m
Radio hidráulico (R)=	<input type="text" value="0.4892"/>	m
Ancho superficial (T)=	<input type="text" value="2"/>	m
Profundidad hidráulica (D)=	<input type="text" value="0.8"/>	m
Factor de sección (Z)=	<input type="text" value="0.8944"/>	m

4.4.6. Ejemplo de cálculo de una sección rectangular con esquinas redondeadas.

En un canal con sección rectangular con esquinas redondeadas con un tirante de 1.5m, un ancho de la solera de 1m y radio de 0.3m. Calcular el radio hidráulico, la profundidad hidráulica, y el factor de la sección.

Datos del problema:



$y=1.5\text{m}$; $b=1\text{m}$; $r=0.3\text{m}$

Uso de SN CANALES:

ELEMENTOS GEOMÉTRICOS DE UNA SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS

Datos de entrada:

Tirante (y)=	<input type="text" value="1.5"/>	m
Ancho de la solera (b)=	<input type="text" value="1"/>	m
radio (r)=	<input type="text" value="0.3"/>	m

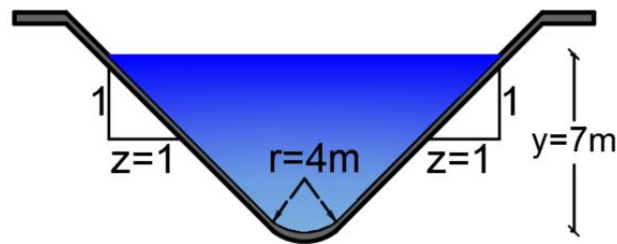
Datos de salida:

Área mojada (A)=	<input type="text" value="2.3614"/>	m ²
Perímetro mojado (P)=	<input type="text" value="4.3425"/>	m
Radio hidráulico (R)=	<input type="text" value="0.5438"/>	m
Ancho superficial (T)=	<input type="text" value="1.6"/>	m
Profundidad hidráulica (D)=	<input type="text" value="1.4759"/>	m
Factor de sección (Z)=	<input type="text" value="1.2148"/>	m

4.4.7. Ejemplo de cálculo de una sección triangular con fondo redondeado.

Calcular los elementos geométricos del canal triangular con fondo redondeado que se indica en la figura.

Datos del problema:



$y=7\text{m}$; $z=1$; $r=4\text{m}$

Uso de SN CANALES:

ELEMENTOS GEOMÉTRICOS DE UNA SECCIÓN TRIANGULAR CON FONDO REDONDEADO

Datos de entrada:

Tirante (y)=	<input type="text" value="7"/>	m
Radio (r)=	<input type="text" value="4"/>	m
Talud (z)=	<input type="text" value="1"/>	

Datos de salida:

Área mojada (A)=	<input type="text" value="71.5075"/>	m ²
Perímetro mojado (P)=	<input type="text" value="22.7685"/>	m
Radio hidráulico (R)=	<input type="text" value="3.1406"/>	m
Ancho superficial (T)=	<input type="text" value="17.3137"/>	m
Profundidad hidráulica (D)=	<input type="text" value="4.1301"/>	m
Factor de sección (Z)=	<input type="text" value="2.0323"/>	m

4.5. SECCIÓN HIDRÁULICA ÓPTIMA

Para el cálculo de la sección hidráulica óptima, el software dispone de cuatro opciones, que corresponden a secciones transversales de canales como son: rectangular, trapezoidal, triangular y semicircular.

Para acceder a las ventanas de cálculo se debe seleccionar del menú principal la opción tirante normal en la cual se presenta un submenú como el mostrado en la siguiente imagen.



Para el cálculo en cualquiera de las secciones disponibles se debe de dar clic sobre la sección elegida, donde a continuación aparecerá la pantalla correspondiente para su cálculo.

Los datos de entrada dependerán de la sección transversal seleccionada como son:

- Sección rectangular: Caudal (Q), pendiente (S) y coeficiente de manning (n).
- Sección trapezoidal: Caudal (Q), talud (z), pendiente (S) y coeficiente de manning (n).
- Sección triangular: Caudal (Q), pendiente (S) y coeficiente de manning (n).

- Sección semicircular: Caudal (Q), pendiente (S) y coeficiente de manning (n).

Los datos son ingresados tecleando sobre cada casilla el valor correspondiente, además se puede utilizar la tecla Tab para usuarios que omitan utilizar el mouse.

Las unidades de los datos de entrada y de salida dependerán del sistema de unidades que se escoja para el cálculo, para lo cual tenemos el sistema métrico e inglés, que deberá ser seleccionado antes de calcular.

Los datos de salida que son calculados varían dependiendo de la sección elegida, entre ellos tenemos:

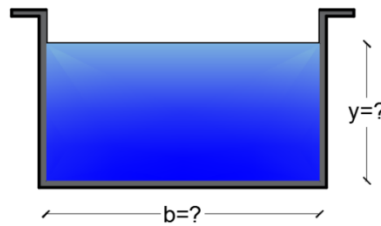
- Tirante (y)
- Ancho de la solera (b)
- Diámetro (d)
- Talud (z)
- Área mojada (A)
- Perímetro mojado (P)
- Radio hidráulico (R)
- Ancho superficial (T)
- Profundidad hidráulica (D)
- Velocidad (V)
- Número de Froude (NF)
- Tipo de flujo
- Energía específica (E)

Para iniciar el cálculo se debe presionar el botón “CALCULAR”, además se dispone del botón “LIMPIAR” que permite borrar los datos de entrada y salida, y del botón “MENÚ PRINCIPAL” el cual permite al usuario salir de la ventana actual y volver a la pantalla principal.

4.5.1. Ejemplo de cálculo de una sección rectangular.

Determinar la sección hidráulica óptima para un canal rectangular con $n=0.01$, si se va a diseñar para transportar $1\text{ m}^3/\text{s}$ y una pendiente de 0.65% .

Datos del problema:



$$Q=1\text{ m}^3/\text{s}; S=0.0065; n=0.01$$

Uso de SN CANALES:

SECCIÓN HIDRÁULICA ÓPTIMA DE UNA SECCIÓN RECTANGULAR

Datos de entrada:

Caudal (Q)=	<input type="text" value="1"/>	m ³ /s
Pendiente (S)=	<input type="text" value="0.0065"/>	m/m
Coefficiente de Manning (n)=	<input type="text" value="0.01"/>	

Datos de salida:

Ancho de la solera (b)=	<input type="text" value="0.8385"/>	m
Tirante (y)=	<input type="text" value="0.4192"/>	m
Área mojada (A)=	<input type="text" value="0.3515"/>	m ²
Perímetro mojado (P)=	<input type="text" value="1.6769"/>	m
Radio hidráulico (R)=	<input type="text" value="0.2096"/>	m
Ancho superficial (T)=	<input type="text" value="0.8385"/>	m
Profundidad hidráulica (D)=	<input type="text" value="0.4192"/>	m
Velocidad (V)=	<input type="text" value="2.8449"/>	m/s
Número de Froude (NF)=	<input type="text" value="1.4028"/>	
Tipo de flujo=	<input type="text" value="Supercrítico"/>	
Energía específica (E)=	<input type="text" value="0.8317"/>	m-Kg/Kg

Diagrama de un canal rectangular con un fondo azul. El ancho del canal está etiquetado como b y la profundidad del agua como y .

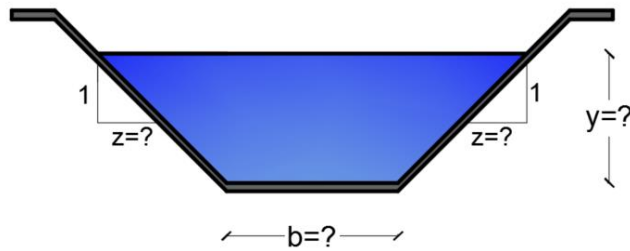
Sistema de unidades:

Sistema métrico
 Sistema inglés

4.5.2. Ejemplo de cálculo de una sección trapezoidal.

Determinar la sección hidráulica óptima para un canal trapezoidal con $n=0.0149$, si se va a diseñar para transportar $2\text{ m}^3/\text{s}$ y una pendiente de 1%.

Datos del problema:



$$Q=2\text{ m}^3/\text{s}; S=0.01; n=0.0149$$

Uso de SN CANALES:

SECCIÓN HIDRÁULICA ÓPTIMA DE UNA SECCIÓN TRAPEZOIDAL

Datos de entrada:

Caudal (Q)= m³/s

Pendiente (S)= m/m

Coefficiente de Manning (n)=

Datos de salida:

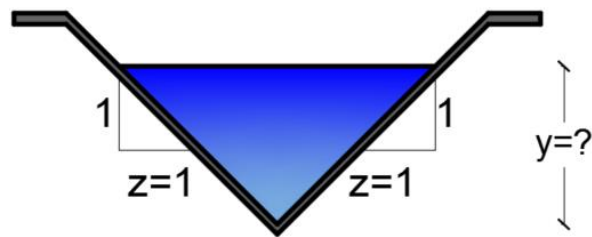
Talud (z)=	0.5774
Tirante (y)=	0.6146 m
Ancho de la solera (b)=	0.7099 m
Área mojada (A)=	0.6544 m ²
Perímetro mojado (P)=	2.1292 m
Radio hidráulico (R)=	0.3073 m
Ancho superficial (T)=	1.4195 m
Profundidad hidráulica (D)=	0.461 m
Velocidad (V)=	3.0564 m/s
Número de Froude (NF)=	1.4373
Tipo de flujo=	Supercrítico
Energía específica (E)=	1.0907 m-Kg/Kg

Sistema métrico
 Sistema inglés

4.5.3. Ejemplo de cálculo de una sección triangular.

Determinar la sección hidráulica óptima para un canal triangular con $n=0.02$ si se va a diseñar para transportar $3 \text{ ft}^3/\text{s}$ y una pendiente de 1.5%.

Datos del problema:



$$Q=3 \text{ ft}^3/\text{s}; S=0.015; n=0.02$$

Uso de SN CANALES:

SECCIÓN HIDRÁULICA ÓPTIMA DE UNA SECCIÓN TRIANGULAR

Datos de entrada:	
Caudal (Q)=	3 ft ³ /s
Pendiente (S)=	0.015 ft/ft
Coficiente de Manning (n)=	0.02

Datos de salida:	
Talud (z)=	1
Tirante (y)=	0.8554 ft
Área mojada (A)=	0.7317 ft ²
Perímetro mojado (P)=	2.4194 ft
Radio hidráulico (R)=	0.3024 ft
Ancho superficial (T)=	1.7108 ft
Profundidad hidráulica (D)=	0.4277 ft
Velocidad (V)=	4.1 ft/s
Número de Froude (NF)=	1.1051
Tipo de flujo=	Supercrítico
Energía específica (E)=	1.1165 ft-lb/lb

Sistema de unidades:

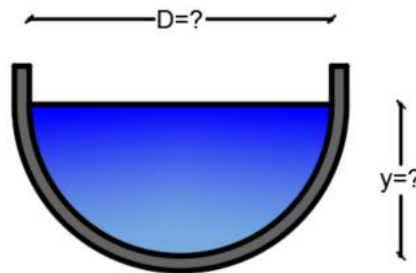
Sistema métrico

Sistema inglés

4.5.4. Ejemplo de cálculo de una sección semicircular.

Determinar la sección hidráulica óptima para un canal semi circular con $n=0.025$ si se va a diseñar para transportar $4\text{m}^3/\text{s}$ y una pendiente de 0.012.

Datos del problema:



$$Q=4\text{m}^3/\text{s}; S=0.012 ; n=0.025$$

Uso de SN CANALES:

SECCIÓN HIDRÁULICA ÓPTIMA DE UNA SECCIÓN SEMICIRCULAR

Datos de entrada:

Caudal (Q)=	<input type="text" value="4"/>	m ³ /s
Pendiente (S)=	<input type="text" value="0.012"/>	m/m
Coefficiente de Manning (n)=	<input type="text" value="0.025"/>	

Datos de salida:

Diámetro (d)=	<input type="text" value="1.9404"/>	m
Tirante (y)=	<input type="text" value="0.9702"/>	m
Área mojada (A)=	<input type="text" value="1.4786"/>	m ²
Perímetro mojado (P)=	<input type="text" value="3.048"/>	m
Radio hidráulico (R)=	<input type="text" value="0.4851"/>	m
Ancho superficial (T)=	<input type="text" value="1.9404"/>	m
Profundidad hidráulica (D)=	<input type="text" value="0.762"/>	m
Velocidad (V)=	<input type="text" value="2.7053"/>	m/s
Número de Froude (NF)=	<input type="text" value="0.9895"/>	
Tipo de flujo=	<input type="text" value="Subcrítico"/>	
Energía específica (E)=	<input type="text" value="1.3432"/>	m-Kg/Kg

Sistema de unidades:

Sistema métrico
 Sistema inglés