



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA CIVIL Y MECÁNICA
CARRERA DE INGENIERÍA CIVIL

**PROYECTO TÉCNICO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO CIVIL**

TEMA:

**“DESARROLLO DE UNA APLICACIÓN PARA EL CÁLCULO Y
DISEÑO DE CANALES ABIERTOS DE DIFERENTES SECCIONES
HIDRAÚLICAS”**

AUTOR: Alex Vinicio Tipán Chancusi

TUTOR: Ing. M. Sc. Fabián Rodrigo Morales Fiallos

AMBATO – ECUADOR

Marzo - 2021

CERTIFICACIÓN

En mi calidad de Tutor del Proyecto Técnico, previo a la obtención del título de Ingeniero Civil, con el tema: “**DESARROLLO DE UNA APLICACIÓN PARA EL CÁLCULO Y DISEÑO DE CANALES ABIERTOS DE DIFERENTES SECCIONES HIDRAÚLICAS**”, elaborado por el Sr. Alex Vinicio Tipán Chancusi, portador de la cedula de ciudadanía: C.I. 0503321580, estudiante de la Carrera de Ingeniería Civil de la Facultad de Ingeniería Civil y Mecánica.

Certifico:

- Que el presente proyecto técnico es original de su autor.
- Ha sido revisado cada uno de sus capítulos componentes.
- Esta concluido en su totalidad.

Ambato, Marzo 2021

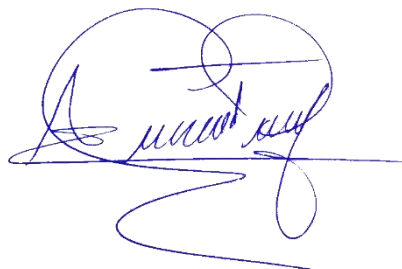
Ing. M. Sc. Fabián Rodrigo Morales Fiallos

TUTOR

AUTORÍA DE LA INVESTIGACIÓN

Yo, **Alex Vinicio Tipán Chancusi**, con C.I. 0503321580 declaro que todas las actividades y contenidos expuestos en el presente proyecto técnico con el tema **“DESARROLLO DE UNA APLICACIÓN PARA EL CÁLCULO Y DISEÑO DE CANALES ABIERTOS DE DIFERENTES SECCIONES HIDRAÚLICAS”**, así como también los análisis estadísticos, gráficos, conclusiones y recomendaciones son de mi exclusiva responsabilidad como autor del proyecto, a excepción de las referencias bibliográficas citadas en el mismo.

Ambato, Marzo 2021



Alex Vinicio Tipán Chancusi

C.I:0503321580

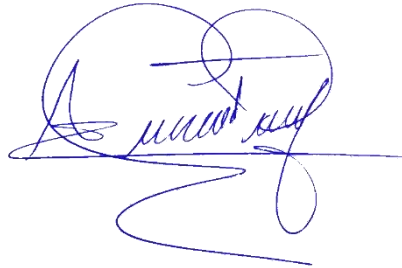
AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que se haga de este proyecto de titulación bajo la modalidad de Trabajo Técnico o parte de ello un documento disponible para su lectura, consulta y proceso de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de nuestro Trabajo Técnico con los fines de difusión pública, además aprobamos la reproducción de este documento dentro de las regulaciones de la Universidad.

Ambato, Marzo 2021



Alex Vinicio Tipán Chancusi

C.I:0503321580

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

Los miembros del Tribunal de Grado aprueban el informe del Proyecto Técnico, realizado por el estudiante Alex Vinicio Tipán Chancusi. De la Carrera de Ingeniería Civil bajo el tema: **“DESARROLLO DE UNA APLICACIÓN PARA EL CÁLCULO Y DISEÑO DE CANALES ABIERTOS DE DIFERENTES SECCIONES HIDRAÚLICAS”**.

Ambato, Marzo 2021

Para constancia firman:

Ing. M. Sc Dylon German Moya Medina
Miembro Calificador

Ing. M. Sc Fidel Alberto Castro Solorzano
Miembro Calificador

DEDICATORIA

*Quiero dedicar principalmente a Dios quien me dio la lucidez
para alcanzar este gran logro.*

*A mis padres quienes por siempre me dieron su apoyo
incondicional para seguir luchando día tras día.*

*A mi esposa, a mis hijos quienes son el motivo por quien siempre
luchare para salir adelante.*

*A mi Hermano Efraín quien estuvo siempre conmigo
apoyándome en la trayectoria de mi vida profesional.*

*A mi familia en general, que siempre estuvieron pendientes
cuando lo necesitaba.*

Alex Típán

AGRADECIMIENTO

Gracias Papito Dios por darme la vida y la salud que es la parte primordial en una persona, gracias por esa maravillosa familia que tengo junto a mí.

A mi madre María Hortensia Chancusi y a mi padre Luis Enrique Típán por formar y hacerme un hombre de bien, inculcándome buenos valores y seguir siendo un ejemplo de esfuerzo y dedicación.

A mi esposa Lilitiana Zurita quien estuvo siempre conmigo, dándome ese apoyo para alcanzar este gran objetivo y a mis hijos Jeremy Típán y Kimberly Típán quienes son el motor por cual me sigo esforzando todos los días, me dan esas fuerzas para seguir luchando y venciendo todos los obstáculos que se presentan en el camino.

Al Ing. Fabián Morales por su asesoría y colaboración en la elaboración de este proyecto.

Gracias a mi querida Universidad por brindarme ese gran conocimiento, a mis compañeros y amigos por haber compartido conmigo una de las etapas maravillosas de la vida.

ÍNDICE GENERAL DE CONTENIDOS

A. PÁGINAS PRELIMINARES

CERTIFICACIÓN	ii
AUTORÍA DE LA INVESTIGACIÓN	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS.....	viii
ÍNDICE DE TABLAS	xiv
ÍNDICE DE FIGURAS.....	xv
RESUMEN.....	xvi
ABSTRACT.....	xvii

B. TEXTO

CAPÍTULO I.....	1
1.1. TEMA DEL TRABAJO TÉCNICO	1
1.2. ANTECEDENTES.....	1
1.3. JUSTIFICACIÓN	2
1.4. OBJETIVOS	3
1.4.1. GENERAL	3
1.4.2. ESPECÍFICOS	3
CAPÍTULO II	4
FUNDAMENTACIÓN.....	4
2.1. FUNDAMENTACIÓN TEÓRICA.....	4
2.1.1. CANALES	5

2.1.2.	GEOMETRÍA DE LOS CANALES.....	5
2.1.2.1.	ELEMENTOS GEOMÉTRICOS DE UNA SECCIÓN DE CANAL.	6
2.1.3.	ESTADO DE FLUJO EN CANALES ABIERTOS	9
2.1.3.1.	EFFECTO DE LA GRAVEDAD.....	9
2.1.4.	FLUJO UNIFORME EN CANALES ABIERTOS	10
2.1.5.	PENDIENTE DEL CANAL	11
2.1.5.1.	FORMAS PARA EXPRESAR LA PENDIENTE.....	12
2.1.6.	ECUACIÓN DE MANNING	12
2.1.6.1.	VALORES DEL COEFICIENTE DE RUGOSIDAD DE MANNING	13
2.1.7.	CÁLCULO DEL FLUJO UNIFORME	14
2.1.7.1.	EL FACTOR DE SECCIÓN PARA EL CÁLCULO DEL FLUJO UNIFORME.....	16
2.1.7.2.	CÁLCULO DE LA PENDIENTE NORMAL Y CRÍTICA	17
2.1.8.	SECCIÓN HIDRÁULICA ÓPTIMA	18
2.1.9.	ENERGÍA.	19
2.1.9.1.	ENERGÍA ESPECÍFICA.....	22
2.1.9.2.	CURVAS DE ENERGÍA ESPECÍFICA	22
2.1.10.	FLUJO CRÍTICO.....	25
2.1.10.1.	DETERMINACIÓN DEL FLUJO CRÍTICO.....	26
2.1.10.2.	EL FACTOR DE SECCIÓN PARA EL CÁLCULO DEL FLUJO CRÍTICO.....	28
2.2.	HIPÓTESIS	29
2.3.	SEÑALAMIENTO DE LAS VARIABLES DE LA HIPÓTESIS	29
2.3.1.	VARIABLE INDEPENDIENTE.....	29
2.3.2.	VARIABLE DEPENDIENTE	29
CAPÍTULO III		30

METODOLOGÍA	30
3.1. NIVEL O TIPO DE INVESTIGACIÓN	30
3.2. POBLACIÓN Y MUESTRA.....	31
3.2.1. POBLACIÓN	31
3.2.2. MUESTRA.....	31
3.3. OPERACIONALIZACIÓN DE VARIABLES	32
3.3.1. VARIABLE INDEPENDIENTE	32
3.3.2. VARIABLE DEPENDIENTE	33
3.4. PLAN DE RECOLECCION DE INFORMACIÓN	35
3.5. PLAN DE PROCESAMIENTO Y ANÁLISIS	35
CAPÍTULO IV	37
ANÁLISIS E INTERPRETACIÓN DE RESULTADOS.....	37
4.1. NIVEL O TIPO DE INVESTIGACIÓN	37
4.1.1. EJEMPLO PARA EL CÁLCULO DE LOS ELEMENTOS GEOMÉTRICOS DE UN CANAL RECTANGULAR CON ESQUINAS REDONDEADAS.....	37
4.1.1.1. MÉTODO MANUAL	38
4.1.1.2. MÉTODO CON UN SOFTWARE EXISTENTE	40
4.1.1.3. MÉTODO CON EL SOFTWARE DESARROLLADO.....	41
4.1.2. EJEMPLO PARA EL CÁLCULO DEL CAUDAL SECCIÓN TRIANGULAR CON FONDO REDONDEADO.....	42
4.1.2.2. MÉTODO CON UN SOFTWARE EXISTENTE	45
4.1.2.3. MÉTODO CON EL SOFTWARE DESARROLLADO.....	46
4.1.3. EJEMPLO PARA EL CÁLCULO DE LA SECCIÓN HIDRAÚLICA ÓPTIMA (SECCIÓN SEMICIRCULAR).....	47
4.1.3.2. MÉTODO CON UN SOFTWARE EXISTENTE	51

4.1.3.3. MÉTODO CON EL SOFTWARE DESARROLLADO.....	52
4.1.4. EJEMPLO PARA EL CÁLCULO DEL TIRANTE NORMAL DE UNA SECCIÓN TRAPEZOIDAL	53
4.1.4.1. MÉTODO MANUAL	53
4.1.4.2. MÉTODO CON EL SOFTWARE SN CANALES V2.0L.....	59
4.1.4.3. MÉTODO CON EL SOFTWARE HYDROEOLM_CANALES.....	60
4.1.4.4. MÉTODO CON EL SOFTWARE DESARROLLADO.....	61
4.1.5. EJEMPLO PARA EL CALCULO DEL TIRANTE CRÍTICO DE UNA SECCIÓN TRIANGULAR.....	62
4.1.5.1. MÉTODO MANUAL	63
4.1.5.2. MÉTODO CON EL SOFTWARE SN CANALES V 2.0L.....	66
4.1.5.3. MÉTODO CON EL SOFTWARE HYDROEOLM_CANALES.....	67
4.1.5.4. MÉTODO CON EL SOFTWARE DESARROLLADO.....	68
4.2. ANALISIS DE RESULTADOS	69
4.3. VERIFICACIÓN DE HIPÓTESIS	71
CAPÍTULO V	72
CONCLUSIONES Y RECOMENDACIONES.....	72
5.1. CONCLUSIONES	72
5.2. RECOMENDACIONES	73
C. MATERIALES DE REFERENCIA	
1. BIBLIOGRAFÍA	74
2. ANEXOS	75
2.1. CODIFICACIÓN DE LA APLICACIÓN MOVIL FICM CANALES V1.0.	75
2.1.1. ELEMENTOS GEOMETRICOS	75
2.1.1.1. SECCIÓN RECTANGULAR.....	75
2.1.1.2. SECCIÓN TRAPEZOIDAL	79

2.1.1.3. SECCIÓN TRIANGULAR.....	83
2.1.1.4. SECCIÓN CIRCULAR	87
2.1.1.5. SECCIÓN PARABÓLICO	92
2.1.1.6. SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS.....	97
2.1.1.7. SECCIÓN TRIANGULAR CON FONDO REDONDEADO.....	101
2.1.2. CAUDALES	107
2.1.2.1 SECCIÓN RECTANGULAR.....	107
2.1.2.2 SECCIÓN TRAPEZOIDAL	112
2.1.2.3 SECCIÓN TRIANGULAR.....	118
2.1.2.4 SECCIÓN CIRCULAR	123
2.1.2.5 SECCIÓN PARABÓLICA	129
2.1.2.6 SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS.....	135
2.1.2.7 SECCIÓN TRIANGULAR CON FONDO REDONDEADO.....	141
2.1.3. SECCIÓN HIDRÁULICA ÓPTIMA	148
2.1.3.1 SECCIÓN RECTANGULAR.....	148
2.1.3.2 SECCIÓN TRAPEZOIDAL CON “Z” CONOCIDO	154
2.1.3.3 SECCIÓN TRAPEZOIDAL CON “Z” NO CONOCIDO.....	161
2.1.3.4 SECCIÓN TRIANGULAR.....	169
2.1.3.5 SECCIÓN SEMICIRCULAR.....	175
2.1.4. TIRANTE NORMAL	183
2.1.4.1 SECCIÓN RECTANGULAR.....	183
2.1.4.2 SECCIÓN TRAPEZOIDAL	189
2.1.4.3 SECCIÓN TRIANGULAR.....	196
2.1.4.4 SECCIÓN CIRCULAR	203
2.1.4.5 SECCIÓN PARABÓLICA CON “T” CONOCIDO	210

2.1.4.6 SECCIÓN PARABÓLICA CON “T” NO CONOCIDO.....	218
2.1.4.7 SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS.....	225
2.1.4.8 SECCIÓN TRIANGULAR CON FONDO REDONDEADO.....	232
2.1.4. TIRANTE CRÍTICO.....	241
2.1.4.1 SECCIÓN RECTANGULAR.....	241
2.1.4.2 SECCIÓN TRAPEZOIDAL	247
2.1.4.3 SECCIÓN TRIANGULAR.....	254
2.1.4.4 SECCIÓN CIRCULAR	260
2.1.4.5 SECCIÓN PARABÓLICA CON “T” CONOCIDO	267
2.1.4.6 SECCIÓN PARABÓLICA CON “T” NO CONOCIDO.....	273
2.1.4.7 SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS.....	280
2.1.4.8 SECCIÓN TRIANGULAR CON FONDO REDONDEADO.....	287
2.2. TUTORIAL PARA SUBIR UNA APP A LA PLAY STORE.....	296

ÍNDICE DE TABLAS

Tabla 1: Elementos geométricos de secciones de canal.....	8
Tabla 2: Ecuaciones para el cálculo del flujo uniforme en unidades métricas e inglesas.	16
Tabla 3: Operacionalización de la variable independiente	32
Tabla 4: Operacionalización de variable dependiente	33
Tabla 5: Plan de recolección de información.....	35
Tabla 6: Comparación de resultados en el cálculo de los elementos geométricos de una canal rectangular con esquinas redondeadas	69
Tabla 7: Comparación de resultados en el cálculo del caudal de una de una sección triangular con fondo redondeado	69
Tabla 8: Comparación de resultados en una sección hidráulica óptima (Sección Semicircular).....	70
Tabla 9: Comparación de resultados en el cálculo del tirante normal de una sección trapezoidal.....	70
Tabla 10: Comparación de resultados en el cálculo del tirante crítico de una sección triangular	71

ÍNDICE DE FIGURAS

Figura 1: Flujo en conductos	5
Figura 2: Flujo Uniforme.....	11
Figura 3: Pendiente de un canal.....	12
Figura 4: Sección hidráulica óptima	19
Figura 5: Energía total en una sección de un canal.....	20
Figura 6: Curva de energía específica.....	23
Figura 7: Profundidad crítica y profundidades alternas.....	24
Figura 8: Gráfica para el cálculo de los elementos geométricos de una sección rectangular con esquinas redondeadas	38
Figura 9: Gráfica para el cálculo del caudal de una sección triangular con fondo redondeado	42
Figura 10: Gráfica para el cálculo de una sección semicircular	47
Figura 11: Gráfica para el cálculo del tirante normal de una sección trapezoidal....	53
Figura 12: Gráfica para el cálculo del tirante crítico de una sección triangular	62

RESUMEN

El propósito de este trabajo técnico es el desarrollo de una aplicación móvil en un entorno Android para el cálculo y diseño de canales abiertos el cual permite agilizar el proceso de cálculos.

Se obtuvo resultados exactos y precisos como son en el caso de los elementos geométricos, caudal, secciones optimas, tirante normal, tirante crítico; se pueden realizar el cálculo de canales en secciones como: rectangular, trapezoidal, triangular, circular, parabólica, rectangular con esquinas redondeadas, triangular con fondo redondeado. También podemos obtener resultados específicos como son: la velocidad, numero de froude, tipo de flujo, energía específica entre otros junto con las fórmulas que plantea Ven Te Chow. La codificación de esta aplicación se lo realizo en Netbeans64 y Android Studio. La eficiencia de esta aplicación móvil en un entorno Android es aceptable ya que se realizaron comprobaciones con diferentes softwares como son: SN CANALES, HydroEolm_Canales y el método manual. Se pudo determinar un margen de error menor a 0.5 por ciento, lo cual le hace más fiable a esta aplicación móvil.

ABSTRACT

The purpose of this technical work is the development of a mobile application for the Android system. The purpose is to design open channels which allows to speed up the process of calculations with exact and precise results such as in the case of geometric elements, flow, sections optimal, normal depth critical depth you can calculate channels in sections such as: rectangular, trapezoidal, triangular, circular, parabolic, rectangular with rounded corners, triangular with rounded bottom. We can also obtain specific results such as: speed, froude number, type of flow, specific energy among others along with the formulas proposed by Ven Te Chow.

The encoding of this application was done in Netbeans64 and Android Studio. The efficiency of this mobile application in an Android environment is acceptable and checks were carried out with different software such as: SN CHANNELS, HydroEolm Channels and the manual method. A margin of error less than 0.5 percent could be determined, which makes this mobile application more reliable.

CAPÍTULO I

1.1.TEMA DEL TRABAJO TÉCNICO

“DESARROLLO DE UNA APLICACIÓN PARA EL CÁLCULO Y DISEÑO DE CANALES ABIERTOS DE DIFERENTES SECCIONES HIDRAÚLICAS.”

1.2.ANTECEDENTES

El software en la ingeniería se remonta a el inicio propio de la era de los computadores pues el desarrollo de estos sistemas siempre tiene el mismo fin, el cual es la solución de problemas de la vida cotidiana bajo modelos matemáticos que modelen y predigan el comportamiento de las variables que se estén tratando [1].

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].

Pero las grandes civilizaciones andinas fueron más allá de desarrollar conceptos que se descubrirían en el siglo XX. Para ellas, el manejo de un recurso básico como el agua estaba ligado a la concepción de la misma sociedad e influía en la tecnología a usar. Es por eso que diseñaron sus sistemas hidráulicos bajo concepciones totalmente diferentes a las de la cultura occidental, que son las que predominan ahora [2].

1.3.JUSTIFICACIÓN

Alrededor del mundo se han realizado muchas obras de ingeniería hidráulica donde los canales abiertos han desempeñado un papel muy importante.

La razón por el cual se realizó esta aplicación móvil en un entorno Android en el área de la hidráulica de canales abiertos, es la facilidad y necesidad que tiene los diseñadores de canales al momento de optimizar el tiempo y recursos.

Es muy importante para el diseño de canales hacer referencia al tipo de canal, sección transversal, pendiente, revestimiento y geometría. También se requiere saber variables como la velocidad, caudal, coeficientes de rugosidad.

Dadas las variables anteriormente mencionadas se realiza un proceso iterativo con el cual convergen valores obtenidos de tablas y nomogramas para los cuales muestran el valor del revestimiento, borde libre y altura de bancas; esto hace referencia si mencionamos de diseño de canales con revestimiento.

Estos valores, formulas, tablas, métodos y ecuaciones; son programadas, almacenados y cargados en una base de datos que son la parte principal de la aplicación móvil para el diseño de canales.

La parte fundamental de este software para el diseño de canales, será el resultado de muy buena calidad generado por la Universidad Técnica de Ambato y por la persona que realizo esta tesis, y esta podrá ser comercializada a la comunidad educativa y profesional en el área de la hidráulica.

1.4.OBJETIVOS

1.4.1. GENERAL

- Desarrollo de una aplicación móvil para el cálculo y diseño de canales abiertos de diferentes secciones hidráulicas.

1.4.2. ESPECÍFICOS

- Desarrollar una aplicación móvil con el software Android Studio que permita el cálculo de Elementos Geométricos, Caudales, Secciones Optimas, Tirante normal, Tirante Crítico.
- Vincular a la Ingeniería Civil una herramienta que permita optimizar los cálculos en el diseño de canales abiertos.
- Realizar el procedimiento legal del software para el uso con la comunidad Universitaria.

CAPÍTULO II

FUNDAMENTACIÓN

2.1. FUNDAMENTACIÓN TEÓRICA

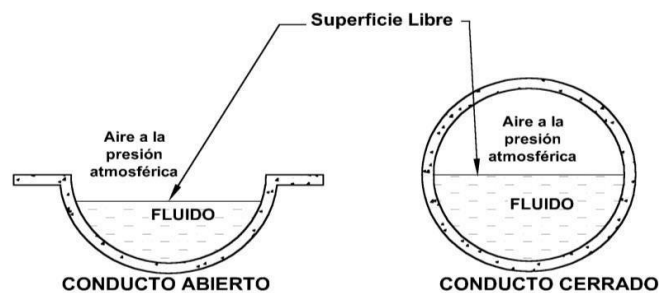
Los canales como estructuras hidráulicas requieren un apropiado diseño para su correcto funcionamiento de la mano de la optimización de recursos, por lo que es necesaria la consideración de ciertas variables que engloban el concepto de máxima eficiencia hidráulica. Dentro del campo de la hidráulica se requiere en los procesos de cálculo estructural el manejo de herramientas tecnológicas de vanguardia que hacen que el trabajo del ingeniero sea mucho más eficiente y productivo, tal es el caso del desarrollo e implementación de softwares que se han convertido en instrumentos que coadyuvan a planificar, optimizar, diseñar y operar los sistemas hídricos [1].

Un canal abierto es un conducto con un flujo el cual tiene una superficie libre, una de las fronteras está expuesta a la atmósfera. La superficie libre es esencialmente una interface entre dos fluidos de diferentes densidades. En el caso de la atmósfera, la densidad del aire es mucho menor que la densidad de un líquido como el agua, además la presión es constante [2]. Los canales pueden ser clasificados como naturales o artificiales. La terminología canal natural se refiere a todos los canales que han sido desarrollados por procesos naturales y que no han tenido una mejoría significativa por parte de los humanos [3]. Mientras tanto, los canales artificiales son todos aquellos construidos o desarrollados mediante el esfuerzo de la mano del hombre, tales como: canales de riego, de navegación, control de inundaciones, canales de centrales hidroeléctricas, alcantarillado pluvial, sanitario, canales de desborde, canaletas de madera, cunetas a lo largo de carreteras, cunetas de drenaje agrícola y canales de modelos construidos en el laboratorio. Los canales artificiales usualmente se diseñan con forma geométricas regulares (prismáticos), un canal construido con una sección transversal invariable y una pendiente de fondo constante se conoce como canal prismático [4].

2.1.1. CANALES

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 [3].

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 [1].

El término sección de canal se refiere a la sección transversal de un canal tomado 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 [1].

2.1.2.1. ELEMENTOS GEOMÉTRICOS DE UNA SECCIÓN DE CANAL.

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 [1]

✓ **Profundidad del flujo.**

Es la distancia vertical desde el punto más bajo de una sección del canal hasta la superficie libre, es decir la profundidad máxima del agua en el canal [1].

✓ **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 [1].

✓ **Ancho Superficial.**

El ancho superficial (T) es el ancho de la sección del canal en la superficie libre [1].

✓ **Área mojada.**

El área mojada (A) es el área de la sección transversal del flujo perpendicular a la dirección del flujo [1].

✓ **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 [1].

✓ **Radio Hidráulico.**

El radio hidráulico (R) es la relación entre el área mojada con respecto a su perímetro mojado [1].

Ecuación 2: Definición del radio hidráulico

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

✓ **Profundidad Hidráulica.**

La profundidad hidráulica (D) es la relación entre el área mojada y el ancho en la superficie [1].

Ecuación 3: Definición de la profundidad hidráulica

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

✓ **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 [1].

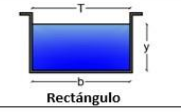
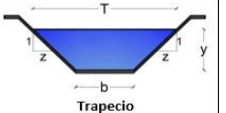
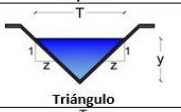
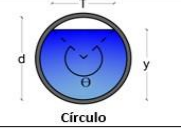

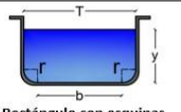

Ecuación 4: Definición del factor de sección para el cálculo del flujo crítico

$$z = A * \sqrt{D} = A * \sqrt{\frac{A}{T}}$$

✓ **Factor de sección para el cálculo del flujo uniforme**

El factor de sección para el cálculo de flujo uniforme $A * Rh^{2/3}$ es el producto del área mojada y el radio hidráulico elevado a la potencia 2/3 [1].

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	$\frac{1}{8}(\theta - \sin \theta)d^2$	$\frac{1}{2}\theta d$	$\frac{1}{4}\left(1 - \frac{\sin \theta}{\theta}\right)d$	$\frac{\left(\sin \frac{\theta}{2}\right)d}{2\sqrt{y(d - y)}}$	$\frac{1}{8}\left(\frac{\theta - \sin \theta}{\sin \frac{\theta}{2}}\right)d$	$\frac{\sqrt{2}}{32}\frac{(\theta - \sin \theta)^{1.5}}{\left(\sin \frac{\theta}{2}\right)^{0.5}}d^{2.5}$
 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	$\frac{T^2}{4z} - \frac{r^2}{z}(1 - z \cot^{-1} z)$	$\frac{T}{z}\sqrt{1 + z^2} - \frac{2r}{z}(1 - z \cot^{-1} z)$	$\frac{A}{P}$	$2\left[z(y - r) + r\sqrt{1 + z^2}\right]$	$\frac{A}{T}$	$A\sqrt{\frac{A}{T}}$

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

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

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 [3].

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 [3].

✓ Número de Froude

Se tiene la ecuación 4 y la clasificación del flujo de acuerdo al número de Froude [3].

Ecuación 4: Número de Froude

$$NF = \frac{V}{\sqrt{g * D}} = \frac{V}{\sqrt{g * \frac{A}{T}}}$$

NF = número de froude

V = velocidad de flujo

D = profundidad hidráulica

A = área hidráulica

T = espejo de agua

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

✓ **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 $NF=1$ [4].

✓ **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 $NF<1$ [4].

✓ **Flujo supercrítico**

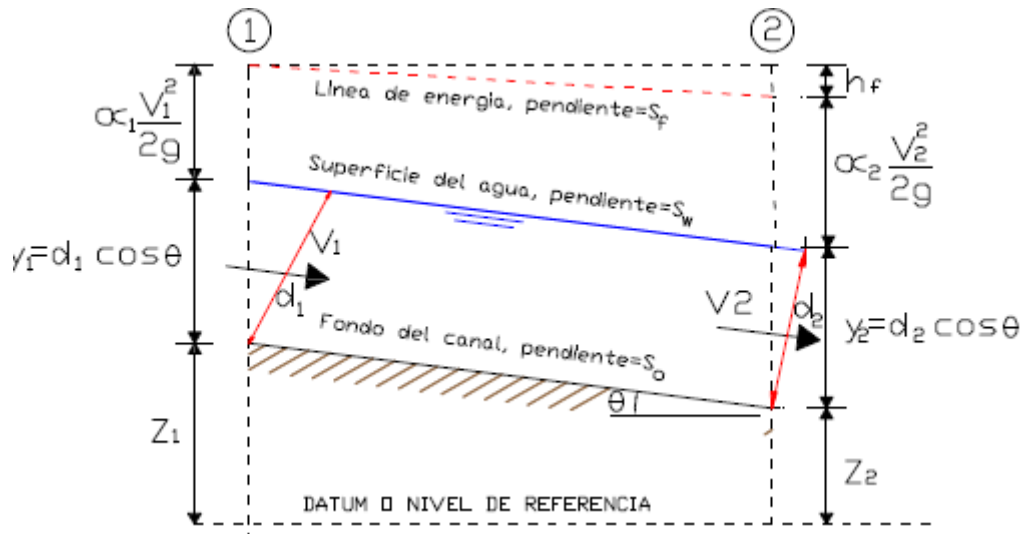
El flujo supercrítico o flujo rápido es aquel en el cual la velocidad del flujo es mayor que la velocidad de la onda de gravedad, el parámetro adimensional $NF>1$ [4].

2.1.4. FLUJO UNIFORME EN CANALES ABIERTOS

En la figura 2 se muestra un canal con una pendiente alta, la línea que representa la elevación de la altura total de flujo es la línea de energía donde la pendiente de esta línea se denomina gradiente de energía, simboliza por S_f .

La pendiente de la superficie del líquido que para el agua se representa por S_w y la pendiente del fondo del canal se representa por $S_o=\text{sen}\theta$, por lo general la pendiente se define como $\tan \theta$, pero por el rango de valores que se manejan los cuales son pequeños se define como $\text{sen}\theta$, donde para el flujo uniforme cumple que $S_f=S_w=S_o= \text{sen}\theta$ [4].

Figura 2: Flujo Uniforme



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 utilizados 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 [1].

Ecuación 6: Pendiente de un canal

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

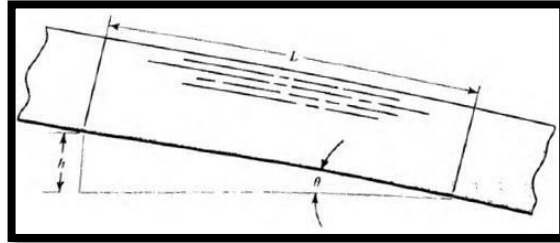
Donde:

$S =$ pendiente del canal

$L =$ longitud del canal

$h = \text{distancia vertical}$

Figura 3: Pendiente de un canal



2.1.5.1.FORMAS PARA EXPRESAR LA PENDIENTE

Es posible expresar una pendiente de 0.001 [12]:

- El canal desciende 1m por cada 1000m que avanza.
- En porcentaje, donde $0.001=0.1\%$.
- 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

Esta fórmula fue desarrollada por el ingeniero Irlandes Robert Manning en el año de 1889. 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 [4].

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

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

Donde:

V= Velocidad del flujo (m/s)

n= Factor de resistencia o n de

Manning R= Radio hidráulico (m)

S= Pendiente del canal.

Más tarde, fue reconvertida a unidades inglesas, dando como resultado [1].

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

$$V = \frac{1.486}{n} * Rh^{\frac{2}{3}} * S^{\frac{1}{2}} \quad (ft/s)$$

2.1.6.1. VALORES DEL COEFICIENTE DE RUGOSIDAD DE MANNING

- **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 [1].
- **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 [1].
- **Irregularidad del canal.** Un canal incluye 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 [1].

- **Alineamiento del canal.** Curvas suaves con radios grandes producirán valores de n relativamente bajos, en tanto que curvas bruscas con meandros severos incrementarán n [1].
- **Sedimentación.** 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 [1].
- **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 [1].
- **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 [1].

2.1.7. CÁLCULO DEL FLUJO UNIFORME

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

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 [1]:

Ecuación 10: Ecuación de Manning para caudal en unidades métricas

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

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).

Según [1] se puede obtener otra forma equivalente a la ecuación 10 para el sistema inglés.

Ecuación 11: Ecuación de Manning para caudal en unidades inglesas

$$Q = \frac{1.486}{n} * A * Rh^{\frac{2}{3}} * S^{1/2}$$

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

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

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 $A * Rh^{\frac{2}{3}}$ 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 [1]:

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

$$A * Rh^{\frac{2}{3}} = (n * Q / S^{\frac{1}{2}})$$

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 $A * Rh^{\frac{2}{3}}$ 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 [1].

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

$$A * Rh^{\frac{2}{3}} = \left(\frac{n * Q}{1.486 * S^{\frac{2}{3}}} \right)$$

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 [1].

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 [1].

Según [3] 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

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

Ecuación 15: Cálculo de pendiente para el sistema inglés

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

Donde:

S = pendiente del canal

V = velocidad de flujo

n = Factor de resistencia o n de Manning

Rh = radio hidráulico

2.1.8. 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 [4].

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 [4].

La conductividad o tránsito de agua en un canal, depende en gran medida de la relación existente entre al área mojada y el perímetro mojado en un canal [4].

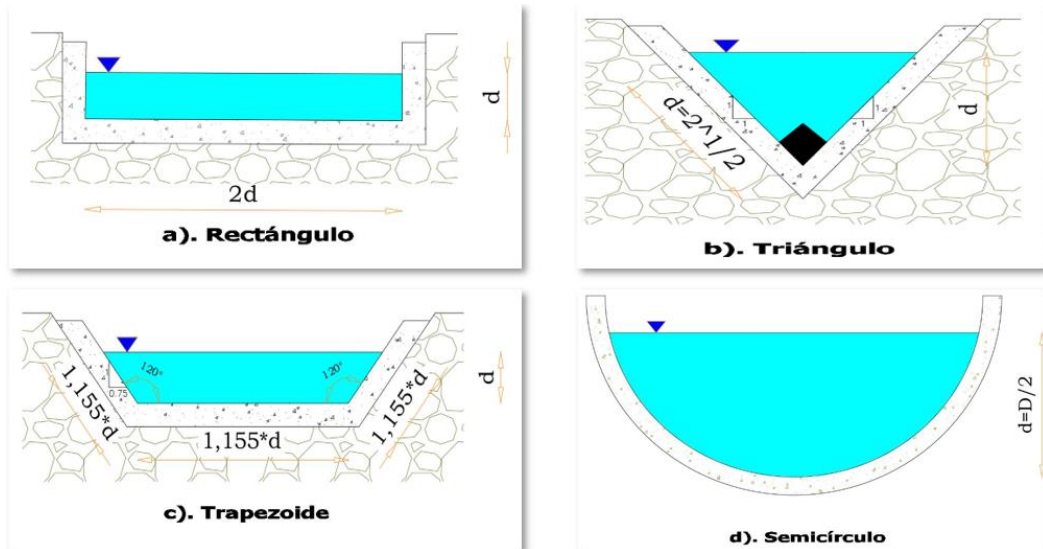
Se sabe que la conductividad de una sección de canal se incrementa con el aumento en el radio hidráulico o la disminución en el perímetro mojado [4].

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.

Figura 4: Sección hidráulica óptima

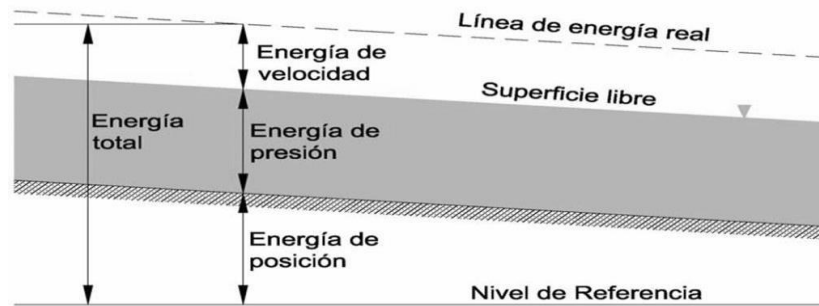


2.1.9. ENERGÍA.

La energía total de cualquier línea de corriente que pasa a través de una sección se define como la suma de las energías de posición, más la de presión y más de la velocidad, es decir:

Energía total = Energía de posición + Energía de presión + Energía de velocidad.

Figura 5: 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) [5].

Ecuación 16: Energía potencial

$$PE = W * z$$

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 W/g (donde W es el peso y g la aceleración de la gravedad) [5].

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 $\frac{W}{\gamma}$ por específico del fluido [5].

Ecuación 18: Energía de presión

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

La energía total (ET) es la suma de PE, KE Y FE [5].

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 [5].

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 [5].

2.1.9.1. 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 [3].

Ecuación 21: Energía específica

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

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

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 [3]:

Ecuación 22: Energía específica en función del caudal

$$E = y + \frac{Q^2}{2 * g * A^2}$$

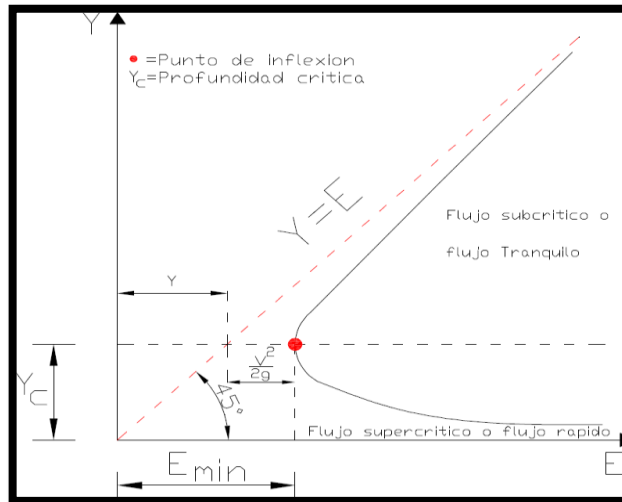
2.1.9.2. CURVAS DE ENERGÍA ESPECÍFICA

Para visualizar los regímenes posibles de flujo en un canal, es útil una gráfica de la profundidad versus la energía específica [1].

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 [4].

Figura 6: Curva de energía específica

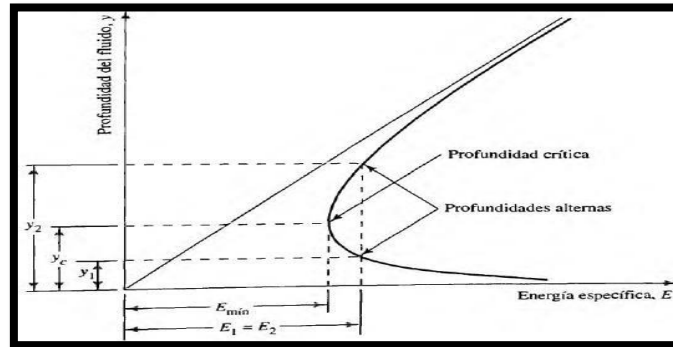


Fuente: L. Acosta, J. Barragan, y A. Reyes. Software para el diseño de canales abiertos. 2007.

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 representa la energía potencial (y). La distancia restante a la curva de la energía específica es la energía cinética $\frac{v^2}{2g}$, aparece un valor mínimo definido de E , y se demuestra que ocurre cuando el flujo se halla en el estado crítico, es decir cuando $NF = 1$ [4].

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 , tiene 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) [4].

Figura 7: Profundidad crítica y profundidades alternas



Fuente: S. Naranjo. “Desarrollo de un software para el cálculo de canales abiertos de flujo uniforme”. Pregrado, Universidad Técnica de Ambato, 2017.

Energía específica mínima (Emín.): 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 [3].

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 $y_n > y_c$ [4].
- 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 $NF = 1$ [4].
- 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 $y_n < y_c$ [4].
- Cada caudal tiene su propia curva de energía específica, es decir que la curva se desplaza [4].
- 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 [4].
- 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 [4].

- Cuando se presente flujo subcrítico, siempre que se aumente la profundidad del flujo, la energía específica aumentara y viceversa [4].
- Cuando se presente flujo supercrítico, siempre que se aumente la profundidad del flujo, la energía específica disminuirá y viceversa [4].

Cuando se presenta 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 [4].

2.1.10. 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 [1].

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 [3].

- **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 [3].

- **Velocidad crítica.**

La velocidad media cuando el caudal es el crítico [3].

- **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 [3].

2.1.10.1. DETERMINACIÓN DEL FLUJO CRÍTICO

Para que se presente un estado de flujo crítico, se ha de cumplir la condición en la cual en 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 [4].

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 $1.1 y_c < y < 0.9 y_c$ con $y \neq y_c$ [4].

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 [1],

$$\frac{dE}{dy} = 1 - \frac{Q^2}{g * A^3} * \frac{dA}{dy}$$

El diferencial de la era mojada dA cerca de la superficie libre es igual $T*dy$. Ahora $\frac{dA}{dy} = T$, luego la anterior ecuación se convierte en [4]:

Ecuación 24: Ecuación general del flujo crítico

$$\frac{Q^2}{g * A^3} * T = 1$$

Con la ecuación 24, dejando el caudal en términos de la velocidad se puede determinar el estado crítico de flujo [4].

Ecuación 25: Estado crítico de flujo

$$\frac{V}{\sqrt{g * D}} = 1$$

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

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 [4].

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 [1].

Ecuación 27: Criterio para flujo crítico

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

2.1.10.2. EL FACTOR DE SECCIÓN PARA EL CÁLCULO DEL FLUJO CRÍTICO

Al sustituir $V = Q/A$ en la ecuación 27 y simplificando se tiene [1]:

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 [1].

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 [1].

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 un Software

2.3.2. VARIABLE DEPENDIENTE

- Calculo y diseño de canales abiertos de diferentes secciones hidráulicas.

CAPÍTULO III

METODOLOGÍA

3.1. NIVEL O TIPO DE INVESTIGACIÓN

Es importante conocer el enfoque práctico del software para el diseño y el aprendizaje de la hidráulica de canales abiertos, complementando 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 que se debe aplicar [4].

Es muy importante conocer las características del canal para el cálculo del flujo uniforme en la cual están involucradas variables como caudal normal, velocidad de flujo, rugosidad del canal, pendiente y las secciones del canal [3].

Existen métodos de comprobación y solución para el cálculo de canales como es el método algebraico, método manual, método gráfico, método de tablas de diseño, etc.

Para el diseño hidráulico de canales se requiere un proceso de cálculo para la obtención de las dimensiones, requiere evaluar área hidráulica, perímetro mojado, radio hidráulico entre otros factores que posteriormente se incorporan a la ecuación del gasto particular de Chezy – Manning [6].

Se desarrollará un procedimiento convencional para el desarrollo del software, como es la estructuración del algoritmo, métodos de cálculo para canales, ecuaciones, formulas hidráulicas e interfaces gráficas.

La metodología mencionada anteriormente se la aplicara en secciones como: rectangular, trapezoidal, parabólica, triangular, rectangular con esquinas redondeadas y triangular con fondo redondeado.

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 una aplicación

Tabla 3: Operacionalización de la variable independiente

Conceptualización	Dimensiones	Indicadores	Ítems	Técnicas e Instrumentos
El Software son los programas de aplicación y los sistemas operativos que permiten que la computadora o el teléfono móvil pueda desempeñar tareas inteligentes	Lenguaje de programación	-Código fuente - Método de cálculo	¿Cómo aplicar el código de programación en el método de cálculo?	Investigación bibliográfica.
	Ayudar y facilitar el proceso de cálculo	-Eficiencia de los resultados - Precisión de los resultados	¿Cómo el software permite agilizar y optimizar el proceso de cálculo?	Investigación bibliográfica

Fuente: Alex Tipán

3.3.2. VARIABLE DEPENDIENTE

- Cálculo de canales abiertos de flujo uniforme.

Tabla 4: Operacionalización de variable dependiente

Conceptualización	Dimensiones	Indicadores	Ítems	Técnicas e Instrumentos
Los canales abiertos tienen lugar cuando los líquidos fluyen por la acción de la gravedad y solo están parcialmente envueltos por un contorno sólido. En el flujo de canales abiertos, el líquido que fluye tiene superficie libre y sobre él no actúa otra presión que la debida a su propio peso y a la presión atmosférica.	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 	<ul style="list-style-type: none"> ¿Cómo se calcula la profundidad del flujo? ¿Cuándo se presenta el tirante normal y el crítico? 	Investigación bibliográfica.

	Diseño de canales	- Sección hidráulica óptima	¿Cuáles son las secciones hidráulicas óptimas?	Investigación bibliográfica.
	Propiedades del flujo	- Energía específica - Número de Froude - Tipo de flujo	¿Cómo se determinan las propiedades del flujo?	Investigación bibliográfica

3.4. PLAN DE RECOLECCION DE INFORMACIÓN

Tabla 5: Plan de recolección de información

Preguntas Básicas	Explicación
1. ¿Para qué?	Facilitar el proceso de cálculo de canales abiertos de flujo uniforme en los que se considera los principios de energía, para obtener resultados precisos y disminuir el tiempo de cálculo.
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?	Alex Vinicio Tipán Chancusi
5. ¿Dónde?	En la Universidad Técnica de Ambato en la biblioteca de la Facultad de Ingeniería Civil y Mecánica.
6. ¿Cómo?	Mediante Investigación Bibliográfica.

3.5. PLAN DE PROCESAMIENTO Y ANÁLISIS

- Para el cumplimiento se seguirá un plan estructurado de la siguiente manera:
 - 1) Revisión de la bibliografía necesaria para entender el presente trabajo.
 - 2) Identificación del tipo de lenguaje de programación.

- 3) Desarrollo del software.
- 4) Cálculo de los ejercicios planteados de forma manual.
- 5) Cálculo de los ejercicios planteados mediante el software existente.
- 6) Cálculo de los ejercicios planteados mediante el software desarrollado.
- 7) Análisis y comprobación de los resultados obtenidos.
- 8) Verificación de Hipótesis.

CAPÍTULO IV

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1. NIVEL O TIPO DE INVESTIGACIÓN

A continuación, se realizar ejercicios por tres métodos: de forma manual, con un software existente y con el software desarrollado para posteriormente realizar un análisis con los resultados obtenidos.

Para la aplicación de ejercicios de manera manual se utilizarán las definiciones y ecuaciones que se trataron en el capítulo II.

Para desarrollar los ejercicios mediante un software existente se utilizará **SN CANALES V2.0L**.

4.1.1. EJEMPLO PARA EL CÁLCULO DE LOS ELEMENTOS GEOMÉTRICOS DE UN CANAL RECTANGULAR CON ESQUINAS REDONDEADAS.

Calcular los elementos geométricos para un canal rectangular con esquinas redondeadas, considerando un ancho de solera $b=1m$, con un tirante de $y=2m$ y un radio $r=0.5m$.

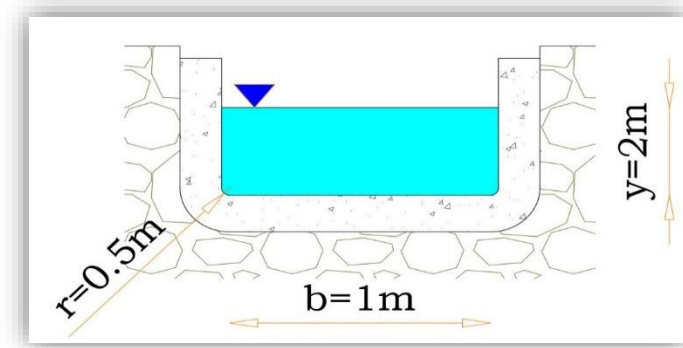
Datos:

$$b = 1m$$

$$y = 2m$$

$$r = 0.5m$$

Figura 8: Gráfica para el cálculo de los elementos geométricos de una sección rectangular con esquinas redondeadas



Fuente: Alex Tipán

4.1.1.1. MÉTODO MANUAL

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

$$A = \left(\frac{\pi}{2} - 2\right) * r^2 + (b + 2 * r) * y$$

$$A = \left(\frac{\pi}{2} - 2\right) * 0.5^2 + (1 + 2 * 0.5) * 2$$

$$A = 3.8927m^2$$

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

$$Pm = (\pi - 2) * r + b + 2 * y$$

$$Pm = (\pi - 2) * 0.5 + 1 + 2 * 2$$

$$Pm = 5.5708m$$

Cálculo del radio hidráulico para una sección rectangular con esquinas redondeadas mediante la siguiente ecuación:

$$Rh = \frac{A}{Pm}$$

$$Rh = \frac{3.8927m^2}{5.5708m}$$

$$Rh = 0.6988m$$

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

$$T = b + 2 * r$$

$$T = 1 + 2 * 0.5$$

$$T = 2m$$

Cálculo de la profundidad hidráulica para una sección rectangular con esquinas redondeadas mediante la siguiente ecuación:

$$D = \frac{\left(\frac{\pi}{2} - 2\right) * r^2}{b + 2 * r} + y$$

$$D = \frac{\left(\frac{\pi}{2} - 2\right) * 0.5^2}{1 + 2 * 0.5} + 2$$

$$D = 1.9463m$$

4.1.1.2. MÉTODO CON UN SOFTWARE EXISTENTE

Cálculo de los elementos geométricos de una sección rectangular con esquinas redondeadas mediante un software existente

The software interface, titled "ELEMENTOS GEOMETRICOS DE UNA SECCION RECTANGULAR REDONDEADO", displays the following data:

Datos de entrada:		
Tirante (y)=	2	m
Ancho de la solera (b)=	1	m
radio (r)=	.5	m

Buttons: CALCULAR (green), LIMPIAR (pink)

Datos de salida:		
Área mojada (A)=	3.8927	m ²
Perímetro mojado (P)=	5.5708	m
Radio hidráulico (R)=	0.6988	m
Ancho superficial (T)=	2	m
Profundidad hidráulica (D)=	1.9463	m
Factor de sección (Z)=	5.4308	m

The diagram on the right shows a cross-section of a channel with rounded corners. The water level is indicated by a blue shaded area. Dimensions are labeled: y (water depth), T (surface width), b (bottom width), and r (corner radius).

4.1.1.3. MÉTODO CON EL SOFTWARE DESARROLLADO.

Determinación de la solución del elemento geométrico rectangular con esquinas redondeadas mediante el software desarrollado.

FICM CANALES V 1.0



Tirante (y) =
2 m

Ancho de la solera (b) =
1 m

Radio (r) =
0.5 m

CALCULAR LIMPIAR

Área Mojada (A): 3,8927m²
Perímetro Mojado (Pm): 5,5708m
Radio Hidráulico (Rh): 0,6988m
Ancho Superficial (T): 2,0000m
Profundidad Hidráulica(D): 1,9463m

4.1.2. EJEMPLO PARA EL CÁLCULO DEL CAUDAL SECCIÓN TRIANGULAR CON FONDO REDONDEADO.

Determinar el caudal de agua en una tubería de alcantarillado con un coeficiente de rugosidad de $n=0.012$, con un tirante de $y=2m$, talud $z=0.75$, la pendiente del canal es del 0.12% , radio= 0.5 . Determine la velocidad de flujo, los elementos geométricos, el tipo de flujo y la energía específica.

Datos:

$$y = 2m$$

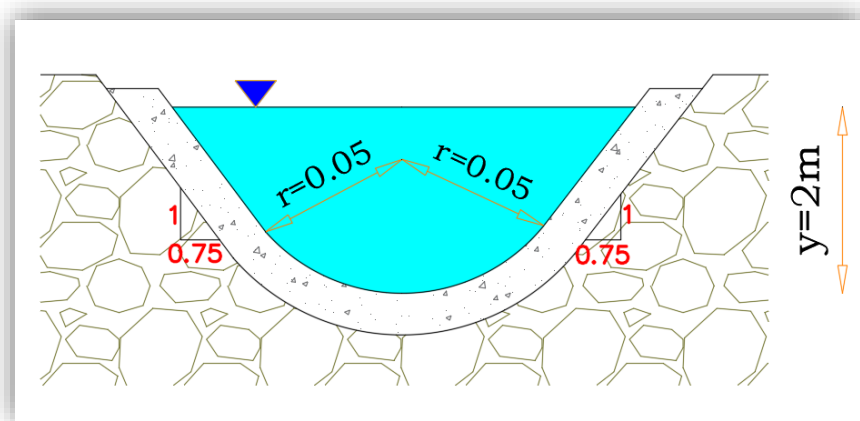
$$n = 0.012$$

$$r = 0.5m$$

$$z = 0.75m$$

$$S = 0.12\% \approx 0.0012$$

Figura 9: Gráfica para el cálculo del caudal de una sección triangular con fondo redondeado



Fuente: Alex Tipán

4.1.2.1. MÉTODO MANUAL

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

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

$$T = 2 * (0.75 * (2 - 0.5) + 0.5 * \sqrt{1 + 0.75^2})$$

$$T = 3.50m$$

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

$$A = \frac{T^2}{4 * z} - \frac{r^2}{z} * \left(1 - \frac{\alpha}{\tan \alpha}\right)$$

$$A = \frac{3.50^2}{4 * 0.75} - \frac{0.5^2}{0.75} * (1 - \cot^{-1}0.75)$$

$$A = 3.9818m^2$$

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

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

$$Pm = \frac{3.50}{0.75} * \sqrt{1 + 0.75^2} - \frac{2 * 0.5}{0.75} * (1 - \cot^{-1}0.75)$$

$$Pm = 5.4273m$$

Cálculo del caudal mediante la siguiente ecuación:

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

$$Q = \frac{1}{0.012} * 3.9818 * 0.7337^{2/3} * 0.0012^{1/2}$$

$$Q = 9.3503 m^3/s$$

Cálculo de la profundidad hidráulica una sección triangular con fondo redondeado mediante la siguiente ecuación:

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

$$D = \frac{3.918 m^2}{3.5 m}$$

$$D = 1.1377 m$$

Cálculo de la velocidad normal mediante la siguiente ecuación:

$$v = \frac{1}{n} * Rh^{\frac{2}{3}} * S^{1/2}$$

$$v = \frac{1}{0.012} * 0.7337^{\frac{2}{3}} * 0.0012^{1/2}$$

$$v = 2.3482 m/s$$

Cálculo del número de Froude mediante la siguiente ecuación:

$$NF = \frac{v}{\sqrt{g * D}}$$

$$NF = \frac{2.3482}{\sqrt{9.81 * 1.1377}}$$

$$NF = 0.7029 \therefore \text{Subcrítico}$$

Cálculo de la energía específica mediante la siguiente ecuación:

$$E = y + \frac{Q^2}{2 * g * A^2}$$

$$E = 2 + \frac{9.3503^2}{2 * 9.81 * 3.9818^2}$$

$$E = 2.2811 \text{ m} - \text{Kg/Kg}$$

4.1.2.2. MÉTODO CON UN SOFTWARE EXISTENTE

Cálculo del caudal de una sección triangular con fondo redondeado mediante un software existente.

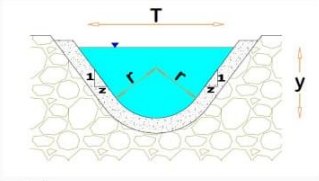
The screenshot shows a software window titled "CAUDAL_DE_UNA_SECCION_TRIANGULAR_REDONDEADO". It is divided into several sections:

- Datos de entrada:**
 - Tirante (y)= 2 m
 - Radio (r)= 0.5 m
 - Talud (z)= 0.75
 - Pendiente (S)= 0.0012 m/m
 - Coefficiente de Manning (n)= 0.012
- Buttons:** "CALCULAR" (green) and "LIMPIAR" (pink).
- Datos de salida:**
 - Caudal (Q)= 9.3503 m³/s
 - Área mojada (A)= 3.9818 m²
 - Perímetro mojado (P)= 5.4273 m
 - Radio hidráulico (R)= 0.7337 m
 - Ancho superficial (T)= 3.5 m
 - Profundidad hidráulica (D)= 1.1377 m
 - Velocidad (V)= 2.3482 m/s
 - Número de Froude (NF)= 0.7029
 - Tipo de flujo= Subcrítico
 - Energía específica (E)= 2.2811 m-Kg/Kg
- Diagrama:** A cross-section of a rounded triangular channel with a blue water surface. Labels include T (top width), Z (side slope), and r (bottom radius).
- Message:** "La velocidad es la adecuada ya que se encuentra en el rango de m/s" (The velocity is adequate as it is within the range of m/s).
- Sistema de unidades:**
 - Sistema métrico
 - Sistema inglés
- MENÚ F** (partially visible)

4.1.2.3. MÉTODO CON EL SOFTWARE DESARROLLADO.

Cálculo del caudal de una sección triangular con fondo redondeado mediante el software desarrollado.

FICM CANALES V 1.0



Tirante (y) = m

Radio (r) = m

Talud (z) = m

Pendiente (S) = m/m

Coefficiente de Manning (n) =

Caudal (Q): 9,3503m³/s
Área Mojada (A): 3,9818m²

FICM CANALES V 1.0

Tirante (y) = m

Pendiente (S) = m/m

Coefficiente de Manning (n) =

Caudal (Q): 9,3503m³/s
Área Mojada (A): 3,9818m²
Perímetro Mojado (Pm): 5,4273m
Radio Hidráulico (Rh): 0,7337m
Ancho Superficial (T): 3,5000m
Profundidad Hidráulica(D): 1,1377m
Velocidad (V): 2,3482m/s
Número de Froude (NF): 0,7030
Tipo de Flujo: Subcrítico
Energía Específica (E): 2,2811m-Kg/Kg

La velocidad es la adecuada porque se encuentra en el rango de 0.6 - 4.5 m/s

4.1.3. EJEMPLO PARA EL CÁLCULO DE LA SECCIÓN HIDRAÚLICA ÓPTIMA (SECCIÓN SEMICIRCULAR)

Un canal abierto de hormigón está diseñado para transportar un caudal de $3m^3/s$, con una pendiente de $S=0.00085$. Determinar las dimensiones de la sección recta de máximo rendimiento.

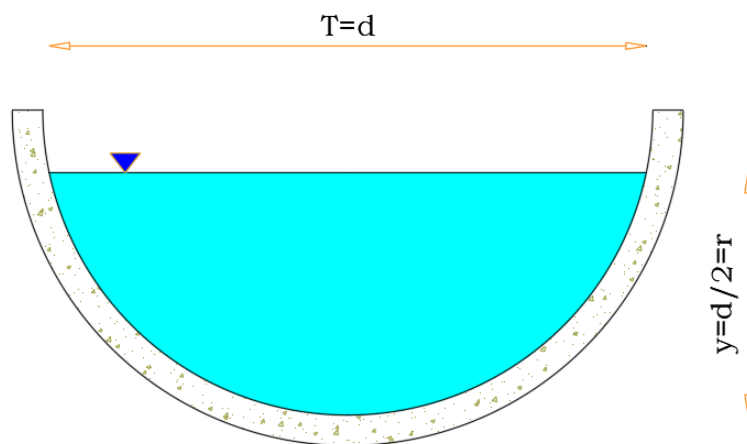
Datos:

$$Q = 3m^3/s$$

$$n = 0.013$$

$$S = 0.00085$$

Figura 10: Gráfica para el cálculo de una sección semicircular



Fuente: Alex Tipán

4.1.3.1. MÉTODO MANUAL

Para una sección semicircular debe cumplir las condiciones de:

$$y = \frac{d}{2} = r$$

Cálculo del diámetro mediante la siguiente ecuación:

$$Q = \frac{1}{n} * A * Rh^{2/3} * S^{1/2}$$
$$\frac{Q}{S^{1/2}} = \frac{\pi * d^2}{8} * \frac{d^{2/3}}{4^{2/3}}$$
$$\frac{Q * n}{S^{1/2}} * \frac{8 * 4^{2/3}}{\pi} = d^{8/3}$$
$$\left(\frac{Q * n}{S^{1/2}} * \frac{8 * 4^{2/3}}{\pi}\right)^{3/8} = (d^{8/3})^{3/8}$$
$$\left(\frac{Q * n}{S^{1/2}} * \frac{8 * 4^{2/3}}{\pi}\right)^{3/8} = d$$
$$d = 2.2394m$$

Cálculo del área hidráulica una sección semicircular mediante la siguiente ecuación:

$$A = \frac{\pi * d^2}{8}$$
$$A = \frac{\pi * 2.2394^2}{8}$$
$$A = 1.9694m^2$$

Cálculo del perímetro mojado mediante la siguiente ecuación:

$$Pm = \frac{\pi * d}{2}$$
$$Pm = \frac{\pi * 2.2394}{2}$$
$$Pm = 3.5176m$$

Cálculo del radio hidráulico mediante la siguiente ecuación:

$$Rh = \frac{A}{Pm}$$

$$Rh = \frac{1.9694m^2}{3.5176m}$$

$$Rh = 0.5599m$$

Cálculo del tirante óptimo mediante la siguiente ecuación:

$$y_o = \frac{d}{2}$$

$$y_o = \frac{2.2394m}{2}$$

$$y_o = 1.1197m$$

Cálculo de la profundidad hidráulica mediante la siguiente ecuación:

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

$$D = \frac{1.9694m^2}{2.2394m}$$

$$D = 0.8794m$$

Cálculo de la velocidad mediante la siguiente ecuación:

$$v = \frac{Q}{A}$$

$$v = \frac{3m^3/s}{1.9694m^2}$$

$$v = 1.5233m/s$$

Cálculo del número de Froude mediante la siguiente ecuación:

$$NF = \frac{v}{\sqrt{g * D}}$$

$$NF = \frac{1.5233m/s}{\sqrt{9.81 * 0.8794}}$$

$$NF = 0.5186 \therefore \text{Subcrítico}$$

Cálculo de la energía específica mediante la siguiente ecuación:

$$E = y + \frac{Q^2}{2 * g * A^2}$$

$$E = 1.1197 + \frac{3^2}{2 * 9.81 * 1.9694^2}$$

$$E = 1.2379m - Kg/Kg$$

4.1.3.2. MÉTODO CON UN SOFTWARE EXISTENTE

Cálculo del caudal de una sección semicircular mediante un software existente

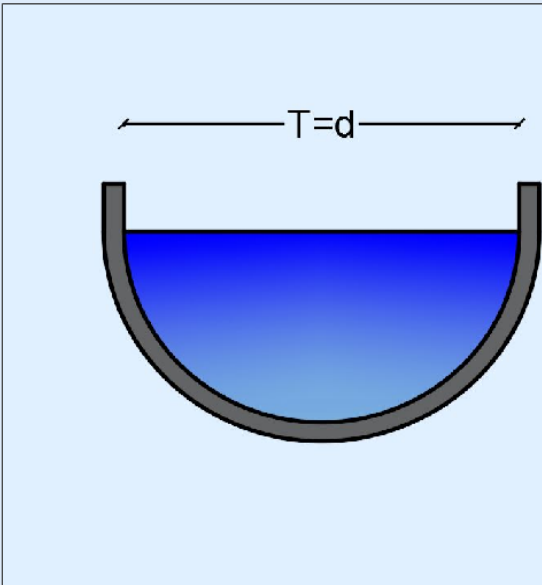
SECCION_HIDRAULICA_OPTIMA_DE_UNA_SECCION_SEMICIRCULAR

Datos de entrada:

Caudal (Q)=	<input type="text" value="3"/>	m³/s
Pendiente (S)=	<input type="text" value="0.00085"/>	m/m
Coficiente de Manning (n)=	<input type="text" value="0.013"/>	

Datos de salida:

Diámetro (d)=	<input type="text" value="2.2394"/>	m
Tirante (y)=	<input type="text" value="1.1197"/>	m
Área mojada (A)=	<input type="text" value="1.9693"/>	m²
Perímetro mojado (P)=	<input type="text" value="3.5176"/>	m
Radio hidráulico (R)=	<input type="text" value="0.5598"/>	m
Ancho superficial (T)=	<input type="text" value="2.2394"/>	m
Profundidad hidráulica (D)=	<input type="text" value="0.8794"/>	m
Velocidad (V)=	<input type="text" value="1.5234"/>	m/s
Número de Froude (NF)=	<input type="text" value="0.5187"/>	
Tipo de flujo=	<input type="text" value="Subcrítico"/>	
Energía específica (E)=	<input type="text" value="1.238"/>	m-Kg/Kg



La velocidad es la adecuada ya que se encuentra en el rango de m/s

Sistema de unidades:

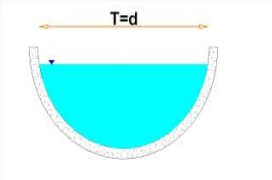
Sistema métrico
 Sistema inglés

MENÚ

4.1.3.3. MÉTODO CON EL SOFTWARE DESARROLLADO.

Cálculo del caudal de una sección triangular con fondo redondeado mediante el software desarrollado.

FICM CANALES V 1.0



Caudal (Q) =
3 m³/s

Pendiente (S) =
0.00085 m/m

Coefficiente de Manning (n) =
0.013

CALCULAR
LIMPIAR

Tirante Óptimo (yo): 1,1197m
Diámetro (d): 2,2394m
Área Mojada (A): 1,9693m²
Perímetro Mojado (Pm): 3,5176m
Radio Hidráulico (Rh): 0,5598m
Ancho Superficial (T): 2,2394m
Profundidad Hidráulica (D): 0,8794m
Velocidad (V): 1,5234m/s

FICM CANALES V 1.0

3 m³/s

Pendiente (S) =
0.00085 m/m

Coefficiente de Manning (n) =
0.013

CALCULAR
LIMPIAR

Tirante Óptimo (yo): 1,1197m
Diámetro (d): 2,2394m
Área Mojada (A): 1,9693m²
Perímetro Mojado (Pm): 3,5176m
Radio Hidráulico (Rh): 0,5598m
Ancho Superficial (T): 2,2394m
Profundidad Hidráulica (D): 0,8794m
Velocidad (V): 1,5234m/s
Número de Froude (NF): 0,5190
Tipo de Flujo: Subcrítico
Energía Específica (E): 1,2380m-Kg/Kg

La velocidad es la adecuada porque se encuentra en el rango de 0.6 - 4.5 m/s

4.1.4. 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 $3\text{m}^3/\text{s}$, el coeficiente de rugosidad es $n=0.012$, el ancho de la solera es $b=1\text{m}$, el talud es $z=0.75$, la pendiente del canal es del 0.12% . Determine también la velocidad normal, los elementos geométricos, el tipo de flujo y la energía específica.

Datos:

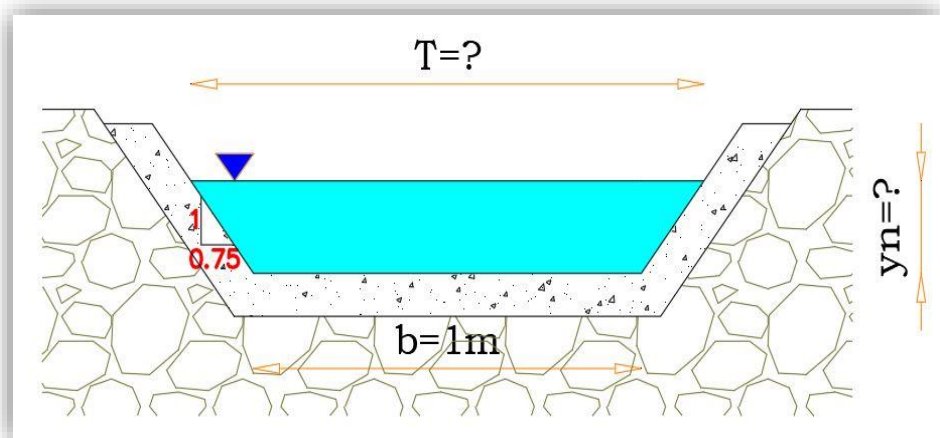
$$Q = 3\text{m}^3/\text{s}$$

$$n = 0.012$$

$$b = 1\text{m}$$

$$S = 0.12\% \approx 0.0012$$

Figura 11: Gráfica para el cálculo del tirante normal de una sección trapezoidal



Fuente: Alex Tipán

4.1.4.1. MÉTODO MANUAL

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

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

Con la cual se debe seguir los siguientes pasos:

- 1) Calcular la relación $\frac{n*Q}{S^{1/2}}$
- 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 $A * Rh^{2/3}$
- 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.

$$\frac{n * Q}{S^{1/2}} = \frac{0.012 * 3}{0.0012^{1/2}} = 1.0392$$

Iteración 1

Para y = 1m

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

$$A = (b + z * y) * y$$

$$A = (1 + (0.75 * 1)) * 1$$

$$A = 1.75m^2$$

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

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

$$P = 1 + 2 * 1\sqrt{1 + 0.75^2}$$

$$P = 3.5m$$

Cálculo del radio hidráulico mediante la siguiente ecuación:

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

$$Rh = \frac{1.75m^2}{3.5m}$$

$$Rh = 0.5m$$

Verificación si con los valores obtenidos se cumple la igualdad.

$$Rh^2 * A = 1.039$$

$$0.5^2 * 1.75 = 1.039$$

$$1.102 \neq 1.039 \therefore NO CUMPLE$$

Iteración 2

$$\text{Para } y = 0.9m$$

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

$$A = (b + z * y) * y$$

$$A = (1 + (0.75 * 0.9)) * 0.9$$

$$A = 1.508m^2$$

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

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

$$P = 1 + 2 * 0.9 \sqrt{1 + 0.75^2}$$

$$P = 3.25m$$

Cálculo del radio hidráulico mediante la siguiente ecuación:

$$Rh \frac{A}{P}$$

$$Rh = \frac{1.508m^2}{3.25m}$$

$$Rh = 0.464m$$

Verificación si con los valores obtenidos se cumple la igualdad.

$$Rh^{\frac{2}{3}} * A = 1.039$$

$$0.464^{\frac{2}{3}} * 1.508 = 1.039$$

$$0.904 \neq 1.039 \therefore \text{NO CUMPLE}$$

Iteración 3

$$\text{Para } y = 0.9694m$$

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

$$A = (b + z * y) * y$$

$$A = (1 + (0.75 * 0.9694)) * 0.9694$$

$$A = 1.6742m^2$$

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

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

$$P = 1 + 2 * 0.9694 * \sqrt{1 + 0.75^2}$$

$$P = 3.4235m$$

Cálculo del radio hidráulico mediante la siguiente ecuación:

$$Rh = \frac{A}{P}$$
$$Rh = \frac{1.6742m^2}{3.4235m}$$
$$Rh = 0.489m$$

Verificación si con los valores obtenidos se cumple la igualdad.

$$Rh^{\frac{2}{3}} * A = 1.039$$

$$0.489^{\frac{2}{3}} * 1.6742 = 1.039$$

$$1.039 = 1.039 \therefore SI \ CUMPLE$$

Como la condición cumple, el tirante normal es $y_n = 0.969m$

Cálculo de la velocidad normal mediante la siguiente ecuación:

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

$$V = \frac{1}{0.012} * 0.489^{\frac{2}{3}} * 0.0012^{1/2}$$

$$V = 1.7919 \ m/s$$

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

$$T = b + 2 * z * y$$

$$T = 1 + 2 * 0.75 * 0.9694$$

$$T = 2.4541m$$

Cálculo de la profundidad hidráulica mediante la siguiente ecuación:

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

$$D = \frac{1.6742}{2.4541}$$

$$D = 0.6822m$$

Cálculo del número de Froude mediante la siguiente ecuación:

$$NF = \frac{V}{\sqrt{g * D}}$$

$$NF = \frac{1.7919}{\sqrt{9.81 * 0.6822}}$$

$$NF = 0.6927 \therefore \text{Subcrítico}$$

Cálculo de la energía específica mediante la siguiente ecuación:

$$E = y + \frac{Q^2}{2 * g * A^2}$$

$$E = 0.9694 + \frac{3^2}{2 * 9.81 * 1.6742^2}$$

$$E = 1.1331 m - Kg/Kg$$

4.1.4.2. MÉTODO CON EL SOFTWARE SN CANALES V2.0L

Determinación de la solución del tirante normal de una sección trapezoidal mediante el programa SN CANALES V2.0L.

The screenshot displays the 'TIRANTE_NORMAL_DE_UNA_SECCION_TRAPEZOIDAL' software window. It is divided into several sections:

- Datos de entrada:** A table of input parameters:

Caudal (Q)=	3	m³/s
Ancho de la solera (b)=	1	m
Talud (z)=	0.75	
Pendiente (S)=	0.0012	m/m
Coficiente de Manning (n)=	0.012	
- Buttons:** 'CALCULAR' (green) and 'LIMPIAR' (orange).
- Datos de salida:** A table of output results:

Tirante normal (y)=	0.9694	m
Área mojada (A)=	1.6742	m²
Perímetro mojado (P)=	3.4235	m
Radio hidráulico (R)=	0.489	m
Ancho superficial (T)=	2.4541	m
Profundidad hidráulica (D)=	0.6822	m
Velocidad (V)=	1.7919	m/s
Número de Froude (NF)=	0.6926	
Tipo de flujo=	Subcrítico	
Energía específica (E)=	1.1331	m-Kg/Kg
- Diagrama:** A trapezoidal channel cross-section with a blue water surface. Labels include 'T' for top width, 'b' for bottom width, 'z' for the slope, and '1' for the vertical height of the slope triangle.
- Nota:** A red text note states: 'La velocidad es la adecuada ya que se encuentra en el rango de m/s'.
- Sistema de unidades:** Radio buttons for 'Sistema métrico' (selected) and 'Sistema inglés'.
- Botón:** 'MENÚ PRINCIPAL' (dark blue).

4.1.4.3. MÉTODO CON EL SOFTWARE HYDROEOLM_CANALES

Determinación de la solución del tirante normal de una sección trapezoidal mediante el programa HYDROEOLM_CANALES.



Canal Trapezoidal Simétrico
Ingeniería Hidráulica

Diagrama de un canal trapezoidal con: Top width (T), Bottom width (b), Side slope (Z:1), and Normal depth (YN).

Caudal (m ³ /s):	3.000000	Base Canal (m):	1.0000
Rugosidad (n):	0.012000	Pendiente (m/m):	0.00120000
Talud (Z):	0.7500	Área Hidráulica (m ²):	1.6742
Profundidad Normal (m):	0.9694	Perímetro (m):	3.4235
Espejo (m):	2.4541	Velocidad (m/s):	1.7919
Radio Hidráulico (m):	0.4890	Número de Froude:	0.6926
Energía Específica (Nm/N):	1.1331		
Tipo de Flujo:	Subcrítico		

HydroEoIm ©

Buttons:  Inicio

4.1.4.4. MÉTODO CON EL SOFTWARE DESARROLLADO.

Determinación de la solución del tirante normal de una sección trapezoidal mediante el software desarrollado.

FICM CANALES V 1.0

Caudal (Q) = 3 m³/s

Ancho de la solera (b) = 1 m

Talud (z) = 0.75 m

Pendiente (S) = 0.0012 m/m

Coefficiente de Manning (n) = 0.012

Tirante Normal (yn): 0,9694m
Área Mojada (A): 1,6742m²

FICM CANALES V 1.0

talud (z) = 0.75 m

Pendiente (S) = 0.0012 m/m

Coefficiente de Manning (n) = 0.012

Tirante Normal (yn): 0,9694m
Área Mojada (A): 1,6742m²
Perímetro Mojado (Pm): 3,4235m
Radio Hidráulico (Rh): 0,4890m
Ancho Superficial (T): 2,4541m
Profundidad Hidráulica (D): 0,6822m
Velocidad (V): 1,7919m/s
Número de Froude (NF): 0,6926
Tipo de Flujo: Subcritico
Energía Específica (E): 1,1331m·kg/kg

La velocidad es la adecuada porque se encuentra en el rango de 0.6 - 4.5 m/s

4.1.5. EJEMPLO PARA EL CALCULO DEL TIRANTE CRÍTICO DE UNA SECCIÓN TRIANGULAR.

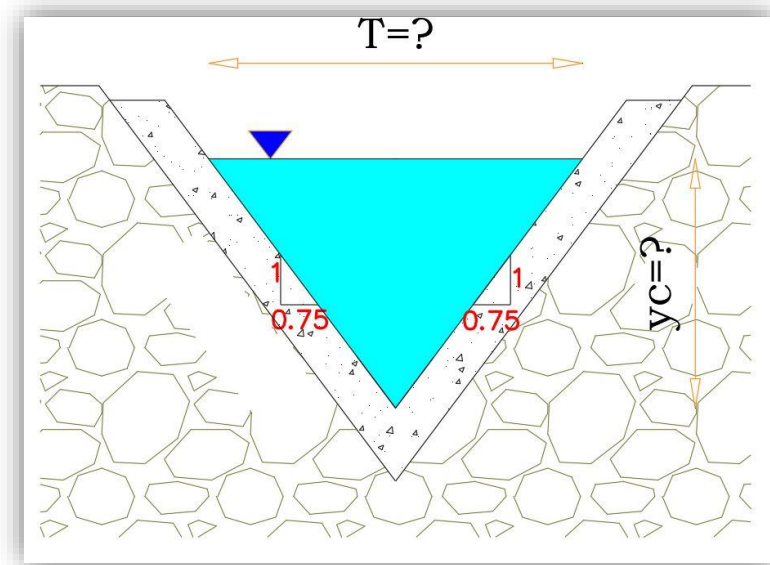
Calcular el tirante crítico, para un canal rectangular, considerando que su caudal de diseño es de $3m^3/s$, el talud es 1:0.75. Determine también la velocidad crítica, los elementos geométricos, el tipo de flujo y la energía específica.

Datos:

$$Q = 3m^3/s$$

$$z = 0.75$$

Figura 12: Gráfica para el cálculo del tirante crítico de una sección triangular



Fuente: Alex Tipán

4.1.5.1. MÉTODO MANUAL

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

$$\left(\frac{Q^2}{g}\right) = \frac{A^3}{T}$$

Con la cual se debe seguir los siguientes pasos:

- 1) Calcular la relación $\left(\frac{Q^2}{g}\right)$
- 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 la relación $\frac{A^3}{T}$
- 5) Calcular $A * Rh^{2/3}$
- 6) 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.

$$\left(\frac{Q^2}{g}\right) = \frac{3^2}{9.81} = \mathbf{0.9174}$$

Iteración 1

Para y = 1.20m

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

$$A = (z * y^2)$$

$$A = (0.75 * 1.20^2)$$

$$A = 1.08m^2$$

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

$$T = 2 * z * y$$

$$T = 2 * 0.75 * 1.20$$

$$T = 1.80m$$

Verificación si los valores obtenidos se cumple la igualdad.

$$\frac{A^3}{T} = 0.9174$$

$$\frac{1.08^3}{1.80} = 0.9174$$

$$0.699 \neq 0.9174 \therefore NO CUMPLE$$

Iteración 2

Para y = 1.2668m

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

$$A = (z * y^2)$$

$$A = (0.75 * 1.2668^2)$$

$$A = 1.2036m^2$$

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

$$T = 2 * z * y$$

$$T = 2 * 0.75 * 1.2668$$

$$T = 1.9002m$$

Verificación si los valores obtenidos se cumple la igualdad.

$$0.9174 = \frac{A^3}{T}$$

$$0.9174 = \frac{1.2036^3}{1.9002}$$

$$0.9174 = 0.9174 \therefore \text{CUMPLE}$$

Como la condición cumple, el tirante crítico es $y_c = 1.2668m$

Cálculo de la velocidad crítica mediante la siguiente ecuación:

$$V = \frac{Q}{A}$$

$$V = \frac{3}{1.2036}$$

$$V = 2.4925 \text{ m/s}$$

Cálculo de la profundidad hidráulica mediante la siguiente ecuación:

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

$$D = \frac{1.2036}{1.9001}$$

$$D = 0.6334m$$

Cálculo del número de Froude mediante la siguiente ecuación:

$$NF = \frac{V}{\sqrt{g * D}}$$

$$NF = \frac{2.4927}{\sqrt{9.81 * 0.6334}}$$

$$NF = 1 \therefore \text{Crítico}$$

Cálculo de la energía específica mediante la siguiente ecuación:

$$E = y + \frac{Q^2}{2 * g * A^2}$$

$$E = 1.2668 + \frac{3^2}{2 * 9.81 * 1.2036^2}$$

$$E = 0.9174 + \frac{3^2}{2} * 9.81 * 1.2036^2$$

$$E = 1.5834 \text{ m} - \text{Kg/Kg}$$

4.1.5.2. MÉTODO CON EL SOFTWARE SN CANALES V 2.0L

Determinación de la solución del tirante crítico de una sección triangular mediante el programa SN CANALES V2.0L.

The screenshot shows a software interface for calculating the critical depth of a triangular channel. The window title is "TIRANTE_CRITICO_DE_UNA_SECCION_TRIANGULAR".

Datos de entrada:

- Caudal (Q)= 3 m³/s
- Talud (z)= 0.75

Buttons: CALCULAR, LIMPIAR

Datos de salida:

- Tirante crítico (y)= 1.2668 m
- Área mojada (A)= 1.2035 m²
- Perímetro mojado (P)= 3.1669 m
- Radio hidráulico (R)= 0.38 m
- Ancho superficial (T)= 1.9001 m
- Profundidad hidráulica (D)= 0.6334 m
- Velocidad (V)= 2.4927 m/s
- Número de Froude (NF)= 1
- Tipo de flujo= Crítico
- Energía específica (E)= 1.5835 m-Kg/Kg

The diagram on the right shows a triangular channel cross-section with a water surface width T, side slopes of 1 vertical to z horizontal, and a water depth y. A red message states: "La velocidad es la adecuada ya que se encuentra en el rango de m/s".

System of units: Sistema métrico, Sistema inglés

Buttons: MENÚ F

4.1.5.3. MÉTODO CON EL SOFTWARE HYDROEOLM_CANALES

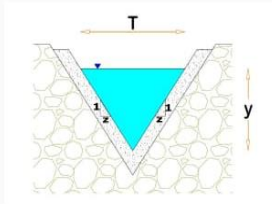
Determinación de la solución del tirante crítico de una sección triangular mediante el programa HYDROEOLM_CANALES.



4.1.5.4. MÉTODO CON EL SOFTWARE DESARROLLADO.

Determinación de la solución del tirante crítico de una sección triangular mediante el software desarrollado.

FICM CANALES V 1.0

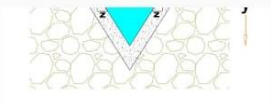


Caudal (Q) =
 m³/s

Talud (z) =
 m

Tirante Crítico (yc): 1,2668m
Área Mojada (A): 1,2035m²
Perímetro Mojado (Pm): 3,1669m
Radio Hidráulico (Rh): 0,3800m
Ancho Superficial (T): 1,9001m
Profundidad Hidráulica (D): 0,6334m
Velocidad (V): 2,4927m/s
Número de Froude (NF): 1,0000
Tipo de Flujo: Subcritico
Energía Específica (E): 1,5835m·kg/kg

FICM CANALES V 1.0



Caudal (Q) =
 m³/s

Talud (z) =
 m

Tirante Crítico (yc): 1,2668m
Área Mojada (A): 1,2035m²
Perímetro Mojado (Pm): 3,1669m
Radio Hidráulico (Rh): 0,3800m
Ancho Superficial (T): 1,9001m
Profundidad Hidráulica (D): 0,6334m
Velocidad (V): 2,4927m/s
Número de Froude (NF): 1,0000
Tipo de Flujo: Subcritico
Energía Específica (E): 1,5835m·kg/kg

La velocidad es la adecuada porque se encuentra en el rango de 0.6 - 4.5 m/s

4.2. ANALISIS DE RESULTADOS

Tabla 6: Comparación de resultados en el cálculo de los elementos geométricos de una canal rectangular con esquinas redondeadas

DESCRIPCIÓN	MÉTODO			UNIDADES	DIFERENCIA CON SOFTWARE DESARROLLADO	
	%					
	MANUAL	SN CANALES V2.0L	SOFTWARE DESARROLLADO		MANUAL	SN CANALES V2.0L
Área Mojada (A)	3.8927	3.8927	3.8927	m ²	0.00%	0.00%
Perímetro Mojado (Pm)	5.5708	5.5708	5.5708	m	0.00%	0.00%
Radio Hidráulico (Rh)	0.6988	0.6988	0.6988	m	0.00%	0.00%
Ancho Superficial (T)	2	2	2	m	0.00%	0.00%
Profundidad Hidráulica (D)	1.9463	1.9463	1.9463	m	0.00%	0.00%

Realizado por: Alex Tipán

Tabla 7: Comparación de resultados en el cálculo del caudal de una de una sección triangular con fondo redondeado

DESCRIPCIÓN	MÉTODO			UNIDADES	DIFERENCIA CON SOFTWARE DESARROLLADO	
	%					
	MANUAL	SN CANALES V2.0L	SOFTWARE DESARROLLADO		MANUAL	SN CANALES V2.0L
Caudal (Q)	9.3503	9.3503	9.3503	m ³	0.00%	0.00%
Área Mojada (A)	3.9818	3.9818	3.9818	m ²	0.00%	0.00%
Perímetro Mojado (Pm)	5.4273	5.4273	5.4273	m	0.00%	0.00%
Radio Hidráulico (Rh)	0.7337	0.7337	0.7337	m	0.00%	0.00%
Ancho Superficial (T)	3.5	3.5	3.5	m	0.00%	0.00%
Profundidad Hidráulica (D)	1.1377	1.1377	1.1377	m	0.00%	0.00%
Velocidad (V)	2.3482	2.3482	2.3482	m/s	0.00%	0.00%
Número de Froude (Fr)	0.7029	0.703	0.703	-	0.01%	0.00%
Tipo de Flujo	Subcrítico	Subcrítico	Subcrítico	-	0.00%	0.00%
Energía Específica (E)	2.2811	2.2811	2.2811	m-Kg/Kg	0.00%	0.00%

Realizado por: Alex Tipán

Tabla 8: Comparación de resultados en una sección hidráulica óptima (Sección Semicircular)

DESCRIPCIÓN	MÉTODO			UNIDADES	DIFERENCIA CON SOFTWARE DESARROLLADO	
	%					
	MANUAL	SN CANALES V2.0L	SOFTWARE DESARROLLADO		MANUAL	SN CANALES V2.0L
Tirante Óptimo (yo)	1.1197	1.1197	1.1197	m	0.00%	0.00%
Diámetro (d)	2.2394	2.2394	2.2394	m	0.00%	0.00%
Área Mojada (A)	1.9694	1.9693	1.9693	m ²	0.01%	0.00%
Perímetro Mojado (Pm)	3.5176	3.5176	3.5176	m	0.00%	0.00%
Radio Hidráulico (Rh)	0.5599	0.5598	0.5598	m	0.01%	0.00%
Ancho Superficial (T)	2.2394	2.2394	2.2394	m	0.00%	0.00%
Profundidad Hidráulica (D)	0.8794	0.8794	0.8794	m	0.00%	0.00%
Velocidad (V)	1.5234	1.5234	1.5234	m/s	0.01%	0.00%
Número de Froude (Fr)	0.5186	0.5187	0.519	-	0.02%	0.00%
Tipo de Flujo	Subcrítico	Subcrítico	Subcrítico	-	0.00%	0.00%
Energía Específica (E)	1.2379	1.238	1.238	m-Kg/Kg	0.01%	0.00%

Realizado por: Alex Tipán

Tabla 9: Comparación de resultados en el cálculo del tirante normal de una sección trapezoidal

DESCRIPCIÓN	MÉTODO				UNIDADES	DIFERENCIA CON SOFTWARE DESARROLLADO		
	%							
	MANUAL	SN CANALES V2.0L	HYDROEOLM	SOFTWARE DESARROLLADO		MANUAL	SN CANALES V2.0L	HYDROEOLM
Tirante Normal (yn)	0.9694	0.9694	0.9694	0.9694	m	0%	0%	0%
Área Mojada (A)	1.6742	1.6742	1.6742	1.6742	m ²	0%	0%	0%
Perímetro Mojado (Pm)	3.4235	3.4235	3.4235	3.4235	m	0%	0%	0%
Radio Hidráulico (Rh)	0.489	0.489	0.489	0.489	m	0%	0%	0%
Ancho Superficial (T)	2.4541	2.4541	2.4541	2.4541	m	0%	0%	0%
Profundidad Hidráulica (D)	0.6822	0.6822	0.6822	0.6822	m	0%	0%	0%
Velocidad (V)	1.7919	1.7919	1.7919	1.7919	m/s	0%	0%	0%
Número de Froude (Fr)	0.6927	0.693	0.6926	0.693	-	0.03%	0%	0.04%
Tipo de Flujo	Subcrítico	Subcrítico	Subcrítico	Subcrítico	-	0%	0%	0%
Energía Específica (E)	1.1331	1.1331	1.1331	1.1331	m-Kg/Kg	0%	0%	0%

Realizado por: Alex Tipán

Tabla 10: Comparación de resultados en el cálculo del tirante crítico de una sección triangular

DESCRIPCIÓN	MÉTODO				UNIDADES	DIFERENCIA CON SOFTWARE DESARROLLADO		
	MANUAL	SN CANALES V2.0L	HYDROEOLM	SOFTWARE DESARROLLADO		%		
						MANUAL	SN CANALES V2.0L	HYDROEOLM
Tirante Crítico (yc)	1.2668	1.2668	1.2668	1.2668	m	0.00%	0.00%	0.00%
Área Mojada (A)	1.2036	1.2035	1.2035	1.2035	m ²	0.01%	0.00%	0.00%
Perímetro Mojado (Pm)	3.1669	3.1669	3.1669	3.1669	m	0.00%	0.00%	0.00%
Radio Hidráulico (Rh)	0.38	0.38	0.38	0.38	m	0.00%	0.00%	0.00%
Ancho Superficial (T)	1.9001	1.9001	1.9001	1.9001	m	0.00%	0.00%	0.00%
Profundidad Hidráulica (D)	0.6334	0.6334	0.6334	0.6334	m	0.00%	0.00%	0.00%
Velocidad (V)	2.4925	2.4927	2.4927	2.4927	m/s	0.02%	0.00%	0.00%
Número de Froude (Fr)	1	1	1	1	-	0.03%	0.00%	0.04%
Tipo de Flujo	Critico	Critico	Critico	Critico	-	0.00%	0.00%	0.00%
Energía Específica (E)	1.5834	1.5835	1.5835	1.5835	m-Kg/Kg	0.01%	0.00%	0.00%

Realizado por: Alex Tipán

4.3. VERIFICACIÓN DE HIPÓTESIS

La verificación de esta hipótesis resulta de la investigación donde aumenta la eficiencia en optimización de recursos como son el tiempo de cálculos y la precisión. Con respecto al porcentaje de error se pudo determinar un análisis de resultados obtenidos mediante métodos como son: manual, con una aplicación móvil denominada HydroEolm_Canales, con el software SNCALES V2.0L, FICM CANALES V1.0 esta última siendo la aplicación móvil desarrollada en el cual arroja valores con un error menor al 0.5%.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- La investigación realizada en este proyecto técnico nos ha dado como resultado una adquisición de conocimiento profundo en el campo de la hidráulica como es el desarrollo de la aplicación móvil.
- Se obtuvo resultados buenos al realizar tablas comparativas tanto como método manuales y software.
- El desarrollo de esta aplicación móvil ha sido satisfactorio por los resultados obtenidos ya que su error es menor a 0.5%, dando una validez para el uso de este software.
- Se determinó que la aplicación móvil para el cálculo de canales abiertos de flujo uniforme ayuda a optimizar el tiempo de ejecución para la realización de un diseño de canales.
- La aplicación móvil FICM CANALES V 1.0 permite una eficiencia de cálculos para los elementos geométricos, caudal, sección hidráulica óptima, tirante normal y tirante crítico.

- La propiedad intelectual de los autores y la patente de la aplicación móvil se lo realizaron con éxito para beneficio de la Universidad y especialmente para la carrera de Ingeniería Civil.

5.2. RECOMENDACIONES

- Para un correcto diseño de canales abiertos de flujo uniforme es importante considerar los parámetros de entrada en la aplicación móvil.
- Es importante tener un conocimiento formado en el área de la hidráulica para un mejor entendimiento e interpretación de resultados en la aplicación móvil.
- Para entender la codificación y el lenguaje de esta aplicación móvil es importante tener en cuenta la programación orientada a objetos.
- Se recomienda ingresar valores coherentes para un óptimo diseño de canales abiertos de flujo uniforme con la aplicación FICM CANALES V 1.0.

1. BIBLIOGRAFÍA

- [1] M.A. Morales, K.K. Romero, R.E. Garzón, «"Aplicación del Software Educativo Cefficiency para el Diseño de Canales de Máxima Eficiencia Hidráulica",» *Dialnet*, vol. 7, pp. 101-103, Enero 2019.
- [2] V. Chow, «"Principios de Energía y Momentum",» de *Hidráulica de Canales*, Tercera ed., Colombia: mcgraw-Hill, 2004, pp. 39-42.
- [3] H.G. Villegas, «"Aspectos Generales Sobre Flujo Permanente en Canales",» de *Hidráulica de Canales*, Mexico: Fundación ICA, 1995, pp. 2-20.
- [4] L.J. Aguilar, «"Algoritmos y Estructura de Datos",» de *Estructura de Datos en Java*, España: mcgraw-Hill, 2008, pp. 1-71.
- [5] P. Rodríguez, *Hidráulica II*, Oxaca: S/N, 2008, pp. 130-153.
- [6] A. R. L. A. J.A. Barragan, *"Software para el Diseño de Canales Abiertos"*, Universidad de la Salle, Bogota, 2007.
- [7] W. C. H. Gálvez, «"Modernizacion de la enseñanza aprendizaje en la asignatura de hidráulica II",» 05 Mayo 2006. [En línea]. Available: https://www.academia.edu/8365506/HIDRAULICA_II, Mayo 2006. [Último acceso: 12 Julio 2020].

2. ANEXOS

2.1. CODIFICACIÓN DE LA APLICACIÓN MOVIL FICM CANALES V1.0

2.1.1. ELEMENTOS GEOMETRICOS

2.1.1.1. SECCIÓN RECTANGULAR

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package elementosgeometricos;
```

```
import static java.lang.Math.pow;
```

```
/**
```

```
*
```

```
* @author User
```

```
*/
```

```
public class ElemGeoSecRec {
```

```
    private double b; //Almacena la base
```

```
    private double y; //Almacena el tirante
```

```
private double D; //Profundidad hidraulica
```

```
private double Rh; //Radio Hidraulico
```

```
private double T; //Ancho superficial
```

```
private double P; //Perimetro mojado
```

```
private double A; // Area
```

```
ElemGeoSecRec(double b, double y) {
```

```
    this.b = b;
```

```
    this.y = y;
```

```
}
```

```
//Calcula el area
```

```
public double CalcularArea(double b, double y) {
```

```
    return this.b*this.y;
```

```
}
```

```
//Calcula el perimetro mojado
```

```
public double CalcularPm(double b, double y) {
```

```
    return (this.b) + (2.0 *this.y);
```

```
}
```

```
//Calcula el radio hidraulico
```

```
public double CalcularRh(double area, double pm) {  
  
    return area / pm;  
  
}
```

```
//Calcula el ancho superficial
```

```
public double CalcularT(double b) {  
  
    return this.b;  
  
}
```

```
//Calcula el profundidad hidraulica
```

```
public double CalcularD(double y) {  
  
    return this.y;  
  
}
```

```
public void CalcularValores() {  
  
    this.A = CalcularArea(this.b, this.y);  
  
    this.P = CalcularPm(this.b, this.y);  
  
}
```

```
    this.Rh = CalculaRh(A, P);  
  
    this.T = b;  
  
    this.D = CalculaD(this.y);  
  
}
```

```
public double getA() {  
    return A;  
}
```

```
public double getP() {  
    return P;  
}
```

```
public double getRh() {  
    return Rh;  
}
```

```
public double getT() {  
    return T;  
}
```

```
public double getD() {
```

```
        return D;
    }

}
```

2.1.1.2. SECCIÓN TRAPEZOIDAL

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package elementosgeometricos;

import static java.lang.Math.pow;
import static java.lang.Math.sqrt;

/**
 *
 * @author User
 */
public class ElemGeoSecTrap {

    private double b; //Almacena la base
    private double z; //Almacena el talud
    private double y; //Almacena el tirante
```



```
private double D; //Profundidad hidraulica
```

```
private double Rh; //Radio Hidraulico
```

```
private double T; //Ancho superficial
```

```
private double P; //Perimetro mojado
```

```
private double A; // Area
```

```
private double Q; //Caudal
```

```
ElemGeoSecTrap(double y, double b, double z) {
```

```
    this.y = y;
```

```
    this.b = b;
```

```
    this.z = z;
```

```
}
```

```
//Calcula el area
```

```
public double CalcularArea(double b, double y, double z) {
```

```
    return (b + z * y) * y;
```

```
}
```

```
// Calcular perimetro mojado
```

```
private double CalcularPm(double b, double y, double z) {  
    return b + 2 * y * (sqrt(1 + pow(z, 2.0)));  
}
```

```
//Calcular el radio hidraulico
```

```
public double CalcularRh(double area, double pm) {  
  
    return area / pm;  
}
```

```
//Calcular el ancho superficial
```

```
public double CalcularT(double b, double y, double z) {  
  
    return (b + 2 * z * y);  
}
```

```
//Calcular la profundidad hidraulica
```

```
public double CalcularD(double area, double t) {  
  
    return area / t;  
}
```

```
public void CalcularValores() {
```

```
    this.A = CalcularArea(b, y, z);  
  
    this.P = CalcularPm(b, y, z);  
  
    this.Rh = CalcularRh(A, P);  
  
    this.T = CalcularT(b, y, z);  
  
    this.D = A / T;  
  
}
```

```
public double getA() {  
    return A;  
}
```

```
public double getD() {  
    return D;  
}
```

```
public double getRh() {  
    return Rh;  
}
```

```
public double getT() {  
    return T;  
}
```

```
public double getP() {  
    return P;  
}
```

```
public double getQ() {  
    return Q;  
}
```

```
}
```

2.1.1.3. SECCIÓN TRIANGULAR

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package elementosgeometricos;
```

```
import static java.lang.Math.pow;
```

```
import static java.lang.Math.sqrt;
```

```
/**
```

```
*
```

```
* @author User
```

```

*/
public class ElemGeoSecTri {

    private double b; //Almacena la base

    private double z; //Almacena el talud

    private double y; //Almacena el tirante

    private double D; //Profundidad hidraulica

    private double Rh; //Radio Hidraulico

    private double T; //Ancho superficial

    private double P; //Perimetro mojado

    private double A; // Area

    ElemGeoSecTri(double y, double z) {

        this.y = y;

        this.z = z;

    }

    //Calcula el area

    public double CalcularArea(double z, double y) {

        return z * (y * y);

    }
}

```

```

// Calcular Perimetro mojado

private double CalcularPm(double y, double z) {

    return (2 * y) * (sqrt(1 + pow(z, 2.0)));

}

//Calcular el radio hidraulico

public double CalcularRh(double area, double pm) {

    return area / pm;

}

//Calcula Ancho Superficial

public double CalcularT(double z, double y) {

    return 2 * z * y;

}

//Calcula Profundidad Hidraulica

public double CalcularD(double y) {

    return y / 2;

}

```

```
public void CalcularValores() {  
  
    this.A = CalcularArea(this.z, this.y);  
  
    this.P = CalcularPm(this.y, this.z);  
  
    this.Rh = CalcularRh(A, P);  
  
    this.T = CalcularT(this.z, this.y);  
  
    this.D = A / T;  
  
}
```

```
public double getA() {  
  
    return A;  
  
}
```

```
public double getD() {  
  
    return D;  
  
}
```

```
public double getRh() {  
  
    return Rh;  
  
}
```

```
public double getT() {  
  
    return T;  
  
}
```

```
}
```

```
public double getP() {
```

```
    return P;
```

```
}
```

```
}
```

2.1.1.4. SECCIÓN CIRCULAR

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package elementosgeometricos;
```

```
import static java.lang.Math.PI;
```

```
import static java.lang.Math.abs;
```

```
import static java.lang.Math.acos;
```

```
import static java.lang.Math.atan;
```

```
import static java.lang.Math.cos;
```

```
import static java.lang.Math.pow;
```

```
import static java.lang.Math.sin;
```

```
import static java.lang.Math.sqrt;
```



```

import static java.lang.Math.tan;

import java.text.DecimalFormat;

/**
 *
 * @author User
 */
public class ElemGeoSecCir {

    private double Do; //Almacena el diametro

    private double S; //Almacena la pendiente

    private double n; //Almacena el coeficiente de Manning

    private double y; //Almacena el tirante

    private double D; //Profundidad hidraulica

    private double Rh; //Radio Hidraulico

    private double T; //Ancho superficial

    private double P; //Perimetro mojado

    private double A; // Area

    private double Q; //Caudal

    private double thetaRad;

    private double thetaDeg;

```

```

ElemGeoSecCir(double Do, double y) {

    this.Do = Do;

    this.y = y;

}

//Calcular el area

public double CalcularArea(double y) {

    return ((y - sin(y)) * pow(this.Do, 2.0)) / 8.0;

}

//Calcular el radio hidraulico

public double CalcularRh(double y) {

    return ((1.0 - (sin(y) / y)) * this.Do) / 4.0;

}

//Calcular angulo

public void CalcularValor1() {

    if (this.y == this.Do / 2.0) {

        y = Pi;
    }
}

```

```

} else if (this.y > Do / 2.0) {

    double y1 = this.y - (Do / 2.0);

    double t = acos(y1 / (Do / 2.0));

    double t1 = 2.0 * t;

    this.y = (2.0 * PI) - t1;

} else {

    double y1 = (Do / 2.0) - y;

    double t = acos(y1 / (Do / 2.0));

    this.y = 2.0 * t;

}

this.thetaRad = this.y;

}

public void CalcularValores() {

    this.A = CalcularArea(this.thetaRad);

    this.P = (this.thetaRad * this.Do) / 2.0;

    this.Rh = A / P;

    this.T = sin(this.thetaRad / 2.0) * this.Do;

    this.D = 0.125 * (thetaRad - sin(thetaRad)) / (sin(thetaRad / 2.0)) * Do;

```

```
}
```

```
public double getA() {  
    return A;  
}
```

```
public double getD() {  
    return D;  
}
```

```
public double getRh() {  
    return Rh;  
}
```

```
public double getT() {  
    return T;  
}
```

```
public double getP() {  
    return P;  
}
```

```
public double getQ() {
```

```
        return Q;
    }

}
```

2.1.1.5. SECCIÓN PARABÓLICO

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package elementosgeometricos;

import static java.lang.Math.log;
import static java.lang.Math.pow;
import static java.lang.Math.sqrt;

/**
 *
 * @author User
 */
public class ElemGeoSecPar {

    private double y; //Almacena el tirante
```

```
private double D; //Profundidad hidraulica
```

```
private double Rh; //Radio Hidraulico
```

```
private double T; //Ancho superficial
```

```
private double P; //Perimetro mojado
```

```
private double A; // Area
```

```
private double a;
```

```
private double b;
```

```
private double c;
```

```
ElemGeoSecPar(double y, double T) {
```

```
    this.y = y;
```

```
    this.T = T;
```

```
}
```

```
ElemGeoSecPar(double a, double b, double c, double T) {
```

```
    this.a = a;
```

```
    this.b = b;
```

```
    this.c = c;
```

```
    this.T = T;
```

```
}
```

```
//Calcula el area
```

```
public double CalcularArea(double T, double y) {
```

```
    return (2.0 * T * y) / 3.0;
```

```
}
```

```
//Calcula el radio hidraulico
```

```
public double CalcularRh(double area, double pm) {
```

```
    return area / pm;
```

```
}
```

```
//Calcula Profundidad Hidraulica
```

```
public double CalcularD(double y) {
```

```
    return y / 2;
```

```
}
```

```
// Funcion que calcula el caudal
```

```
public void Calcular1() {
```

```
    double x, aux0, aux1, Pm;
```

```
x = (4 * y) / T;
```

```
aux0 = sqrt(1.0 + pow(x, 2.0));
```

```
aux1 = log(x + aux0);
```

```
Pm = (T / 2.0) * (aux0 + aux1 / x);
```

```
this.P = Pm;
```

```
}
```

```
public void Calcular2() {
```

```
double x = 0;
```

```
double y = this.a * pow(x, 2.0) + this.b * x + c;
```

```
}
```

```
public void CalcularValores() {
```

```
this.A = CalcularArea(this.T, this.y);
```

```
this.Rh = A / P;
```

```
this.D = A / T;
```

```
}
```

```
public double getA() {
```

```
return A;
```



```
}

public double getP() {
    return P;
}

public double getRh() {
    return Rh;
}

public double getT() {
    return T;
}

public double getD() {
    return D;
}

}
```

2.1.1.6. SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package elementosgeometricos;
```

```
import static java.lang.Math.PI;
```

```
import static java.lang.Math.pow;
```

```
/**
```

```
*
```

```
* @author User
```

```
*/
```

```
public class ElemGeoSecRecEsqRed {
```

```
    private double b; //Almacena la base
```

```
    private double r; //Almacena el radio
```

```
    private double y; //Almacena el tirante
```

```
    private double D; //Profundidad hidraulica
```

```
    private double Rh; //Radio Hidraulico
```

```
    private double T; //Ancho superficial
```

```

private double P; //Perimetro mojado

private double A; // Area

ElemGeoSecRecEsqRed(double y, double b, double r) {

    this.y = y;

    this.b = b;

    this.r = r;

}

//Calcula el area

public double CalcularArea(double r, double b, double y) {

    double part1 = (PI / 2 - 2) * pow(r, 2.0);

    double part2 = (b + 2 * r) * y;

    return part1 + part2;

}

//Calcula el perimetro mojado

public double CalcularPm(double r, double b, double y) {

    double aux1 = (PI - 2) * r;

    double aux2 = (b + 2 * y);

```

```

        return aux1 + aux2;
    }

    //Calcula el radio hidraulico
    public double CalcularRh(double area, double Pm) {

        return area / Pm;
    }

    //Calcula Ancho Superficial
    public double CalcularT(double b, double r) {

        return b + 2 * r;
    }

    //Calcula la profundidad hidraulica
    public double CalcularD(double r, double b, double y) {

        return (((PI / 2 - 2) * pow(r, 2.0)) / (b + 2 * r)) + y;

    }

    public void CalcularValores() {

        this.A = CalcularArea(this.r, this.b, this.y);
    }

```

```
this.P = CalculaPm(this.r, this.b, this.y);
```

```
this.Rh = CalculaRh(A, P);
```

```
this.T = CalculaT(this.b, this.r);
```

```
this.D = CalculaD(this.r, this.b, this.y);
```

```
}
```

```
public double getA() {
```

```
    return A;
```

```
}
```

```
public double getP() {
```

```
    return P;
```

```
}
```

```
public double getRh() {
```

```
    return Rh;
```

```
}
```

```
public double getT() {
```

```
    return T;
```

```
}
```

```
public double getD() {
```

```
        return D;
    }

}
```

2.1.1.7. SECCIÓN TRIANGULAR CON FONDO REDONDEADO

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package elementosgeometricos;

import static java.lang.Math.PI;
import static java.lang.Math.pow;
import static java.lang.Math.sqrt;
import static java.lang.Math.tan;

/**
 *
 * @author User
 */
public class ElemGeoSecTriFonRed {
```

```

private double Q; //Almacena el caudal

private double r; //Almacena el radio

private double z; //Almacena el talud

private double S; //Almacena la pendiente

private double n; //Almacena el coeficiente de Manning

private double y; //Almacena el tirante

private double D; //Profundidad hidraulica

private double Rh; //Radio Hidraulico

private double T; //Ancho superficial

private double P; //Perimetro mojado

private double A; // Area

private double alfaRad;

private double alfaDeg;

ElemGeoSecTriFonRed(double y, double r, double z) {

    this.y = y;

    this.r = r;

    this.z = z;

}

//Calcula el area

```

```

public double CalcularArea(double ar) {

    return ar;
}

//Calcula el perimetro mojado
public double CalcularPm(double pm) {

    return pm;
}

//Calcula el radio hidraulico
public double CalcularRh(double rh) {

    return rh;
}

//Calcula el ancho superficial
public double CalcularT(double r, double z, double y) {

    double t = 2 * (z * (y - r) + r * (sqrt(1 + pow(z, 2.0))));
}

```



```

    return t;
}

//Calcula la profundidad hidraulica
public double CalcularD(double area, double t) {

    return area / t;
}

public double CalcularAlfa(double z) {

    return Math.atan(1.0 / z);
}

// Funcion calcular
public void Calcular() {

    double aux1, area, ter1, ter2, pm, rh;

    double ang = CalcularAlfa(this.z);

    double T = CalcularT(this.r, this.z, this.y);

    aux1 = (1 - ang / tan(ang));

    area = (pow(T, 2.0) / (4.0 * this.z)) - ((pow(r, 2.0) / this.z) * aux1);
}

```

```

ter1 = (T / this.z) * sqrt(1 + pow(this.z, 2.0));

ter2 = 2 * this.r / this.z;

pm = ter1 - (ter2 * aux1);

rh = area / pm;

this.alfaRad = ang;

this.alfaDeg = (ang * 180.0) / PI;

this.A = area;

this.P = pm;

}

public void CalcularValores() {

    this.Rh = A / P;

    this.T = CalcularT(this.r, this.z, this.y);

    this.D = A / T;

}

public double getA() {

    return A;

}

```

```
public double getP() {  
    return P;  
}  
  
public double getRh() {  
    return Rh;  
}  
  
public double getT() {  
    return T;  
}  
  
public double getD() {  
    return D;  
}  
  
}
```

2.1.2. CAUDALES

2.1.2.1 SECCIÓN RECTANGULAR

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package caudal;
```

```
import static java.lang.Math.pow;
```

```
import java.text.DecimalFormat;
```

```
/**
```

```
*
```

```
* @author User
```

```
*/
```

```
public class CauSecRec {
```

```
    private double Q; //Almacena el caudal
```

```
    private double b; //Almacena la base
```

```
    private double S; //Almacena la pendiente
```

```
    private double n; //Almacena el coeficiente de Manning
```

```
    private double y; //Almacena el tirante
```

```

private double D; //Profundidad hidraulica

private double Rh; //Radio Hidraulico

private double T; //Ancho superficial

private double P; //Perimetro mojado

private double A; // Area

private Velocidad Vel;

private EEspecifica EE;

private Froude Fro;

CauSecRec(double y, double b, double S, double n) {

    this.y = y;

    this.b = b;

    this.S = S;

    this.n = n;

}

//Calcula el area

public double CalcularArea(double b, double y) {

    return b * y;

}

//Calcula el perimetro mojado

```

```
public double CalcularPm(double b, double y) {  
  
    return b + (2 * y);  
}  
  
//Calcula el radio hidraulico  
  
public double CalcularRh(double area, double pm) {  
  
    return area / pm;  
}  
  
//Calcula el ancho superficial  
  
public double CalcularD(double y) {  
  
    return y;  
}  
  
//Calcula el caudal  
  
public double CalcularQ(double caudal) {  
  
    return caudal;  
}
```

```
}
```

```
// Funcion que calcula el caudal
```

```
public void CalcularCaudal() {
```

```
    double area = CalcularArea(b, y);
```

```
    double pm = CalcularPm(b, y);
```

```
    double rh = area / pm;
```

```
    double caudal = area * pow((rh * rh), 1.0 / 3.0) * pow(S, 0.5) / n;
```

```
    this.Q = caudal;
```

```
}
```

```
public void CalcularValores() {
```

```
    this.A = CalcularArea(this.b, this.y);
```

```
    this.P = CalcularPm(this.b, this.y);
```

```
    this.Rh = CalcularRh(A, P);
```

```
    this.T = b;
```

```
    this.D = CalcularD(this.y);
```

```
    this.Vel = new Velocidad(this.Q, this.A);
```

```
    this.EE = new EEspecifica(this.y, Vel.getVelocidad(), 9.81);
```

```
    this.Fro = new Froude(Vel.getVelocidad(), this.Rh);
```

```
}
```

```
public double getQ() {  
    return Q;  
}
```

```
public double getA() {  
    return A;  
}
```

```
public double getP() {  
    return P;  
}
```

```
public double getRh() {  
    return Rh;  
}
```

```
public double getT() {  
    return T;  
}
```

```
public double getD() {  
    return D;  
}
```



```

public double getVel() {
    return Vel.getVelocidad();
}

public double getEE() {
    return EE.getEnergia();
}

public String getFro() {
    return Fro.getTipo();
}

}

```

2.1.2.2 SECCIÓN TRAPEZOIDAL

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package caudal;

import static java.lang.Math.pow;

import static java.lang.Math.sqrt;

```

```

/**
 *
 * @author User
 */
public class CauSecTrap {

    private double b; //Almacena la base

    private double z; //Almacena el talud

    private double S; //Almacena la pendiente

    private double n; //Almacena el coeficiente de Manning

    private double y; //Almacena el tirante

    private double D; //Profundidad hidraulica

    private double Rh; //Radio Hidraulico

    private double T; //Ancho superficial

    private double P; //Perimetro mojado

    private double A; // Area

    private double Q; //Caudal

    private Velocidad Vel;

    private EEspecifica EE;

    private Froude Fro;

    CauSecTrap(double y, double b, double z, double S, double n) {

```

```

    this.y = y;

    this.b = b;

    this.z = z;

    this.S = S;

    this.n = n;

}

//Calcula el area

public double CalcularArea(double b, double y, double z) {

    return (b + z * y) * y;

}

// Calcular perimetro mojado

private double CalcularPm(double b, double y, double z) {

    return b + 2 * y * (sqrt(1 + pow(z, 2.0)));

}

//Calcular el radio hidraulico

public double CalcularRh(double area, double pm) {

    return area / pm;

```

```

}

//Calcular el ancho superficial

public double CalcularT(double b, double y, double z) {

    return (b + 2 * z * y);

}

//Calcular la profundidad hidraulica

public double CalcularD(double area, double t) {

    return area / t;

}

//Calcula el caudal

public double CalcularQ(double caudal) {

    return caudal;

}

// Funcion que calcula el caudal

public void CalcularCaudal() {

    double area = CalcularArea(b, y, z);

```

```

double pm = CalcularPm(b, y, z);

double rh = area / pm;

double caudal = area * pow(rh * rh, 1.0 / 3.0) * (pow(this.S, 0.5)) / this.n;

this.Q = caudal;
}

public void CalcularValores() {

    this.A = CalcularArea(b, y, z);

    this.P = CalcularPm(b, y, z);

    this.Rh = CalcularRh(A, P);

    this.T = CalcularT(b, y, z);

    this.D = A / T;

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.y, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.Rh);

}

public double getQ() {

    return Q;

}

public double getA() {

```

```
    return A;  
}
```

```
public double getP() {  
    return P;  
}
```

```
public double getRh() {  
    return Rh;  
}
```

```
public double getT() {  
    return T;  
}
```

```
public double getD() {  
    return D;  
}
```

```
public double getVel() {  
    return Vel.getVelocidad();  
}
```

```
public double getEE() {
```

```

        return EE.getEnergia();
    }

    public String getFro() {
        return Fro.getTipo();
    }
}

```

2.1.2.3 SECCIÓN TRIANGULAR

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package caudal;

import static java.lang.Math.pow;
import static java.lang.Math.sqrt;

/**
 *
 * @author User
 */
public class CauSecTri {

```

```

private double b; //Almacena la base

private double z; //Almacena el talud

private double S; //Almacena la pendiente

private double n; //Almacena el coeficiente de Manning

private double y; //Almacena el tirante

private double D; //Profundidad hidraulica

private double Rh; //Radio Hidraulico

private double T; //Ancho superficial

private double P; //Perimetro mojado

private double A; // Area

private double Q; //Caudal

private Velocidad Vel;

private EEspecifica EE;

private Froude Fro;

CauSecTri(double y, double z, double S, double n) {

    this.y = y;

    this.z = z;

    this.S = S;

    this.n = n;

```



```
}
```

```
//Calcula el area
```

```
public double CalcularArea(double z, double y) {
```

```
    return z * (y * y);
```

```
}
```

```
// Calcular Perimetro mojado
```

```
private double CalcularPm(double y, double z) {
```

```
    return (2 * y) * (sqrt(1 + pow(z, 2.0)));
```

```
}
```

```
//Calcular el radio hidraulico
```

```
public double CalcularRh(double area, double pm) {
```

```
    return area / pm;
```

```
}
```

```
//Calcula Ancho Superficial
```

```
public double CalcularT(double z, double y) {
```

```
    return 2 * z * y;
```

```

}

//Calcula Profundidad Hidraulica

public double CalcularD(double y) {

    return y / 2;

}

// Funcion que calcula el caudal

public void CalcularCaudal() {

    double area = CalcularArea(this.z, this.y);

    double pm = CalcularPm(this.y, this.z);

    double rh = area / pm;

    double caudal = area * pow(rh * rh, 1.0 / 3.0) * (pow(this.S, 0.5)) / this.n;

    this.Q = caudal;

}

public void CalcularValores() {

    this.A = CalcularArea(this.z, this.y);

    this.P = CalcularPm(this.y, this.z);

    this.Rh = CalcularRh(A, P);

    this.T = CalcularT(this.z, this.y);

    this.D = A / T;

```

```
    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.y, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.Rh);
}

public double getQ() {

    return Q;
}

public double getA() {

    return A;
}

public double getP() {

    return P;
}

public double getRh() {

    return Rh;
}

public double getT() {

    return T;
}
```

```
}

public double getD() {

    return D;

}

public double getVel() {

    return Vel.getVelocidad();

}

public double getEE() {

    return EE.getEnergia();

}

public String getFro() {

    return Fro.getTipo();

}

}
```

2.1.2.4 SECCIÓN CIRCULAR

/*

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

```

*/

package caudal;

import static java.lang.Math.PI;

import static java.lang.Math.pow;

import static java.lang.Math.sin;

import static java.lang.Math.sqrt;

/**
 *
 * @author User
 */

public class CauSecCir {

    private double Do; //Almacena el diametro

    private double S; //Almacena la pendiente

    private double n; //Almacena el coeficiente de Manning

    private double y; //Almacena el tirante

    private double D; //Profundidad hidraulica

    private double Rh; //Radio Hidraulico

    private double T; //Ancho superficial

    private double P; //Perimetro mojado

    private double A; // Area

```

```

private double Q; //Caudal

private Velocidad Vel;

private EEspecifica EE;

private Froude Fro;

private double thetaRad;

private double thetaDeg;

CauSecCir(double y, double Do, double S, double n) {

    this.y = y;

    this.Do = Do;

    this.S = S;

    this.n = n;

}

//Calcular el area

public double CalcularArea(double ang) {

    return ((ang - sin(ang)) * pow(this.Do, 2.0)) / 8;

}

```

```

//Calcular el radio hidraulico

public double CalcularRh(double ang) {

    return ((1 - (sin(ang) / ang)) * this.Do) / 4;

}

//Calcular angulo

public double CalcularTheta(double yn) {

    double m = 0;

    double l = 0;

    double r = this.Do / 2;

    double alpha;

    m = this.Do - yn;

    l = r - m;

    alpha = Math.acos(l / r); // alpha esta en grados

    return 2 * PI - 2 * alpha;

}

// Calcular Caudal

public void CalcularCaudal() {

    double ang;

    ang = 0;

```

```

    ang = CalcularTheta(this.y);

    double area = CalcularArea(ang);

    double rh = CalcularRh(ang);

    double caudal = area * pow(rh * rh, 1.0 / 3.0) * (pow(this.S, 0.5)) / this.n;

    this.Q = caudal;

    this.thetaRad = ang;

    this.thetaDeg = (ang * 180.0) / Pi;
}

public void CalcularValores() {

    this.A = CalcularArea(this.thetaRad);

    this.P = (this.thetaRad * this.Do) / 2;

    this.Rh = A / P;

    this.T = sin(this.thetaRad / 2.0) * this.Do;

    this.D = 0.125 * (thetaRad - sin(thetaRad)) / (sin(thetaRad / 2.0)) * Do;

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.y, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.Rh);
}

public double getQ() {

    return Q;
}

```



```
}
```

```
public double getA() {
```

```
    return A;
```

```
}
```

```
public double getP() {
```

```
    return P;
```

```
}
```

```
public double getRh() {
```

```
    return Rh;
```

```
}
```

```
public double getT() {
```

```
    return T;
```

```
}
```

```
public double getD() {
```

```
    return D;
```

```
}
```

```
public double getVel() {
```

```
    return Vel.getVelocidad();
```

```

    }

    public double getEE() {
        return EE.getEnergia();
    }

    public String getFro() {
        return Fro.getTipo();
    }

}

```

2.1.2.5 SECCIÓN PARABÓLICA

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package caudal;

import static java.lang.Math.log;
import static java.lang.Math.pow;
import static java.lang.Math.sqrt;

/**

```

```

*
* @author User
*/
public class CauSecPar {

    private double Q; //Almacena el caudal

    private double S; //Almacena la pendiente

    private double n; //Almacena el coeficiente de Manning

    private double y; //Almacena el tirante

    private double D; //Profundidad hidraulica

    private double Rh; //Radio Hidraulico

    private double T; //Ancho superficial

    private double P; //Perimetro mojado

    private double A; // Area

    private Velocidad Vel;

    private EEspecifica EE;

    private Froude Fro;

    CauSecPar(double y, double T, double S, double n) {

        this.y = y;

        this.T = T;

```

```

    this.S = S;

    this.n = n;

}

public double CalcularArea(double T, double y) {

    return (2.0 * T * y) / 3.0;

}

//Calcula el perimetro mojado

/* public double CalcularPm(double T, double y) {

    return T + ((8 * pow(y, 2.0)) / (3 * T));

}*/

//Calcula el radio hidraulico

public double CalcularRh(double area, double pm) {

    return area / pm;

}

//Calcula Profundidad Hidraulica

public double CalcularD(double y) {

```

```

        return 2 * y / 3;
    }

//Calcula el caudal

    public double CalcularQ(double caudal) {

        return caudal;

    }

// Funcion que calcula el caudal

    public void CalcularCaudal() {

        double x, aux0, aux1, Pm, area, rh, caudal;

        x = (4 * y) / T;

        aux0 = sqrt(1.0 + pow(x, 2.0));

        aux1 = log(x + aux0);

        Pm = (T / 2.0) * (aux0 + aux1 / x);

        area = CalcularArea(this.T, this.y);

        rh = area / Pm;

        caudal = area * pow((rh * rh), 1.0 / 3.0)*pow(S, 1.0 / 2.0)*(1/n);

```

```
    this.Q = caudal;

    this.P = Pm;

}

public void CalcularValores() {

    this.A = CalcularArea(T, y);

    this.Rh = CalcularRh(A, P);

    this.D = A / T;

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.y, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.Rh);

}

public double getQ() {

    return Q;

}

public double getA() {

    return A;

}
```

```
public double getP() {  
    return P;  
}
```

```
public double getRh() {  
    return Rh;  
}
```

```
public double getT() {  
    return T;  
}
```

```
public double getD() {  
    return D;  
}
```

```
public double getVel() {  
    return Vel.getVelocidad();  
}
```

```
public double getEE() {  
    return EE.getEnergia();  
}
```

```

public String getFro() {

    return Fro.getTipo();

}

}

```

2.1.2.6 SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package caudal;
```

```
import static java.lang.Math.PI;
```

```
import static java.lang.Math.pow;
```

```
/**
```

```
*
```

```
* @author User
```

```
*/
```

```
public class CauSecRecEsqRed {
```

```
    private double Q; //Almacena el caudal
```

```
    private double b; //Almacena la base
```



```

private double S; //Almacena la pendiente

private double r; //Almacena el radio

private double n; //Almacena el coeficiente de Manning

private double y; //Almacena el tirante

private double D; //Profundidad hidraulica

private double Rh; //Radio Hidraulico

private double T; //Ancho superficial

private double P; //Perimetro mojado

private double A; // Area

private Velocidad Vel;

private EEspecifica EE;

private Froude Fro;

CauSecRecEsqRed(double y, double b, double r, double S, double n) {

    this.y = y;

    this.b = b;

    this.r = r;

    this.S = S;

    this.n = n;

}

```

```
//Calcula el area
```

```
public double CalcularArea(double r, double b, double y) {  
    double part1 = (PI / 2 - 2) * pow(r, 2.0);  
    double part2 = (b + 2 * r) * y;  
  
    return part1 + part2;  
}
```

```
public double CalcularPm(double r, double b, double y) {  
    double aux1 = (PI - 2) * r;  
    double aux2 = (b + 2 * y);  
  
    return aux1 + aux2;  
}
```

```
public double CalcularRh(double area, double Pm) {  
  
    return area / Pm;  
}
```

```
//Calcula Ancho Superficial
```

```
public double CalcularT(double b, double r) {
```

```

        return b + 2 * r;
    }

    public double CalcularD(double r, double b, double y) {

        return (((PI / 2 - 2) * pow(r, 2.0)) / (b + 2 * r)) + y;

    }

//Calcula el caudal

    public double CalcularQ(double caudal) {

        return caudal;

    }

// Funcion que calcula el caudal

    public void CalcularCaudal() {

        double area = CalcularArea(this.r, this.b, this.y);

        double pm = CalcularPm(this.r, this.b, this.y);

        double rh = area / pm;

        double caudal = area * pow(rh * rh, 1.0 / 3.0) * pow(S, 0.50) / n;

        this.Q = caudal;
    }

```

```
}
```

```
public void CalcularValores() {
```

```
    this.A = CalcularArea(this.r, this.b, this.y);
```

```
    this.P = CalcularPm(this.r, this.b, this.y);
```

```
    this.Rh = CalcularRh(A, P);
```

```
    this.T = CalcularT(this.b, this.r);
```

```
    this.D = CalcularD(this.r, this.b, this.y);
```

```
    this.Vel = new Velocidad(this.Q, this.A);
```

```
    this.EE = new EEspecifica(this.y, Vel.getVelocidad(), 9.81);
```

```
    this.Fro = new Froude(Vel.getVelocidad(), this.Rh);
```

```
}
```

```
public double getQ() {
```

```
    return Q;
```

```
}
```

```
public double getA() {
```

```
    return A;
```

```
}
```

```
public double getP() {
```

```
    return P;
```

```
}
```

```
public double getRh() {
```

```
    return Rh;
```

```
}
```

```
public double getT() {
```

```
    return T;
```

```
}
```

```
public double getD() {
```

```
    return D;
```

```
}
```

```
public double getVel() {
```

```
    return Vel.getVelocidad();
```

```
}
```

```
public double getEE() {
```

```
    return EE.getEnergia();
```

```
}
```

```
public String getFro() {
```

```
    return Fro.getTipo();
```

```
}
```

```
}
```

2.1.2.7 SECCIÓN TRIANGULAR CON FONDO REDONDEADO

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package caudal;
```

```
import static java.lang.Math.PI;
```

```
import static java.lang.Math.pow;
```

```
import static java.lang.Math.sqrt;
```

```
import static java.lang.Math.tan;
```

```
/**
```

```
*
```

```
* @author User
```

```
*/
```

```
public class CauSecTriFonRed {
```

```
    private double Q; //Almacena el caudal
```

```
    private double r; //Almacena el radio
```

```

private double z; //Almacena el talud

private double S; //Almacena la pendiente

private double n; //Almacena el coeficiente de Manning

private double y; //Almacena el tirante

private double D; //Profundidad hidraulica

private double Rh; //Radio Hidraulico

private double T; //Ancho superficial

private double P; //Perimetro mojado

private double A; // Area

private Velocidad Vel;

private EEspecifica EE;

private Froude Fro;

private double alfaRad;

private double alfaDeg;

CauSecTriFonRed(double y, double r, double z, double S, double n) {

    this.y = y;

    this.r = r;

    this.z = z;

    this.S = S;

```

```
this.n = n;
```

```
}
```

```
public double CalcularArea(double ar) {
```

```
    return ar;
```

```
}
```

```
public double CalcularPm(double pm) {
```

```
    return pm;
```

```
}
```

```
public double CalcularRh(double rh) {
```

```
    return rh;
```

```
}
```

```
public double CalcularT(double r, double z, double y) {
```

```
    double t = 2 * (z * (y - r) + r * (sqrt(1 + pow(z, 2.0))));
```



```

        return t;
    }

    public double CalcularD(double area, double t) {

        return area / t;
    }

    public double CalcularAlfa(double z) {

        return Math.atan(1.0 / z);

    }

    public double CalcularQ(double caudal) {

        return caudal;

    }

// Funcion que calcula el caudal

    public void CalcularCaudal() {

        double aux1, area, ter1, ter2, Pm, rh;

```

```

double ang = CalcularAlfa(z);

double T = CalcularT(r, z, y);

aux1 = (1 - ang / tan(ang));

area = (pow(T, 2.0) / (4.0 * z)) - ((pow(r, 2.0) / z) * aux1);

ter1 = (T / z) * sqrt(1 + pow(z, 2.0));

ter2 = 2 * r / z;

Pm = ter1 - (ter2 * aux1);

rh = area / Pm;

double caudal = area * pow((rh * rh), 1.0 / 3.0) * pow(S, 0.5) / n;

this.Q = caudal;

this.alfaRad = ang;

this.alfaDeg = (ang * 180.0) / PI;

this.A = area;

this.P = Pm;

}

public void CalcularValores() {

this.Rh = A / P;

this.T = CalcularT(this.r, this.z, this.y);

```

```
this.D = A / T;
```

```
this.Vel = new Velocidad(this.Q, this.A);
```

```
this.EE = new EEspecifica(this.y, Vel.getVelocidad(), 9.81);
```

```
this.Fro = new Froude(Vel.getVelocidad(), this.Rh);
```

```
}
```

```
public double getQ() {
```

```
    return Q;
```

```
}
```

```
public double getA() {
```

```
    return A;
```

```
}
```

```
public double getP() {
```

```
    return P;
```

```
}
```

```
public double getRh() {
```

```
    return Rh;
```

```
}
```

```
public double getT() {
```

```
        return T;
    }

    public double getD() {
        return D;
    }

    public double getVel() {
        return Vel.getVelocidad();
    }

    public double getEE() {
        return EE.getEnergia();
    }

    public String getFro() {
        return Fro.getTipo();
    }
}
```

2.1.3. SECCIÓN HIDRÁULICA ÓPTIMA

2.1.3.1 SECCIÓN RECTANGULAR

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package seccionsoptimas;
```

```
import static java.lang.Math.abs;
```

```
import static java.lang.Math.pow;
```

```
import java.text.DecimalFormat;
```

```
/**
```

```
*
```

```
* @author User
```

```
*/
```

```
public class SecOpRec {
```

```
    private double Q; //Almacena el caudal
```

```

private double S; //Almacena la pendiente

private double n; //Almacena el coeficiente de Manning

private double yo; //Almacena el tirante normal


private double A; //Area

private double P; //Perimetro mojado

private double Rh; //Radio Hidraulico

private double T; //Ancho superficial

private double D; //Profundidad hidraulica

private double b; //Base

private Velocidad Vel;

private EEspecifica EE;

private Froude Fro;

private final double err = 0.000001; //Error maximo que puede tener la iteraccion

SecOpRec(double Q, double S, double n) {

    this.Q = Q;

    this.S = S;

    this.n = n;

}

public double CalcularArea(double b, double y) {

```

```
    return b * y;  
}
```

```
public double CalcularPm(double b, double y) {
```

```
    return b + (2 * y);  
}
```

```
public double CalcularRh(double area, double pm) {
```

```
    return area / pm;  
}
```

```
public double CalcularD(double y) {
```

```
    return y;  
}
```

```
public void CalcularYo() {
```

```
    double yo = 100;  
  
    double M = 100000;
```

```

DecimalFormat formato1 = new DecimalFormat("#0.000");

for (int i = 0; i <= 100000; i++) {

    double FY = ((1 / n) * (2 * yo * yo) * pow(yo / 2.0, 2.0 / 3.0) * pow(S, 1.0 / 2.0)) -
(this.Q);

    if (FY == 0) {

        break;

    } else {

        double FDY = (1 / n) * pow(S, 1 / 2) * pow(2.0, 1.0 / 3.0) * (8.0 / 3.0) * pow(yo, 5.0
/ 3.0);

        double ER = FY / FDY;

        yo = yo - ER;

        if (abs(ER) <= err) {

            break;

        }

        System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyn: "
+ formato1.format(yo));
    }
}

```



```

    }

}

this.yo = yo;

b = yo * 2;

}

public void CalcularValores() {

    this.A = CalcularArea(this.b, this.yo);

    this.P = CalcularPm(this.b, this.yo);

    this.Rh = CalcularRh(A, P);

    this.T = b;

    this.D = CalcularD(this.yo);

    this.b = b;

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.yo, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.Rh);

}

public double getb() {

    return b;

}

```

```
public double getYo() {  
    return yo;  
}
```

```
public double getA() {  
    return A;  
}
```

```
public double getP() {  
    return P;  
}
```

```
public double getRh() {  
    return Rh;  
}
```

```
public double getT() {  
    return T;  
}
```

```
public double getD() {  
    return D;  
}
```

```

public double getVel() {
    return Vel.getVelocidad();
}

public double getEE() {
    return EE.getEnergia();
}

public String getFro() {
    return Fro.getTipo();
}

}

```

2.1.3.2 SECCIÓN TRAPEZOIDAL CON “Z” CONOCIDO

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

*/

```

package seccionsoptimas;

import static java.lang.Math.abs;

import static java.lang.Math.pow;

import static java.lang.Math.sqrt;

```

```

import java.text.DecimalFormat;

/**
 *
 * @author User
 */
public class SecOpTrap {

    private double Q; //Almacena el caudal

    private double b; //Almacena la base

    private double z; //Almacena el talud

    private double S; //Almacena la pendiente

    private double n; //Almacena coeficiente de Manning

    private double yo; //Almacena el tirante optimo

    private double A; //Area

    private double P; //Perimetro mojado

    private double Rh; //Radio Hidraulico

    private double T; //Ancho superficial

    private double D; //Profundidad hidraulica

    private Velocidad Vel;

    private EEspecifica EE;

    private Froude Fro;

```

```
private final double err = 0.000001; //error maximo que puede tener la iteracion
```

```
SecOpTrap(double Q, double z, double S, double n) {
```

```
    this.Q = Q;
```

```
    this.z = z;
```

```
    this.S = S;
```

```
    this.n = n;
```

```
}
```

```
// Calcula el area
```

```
public double CalcularArea(double b, double y, double z) {
```

```
    return (b + z * y) * y;
```

```
}
```

```
// Calcular perimetro mojado
```

```
private double CalcularPm(double b, double y, double z) {
```

```
    return b + 2 * y * (sqrt(1 + pow(z, 2.0)));
```

```
}
```

```
//Calcular el radio hidraulico
```

```
public double CalcularRh(double area, double pm) {
```

```

    return area / pm;
}

//Calcular el ancho superficial
public double CalcularT(double b, double y, double z) {

    return (b + 2 * z * y);
}

//Calcular la profundidad hidraulica

public double CalcularD(double area, double t) {

    return area / t;
}

public void CalcularYo() {

    double y = 100;

    double M = 100000;

    double FDY1, FDY2, FDY3, FDY;

    DecimalFormat formato1 = new DecimalFormat("#0.000000");

```

```

for (int i = 0; i <= 1000000; i++) {

    double FY = (1.0 / this.n) * (((2.0 * y * (sqrt(1.0 + pow(this.z, 2.0)) - this.z)) + this.z * y)
* y) * pow(y / 2.0, 2.0 / 3.0) * pow(this.S, 1.0 / 2.0) - (this.Q);

    if (FY == 0) {

        break;

    } else {

        FDY1 = (pow(this.S, 1.0 / 2.0) * (y * this.z - 2.0 * y * (this.z - pow((pow(this.z, 2.0) +
1.0), 1.0 / 2.0))) * pow(y / 2.0, 2.0 / 3.0)) / this.n;

        FDY2 = (pow(this.S, 1.0 / 2.0) * y * (this.z - 2.0 * pow((pow(this.z, 2.0) + 1.0), 1.0 /
2.0)) * pow(y / 2.0, 2.0 / 3.0)) / this.n;

        FDY3 = (pow(this.S, 1.0 / 2.0) * y * (y * z - 2.0 * y * (z - pow((pow(this.z, 2.0) + 1.0),
1.0 / 2.0)))) / (3.0 * this.n * pow((y / 2.0), 1.0 / 3.0));

        FDY = FDY1 - FDY2 + FDY3;

        double ER = FY / FDY;

        y = y - ER;

        if (abs(ER) <= err) {

            break;

```

```

    }

    System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyo: "
        + formato1.format(y));
    }

}

this.yo = y;

this.b=2.0*y*(sqrt(1.0+pow(z,2.0))-z);

}

public void CalcularValores() {

    this.A = CalcularArea(this.b, this.yo, this.z);

    this.P = CalcularPm(this.b, this.yo, this.z);

    this.Rh = CalcularRh(A, P);

    this.T = CalcularT(this.b,this.yo, this.z);

    this.D = CalcularD(A, T);

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.yo, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.Rh);

}

public double getb() {

```



```
        return b;
    }

    public double getYo() {
        return yo;
    }

    public double getA() {
        return A;
    }

    public double getP() {
        return P;
    }

    public double getRh() {
        return Rh;
    }

    public double getT() {
        return T;
    }

    public double getD() {
```

```

        return D;
    }

    public double getVel() {
        return Vel.getVelocidad();
    }

    public double getEE() {
        return EE.getEnergia();
    }

    public String getFro() {
        return Fro.getTipo();
    }
}

```

2.1.3.3 SECCIÓN TRAPEZOIDAL CON “Z” NO CONOCIDO

```
/*
```

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

```
*/
```

```
package seccionsoptimas;
```

```

import static java.lang.Math.abs;

import static java.lang.Math.atan;

import static java.lang.Math.pow;

import static java.lang.Math.sqrt;

import java.text.DecimalFormat;

/**
 *
 * @author User
 */
public class SecOpTrapNo {

    private double Q; //Almacena el caudal

    private double b; //Almacena la base

    private double z=(1/1.73205080757); //Almacena el talud

    private double S; //Almacena la pendiente

    private double n; //Almacena coeficiente de Manning

    private double yo;//Almacena el tirante optimo

    private double A; //Area

    private double P; //Perimetro mojado

    private double Rh; //Radio Hidraulico

    private double T; //Ancho superficial

    private double D; //Profundidad hidraulica

```

```

private Velocidad Vel;

private EEspecifica EE;

private Froude Fro;

private final double err = 0.000001; //error maximo que puede tener la iteracion

SecOpTrapNo(double Q, double S, double n) {

    this.Q = Q;

    this.S = S;

    this.n = n;

}

// Calcula el area

public double CalcularArea(double b, double y, double z) {

    return (b + z * y) * y;

}

// Calcular perimetro mojado

private double CalcularPm(double b, double y, double z) {

    return b + 2 * y * (sqrt(1 + pow(z, 2.0)));

}

```

```

//Calcular el radio hidraulico

public double CalcularRh(double area, double pm) {

    return area / pm;

}

//Calcular el ancho superficial

public double CalcularT(double b, double y, double z) {

    return (b + 2 * z * y);

}

//Calcular la profundidad hidraulica

public double CalcularD(double area, double t) {

    return area / t;

}

public void CalcularYo() {

    double y = 100;

    double M = 100000;

    double FDY1, FDY2, FDY3, FDY4, FDY;

```

```

double z=(1/1.73205080757);

DecimalFormat formato1 = new DecimalFormat("#0.000000");

for (int i = 0; i <= 1000000; i++) {

    double FY = ((1 / this.n) * (((1.155 * y) + z * y) * y) * pow((((1.155 * y) + z * y) * y) /
    ((1.155 * y) + 2 * y * sqrt(1 + pow(z, 2.0))), 2.0 / 3.0)) * pow(this.S, 1.0 / 2.0) - (this.Q);

    if (FY == 0) {

        break;

    } else {

        FDY1 = (pow(this.S, 1.0 / 2.0) * (1.155 * y + y * z) * pow(((y * (1.155 * y + y * z)) /
        (1.155 * y + 2.0 * y * pow((pow(z, 2.0) + 1.0), 1.0 / 2.0))), 2.0 / 3.0)) / this.n;

        FDY2 = (pow(this.S, 1.0 / 2.0) * y * (z + 1.155) * pow(((y * (1.155 * y + y * z)) /
        (1.155 * y + 2.0 * y * pow((pow(z, 2.0) + 1.0), 1.0 / 2.0))), 2.0 / 3.0)) / this.n;

        FDY3 = (2.0 * pow(S, 1.0 / 2.0) * y * (1.155 * y + y * z) * ((1.155 * y + y * z) / (1.155
        * y + 2.0 * y * pow((pow(z, 2.0) + 1.0), 1.0 / 2.0)) + (y * (z + 1.155)) / (1.155 * y + 2.0 * y *
        pow((pow(z, 2.0) + 1.0), 1.0 / 2.0)) - (y * (2.0 * pow((pow(z, 2.0) + 1.0), 1.0 / 2.0) + 1.155) *
        (1.155 * y + y * z)) / pow((1.155 * y + 2.0 * y * pow((pow(z, 2.0) + 1.0), 1.0 / 2.0)), 2.0));

        FDY4 = (3 * n * pow(((y * (1.155 * y + y * z)) / (1.155 * y + 2.0 * y * pow((pow(z,
        2.0) + 1.0), 1.0 / 2.0))), 1.0 / 3.0));

        FDY = FDY1 + FDY2 + (FDY3 / FDY4);
    }
}

```

```

        double ER = FY / FDY;

        y = y - ER;

        if (abs(ER) <= err) {

            break;

        }

        System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyo: "
            + formato1.format(y));

    }

}

this.yo = y;

this.b = 1.155*yo;

}

public void CalcularValores() {

    this.A = CalcularArea(this.b, this.yo, this.z);

    this.P = CalcularPm(this.b, this.yo, this.z);

    this.Rh = CalcularRh(A, P);

    this.T = CalcularT(this.b, this.yo, this.z);

    this.D = CalcularD(A, T);

    this.Vel = new Velocidad(this.Q, this.A);

```

```
        this.EE = new EEspecifica(this.yo, Vel.getVelocidad(), 9.81);

        this.Fro = new Froude(Vel.getVelocidad(), this.Rh);

    }

    public double getz() {

        return z;

    }

    public double getb() {

        return b;

    }

    public double getYo() {

        return yo;

    }

    public double getA() {

        return A;

    }

    public double getP() {

        return P;

    }

    public double getRh() {
```



```
        return Rh;
    }

    public double getT() {
        return T;
    }

    public double getD() {
        return D;
    }

    public double getVel() {
        return Vel.getVelocidad();
    }

    public double getEE() {
        return EE.getEnergia();
    }

    public String getFro() {
        return Fro.getTipo();
    }
}
```

2.1.3.4 SECCIÓN TRIANGULAR

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package seccionsoptimas;
```

```
import static java.lang.Math.abs;
```

```
import static java.lang.Math.pow;
```

```
import static java.lang.Math.sqrt;
```

```
import java.text.DecimalFormat;
```

```
/**
```

```
*
```

```
* @author User
```

```
*/
```

```
public class SecOpTri {
```

```
    private double Q; //Almacena el caudal
```

```
    private double z; //Almacena el talud
```

```
    private double S; //Almacena la pendiente
```

```
    private double n; //Almacena el coeficiente de Manning
```

```
    private double yn; //Almacena el tirante normal
```

```

private double A; //Area

private double P; //Perimetro mojado

private double Rh; //Radio Hidraulico

private double T; //Ancho superficial

private double D; //Profundidad hidraulica

private Velocidad Vel;

private EEspecifica EE;

private Froude Fro;

private double yo;

SecOpTri(double Q, double S, double n) {

    this.Q = Q;

    this.S = S;

    this.n = n;

}

//Calcula el area

public double CalcularArea(double z, double y) {

    return z * (y * y);

```

```
}
```

```
// Calcular Perimetro mojado
```

```
private double CalcularPm(double y, double z) {
```

```
    return (2 * y) * (sqrt(1 + pow(z, 2.0)));
```

```
}
```

```
//Calcular el radio hidraulico
```

```
public double CalcularRh(double area, double pm) {
```

```
    return area / pm;
```

```
}
```

```
//Calcula Ancho Superficial
```

```
public double CalcularT(double z, double y) {
```

```
    return 2 * z * y;
```

```
}
```

```
//Calcula Profundidad Hidraulica
```

```
public double CalcularD(double y) {
```

```
    return y / 2;
```

```

}

public void CalculaYo() {

    double yo = 100;

    double M = 100000;

    z = 1;

    double err = 0.000001;

    DecimalFormat formato1 = new DecimalFormat("#0.0000");

    for (int i = 0; i <= 1000000; i++) {

        double a0 = this.z * pow(yo, 2.0);

        double p0 = 2.0 * yo * sqrt(1 + pow(this.z, 2.0));

        double FY = (1 / this.n) * a0 * pow(a0 / p0, 2.0 / 3.0) * pow(S, 1.0 / 2.0) - (this.Q);

        if (FY == 0) {

            break;

        } else {

            double FDY = (2.0 * pow(this.S, 1.0 / 2.0) * yo * this.z * pow(a0 / p0, 2 / 3)) / this.n
+ (pow(this.S, 1.0 / 2.0) * pow(yo, 2.0) * pow(z, 2.0)) / (3.0 * this.n * sqrt(pow(this.z, 2.0) +
1.0) * pow(a0 / p0, 1 / 3));

```

```

        double ER = FY / FDY;

        yo = yo - ER;

        if (abs(ER) <= err) {

            break;

        }

        System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyn: "

            + formato1.format(yo));

    }

}

this.yo = yo;

}

public void CalcularValores() {

    this.A = CalcularArea(this.z, yo);

    this.P = CalcularPm(yo, z);

    this.Rh = CalcularRh(this.A, this.P);

    this.T = CalcularT(this.z, yo);

    this.D = CalcularD(yo);

```

```
    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.yo, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.Rh);

}

public double getYo() {

    return yo;

}

public double getA() {

    return A;

}

public double getP() {

    return P;

}

public double getRh() {

    return Rh;

}

public double getT() {

    return T;

}
```

```

public double getD() {
    return D;
}

public double getVel() {
    return Vel.getVelocidad();
}

public double getEE() {
    return EE.getEnergia();
}

public String getFro() {
    return Fro.getTipo();
}

}

```

2.1.3.5 SECCIÓN SEMICIRCULAR

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```



```

package seccionsoptimas;

import static java.lang.Math.PI;

import static java.lang.Math.abs;

import static java.lang.Math.pow;

import static java.lang.Math.sin;

import java.text.DecimalFormat;

/**
 *
 * @author User
 */
public class SecOpSemiCir {

    private double Q; //Almacena caudal

    private double S; //Almacena pendiente

    private double n; //Almacena coeficiente de maning

    private double A; //Area

    private double P; //Perimetro mojado

    private double Rh; //Radio Hidraulico

    private double T; //Ancho superficial

    private double D; //Profundidad hidraulica

    private Velocidad Vel;

```

```

private EEspecifica EE;

private Froude Fro;

//----- valores de configuracion

private final double err = 0.000001;

private double yo; //Almacena el yo Optimo

private double d; //Almacena el diametro

private double a = PI;

SecOpSemiCir(double Q, double S, double n) {

    this.Q = Q;

    this.S = S;

    this.n = n;

}

//Calcular el area

public double CalcularArea(double x) {

    return ((x - sin(x)) * pow(this.d, 2.0)) / 8.0;

}

//Calcular el perimetro

public double CalcularPm(double x) {

```

```

        return (1.0 / 2.0) * x * this.d;
    }

    //Calcular el radio hidraulico
    public double CalcularRh(double area, double perimetro) {

        return area / perimetro;
    }

    //Calcular el ancho superficial

    public double CalcularT(double x) {

        return d * sin(x / 2.0);
    }

    //Calcular la profundidad hidraulica

    public double CalcularD(double area, double t) {

        return area / t;
    }

    //Iteracion Calculo yo
    public void CalcularYo() {

```

```

double y = 100;

double M = 100000;

DecimalFormat formato1 = new DecimalFormat("#0.000");

for (int i = 0; i <= 100000; i++) {

    double FY1 = (1.0 / n) * (1.0 / 8.0) * (a - sin(a)) * pow(y, 2.0);

    double FY2 = pow((((1.0 / 8.0) * (a - sin(a)) * pow(y, 2.0)) / ((1.0 / 2.0) * a * y)), 2.0 /
3.0) * pow(S, 1.0 / 2.0);

    double FY = FY1 * FY2 - this.Q;

    if (FY == 0) {

        break;

    } else {

        double FDY1 = ((pow(S, 1.0 / 2.0) * y * (a - sin(a)) * pow(((y * (a - sin(a))) / (4 * a)),
2.0 / 3.0)) / (4 * n));

        double FDY2 = (pow(S, 1.0 / 2.0) * pow(y, 2.0) * pow((a - sin(a)), 2.0) / (48 * a * n *
pow(((y * (a - sin(a))) / (4 * a)), 1.0 / 3.0)));

        double FDY = FDY1 + FDY2;

        double ER = FY / FDY;

```

```

        y = y - ER;
        if (abs(ER) <= err) {
            break;
        }

        System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyn: "
            + formato1.format(yo));
    }

}

this.d = y;

this.yo = d / 2;
}

//Funcion para calcular los parametros
public void CalcularValores() {
    this.A = CalcularArea(PI);
    this.P = CalcularPm(PI);
    this.Rh = A / P;
    this.T = CalcularT(a);
    this.D = A / T;
    this.d = T;
    this.Vel = new Velocidad(this.Q, this.A);
    this.EE = new EEspecifica(this.yo, Vel.getVelocidad(), 9.81);
}

```

```
        this.Fro = new Froude(Vel.getVelocidad(), this.Rh);
    }
public double getd() {
    return d;
}
public double getYo() {
    return yo;
}

public double getA() {
    return A;
}

public double getP() {
    return P;
}

public double getRh() {
    return Rh;
}

public double getT() {
    return T;
}
```

```
public double getD() {  
    return D;  
}  
  
public double getVel() {  
    return Vel.getVelocidad();  
}  
  
public double getEE() {  
    return EE.getEnergia();  
}  
  
public String getFro() {  
    return Fro.getTipo();  
}  
  
}
```

2.1.4. TIRANTE NORMAL

2.1.4.1 SECCIÓN RECTANGULAR

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package tirantenormal;
```

```
import static java.lang.Math.PI;
```

```
import static java.lang.Math.abs;
```

```
import static java.lang.Math.pow;
```

```
import java.text.DecimalFormat;
```

```
/**
```

```
*
```

```
* @author User
```

```
*/
```

```
public class TirNorSecRec {
```

```
    private double Q; //Almacena el caudal
```

```
    private double b; //Almacena la base
```

```
    private double S; //Almacena la pendiente
```

```
    private double n; //Almacena el coeficiente de Manning
```



```

private double yn; //Almacena el tirante normal

private double A; // Area

private double P; //Perimetro mojado

private double Rh; //Radio Hidraulico

private double T; //Ancho superficial

private double D; //Profundidad hidraulica

private Velocidad Vel;

private EEspecifica EE;

private Froude Fro;

private final double errM = 0.000001; //error maximo que puede tener la iteraccion

TirNorSecRec(double Q, double b, double S, double n) {

    this.Q = Q;

    this.b = b;

    this.S = S;

    this.n = n;

}

//Calcula el area

public double CalcularArea(double b, double y) {

    return b * y;
}

```

```

}

// Calcula Perimetro mojado

public double CalcularPm(double b, double y) {

    return b + (2 * y);

}

//Calcula el radio hidraulico

public double CalcularRh(double area, double pm) {

    return area / pm;

}

//Calcula Profundidad Hidraulica

public double CalcularD(double y) {

    return y;

}

//Iteracion para el Calculo de yn

public void CalcularYn() {

```

```

//Variables locales

double yn = 100; //Valor inicial para yc

DecimalFormat formato1 = new DecimalFormat("#0.000");

for (int i = 0; i <= 100000; i++) {

    double a0 = b * yn;

    double p0 = (b + 2 * yn);

    double FY = ((1.0 / this.n) * (a0) * (pow(a0 / p0, 2.0 / 3.0)) * pow(S, 1.0 / 2.0)) -
(this.Q);

    if (FY == 0) {

        break;

    } else {

        double FDY = (pow(this.S, 1.0 / 2.0) * b * pow(a0 / p0, 2.0 / 3.0)) / this.n + (2 *
pow(S, 1.0 / 2.0) * a0 * (this.b / p0 - (2 * a0) / pow(p0, 2.0))) / (3.0 * this.n * pow(a0 / p0,
1.0 / 3.0));

        double ER = FY / FDY;

        yn = yn - ER;

        if (abs(ER) <= errM) {

            break;

```

```

    }

    System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyn: "
        + formato1.format(yn));
    }

}

//Si no hay ninguna falla añadir los valores a los miembros de clase
this.yn = yn;

}

//Funcion para calcular los parametros

public void CalcularValores() {

    this.A = CalcularArea(this.b, this.yn);

    this.P = CalcularPm(this.b, this.yn);

    this.Rh = CalcularRh(A, P);

    this.T = b;

    this.D = CalcularD(this.yn);

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.yn, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.Rh);

}

```

```
public double getYn() {  
    return yn;  
}
```

```
public double getA() {  
    return A;  
}
```

```
public double getP() {  
    return P;  
}
```

```
public double getRh() {  
    return Rh;  
}
```

```
public double getT() {  
    return T;  
}
```

```
public double getD() {  
    return D;  
}
```

```

public double getVel() {
    return Vel.getVelocidad();
}

public double getEE() {
    return EE.getEnergia();
}

public String getFro() {
    return Fro.getTipo();
}

}

```

2.1.4.2 SECCIÓN TRAPEZOIDAL

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package tirantenormal;

import static java.lang.Math.PI;

import static java.lang.Math.abs;

import static java.lang.Math.pow;

```

```

import static java.lang.Math.sqrt;

import java.text.DecimalFormat;

public class TirNorSecTrap {

    private double Q; //Almacena el caudal

    private double b; //Almacena la base

    private double z; //Almacena el talud

    private double S; //Almacena la pendiente

    private double n; //Almacena coeficiente de Manning

    private double yn; //Almacena el tirante normal

    private double A; //Area

    private double P; //Perimetro mojado

    private double Rh; //Radio Hidraulico

    private double T; //Ancho superficial

    private double D; //Profundidad hidraulica

    private Velocidad Vel;

    private EEspecifica EE;

    private Froude Fro;

    private final double err = 0.000001; //error maximo que puede tener la iteraccion

```

```

TirNorSecTrap(double Q, double b, double z, double S, double n) {

    this.Q = Q;

    this.b = b;

    this.z = z;

    this.S = S;

    this.n = n;

    this.D = 0.0;

    this.A = 0.0;

}

// Calcula el area

public double CalcularArea(double b, double y, double z) {

    return (b + z * y) * y;

}

// Calcular perimetro mojado

private double CalcularPm(double b, double y, double z) {

    return b + 2 * y * (sqrt(1 + pow(z, 2.0)));

}

```



```

//Calcular el radio hidraulico

public double CalcularRh(double area, double pm) {

    return area / pm;
}

//Calcular el ancho superficial

public double CalcularT(double b, double y, double z) {

    return (b + 2 * z * y);
}

//Calcular la profundidad hidraulica

public double CalcularD(double area, double t) {

    return area / t;
}

public void CalcularYn() {

    double yn = 100;

    double M = 100000;

    DecimalFormat formato1 = new DecimalFormat("#0.000000");

```

```

for (int i = 0; i <= 1000000; i++) {

    double a0 = ((b + z * yn) * yn);

    double p0 = (b + 2 * yn * sqrt(1 + pow(z, 2.0)));

    double FY = ((1.0 / n) * (a0) * (pow(a0 / p0, 2.0 / 3.0)) * pow(S, 1.0 / 2.0)) - Q;

    if (FY == 0) {

        break;

    } else {

        double FDY = (pow(S, 1.0 / 2.0) * (b + yn * z) * pow(a0 / p0, 2.0 / 3.0)) / n + (pow(S,
1.0 / 2.0) * yn * z * pow(a0 / p0, 2.0 / 3.0)) / n + (2.0 * pow(S, 1.0 / 2.0) * yn * (b + yn * z) *
((b + yn * z) / p0 + (yn * z) / p0 - (2.0 * yn * (b + yn * z) * sqrt(pow(z, 2.0) + 1.0)) / pow(p0,
2.0))) / (3.0 * n * pow(a0 / p0, 1.0 / 3.0));

        double ER = FY / FDY;

        yn = yn - ER;

        if (abs(ER) <= err) {

            break;

        }

        System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyn: "
+ formato1.format(yn));

    }
}

```

```

    }

    this.yn = yn;

}

public void CalcularValores() {

    this.A = CalcularArea(b, yn, z);

    this.P = CalcularPm(b, yn, z);

    this.Rh = CalcularRh(A, P);

    this.T = CalcularT(b, yn, z);

    this.D = CalcularD(A, T);

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.yn, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.Rh);

}

public double getYn() {

    return yn;

}

public double getA() {

    return A;

```

```
}
```

```
public double getP() {
```

```
    return P;
```

```
}
```

```
public double getRh() {
```

```
    return Rh;
```

```
}
```

```
public double getT() {
```

```
    return T;
```

```
}
```

```
public double getD() {
```

```
    return D;
```

```
}
```

```
public double getVel() {
```

```
    return Vel.getVelocidad();
```

```
}
```

```
public double getEE() {
```

```
    return EE.getEnergia();
```

```

    }

    public String getFro() {
        return Fro.getTipo();
    }
}

```

2.1.4.3 SECCIÓN TRIANGULAR

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package tirantenormal;

import static java.lang.Math.PI;

import static java.lang.Math.abs;

import static java.lang.Math.pow;

import static java.lang.Math.sqrt;

import java.text.DecimalFormat;

/**
 *
 * @author User
 */

```

```

public class TirNorSecTri {

    private double Q; //Almacena el caudal

    private double z; //Almacena el talud

    private double S; //Almacena la pendiente

    private double n; //Almacena el coeficiente de Manning

    private double yn; //Almacena el tirante normal

    private double A; //Area

    private double P; //Perimetro mojado

    private double Rh; //Radio Hidraulico

    private double T; //Ancho superficial

    private double D; //Profundidad hidraulica

    private Velocidad Vel;

    private EEspecifica EE;

    private Froude Fro;

    private final double errM = 0.000001; //error maximo que puede tener la iteracion

    TirNorSecTri(double Q, double z, double S, double n) {

        this.Q = Q;

        this.z = z;

```

```

    this.S = S;

    this.n = n;

}

//Calcula el area

public double CalcularArea(double z, double y) {

    return z * (y * y);

}

// Calcular Perimetro mojado

private double CalcularPm(double y, double z) {

    return (2 * y) * (sqrt(1 + pow(z, 2.0)));

}

//Calcular el radio hidraulico

public double CalcularRh(double area, double pm) {

    return area / pm;

}

//Calcula Ancho Superficial

```

```

public double CalcularT(double z, double y) {

    return 2 * z * y;

}

//Calcula Profundidad Hidraulica

public double CalcularD(double y) {

    return y / 2;

}

//Iteracion para el Calculo de yn

public void CalcularYn() {

    //Variables locales

    double yn = 100;

    double M = 100000;

    DecimalFormat formato1 = new DecimalFormat("#0.0000");

    for (int i = 0; i <= 1000000; i++) {

        double a0 = this.z * pow(yn, 2.0);

        double p0 = 2.0 * yn * sqrt(1 + pow(this.z, 2.0));

        double FY = (1 / this.n) * a0 * pow(a0 / p0, 2.0 / 3.0) * pow(S, 1.0 / 2.0) - (this.Q);

```



```

if (FY == 0) {

    break;

} else {

    double FDY = (2.0 * pow(this.S, 1.0 / 2.0) * yn * this.z * pow(a0 / p0, 2 / 3)) / this.n
+ (pow(this.S, 1.0 / 2.0) * pow(yn, 2.0) * pow(z, 2.0)) / (3.0 * this.n * sqrt(pow(this.z, 2.0) +
1.0) * pow(a0 / p0, 1 / 3));

    double ER = FY / FDY;

    yn = yn - ER;

    if (abs(ER) <= errM) {

        break;

    }

    System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyn: "
        + formato1.format(yn));

}

}

//Si no hay ninguna falla añadir los valores a los miembros de clase

this.yn = yn;

```

```

}

//Funcion para calcular los parametros

public void CalcularValores() {

    this.A = CalcularArea(this.z, yn);

    this.P = CalcularPm(yn, z);

    this.Rh = CalcularRh(this.A, this.P);

    this.T = CalcularT(this.z, yn);

    this.D = CalcularD(yn);

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.yn, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.Rh);

}

public double getYn() {

    return yn;

}

public double getA() {

    return A;

}

public double getP() {

```

```
        return P;
    }

    public double getRh() {
        return Rh;
    }

    public double getT() {
        return T;
    }

    public double getD() {
        return D;
    }

    public double getVel() {
        return Vel.getVelocidad();
    }

    public double getEE() {
        return EE.getEnergia();
    }

    public String getFro() {
```

```
        return Fro.getTipo();
    }

}
```

2.1.4.4 SECCIÓN CIRCULAR

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package tirantenormal;

import static java.lang.Math.PI;

import static java.lang.Math.abs;

import static java.lang.Math.atan;

import static java.lang.Math.cos;

import static java.lang.Math.pow;

import static java.lang.Math.sin;

import static java.lang.Math.sqrt;

import static java.lang.Math.tan;

import java.text.DecimalFormat;

public class TirNorSecCir {
```

```

//Variables

private double Q; //Almacena caudal

private double Do; //Almacena dimetro

private double S; //Almacena pendiente

private double n; //Almacena coeficiente de maning

private double D; //Almacena dimetro

private double Rh; // Radio Hidraulico

private double T; //Ancho Superficial

private double P; //Perimetro mojado

private double A; //Area

private Velocidad Vel;

private EEspecifica EE;

private Froude Fro;

//Valores de configuracion

private final double errM = 0.000001; //error maximo que puede tener la iteracion

private double yn; //Almacena el tirante normal

private double alfaRad;

private double alfaDeg;

TirNorSecCir(double Q, double Do, double S, double n) {

```

```

this.Q = Q;

this.Do = Do;

this.S = S;

this.n = n;

}

//Calcular el area

public double CalcularArea(double yn) {

    return ((yn - sin(yn)) * pow(this.Do, 2.0)) / 8;

}

// Calcula Perimetro mojado

private double CalcularPm(double y) {

    return (y * this.Do) / 2;

}

//Calcular el radio hidraulico

public double CalcularRh(double ang) {

    return ((1 - (sin(ang) / ang)) * this.Do) / 4;

}

```

```
//Calcula el Ancho Superficial
```

```
public double CalcularT(double y) {
```

```
    return (sin(y / 2)) * this.Do;
```

```
}
```

```
//Calcula Profundidad Hidraulica
```

```
public double CalcularD(double a, double t) {
```

```
    return a / t;
```

```
}
```

```
//Iteracion Calculo yn
```

```
public void CalcularYn() {
```

```
    //Variables locales
```

```
    double yn = Pi; //Valor inicial para yc
```

```
    DecimalFormat formato1 = new DecimalFormat("#0.000");
```

```
    for (int i = 0; i <= 100000; i++) {
```

```

double FY = ((1.0 / this.n) * ((1.0 / 8.0) * (yn - sin(yn)) * pow(this.Do, 2.0)) *
(pow((((1.0 / 8.0) * (yn - sin(yn)) * pow(this.Do, 2.0)) / ((1.0 / 2.0) * yn * this.Do)), 2.0 / 3.0))
* pow(S, 1.0 / 2.0)) - (this.Q);

```

```

if (FY == 0) {

```

```

    break;

```

```

} else {

```

```

    double FDY1 = (pow(S, 1.0 / 2.0) * pow(this.Do, 2.0) * (cos(yn) - 1.0) *
pow(((this.Do * (yn - sin(yn))) / (4.0 * yn)), 2.0 / 3.0)) / (8.0 * this.n);

```

```

    double FDY2 = (pow(S, 1.0 / 2.0) * pow(this.Do, 2.0) * (yn - sin(yn)) * ((this.Do * (yn
- sin(yn))) / (4.0 * pow(yn, 2.0)) + (this.Do * (cos(yn) - 1.0)) / (4.0 * yn))) / (12 * n *
pow(((this.Do * (yn - sin(yn))) / (4.0 * yn)), 1.0 / 3.0));

```

```

double FDY = -FDY1 - FDY2;

```

```

double ER = FY / FDY;

```

```

yn = yn - ER;

```

```

if (abs(ER) <= errM) {

```

```

    break;

```

```

}

```

```

System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyn: "

```



```

        + formato1.format(yn));
    }

}

//Si no hay ninguna falla añadir los valores a los miembros de clase
this.yn = (this.Do / 2.0) * (1.0 - cos(yn / 2.0));

this.alfaRad = yn;
}

//Funcion para calcular los parametros
public void CalcularValores() {

    this.A = CalcularArea(this.alfaRad);

    this.P = CalcularPm(this.alfaRad);

    this.Rh = A / P;

    this.T = CalcularT(this.alfaRad);

    this.D = A / T;

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.yn, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.Rh);
}

public double getYn() {

    return yn;
}

```

```
public double getA() {  
    return A;  
}
```

```
public double getP() {  
    return P;  
}
```

```
public double getRh() {  
    return Rh;  
}
```

```
public double getT() {  
    return T;  
}
```

```
public double getD() {  
    return D;  
}
```

```
public double getVel() {  
    return Vel.getVelocidad();  
}
```

```

public double getEE() {
    return EE.getEnergia();
}

public String getFro() {
    return Fro.getTipo();
}

}

```

2.1.4.5 SECCIÓN PARABÓLICA CON “T” CONOCIDO

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package tirantenormal;

import static java.lang.Math.PI;

import static java.lang.Math.abs;

import static java.lang.Math.cos;

import static java.lang.Math.log;

import static java.lang.Math.pow;

import static java.lang.Math.sin;

```

```

import static java.lang.Math.sqrt;

import java.text.DecimalFormat;

/**
 *
 * @author User
 */
public class TirNorSecParT {

    private double Q; //Almacena el caudal

    private double S; //Almacena la pendiente

    private double T; // Almacena el ancho superficial

    private double n; //Almacena el coeficiente de Manning

    private double yn; //Almacena el tirante normal

    private double D; //Almacena Profundidad Hidraulic

    private double Rh; //Almacena el radio hidraulico

    private double P; //Almacena Perimetro mojado

    private double A; //Almacena el Area

    private Velocidad Vel;

    private EEspecifica EE;

    private Froude Fro;

```

```

private FuncionObjetivo mFuncionObjetivo;

private final double errM = 0.000001;

private final double ap = 0.0001;

TirNorSecParT(double Q, double T, double S, double n) {

    this.Q = Q;

    this.T = T;

    this.S = S;

    this.n = n;

    mFuncionObjetivo = new FuncionObjetivo(Q, S, n);

}

```

```
//Calcula el area
```

```

public double CalcularArea(double t, double yn) {

    return (2.0 * t * yn) / 3.0;

}

```

```
//Calcula el perimetro mojado
```

```

public double CalcularPm(double T, double yn) {

    //return (T + (8.0 * pow(yn, 2.0)) / (3.0 * T));

    double pm;

```

```

double x = (4.0 * yn) / T;

double aux0 = sqrt(1.0 + pow(x, 2.0));

double aux1 = log(x + aux0);

return pm = (T / 2.0) * (aux0 + aux1 / x);

}

//Calcula el radio hidraulico

public double CalcularRh(double area, double pm) {

return area / pm;

}

//Calcula Ancho Superficial

public double CalcularT(double area, double yn) {

return 3 * area / 2 * yn;

}

//Calcula Profundidad Hidraulica

public double CalcularD(double yn) {

return 2 * yn / 3;

}

public void CalcularYn() {

```

```
double yn = 0.0001;

double err;

double segTermino;

double PrimerTermino = mFuncionObjetivo.getObjetivo();

int cont = 0;

boolean fail = false;

boolean fin = false;

DecimalFormat formato1 = new DecimalFormat("#0.000000");

do {

    cont++;

    double area = CalcularArea(this.T, yn);

    double pm = CalcularPm(this.T, yn);

    double rh = area / pm;

    segTermino = area * pow(rh * rh, 1.0 / 3.0);

    err = PrimerTermino - segTermino;

    System.out.println("i: " + cont + "\terr: " + formato1.format(err) + "\tyn: "
```

```
        + formato1.format(yn) + "\tSegundo termino: " +  
formato1.format(segTermino));
```

```
    if (Math.abs(err) > errM) {
```

```
        yn += ap;
```

```
    } else {
```

```
        fin = true;
```

```
    }
```

```
    if (fail == false) {
```

```
    }
```

```
    } while (!fin && err > 0);
```

```
    if (fail == false) {
```

```
        //si no hay ninguna falla añadir los valores a los miembros de clase
```

```
        this.yn = yn;
```

```
    }
```

```
}
```



```
public void CalcularValores() {

    this.A = CalcularArea(this.T, this.yn);

    this.P = CalcularPm(T, yn);

    this.Rh = A / P;

    this.D = A / T;

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.yn, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.D);
}

public double getYn() {

    return yn;

}

public double getA() {

    return A;

}

public double getP() {

    return P;

}
```

```
public double getRh() {  
    return Rh;  
}
```

```
public double getT() {  
    return T;  
}
```

```
public double getD() {  
    return D;  
}
```

```
public double getVel() {  
    return Vel.getVelocidad();  
}
```

```
public double getEE() {  
    return EE.getEnergia();  
}
```

```
public String getFro() {  
    return Fro.getTipo();  
}
```

```
}
```

2.1.4.6 SECCIÓN PARABÓLICA CON “T” NO CONOCIDO

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package tirantenormal;
```

```
import static java.lang.Math.log;
```

```
import static java.lang.Math.log10;
```

```
import static java.lang.Math.pow;
```

```
import static java.lang.Math.sqrt;
```

```
import java.text.DecimalFormat;
```

```
/**
```

```
*
```

```
* @author User
```

```
*/
```

```
public class TirNorSecPar {
```

```
    private double Q; //Almacena el caudal
```

```
    private double S; //Almacena la pendiente
```

```

private double n; //Almacena el coeficiente de Manning

private double yn; //Almacena el tirante normal

private double D; //Almacena Profundidad Hidraulic

private double Rh; //Almacena el radio hidraulico

private double T; //Almacena Ancho superficial

private double P; //Almacena Perimetro mojado

private double A; //Almacena el Area

private Velocidad Vel;

private EEspecifica EE;

private Froude Fro;

private FuncionObjetivo mFuncionObjetivo;

private final double errM = 0.001;

private final double ap = 0.001;

TirNorSecPar(double Q, double S, double n) {

    this.Q = Q;

    this.S = S;

    this.n = n;

    mFuncionObjetivo = new FuncionObjetivo(Q, S, n);

}

```

```
//Calcula el area
```

```
public double CalcularArea(double T, double yn) {
```

```
    return (2.0 * T * yn) / 3.0;
```

```
}
```

```
//Calcula el perimetro mojado
```

```
public double CalcularPm(double Pm) {
```

```
    return Pm;
```

```
}
```

```
//Calcula el radio hidraulico
```

```
public double CalcularRh(double area, double pm) {
```

```
    return area / pm;
```

```
}
```

```
//Calcula Ancho Superficial
```

```
public double CalcularT(double y) {
```

```
    return 2 * (sqrt(y));
```

```
}
```

```
//Calcula Profundidad Hidraulica
```

```
public double CalcularD(double y) {
```

```
    return y / 2;
```

```
}
```

```
public void CalcularYn() {
```

```
    double yn = 0.001;
```

```
    double err;
```

```
    double segTermino;
```

```
    double T;
```

```
    double x;
```

```
    double Pm;
```

```
    double PrimerTermino = mFuncionObjetivo.getObjetivo();
```

```
    int cont = 0;
```

```
    boolean fail = false;
```

```
    boolean fin = false;
```

```
    DecimalFormat formato1 = new DecimalFormat("#0.000000");
```

```
    do {
```

```

cont++;

T = 2.0 * (sqrt(yn));

x = (4.0 * yn) / T;

double aux0 = sqrt(1.0 + pow(x, 2.0));

double aux1 = log(x + aux0);

Pm = (T / 2.0) * (aux0 + aux1 / x);

double area = CalcularArea(T, yn);

double rh = CalcularRh(area, Pm);

segTermino = area * pow(rh, 2.0 / 3.0);

err = PrimerTermino - segTermino;

System.out.println("i: " + cont + "\terr: " + formato1.format(err) + "\t: "
    + formato1.format(yn) + "\tSegundo termino: " +
formato1.format(segTermino));

if (Math.abs(err) > errM) {

    yn += ap;

} else {

    fin = true;

}

```

```

        if (fail == false) {

            this.yn = yn;

            this.P = Pm;

        }

    } while (!fin && err > 0);

}

public void CalcularValores() {

    this.T = CalcularT(yn);

    this.A = CalcularArea(this.T, this.yn);

    this.P = CalcularPm(P);

    this.Rh = A / P;

    this.D = A / T;

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.yn, Vel.getVelocidad(), 9.81);

    this.Froude = new Froude(Vel.getVelocidad(), this.D);

}

public double getYn() {

    return yn;

}

```



```
public double getA() {  
    return A;  
}
```

```
public double getP() {  
    return P;  
}
```

```
public double getRh() {  
    return Rh;  
}
```

```
public double getT() {  
    return T;  
}
```

```
public double getD() {  
    return D;  
}
```

```
public double getVel() {  
    return Vel.getVelocidad();  
}
```

```

public double getEE() {
    return EE.getEnergia();
}

public String getFro() {
    return Fro.getTipo();
}
}

```

2.1.4.7 SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package tirantenormal;
```

```
import static java.lang.Math.PI;
```

```
import static java.lang.Math.abs;
```

```
import static java.lang.Math.acos;
```

```
import static java.lang.Math.pow;
```

```
import static java.lang.Math.sin;
```

```
import static java.lang.Math.sqrt;
```

```
import java.text.DecimalFormat;
```

```

public class TirNorSecRecEsqRed {

    private double Q; //Almacena el caudal

    private double b; //Almacena la base

    private double r; //Almacena el radio

    private double S; //Almacena la pendiente

    private double n; //Almacena el coeficiente de Manning

    private double yn; //Almacena el tirante normal

    private double D; //Profundidad hidraulica

    private double Rh; //Radio Hidraulico

    private double T; //Ancho superficial

    private double P; //Perimetro mojado

    private double A; //Area

    private Velocidad Vel;

    private EEspecifica EE;

    private Froude Fro;

    TirNorSecRecEsqRed(double Q, double b, double r, double S, double n) {

        this.Q = Q;

        this.b = b;

        this.r = r;

```

```

this.S = S;

this.n = n;

this.D = 0.0;

this.A = 0.0;

}

public double CalculaArea(double r, double b, double y) {

    double part1 = (PI / 2 - 2) * pow(r, 2.0);

    double part2 = (b + 2 * r) * y;

    return part1 + part2;

}

public double CalculaPm(double r, double b, double y) {

    double aux1 = (PI - 2) * r;

    double aux2 = (b + 2 * y);

    return aux1 + aux2;

}

public double CalculaRh(double area, double Pm) {

```

```

    return area / Pm;
}

//Calcula Ancho Superficial
public double CalcularT(double b, double r) {

    return b + 2 * r;
}

public double CalcularD(double r, double b, double y) {

    return (((PI / 2 - 2) * pow(r, 2.0)) / (b + 2 * r)) + y;

}

public void CalcularYn() {

    double yn = 100;

    double M = 100000;

    double err = 0.001;

    DecimalFormat formato1 = new DecimalFormat("#0.000000");

```

```

for (int i = 0; i <= 1000000; i++) {

    double a0 = (((PI / 2.0) - 2.0) * pow(r, 2.0)) + ((b + 2 * r) * yn);

    double p0 = ((PI - 2.0) * r) + b + (2 * yn);

    double FY = (1 / n) * a0 * (pow(a0 / p0, 2.0 / 3.0)) * pow(S, 1.0 / 2.0) - (this.Q);

    if (FY == 0) {

        break;

    } else {

        double FDY = (pow(S, 1.0 / 2.0) * (b + 2 * r) * (pow(a0 / p0, 2.0 / 3.0))) / n + (2 *
pow(S, 1.0 / 2.0) * a0 * ((b + 2 * r) / p0 - (2.0 * a0) / pow(p0, 2.0))) / (3 * n * pow(a0 / p0,
1.0 / 3.0));

        double ER = FY / FDY;

        yn = yn - ER;

        if (abs(ER) <= err) {

            break;

        }

        System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyn: "
+ formato1.format(yn));

    }
}

```

```

    }

    this.yn = yn;

}

public void CalcularValores() {

    this.A = CalcularArea(this.r, this.b, this.yn);

    this.P = CalcularPm(r, b, yn);

    this.Rh = CalcularRh(A, P);

    this.T = CalcularT(b, r);

    this.D = CalcularD(this.r, this.b, this.yn);

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.yn, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.Rh);

}

public double getYn() {

    return yn;

}

public double getA() {

    return A;

```

```
}
```

```
public double getP() {
```

```
    return P;
```

```
}
```

```
public double getRh() {
```

```
    return Rh;
```

```
}
```

```
public double getT() {
```

```
    return T;
```

```
}
```

```
public double getD() {
```

```
    return D;
```

```
}
```

```
public double getVel() {
```

```
    return Vel.getVelocidad();
```

```
}
```

```
public double getEE() {
```

```
    return EE.getEnergia();
```



```

    }

    public String getFro() {
        return Fro.getTipo();
    }

}

```

2.1.4.8 SECCIÓN TRIANGULAR CON FONDO REDONDEADO

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package tirantenormal;

import static java.lang.Math.PI;

import static java.lang.Math.abs;

import static java.lang.Math.atan;

import static java.lang.Math.cos;

import static java.lang.Math.pow;

import static java.lang.Math.sin;

import static java.lang.Math.sqrt;

import static java.lang.Math.tan;

import java.text.DecimalFormat;

```

```

/**
 *
 * @author User
 */
public class TirNorSecTriFonRed {

    private double Q; // Almacena el caudal

    private double r; //Almacena el radio

    private double z; //Almacena inclinacion de talud

    private double S; //Almacena la pendiente

    private double n; //Almacena Coeficiente de Manning

    private double A; //Area

    private double P; //Perimetro Mojado

    private double Rh; //Radio Hidraulico

    private double T; //Ancho Superficial

    private double D; //Profundidad Hidraulica

    private Velocidad Vel;

    private EEspecifica EE;

    private Froude Fro;

    private double errM = 0.000001; //error maximo que puede tener la iteraccion

```

```
private double yn; //Almacena en tirante normal
```

```
TirNorSecTriFonRed(double Q, double r, double z, double S, double n) {
```

```
    this.Q = Q;
```

```
    this.r = r;
```

```
    this.z = z;
```

```
    this.S = S;
```

```
    this.n = n;
```

```
}
```

```
public double CalcularArea(double r, double T, double z) {
```

```
    double ar = (((pow(T, 2.0)) / (4 * z)) - (((pow(r, 2.0)) / z) * (1 - (z * acot(z)))));
```

```
    return ar;
```

```
}
```

```
// Calcula perimetro mojado
```

```
private double CalcularPm(double T, double r, double z) {
```

```
    double pm1 = ((T / z) * sqrt(1 + pow(z, 2.0))) - (((2 * r) / z) * (1 - (z * acot(z))));
```

```
    return pm1;
```

```
}
```

```
//Calcula el radio hidraulico
```

```
public double CalcularRh(double ar, double pm) {
```

```

    return ar / pm;
}

//Calcula el ancho superficial
public double CalcularT(double r, double z, double yn) {

    double t = 2 * (z * (yn - r) + r * (sqrt(1 + pow(z, 2.0))));

    return t;
}

//Calcular la profundidad hidraulica

public double CalcularD(double ar, double t) {

    return ar / t;
}

public double acot(double x) {

    return PI / 2.0 - atan(x);
}

//Iteracion para el Calculo de yn
public void CalcularYn() {

```

```

//Variables locales

double y = 100;

double FY1, FY2, FY, FDY, FDY1, FDY2, FDY3, FDY4, FDY5;

DecimalFormat formato1 = new DecimalFormat("#0.000000");

//Inicio de las iteraciones

for (int i = 0; i <= 1000000; i++) {

    FY1 = (1 / n) * ((pow(y, 2.0) / (4.0 * z)) - ((pow(r, 2.0) / z) * (1.0 - (z * acot(z)))));

    FY2 = (pow((((pow(y, 2.0) / (4.0 * z)) - ((pow(r, 2.0) / z) * (1.0 - (z * acot(z)))))) / ((y / z)
* sqrt(1 + pow(z, 2.0))) - (((2.0 * r) / z) * (1.0 - z * acot(z))))), 2.0 / 3.0) * (pow(S, 1.0 / 2.0));

    FY = FY1 * FY2 - this.Q;

    if (FY == 0) {

        break;

    } else {

        FDY1 = (pow(S, 1.0 / 2.0) * y * (pow((((z * acot(z) - 1.0) * pow(r, 2.0) / z + pow(y,
2.0) / (4.0 * z)) / ((y * pow((pow(z, 2.0) + 1.0), 1.0 / 2.0)) / z + (2.0 * r * (z * acot(z) - 1.0)) /
z)), 2.0 / 3.0))) / (2.0 * n * z);

```

```
FDY2 = ((y / (2 * z * ((y * sqrt(1 + pow(z, 2.0))) / z + (2 * r * (z * acot(z) - 1)) / z)))));
```

```
FDY3 = (pow((pow(z, 2.0) + 1.0), 1.0 / 2.0) * (((z * acot(z) - 1.0) * pow(r, 2.0) / z +  
pow(y, 2.0) / (4.0 * z)))) / (z * pow(((y * pow((pow(z, 2.0) + 1.0), 1.0 / 2.0)) / z + (2.0 * r * (z  
* acot(z) - 1.0)) / z), 2.0));
```

```
FDY4 = (((z * acot(z) - 1) * pow(r, 2)) / z + pow(y, 2) / (4 * z));
```

```
FDY5 = 3.0 * n * (pow(((pow(y, 2.0) / (4.0 * z) + (pow(r, 2.0) * (z * acot(z) - 1.0)) / z)  
/ ((y * pow((pow(z, 2.0) + 1.0), 1.0 / 2.0)) / z + (2.0 * r * (z * acot(z) - 1.0)) / z)), 1.0 / 3.0));
```

```
FDY = FDY1 + ((2.0 * pow(S, 1.0 / 2.0)) * (FDY2 - FDY3) * (FDY4)) / FDY5;
```

```
double ER = FY / FDY;
```

```
y = y - ER;
```

```
if (abs(ER) <= errM) {
```

```
    break;
```

```
}
```

```
System.out.println("i: " + i
```

```
    + "\tFy: " + formato1.format(FY) + "\tFDY: " + formato1.format(FDY)
```

```
    + "\tER: " + formato1.format(ER) + "\tyn: " + formato1.format(y));
```

```
}
```

```

}

T = y;

double w = atan(1 / z);

double w1 = PI - (2 * w);

double B = ((2 * PI) - (2 * w1)) / 2;

double Y2 = r * (1 - cos(B / 2));

double M = 2 * r * sin(B / 2);

double C = (T - M) / 2;

double Y1 = tan(w) * C;

//Si no hay ninguna falla añadir los valores a los miembros de clase

this.yn = Y1 + Y2;

}

//Funcion para calcular los parametros

public void CalcularValores() {

    this.A = CalcularArea(r, T, z);

    this.P = CalcularPm(T, r, z);

    this.D = CalcularD(A, T);

    this.Rh = CalcularRh(A, P);

    this.T = CalcularT(r, z, yn);

    this.Vel = new Velocidad(this.Q, this.A);

```

```
    this.EE = new EEspecifica(this.yn, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.Rh);
}

public double getYn() {

    return yn;
}

public double getA() {

    return A;
}

public double getP() {

    return P;
}

public double getRh() {

    return Rh;
}

public double getT() {

    return T;
}
```



```
public double getD() {  
    return D;  
}  
  
public double getVel() {  
    return Vel.getVelocidad();  
}  
  
public double getEE() {  
    return EE.getEnergia();  
}  
  
public String getFro() {  
    return Fro.getTipo();  
}  
  
}
```

2.1.4. TIRANTE CRÍTICO

2.1.4.1 SECCIÓN RECTANGULAR

```
/*  
  
* To change this license header, choose License Headers in Project Properties.  
  
* To change this template file, choose Tools | Templates  
  
* and open the template in the editor.  
  
*/  
  
package tirantecritico;  
  
  
import static java.lang.Math.abs;  
  
import static java.lang.Math.pow;  
  
import static java.lang.Math.tan;  
  
import java.text.DecimalFormat;  
  
  
/**  
  
*  
  
* @author User  
  
*/  
  
public class TirCriSecRec {  
  
    //Variables  
  
    private double Q; //Almacena caudal  
  
    private double b; //Almacena la base
```

```

private double A; //Area

private double P; //Perimetro Mojado

private double Rh; //Radio Hidraulico

private double T; //Ancho Superficial

private double D; //Profundidad Hidraulica

private Velocidad Vel;

private EEspecifica EE;

private Froude Fro;

// valores de configuracion

private double errM = 0.000001; //error maximo que puede tener la iteracion

private double yc; //Almacena el tirante critico

TirCriSecRec(double Q, double b, double g) {

    this.Q = Q;

    this.b = b;

}

//Calcula el area

public double CalcularArea(double b, double y) {

```

```

    return b * y;
}

// Calcula Perimetro mojado
private double CalcularPm(double b, double yc) {
    return b + 2 * yc;
}

//Calcula el radio hidraulico
public double CalcularRh(double area, double pm) {
    return area / pm;
}

//Calcula Ancho Superficial
public double CalcularT(double b) {
    return b;
}

//Calcula Profundidad Hidraulica

public double CalcularD(double y) {

```

```

    return y;
}

//Iteracion para el Calculo de yc
public void CalcularYc() {

    //Variables locales

    double yc = 100;    //valor inicial para yc

    DecimalFormat formato1 = new DecimalFormat("#0.000000");

    for (int i = 0; i <= 1000000; i++) {

        double FY = (pow((this.b * yc), 3.0) / this.b) - (pow(this.Q, 2.0) / 9.81);

        if (FY == 0) {

            break;

        } else {

            double FDY = 3.0 * pow(this.b, 2.0) * pow(yc, 2.0);

            double ER = FY / FDY;

            yc = yc - ER;

            if (abs(ER) <= errM) {

```

```

        break;
    }

    System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyc: "
        + formato1.format(yc));
    }

}

//Si no hay ninguna falla añadir los valores a los miembros de clase
this.yc = yc;
}

//Funcion para calcular los parametros
public void CalcularValores() {
    this.A = CalcularArea(this.b, this.yc);
    this.P = CalcularPm(this.b, this.yc);
    this.Rh = CalcularRh(this.A, this.P);
    this.T = CalcularT(this.b);
    this.D = this.yc;

    this.Vel = new Velocidad(this.Q, this.A);
    this.EE = new EEspecifica(this.yc, Vel.getVelocidad(), 9.81);
    this.Fro = new Froude(Vel.getVelocidad(), this.D);
}

```

```
public double getYc() {  
    return yc;  
}
```

```
public double getA() {  
    return A;  
}
```

```
public double getP() {  
    return P;  
}
```

```
public double getRh() {  
    return Rh;  
}
```

```
public double getT() {  
    return T;  
}
```

```
public double getD() {  
    return D;  
}
```

```

public double getVel() {
    return Vel.getVelocidad();
}

public double getEE() {
    return EE.getEnergia();
}

public String getFro() {
    return Fro.getTipo();
}
}

```

2.1.4.2 SECCIÓN TRAPEZOIDAL

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package tirantecritico;

import static java.lang.Math.abs;

import static java.lang.Math.pow;

import static java.lang.Math.sqrt;

```



```

import java.text.DecimalFormat;

public class TirCriSecTrap {

    private double Q; //Almacena caudal

    private double b; //Almacena base

    private double z; //Almacena inclinacion de talud

    private double A; //Area

    private double P; //Perimetro Mojado

    private double Rh; //Radio Hidraulico

    private double T; //Ancho Superficial

    private double D; //Profundidad Hidraulica

    private Velocidad Vel;

    private EEspecifica EE;

    private Froude Fro;

    //Valores de configuracion

    private final double errM = 0.000001; //error maximo que puede tener la iteraccion

    private double yc; // Almacena el tirante critico

    TirCriSecTrap(double Q, double b, double z, double g) {

```

```

    this.Q = Q;

    this.b = b;

    this.z = z;

}

//Calcular el area

public double CalcularArea(double b, double y, double z) {

    return (b + z * y) * y;

}

// Calcular perimetro mojado

private double CalcularPm(double b, double y, double z) {

    return b + 2 * y * (sqrt(1 + pow(z, 2.0)));

}

//Calcular el radio hidraulico

public double CalcularRh(double area, double pm) {

    return area / pm;

}

```

```

//Calcular el ancho superficial

public double CalcularT(double b, double y, double z) {

    return (b + 2 * z * y);

}

//Calcular la profundidad hidraulica

public double CalcularD(double area, double t) {

    return area / t;

}

//Iteracion Calculo yn

public void CalcularYc() {

    //Variables locales

    double yc = 100; //valor inicial para yn

    DecimalFormat formato1 = new DecimalFormat("#0.000000");

    for (int i = 0; i <= 1000000; i++) {

        double FY = (pow(((b + z * yc) * yc), 3.0) / (b + 2 * z * yc)) - (pow(this.Q, 2.0) / 9.81);

        if (FY == 0) {

```

```

        break;

    } else {

        double FDY1 = (3.0 * pow(yc, 2.0) * pow((b + yc * z), 3.0)) / (b + 2.0 * yc * z);

        double FDY2 = (3.0 * pow(yc, 3.0) * z * pow((b + yc * z), 2.0)) / (b + 2.0 * yc * z);

        double FDY3 = (2.0 * pow(yc, 3.0) * z * pow((b + yc * z), 3.0)) / pow((b + 2.0 * yc *
z), 2.0);

        double FDY = FDY1 + FDY2 - FDY3;

        double ER = FY / FDY;

        yc = yc - ER;

        if (abs(ER) <= errM) {

            break;

        }

        System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyc: "

            + formato1.format(yc));

    }

}

//Si no hay ninguna falla añadir los valores a los miembros de clase

this.yc = yc;

}

```

```

//Funcion para calcular los parametros

public void CalcularValores() {

    this.A = CalcularArea(this.b, this.yc, this.z);

    this.P = CalcularPm(this.b, this.yc, this.z);

    this.Rh = CalcularRh(A, P);

    this.T = CalcularT(this.b, this.yc, this.z);

    this.D = CalcularD(A, T);

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.yc, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.D);

}

public double getYc() {

    return yc;

}

public double getA() {

    return A;

}

public double getP() {

    return P;

```

```
}
```

```
public double getRh() {
```

```
    return Rh;
```

```
}
```

```
public double getT() {
```

```
    return T;
```

```
}
```

```
public double getD() {
```

```
    return D;
```

```
}
```

```
public double getVel() {
```

```
    return Vel.getVelocidad();
```

```
}
```

```
public double getEE() {
```

```
    return EE.getEnergia();
```

```
}
```

```
public String getFro() {
```

```
    return Fro.getTipo();
```

```
}
```

```
}
```

2.1.4.3 SECCIÓN TRIANGULAR

```
/*
```

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

```
*/
```

```
package tirantecritico;
```

```
import static java.lang.Math.abs;
```

```
import static java.lang.Math.pow;
```

```
import static java.lang.Math.sqrt;
```

```
import java.text.DecimalFormat;
```

```
public class TirCriSecTri {
```

```
    //Variables
```

```
    private double Q; //Almacena Caudal
```

```
    private double z; //Almacena la inclinacion del Talud
```

```
    private double A; //Area
```

```
    private double P; //Perimetro Mojado
```

```
    private double Rh; //Radio Hidraulico
```

```

private double T; //Ancho Superficial

private double D; //Profundidad Hidraulica

private Velocidad Vel;

private EEspecifica EE;

private Froude Fro;

//Valores de Configuracion

private final double errM = 0.001; //Error maximo que puede tener la iteracion

private double yc;

TirCriSecTri(double Q, double z, double g) {

    this.Q = Q;

    this.z = z;

}

//Calcula el area

public double CalcularArea(double z, double y) {

    return z * (y * y);

}

// Calcula Perimetro mojado

```



```

private double CalcularPm(double y, double z) {

    return (2 * y) * (sqrt(1 + pow(z, 2.0)));

}

//Calcula el radio hidraulico

public double CalcularRh(double area, double pm) {

    return area / pm;

}

//Calcula Ancho Superficial

public double CalcularT(double z, double y) {

    return 2 * z * y;

}

//Calcula Profundidad Hidraulica

public double CalcularD(double y) {

    return y / 2;

}

//Iteracion para el Calculo de yc

```

```

public void CalcularYc() {

    //Variables locales

    double yc = 100;    //valor inicial para yc

    DecimalFormat formato1 = new DecimalFormat("#0.000000");

    for (int i = 0; i <= 1000000; i++) {

        double FY = (pow((z * pow(yc, 2.0)), 3.0) / (2.0 * z * yc)) - (pow(this.Q, 2.0) / 9.81);

        if (FY == 0) {

            break;

        } else {

            double FDY = (5.0 * pow(yc, 4.0) * pow(z, 2.0)) / 2.0;

            double ER = FY / FDY;

            yc = yc - ER;

            if (abs(ER) <= errM) {

                break;

            }

        }

    }

}

```

```

        System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyc: "
            + formato1.format(yc));
    }

}

//Si no hay ninguna falla añadir los valores a los miembros de clase
this.yc = yc;
}

//Funcion para calcular los parametros
public void CalcularValores() {
    this.A = CalcularArea(this.z, this.yc);
    this.P = CalcularPm(yc, this.z);
    this.Rh = CalcularRh(A, P);
    this.T = CalcularT(this.z, yc);
    this.D = CalcularD(yc);

    this.Vel = new Velocidad(this.Q, this.A);
    this.EE = new EEspecifica(this.yc, Vel.getVelocidad(), 9.81);
    this.Fro = new Froude(Vel.getVelocidad(), this.D);
}

public double getYc() {
    return yc;
}

```

```
}
```

```
public double getA() {
```

```
    return A;
```

```
}
```

```
public double getP() {
```

```
    return P;
```

```
}
```

```
public double getRh() {
```

```
    return Rh;
```

```
}
```

```
public double getT() {
```

```
    return T;
```

```
}
```

```
public double getD() {
```

```
    return D;
```

```
}
```

```
public double getVel() {
```

```
    return Vel.getVelocidad();
```

```

    }

    public double getEE() {
        return EE.getEnergia();
    }

    public String getFro() {
        return Fro.getTipo();
    }

}

```

2.1.4.4 SECCIÓN CIRCULAR

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package tirantecritico;

import static java.lang.Math.PI;

import static java.lang.Math.abs;

import static java.lang.Math.atan;

import static java.lang.Math.cos;

import static java.lang.Math.pow;

```

```

import static java.lang.Math.sin;

import static java.lang.Math.sqrt;

import static java.lang.Math.tan;

import java.text.DecimalFormat;

public class TirCriSecCir {

    //Variables

    private double Q; //Almacena Caudal

    private double Do; //Almacena el diametro

    private double A; //Area

    private double P; //Perimetro Mojado

    private double Rh; //Radio Hidraulico

    private double T; //Ancho Superficial

    private double D; //Profundidad Hidraulica

    private Velocidad Vel;

    private EEspecifica EE;

    private Froude Fro;

    //-----Valores de Configuracion

    private final double errM = 0.000001; //Error maximo que puede tener la iteraccion

```

```

private double yc;

private double alfaRad;

private double alfaDeg;

TirCriSecCir(double Q, double Do, double g) {

    this.Q = Q;

    this.Do = Do;

}

//Calcula el area

public double CalcularArea(double y) {

    return ((y - sin(y)) * pow(this.Do, 2.0)) * (1.0 / 8.0);

}

// Calcula Perimetro mojado

private double CalcularPm(double y) {

    return (y * this.Do) / 2;

}

//Calcula el radio hidraulico

public double CalcularRh(double y) {

```

```

        return ((1 - (sin(y) / y)) * this.Do) / 4;
    }

    //Calcula el Ancho Superficial
    public double CalcularT(double y) {

        return (sin(y / 2)) * this.Do;
    }

    //Calcula Profundidad Hidraulica

    public double CalcularD(double a, double t) {

        return a / t;
    }

    //Iteracion Calculo yn
    public void CalcularYc() {

        //Variables locales

        double yc = PI;    //Valor inicial para yc

        DecimalFormat formato1 = new DecimalFormat("#0.000000");

        for (int i = 0; i <= 1000000; i++) {

```



```
double FY = (pow(((1.0 / 8.0) * (yc - sin(yc)) * pow(this.Do, 2.0)), 3.0) / (this.Do *  
sin(yc / 2.0))) - (pow(this.Q, 2.0) / 9.81);
```

```
if (FY == 0) {
```

```
    break;
```

```
} else {
```

```
    double FDY1 = (pow(this.Do, 5.0) * cos(yc / 2.0) * pow((yc - sin(yc)), 3.0)) / (1024 *  
pow(sin(yc / 2.0), 2.0));
```

```
    double FDY2 = (3.0 * pow(this.Do, 5.0) * pow((yc - sin(yc)), 2.0) * (cos(yc) - 1.0)) /  
(512 * sin(yc / 2.0));
```

```
    double FDY = -FDY1 - FDY2;
```

```
    double ER = FY / FDY;
```

```
    yc = yc - ER;
```

```
    if (abs(ER) <= errM) {
```

```
        break;
```

```
    }
```

```
    System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyc: "
```

```
        + formato1.format(yc));
```

```
    }
```

```
}
```

```

//Si no hay ninguna falla añadir los valores a los miembros de clase

this.yc = (this.Do / 2.0) * (1.0 - cos(yc / 2.0));

this.alfaRad = yc;
}

//Funcion para calcular los parametros

public void CalcularValores() {

    this.A = CalcularArea(this.alfaRad);

    this.P = CalcularPm(this.alfaRad);

    this.Rh = A / P;

    this.T = CalcularT(this.alfaRad);

    this.D = A / T;

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.yc, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.D);

}

public double getYc() {

    return yc;

}

public double getA() {

    return A;
}

```

```
}
```

```
public double getP() {
```

```
    return P;
```

```
}
```

```
public double getRh() {
```

```
    return Rh;
```

```
}
```

```
public double getT() {
```

```
    return T;
```

```
}
```

```
public double getD() {
```

```
    return D;
```

```
}
```

```
public double getVel() {
```

```
    return Vel.getVelocidad();
```

```
}
```

```
public double getEE() {
```

```
    return EE.getEnergia();
```

```
}

public String getFro() {
    return Fro.getTipo();
}
}
```

2.1.4.5 SECCIÓN PARABÓLICA CON “T” CONOCIDO

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package tirantecritico;

import static java.lang.Math.abs;
import static java.lang.Math.log;
import static java.lang.Math.pow;
import static java.lang.Math.sqrt;
import java.text.DecimalFormat;

/**
 *
 * @author User
 */
```

```

public class TirCriSecParT {

    //Variables

    private double Q; //Almacena Caudal

    private double A; //Area

    private double P; //Perimetro Mojado

    private double Rh; //Radio Hidraulico

    private double T; //Ancho Superficial

    private double D; //Profundidad Hidraulica

    private Velocidad Vel;

    private EEspecifica EE;

    private Froude Fro;

    private FuncionObjetivoTCritico mFuncionObjetivoTCritico;

    //Valores de Configuracion

    private final double errM = 0.00001; //Error maximo que puede tener la iteracion

    private final double ap = 0.001; // Valor de la apertura (valor que varia)

    private double yc;

    TirCriSecParT(double Q, double T, double g) {

```

```

this.Q = Q;

this.T = T;

    mFuncionObjetivoTCritico = new FuncionObjetivoTCritico(Q, g);
}

//Calcula el Area

public double CalcularArea(double t1, double y) {

    return (2.0 * t1 * y) / 3.0;
}

//Calcular el perimetro mojado

public double CalcularPm(double T, double yn) {

    double pm;

    double x = (4.0 * yn) / T;

    double aux0 = sqrt(1.0 + pow(x, 2.0));

    double aux1 = log(x + aux0);

    return pm = (T / 2.0) * (aux0 + aux1 / x);
}

//Calcular el radio hidraulico

public double CalcularRh(double area, double pm) {

```

```

    return area / pm;
}

//Calcula Profundidad Hidraulica
public double CalcularD(double y) {

    return 2.0 * y / 3;
}

//Iteracion para el calculo de yc
public void CalcularYc() {

    //Variables locales

    double yc = 100;    //valor inicial para yc

    DecimalFormat formato1 = new DecimalFormat("#0.000000");

    for (int i = 0; i <= 1000000; i++) {

        double FY = pow(((2.0 / 3.0) * T * yc), 3.0) * (1.0 / T) - (pow(this.Q, 2.0) / 9.81);

        if (FY == 0) {

            break;

```

```

    } else {

        double FDY = 2.0 * pow(((2.0 / 3.0) * T * yc), 2.0);

        double ER = FY / FDY;

        yc = yc - ER;

        if (abs(ER) <= errM) {

            break;

        }

        System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyc: "

            + formato1.format(yc));

    }

}

//Si no hay ninguna falla añadir los valores a los miembros de clase

this.yc = yc;

}

//Funcion para calcular los parametros

public void CalcularValores() {

    this.A = CalcularArea(T, this.yc);

    this.P = CalcularPm(this.T, this.yc);

```



```
    this.Rh = CalcularRh(A, P);

    this.D = CalcularD(this.yc);

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.yc, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.D);

}

public double getYc() {

    return yc;

}

public double getA() {

    return A;

}

public double getP() {

    return P;

}

public double getRh() {

    return Rh;

}
```

```

public double getT() {
    return T;
}

public double getD() {
    return D;
}

public double getVel() {
    return Vel.getVelocidad();
}

public double getEE() {
    return EE.getEnergia();
}

public String getFro() {
    return Fro.getTipo();
}

}

```

2.1.4.6 SECCIÓN PARABÓLICA CON “T” NO CONOCIDO

/*

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

*/

```
package tirantecritico;
```

```
import static java.lang.Math.log;
```

```
import static java.lang.Math.pow;
```

```
import static java.lang.Math.sqrt;
```

```
import java.text.DecimalFormat;
```

```
public class TirCriSecPar {
```

```
    //Variables
```

```
    private double Q; //Almacena Caudal
```

```
    private double A; //Area
```

```
    private double P; //Perimetro Mojado
```

```
    private double Rh; //Radio Hidraulico
```

```
    private double T; //Ancho Superficial
```

```
    private double D; //Profundidad Hidraulica
```

```
    private Velocidad Vel;
```

```
    private EEspecifica EE;
```

```
    private Froude Fro;
```

```

private FuncionObjetivoTCritico mFuncionObjetivoTCritico;

//Valores de Configuracion

private final double errM = 0.00001; //Error maximo que puede tener la iteraccion

private final double ap = 0.00001; // Valor de la apertura (valor que varia)

private double yc;

TirCriSecPar(double Q, double g) {

    this.Q = Q;

    mFuncionObjetivoTCritico = new FuncionObjetivoTCritico(Q, g);

}

//Calcula el Area

public double CalcularArea(double t1, double y) {

    return (2.0 * t1 * y) / 3.0;

}

//Calcular el radio hidraulico

public double CalcularRh(double area, double pm) {

```

```

        return area / pm;
    }

    //Calcula Ancho Superficial
    public double CalcularT(double y) {

        return 2 * (sqrt(y));
    }

    //Calcula Profundidad Hidraulica

    public double CalcularD(double y) {

        return 2.0 * y / 3;
    }

    //Iteracion para el calculo de yc
    public void CalcularYc() {

        //Variables locales

        double yc = 0.001; //valor inicial para yn

        double err; //error

        double segTermino; //valor inicial

        //Inicio de las iteraciones

        int cont = 0; //Inicio de contador de iteraciones

```

```

boolean fail = false;

boolean fin = false;

double primerTermino = mFuncionObjetivoTCritico.getObjetivo();

DecimalFormat formato1 = new DecimalFormat("#0.000000");

do {

    //incremento del contador

    cont++; //Incrementar el contador para medir las iteraciones

    double T = 2 * (sqrt(yc));

    double x = (4 * yc) / T;

    double aux0 = sqrt(1.0 + pow(x, 2.0));

    double aux1 = log(x + aux0);

    double Pm = (T / 2.0) * (aux0 + aux1 / x);

    //Calculo del segundo termino

    double area = CalcularArea(T, yc);

    double t = CalcularT(yc);

    segTermino = (pow(area, 1.50)) / (pow(t, 0.50));

    err = primerTermino - segTermino;

    System.out.println("i: " + cont + "\terr: " + formato1.format(err))

```

```
        + "\\tyc: " + formato1.format(yc) + "\\tSegundo termino: " +  
formato1.format(segTermino));
```

```
    if (Math.abs(err) > errM) {
```

```
        yc += ap;
```

```
    } else {
```

```
        fin = true;
```

```
    }
```

```
    this.P = Pm;
```

```
    } while (!fin && err > 0);
```

```
    if (fail == false) {
```

```
        //Si no hay ninguna falla añadir los valores a los miembros de clase
```

```
        this.yc = yc;
```

```
    }
```

```
}
```

```
//Funcion para calcular los parametros
```

```
public void CalcularValores() {
```

```
    this.T = CalcularT(this.yc);
```

```
    this.A = CalcularArea(T, this.yc);
```

```
    this.Rh = CalcularRh(A, P);
```

```
    this.T = CalcularT(this.yc);
```

```

this.D = CalcularD(this.yc);

this.Vel = new Velocidad(this.Q, this.A);

this.EE = new EEspecifica(this.yc, Vel.getVelocidad(), 9.81);

this.Fro = new Froude(Vel.getVelocidad(), this.D);
}

public double getYc() {

    return yc;

}

public double getA() {

    return A;

}

public double getP() {

    return P;

}

public double getRh() {

    return Rh;

}

public double getT() {

```



```

        return T;
    }

    public double getD() {

        return D;
    }

    public double getVel() {

        return Vel.getVelocidad();
    }

    public double getEE() {

        return EE.getEnergia();
    }

    public String getFro() {

        return Fro.getTipo();
    }
}

```

2.1.4.7 SECCIÓN RECTANGULAR CON ESQUINAS REDONDEADAS

/*

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

```

*/

package tirantecritico;

import static java.lang.Math.PI;

import static java.lang.Math.abs;

import static java.lang.Math.pow;

import static java.lang.Math.sqrt;

import java.text.DecimalFormat;

public class TirCriSecRecEsqRed {

    //Variables

    private double Q; //Almacena caudal

    private double b; //Almacena la base

    private double r; //Almacena el radio

    private double D; //Profundidad hidraulica

    private double Rh; //Radio Hidraulico

    private double T; //Ancho superficial

    private double P; //Perimetro mojado

    private double A; //Area

    private Velocidad Vel;

    private EEspecifica EE;

```

```

private Froude Fro;

// Valor de configuracion

private final double errM = 0.000001; //Error maximo que puede tener la iteracion

private double yc; //Almacena el tirante critico

TirCriSecRecEsqRed(double Q, double b, double r, double g) {

    this.Q = Q;

    this.b = b;

    this.r = r;

}

//Calcula el area

public double CalcularArea(double b, double r, double y) {

    double part1 = (PI / 2 - 2) * pow(r, 2.0);

    double part2 = (b + 2 * r) * y;

    return part1 + part2;

}

// Calcula el Perimetro mojado

public double CalcularPm(double r, double b, double y) {

```

```

double aux1 = (PI - 2) * r;

double aux2 = (b + 2 * y);

return aux1 + aux2;

}

//Calcula el radio hidraulico
public double CalcularRh(double area, double pm) {

return area / pm;

}

//Calcula Ancho Superficial
public double CalcularT(double b, double r) {

return b + 2 * r;

}

//Calcula Profundidad Hidraulica

public double CalcularD(double b, double r, double y) {

return (((PI / 2 - 2) * pow(r, 2.0)) / (b + 2 * r)) + y;

}

```

```

//Iteracion para el calculo de yn

public void CalcularYc() {

    //Variables locales

    double yc = 100;    //valor inicial para yn

    //Inicio de las iteraciones

    DecimalFormat formato1 = new DecimalFormat("#0.000000");

    for (int i = 0; i <= 1000000; i++) {

        double a0 = (((PI / 2.0) - 2.0) * pow(r, 2.0)) + ((b + 2 * r) * yc);

        double t0 = b + 2 * r;

        double FY = (pow(a0, 3.0) / t0) - (pow(Q, 2.0) / 9.81);

        if (FY == 0) {

            break;

        } else {

            double FDY = 3 * pow(a0, 2.0);

            double ER = FY / FDY;

```

```

        yc = yc - ER;

        if (abs(ER) <= errM) {

            break;

        }

        System.out.println("i: " + i + "\tER: " + formato1.format(ER) + "\tyn: "
            + formato1.format(yc));

    }

}

//Si no hay ninguna falla añadir los valores a los miembros de clase

this.yc = yc;

}

//Funcion para calcular los parametros

public void CalcularValores() {

    this.A = CalcularArea(b, r, this.yc);

    this.P = CalcularPm(r, b, this.yc);

    this.Rh = CalcularRh(this.A, this.P);

    this.T = CalcularT(b, r);

    this.D = CalcularD(this.b, this.r, this.yc);

    this.Vel = new Velocidad(this.Q, this.A);

```

```
    this.EE = new EEspecifica(this.yc, Vel.getVelocidad(), 9.81);  
    this.Fro = new Froude(Vel.getVelocidad(), this.D);  
}
```

```
public double getYc() {  
    return yc;  
}
```

```
public double getA() {  
    return A;  
}
```

```
public double getD() {  
    return D;  
}
```

```
public double getRh() {  
    return Rh;  
}
```

```
public double getT() {  
    return T;  
}
```

```

public double getP() {
    return P;
}

public double getVel() {
    return Vel.getVelocidad();
}

public double getEE() {
    return EE.getEnergia();
}

public String getFro() {
    return Fro.getTipo();
}
}

```

2.1.4.8 SECCIÓN TRIANGULAR CON FONDO REDONDEADO

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package tirantecritico;

```



```
import static java.lang.Math.PI;

import static java.lang.Math.abs;

import static java.lang.Math.atan;

import static java.lang.Math.cos;

import static java.lang.Math.pow;

import static java.lang.Math.sin;

import static java.lang.Math.sqrt;

import static java.lang.Math.tan;

import java.text.DecimalFormat;

public class TirCriSecTriFonRed {

    private double Q; //Almacena caudal

    private double r; //Almacena el radio

    private double z; //Almacena inclinacion talud

    private double g;

    private double A; //Area

    private double P; //Perimetro Mojado

    private double Rh; //Radio Hidraulico

    private double T; //Ancho Superficial

    private double D; //Profundidad Hidraulica
```

```

private Velocidad Vel;

private EEspecifica EE;

private Froude Fro;

//Valores de configuracion

private final double errM = 0.000001; //error maximo que puede tener la iteracion

private double yc; // Tirante critico

TirCriSecTriFonRed(double Q, double r, double z, double g) {

    this.Q = Q;

    this.r = r;

    this.z = z;

    this.g = g;

}

//Calcular el area

public double CalcularArea(double r, double T, double z) {

    double ar = (((pow(T, 2.0)) / (4 * z)) - (((pow(r, 2.0)) / z) * (1 - (z * acot(z)))));

    return ar;

}

// Calcular perimetro mojado

```

```

private double CalcularPm(double T, double r, double z) {

    double pm1 = ((T / z) * sqrt(1 + pow(z, 2.0))) - (((2 * r) / z) * (1 - (z * acot(z))));

    return pm1;

}

```

//Calcular el radio hidraulico

```

public double CalcularRh(double ar, double pm) {

    return ar / pm;

}

```

//Calcular el ancho superficial

```

public double CalcularT(double r, double z, double yc) {

    double t = 2 * (z * (yc - r) + r * (sqrt(1 + pow(z, 2.0))));

    return t;

}

```

//Calcular la profundidad hidraulica

```

public double CalcularD(double ar, double t) {

    return ar / t;

}

```

```

}

public double acot(double x) {
    return PI / 2.0 - atan(x);
}

//Iteracion Calculo yn
public void CalcularYc() {

    //Variables locales

    double y = 100; //valor inicial para yn

    double FY;

    double FDY;

    double FDY1;

    double FDY2;

    DecimalFormat formato1 = new DecimalFormat("#0.000");

    //Inicio de las iteraciones
    for (int i = 0; i <= 1000000; i++) {

        FY = (pow(((pow(y, 2.0)) / (4 * z)) - (((pow(r, 2.0)) / z) * (1 - (z * acot(z))))), 3.0) / y) -
        (pow(Q, 2.0) / g);
    }
}

```

```

if (FY == 0) {
    break;

} else {

    FDY1 = 3 * pow((((z * acot(z) - 1) * pow(r, 2.0)) / z + pow(y, 2.0) / (4 * z)), 2.0) / (2 *
z);

    FDY2 = pow((((z * acot(z) - 1) * pow(r, 2.0)) / z + pow(y, 2.0) / (4 * z)), 3.0) / pow(y,
2.0);

    FDY = FDY1 - FDY2;

    double ER = FY / FDY;

    y = y - ER;

    if (abs(ER) <= errM) {
        break;
    }

    System.out.println("i: " + i
        + "\tFy: " + formato1.format(FY) + "\tFDY: " + formato1.format(FDY)
        + "\tER: " + formato1.format(ER) + "\ty: " + formato1.format(y));
    }

}

```

```

T = y;

double w = atan(1 / z);

double w1 = PI - (2 * w);

double B = ((2 * PI) - (2 * w1)) / 2;

double Y2 = r * (1 - cos(B / 2));

double M = 2 * r * sin(B / 2);

double C = (T - M) / 2;

double Y1 = tan(w) * C;

this.yc = Y1 + Y2;

}

//Funcion para calcular los parametros

public void CalcularValores() {

    this.A = CalcularArea(r, T, z);

    this.P = CalcularPm(T, r, z);

    this.Rh = CalcularRh(A, P);

    this.T = CalcularT(r, z, yc);

    this.D = CalcularD(A, T);

    this.Vel = new Velocidad(this.Q, this.A);

    this.EE = new EEspecifica(this.yc, Vel.getVelocidad(), 9.81);

    this.Fro = new Froude(Vel.getVelocidad(), this.D);

```

```
}
```

```
public double getYc() {
```

```
    return yc;
```

```
}
```

```
public double getA() {
```

```
    return A;
```

```
}
```

```
public double getP() {
```

```
    return P;
```

```
}
```

```
public double getRh() {
```

```
    return Rh;
```

```
}
```

```
public double getT() {
```

```
    return T;
```

```
}
```

```
public double getD() {
```

```
    return D;
```

```
}


public double getVel() {
    return Vel.getVelocidad();
}


public double getEE() {
    return EE.getEnergia();
}

public String getFro() {
    return Fro.getTipo();
}
}
```


2.2. TUTORIAL PARA SUBIR UNA APP A LA PLAY STORE

- ❖ Crear una cuenta en **Gmail** para poder acceder a la **Google Play Console**
- ❖ Crear una cuenta en la **Google Play Console** como se muestra a continuación.


 Google Play Console

 alextipan1580@gmail.com ▾

Crear una nueva cuenta de desarrollador

La Cuenta de Google seleccionada será propietaria de esta nueva cuenta de desarrollador. Si quieres unirte a un desarrollador existente, solicita una invitación del administrador.

Si representas a una organización, te recomendamos que no uses cuentas personales para configurar cuentas de desarrollador. Puedes configurar Cuentas de Google con cualquier dirección existente de correo electrónico. [Más información](#)

 Para crear una cuenta, deberás pagar una tarifa de registro única de USD 25. Es posible que se te solicite que verifiques tu identidad con un ID válido para completar el registro de la cuenta. Si no logramos verificar tu identidad, no se reembolsará la tarifa de registro.

Nombre público del desarrollador *	<input type="text" value="Alex Tipán"/>
	Los usuarios podrán verlo en Google Play. 10/50
Dirección secundaria de correo electrónico de contacto *	<input type="text" value="construcadtipan@gmail.com"/>
	Es posible que lo usemos para comunicarnos contigo, además del correo electrónico asociado con tu Cuenta de Google. No estará visible para los demás usuarios de Google Play.
Número de teléfono de contacto *	<input type="text" value="+593988828202"/>
	Incluye el signo más (+), el código de país y el código de área. Es posible que lo usemos para comunicarnos contigo, pero no estará visible para los usuarios en Google Play.
Acuerdo para Desarrolladores y Condiciones del Servicio *	<input checked="" type="checkbox"/> Confirmando que he leído y acepto el Acuerdo de Distribución para Desarrolladores de Google Play . Acepto asociar mi Cuenta de Google con el Acuerdo de Distribución para Desarrolladores de Google Play y confirmo que tengo al menos 18 años de edad.
	<input checked="" type="checkbox"/> Confirmando que he leído y acepto las Condiciones del Servicio de Google Play Console . Acepto asociar mi Cuenta de Google con las Condiciones del Servicio de Google Play Console.

- ❖ Ingresar el número de la tarjeta de **Credito o Debito** en la cual se descontara el valor de \$25 dolares Americanos, y seguir con el proceso de registro.

The image shows a payment modal titled "Finaliza la compra" (Finalize purchase) overlaid on a registration page. The modal displays a "Developer Registration Fee" of USD 25.00. Below this, there is a section for adding a credit or debit card. The card number field is partially filled with "438" and has a "VISA" logo. The name of the cardholder is "Alex Tipán". The location is set to "Provincia de Cotopaxi, Cantón Latacunga, Parroquia ...". At the bottom, there is a "Comprar" (Buy) button and a disclaimer about Google Payments terms and privacy policy.

alextipan1580@gmail.com

Crear una nueva cuenta de desarrollador

La Cuenta de Google seleccionada será propietaria de esta nueva cuenta de desarrollador. Si quieres unirte a un desarrollador existente, solicita una invitación del administrador.

Si representas a una organización, te recomendamos que no uses cuentas personales para configurar cuentas de desarrollador. Puedes configurar Cuentas de Google con cualquier dirección existente de correo electrónico. [Más información](#)

Finaliza la compra

Developer Registration Fee USD 25.00

Agregar tarjeta de crédito o débito

Número de tarjeta: 438 [] VISA MM AA CVC 1 []

Nombre del titular: Alex Tipán

Provincia de Cotopaxi, Cantón Latacunga, Parroquia ...

Si continúas, aceptas las [Condiciones del Servicio](#) de Google Payments. En el [Aviso de privacidad](#) se describe cómo se usan tus datos.

Comprar

- ❖ Ingresar la dirección y el código postal de su localidad y dar clic en **Comprar**

unirte a un desarrollador existente, solicita una invitación del administrador.

Si representas cuentas de correo electrónico de un desarrollador existente de configuración de cuentas de correo electrónico.

Nombre público del desarrollador

Dirección de correo electrónico *

Número de contacto *



Acuerdo para y Condiciones

Finaliza la compra

Developer Registration Fee USD 25.00

Agregar tarjeta de crédito o débito

Visa

  Ecuador (EC)

Línea 1 de la dirección
Provincia de Cotopaxi, Cantón Latacunga

Línea 2 de la dirección
Parroquia Guaytacama, Barrio San Sebastian

Código postal
050153

Ciudad
Latacunga

Si continúas, aceptas las [Condiciones del Servicio](#) de Google Payments. En el [Aviso de privacidad](#) se describe cómo se usan tus datos.

Comprar

- ❖ A su correo **Gmail** le llegara un mensaje en donde menciona que su registro a sido exitoso como tambien el costo por el registro en **Google Play Console**.

Google

6 de enero de 2021 18:36:11 GMT-8

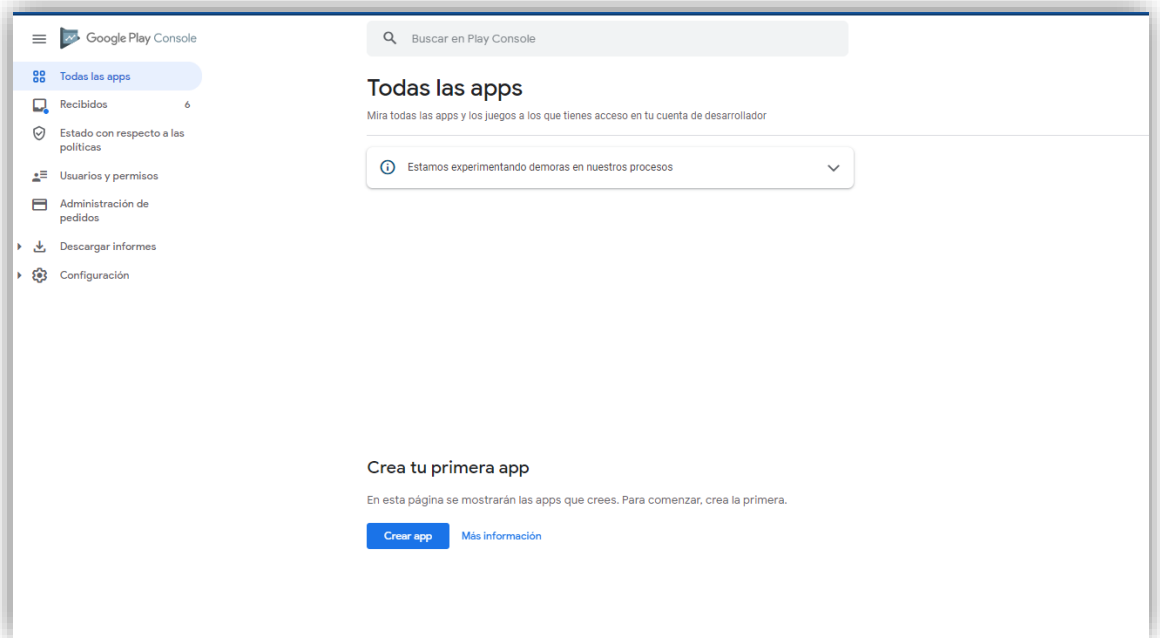
Artículo	Cantidad	Precio
Developer Registration Fee	1	USD 25.00
		Impuesto USD 0.00
Total		USD 25.00

Forma de pago
V

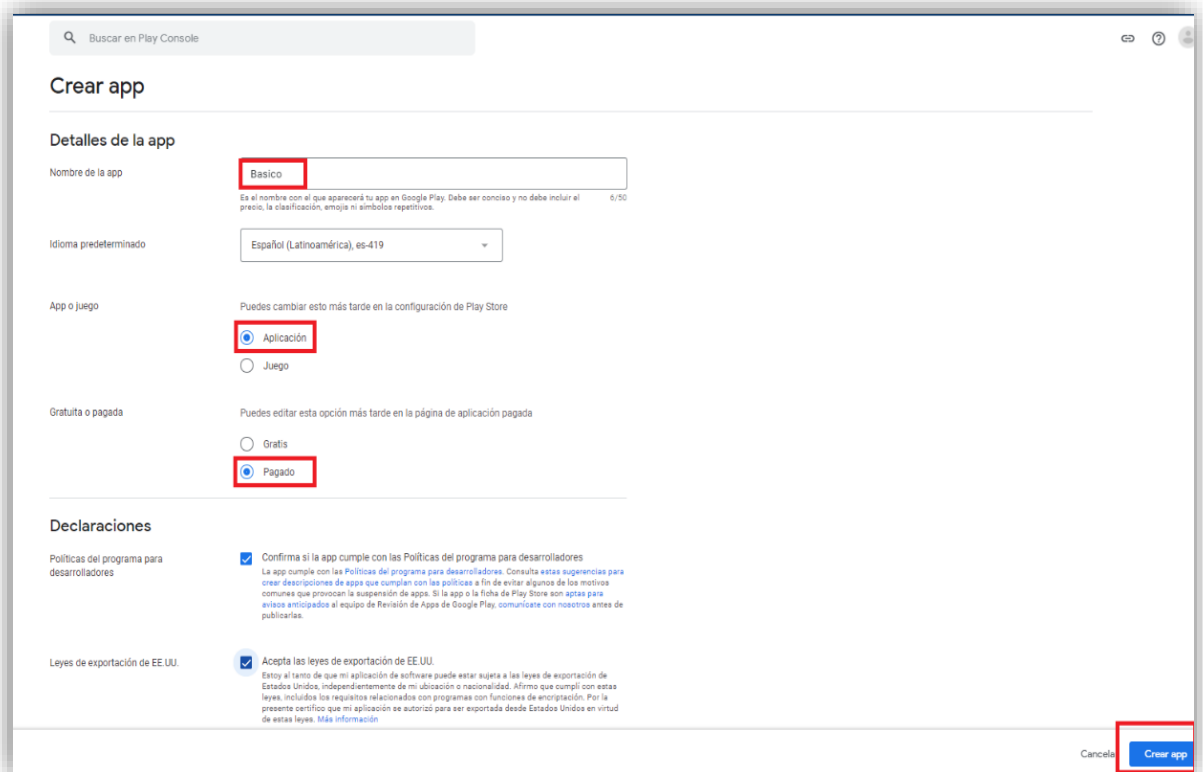
Número de pedido de Google
PDS.5887-2358-7808-49963

[DETALLES DE LA TRANSACCIÓN](#)

- ❖ La plataforma **Google Play Console** estará lista para el registro de datos
- ❖ Dar clic en **Crear app**.



- ❖ Llenar los requerimientos y detalles para la aplicación y dar clic en **Crear app**.



- ❖ En la sección **Crecimiento** damos clic en **Ficha de Play Store principal** y llenamos los detalles de la aplicación.

Ficha de Play Store principal

Edita el nombre, el ícono, las capturas de pantalla y otros elementos de la app para presentar el aspecto que tendrá la app para los usuarios en Google Play. [Mostrar más](#)

Predefinido: Español (Latinoamérica), es-419 [Administrar traducciones](#)

* Campos obligatorios. Ingresar todos los campos en Español (Latinoamérica), es-419

Detalles de la app

Para evitar problemas comunes con la ficha de Play Store, consulta la [Política de Metadatos](#). Antes de enviar tu app, revisa todas las [Políticas del Programa](#). Si eres apto para [proporcionar avisos anticipados](#) al equipo de Revisión de apps de Google Play, comunícate con nosotros antes de publicar la ficha de Play Store.

Nombre de la app * 6/50
Es el nombre con el que aparecerá tu app en Google Play. Debe ser conciso y no debe incluir el precio, la clasificación, emojis ni símbolos repetitivos.

Descripción breve * 24/80
Una descripción breve de tu app. Los usuarios pueden expandirla para ver la descripción completa.

Descripción completa * 28/4000

- ❖ Cargar imágenes de la aplicación con sus requerimientos de tamaño y dar clic en **Guardar**.

Gráficos

Antes de subir gráficos nuevos, revisa la [Política de Robo de Identidad y Propiedad Intelectual](#). Si agregas traducciones a la ficha de Play Store, pero no incluyes gráficos localizados, se usarán los del idioma predeterminado.

Ícono de la app *



Es el ícono de la app como se muestra en Google Play. No uses insignias ni texto que sugiera la clasificación de la tienda, el precio o las categorías de Google Play (por ejemplo, "popular", "oferta" o "nueva").

JPEG o PNG de 32 bits
512 px por 512 px
Hasta 1 MB

Gráfico de funciones *



- ❖ Realizar capturas del entorno de la pantalla de su aplicación con el formato requerido.

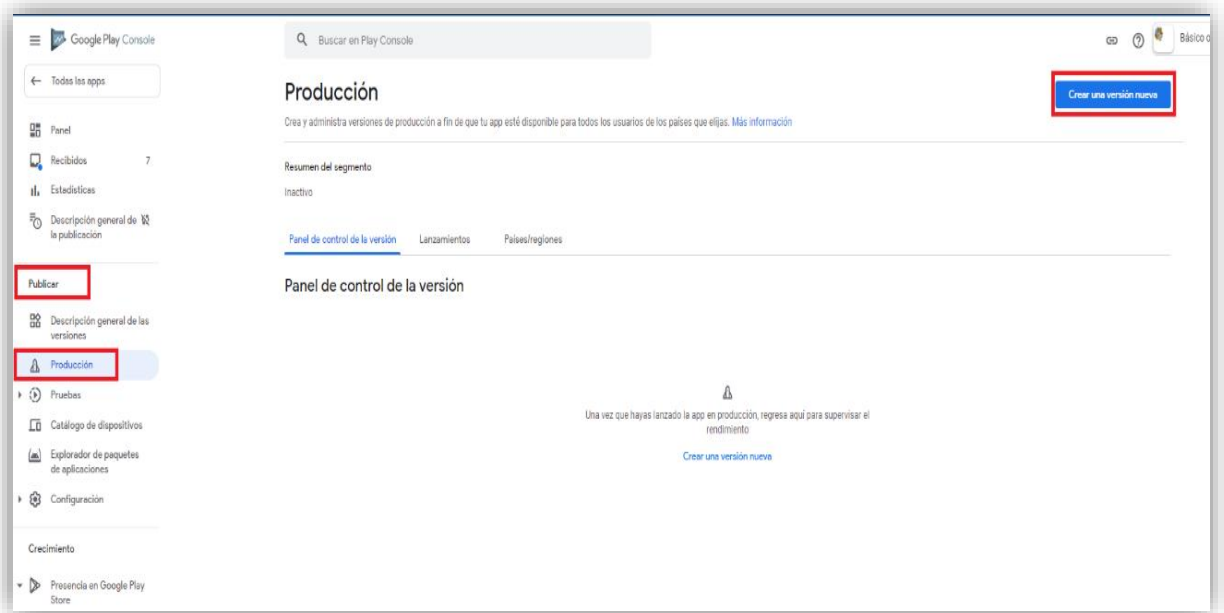
Capturas de pantalla de teléfono *



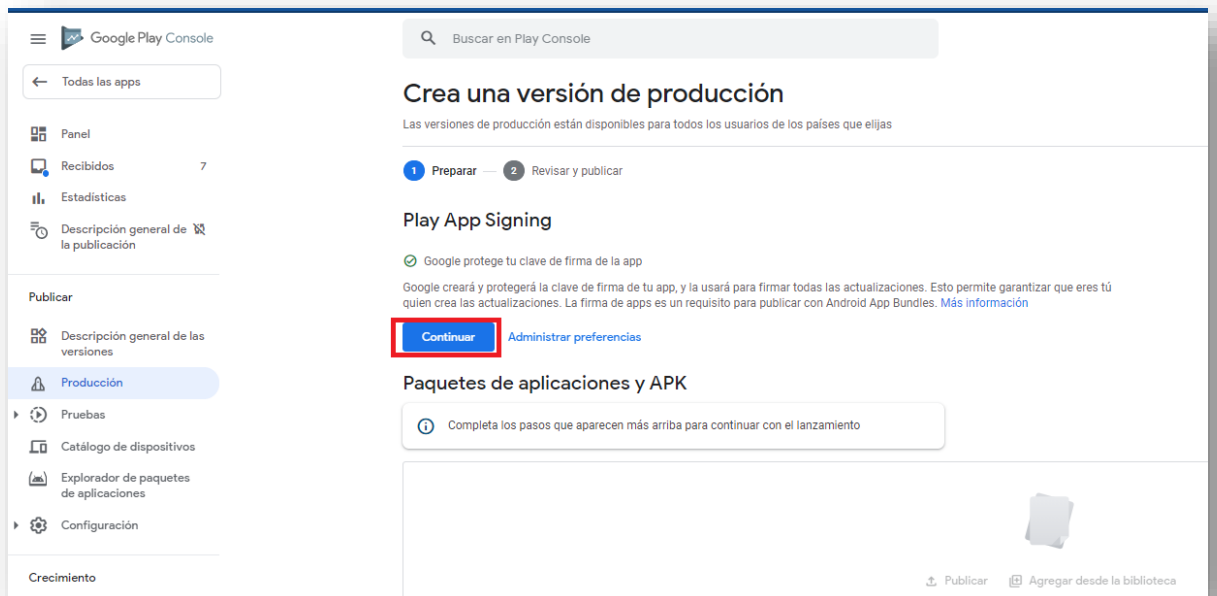
Carga de 2 a 8 capturas de pantalla de teléfonos.

JPEG o PNG de 24 bits (no transparente)
Entre 320 y 3,840 píxeles
Relación de aspecto de 16:9 (para capturas de pantalla horizontales)
Hasta 8 MB

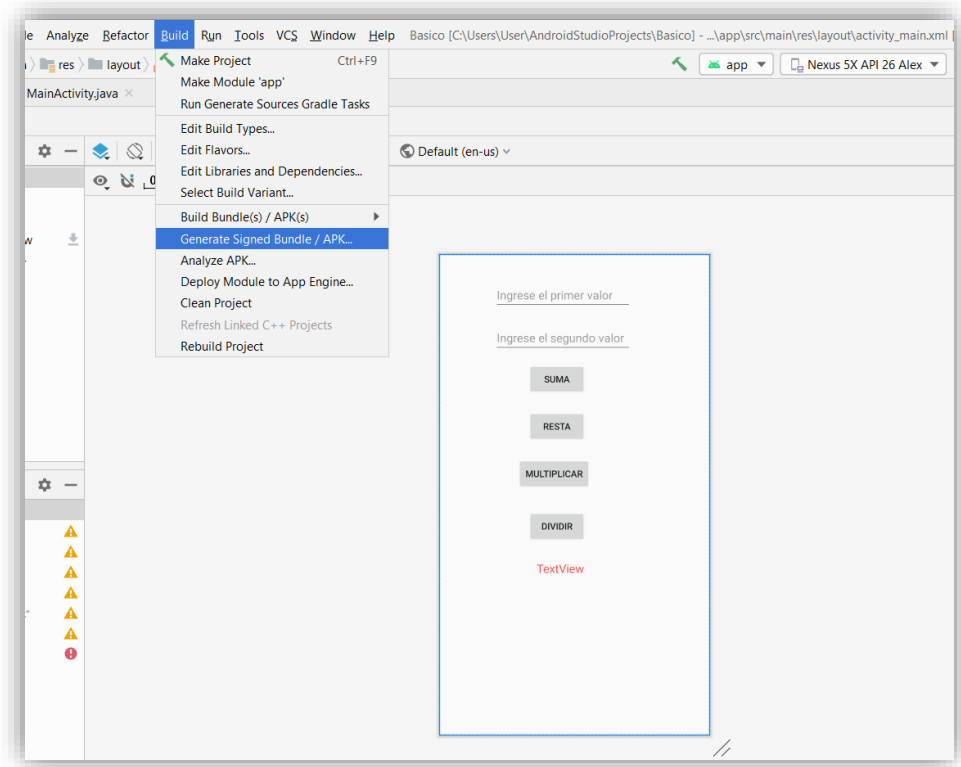
- ❖ Para cargar la versión de la aplicación (APK) seguir esta secuencia **Aplicar - Producción – Crear Una Nueva Versión**

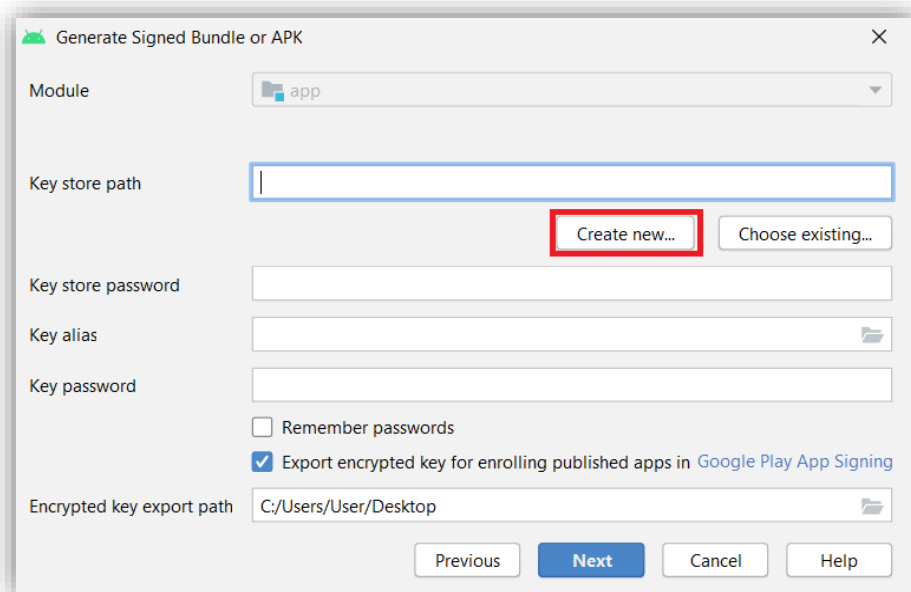
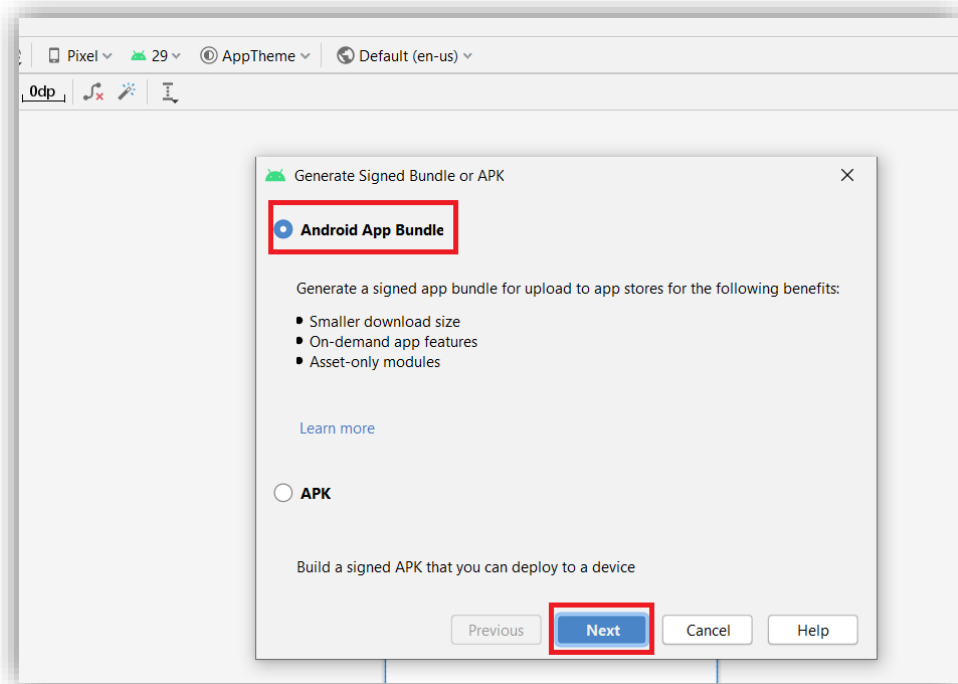


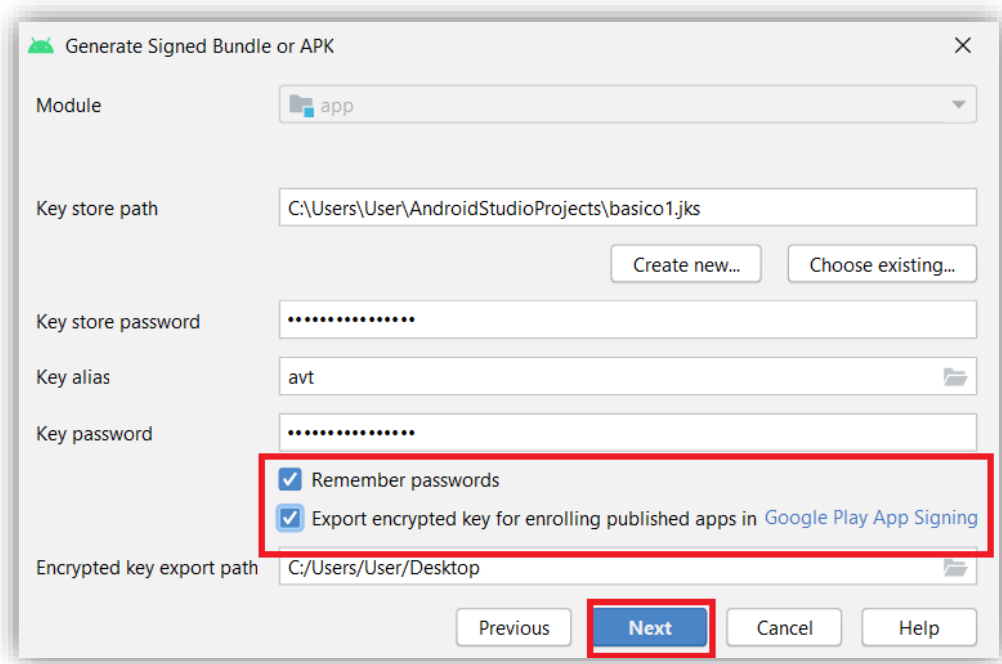
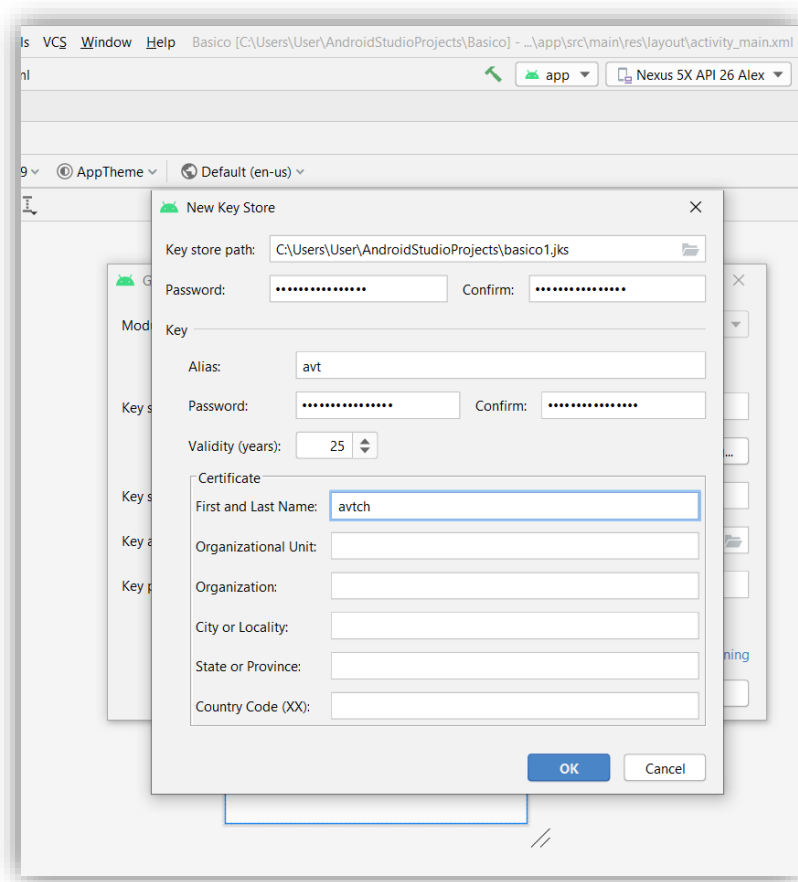
- ❖ Dar clic en **Continuar** para actualizar la versión cada vez que se suba la aplicación.

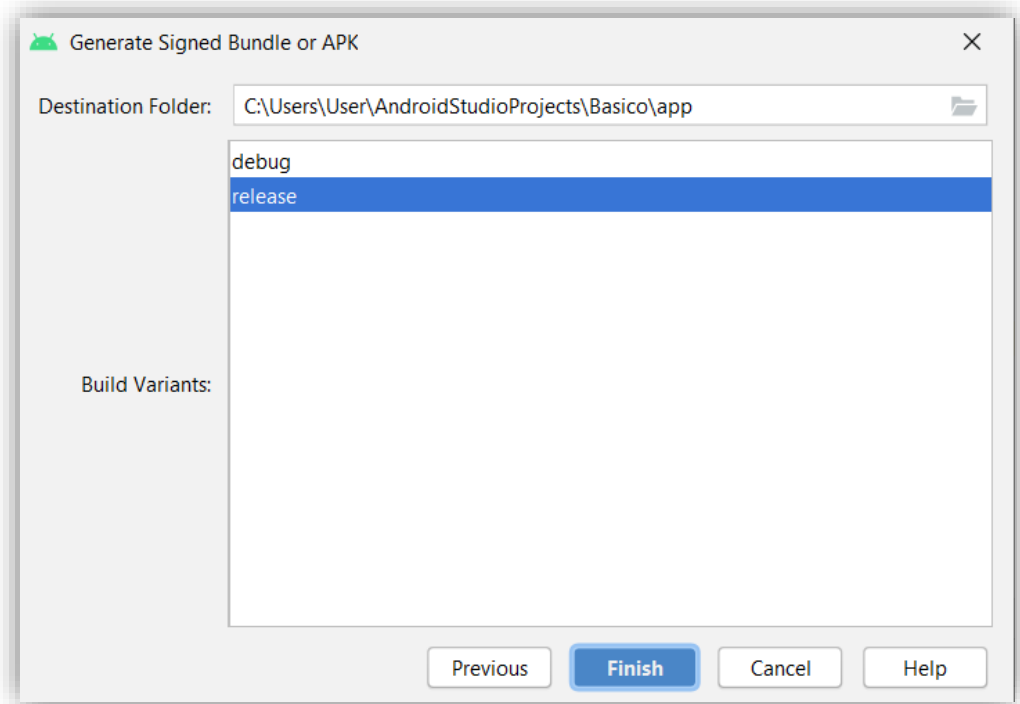


- ❖ Subir el fichero de nuestra aplicación ingresando al **Software Android Studio** como se muestra a continuación.

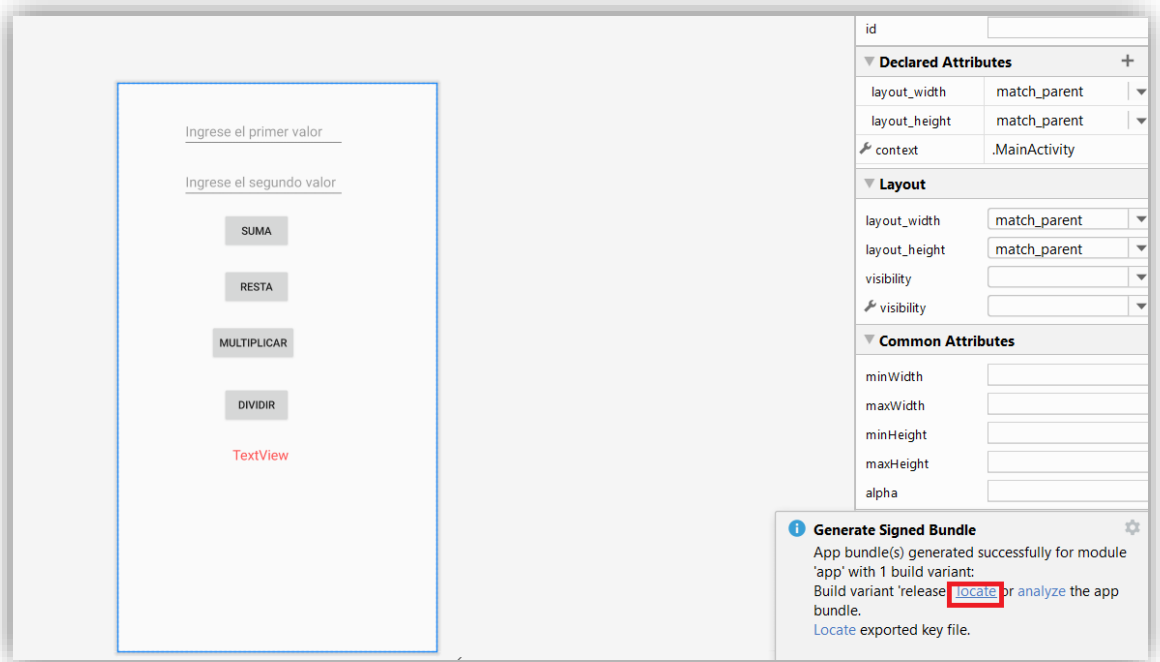




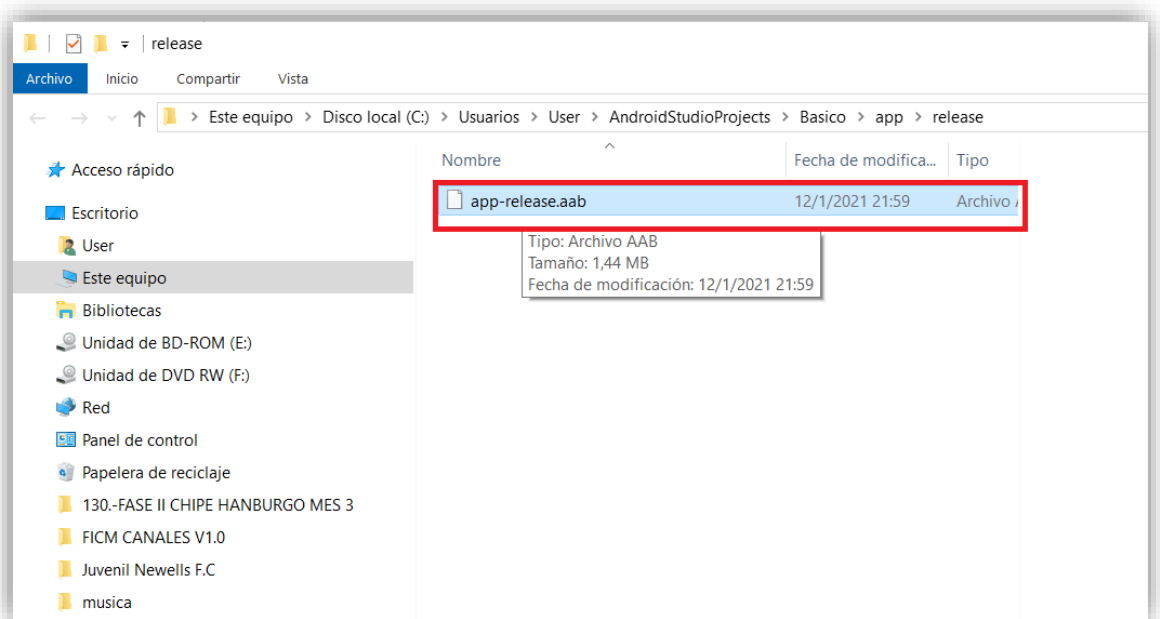




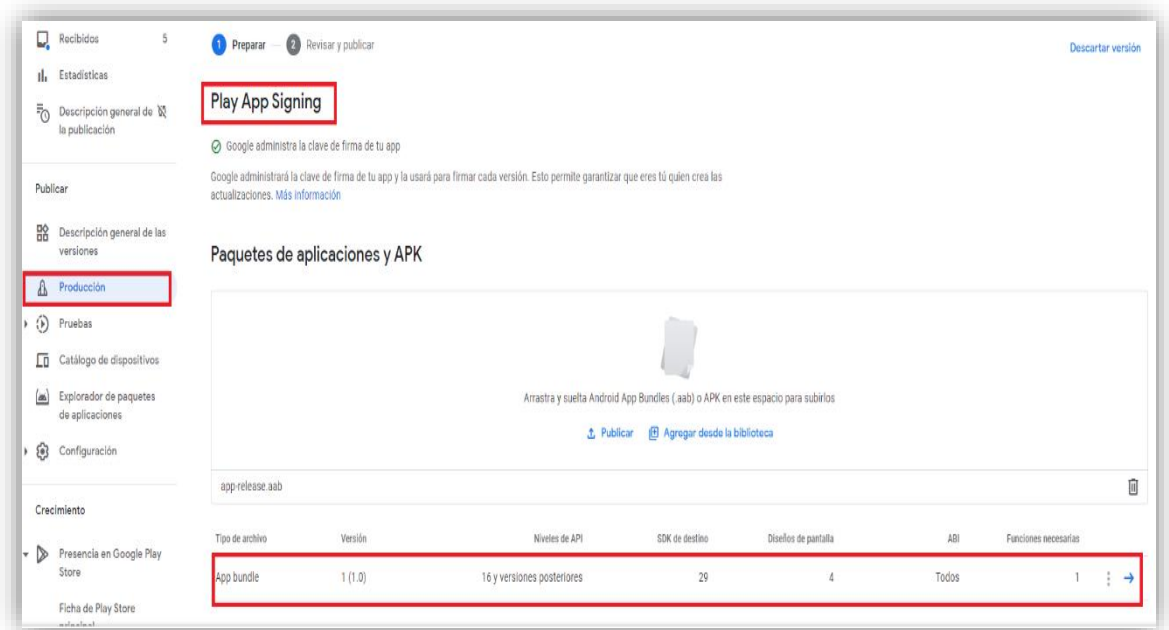
- ❖ Finalizando el proceso anterior nos aparecera un recuadro que dice **Generate Signed Bundle** y a continuación dar clic en **locate** donde nos aparecera la ubicación del paquete **APK**.



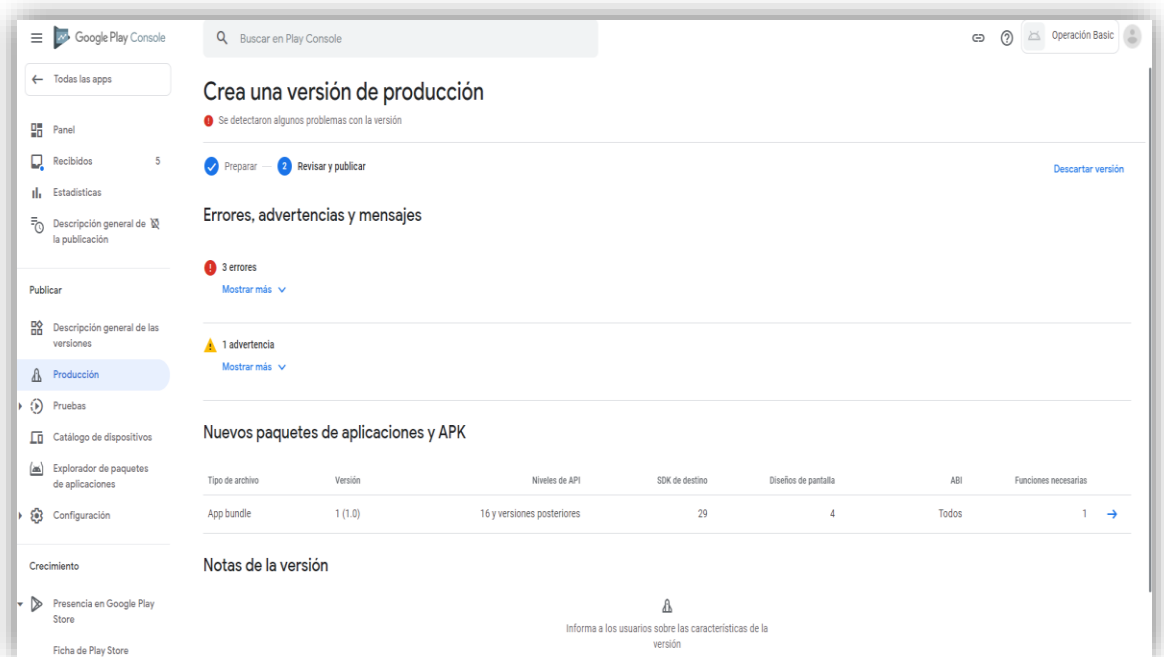
❖ Se generará un paquete con extensión **app-release.aab**



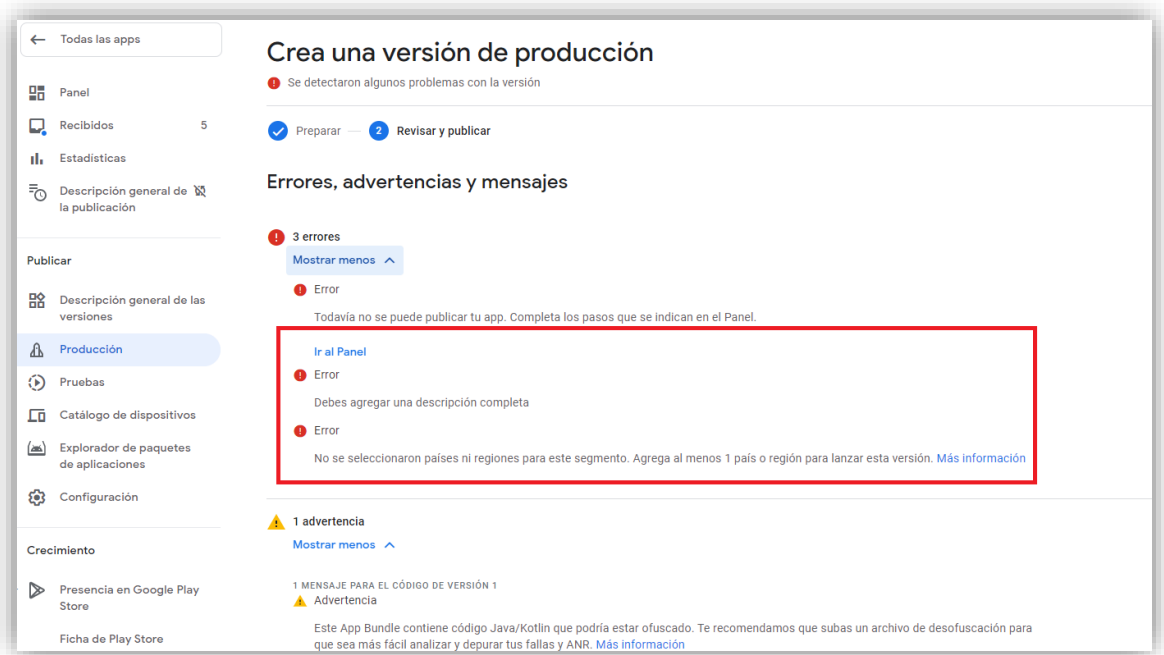
- ❖ Una vez generado el fichero en Android Studio arrastramos el fichero hacia el espacio que nos muestra la **Google Play Console** y dar clic en **Guardar** y luego en **Revisar Versión**.



- ❖ A continuación, nos mostrará este cuadro con una multitud de errores en donde la **Google Play Console** nos pedirá llenar más información.



❖ Dar clic en **Panel** para ingresar la información.



❖ Seguir la siguiente secuencia para completar la información requerida al finalizar dar clic en **Aplicar**.

Google Play Console

Buscar en Play Console

Todas las apps

Panel

Estamos experimentando demoras en nuestros procesos

Configura la app

Brinda información sobre la app y configura la ficha de Play Store

Cuéntanos de qué se trata el contenido de la app y administra cómo se organiza y presenta en Google Play

Ocultar tareas

CUÉNTANOS DE QUÉ SE TRATA EL CONTENIDO DE LA APP

- Acceso a apps >
- Anuncios >
- Clasificación de contenido >
- Público objetivo >
- Apps de noticias >

ADMINISTRA CÓMO SE ORGANIZA Y PRESENTA LA APP

- Selecciona la categoría de app y proporcionar detalles de contacto >
- Configura la ficha de Play Store >

Publicar

- Descripción general de las versiones
- Producción
- Pruebas
- Catálogo de dispositivos
- Explorador de paquetes de aplicaciones
- Configuración

Crecimiento

- Presencia en Google Play Store
- Ficha de Play Store

Google Play Console

Buscar en Play Console

Todas las apps

Panel

Acceso a apps

Si hay partes de la app que están restringidas por credenciales de acceso, membresías, ubicación o cualquier otra forma de autenticación, proporciona instrucciones sobre cómo acceder a ella. Asegúrate de que esta información esté actualizada.

Es posible que Google use esta información para revisar la app. No se compartirá ni utilizará con ningún otro motivo. [Más información](#)

Todas las funcionalidades se encuentran disponibles sin acceso especial

Algunas o todas las funcionalidades están restringidas

[+ Agrega instrucciones nuevas](#)

Puedes agregar hasta 5 instrucciones

Publicar

- Descripción general de las versiones
- Producción
- Pruebas
- Catálogo de dispositivos

holamundo

X

Nombre *

holamundo

Ingresar un nombre para estas instrucciones, de manera que Google pueda reconocer el contexto al que se aplican. Por ejemplo, "Para usuarios de EE.UU."

9/60

Nombre de usuario/número de teléfono

0/100

Contraseña

269078220alexavi

16/100

Otras instrucciones

0/500

Descartar cambios

Aplicar

❖ Regresamos al **Panel** y dar clic en **Anuncios**

Google Play Console

Buscar en Play Console

← Todas las apps

Panel

Estamos experimentando demoras en nuestros procesos

Configura la app

Brinda información sobre la app y configura la ficha de Play Store

Cuéntanos de qué se trata el contenido de la app y administra cómo se organiza y presenta en Google Play

Tareas completadas: 1 de 9 ^

CUÉNTANOS DE QUÉ SE TRATA EL CONTENIDO DE LA APP

- Acceso a apps
- Anuncios >**
- Clasificación de contenido >
- Público objetivo >
- Apps de noticias >

Google Play Console

Buscar en Play Console

← Panel

Anuncios

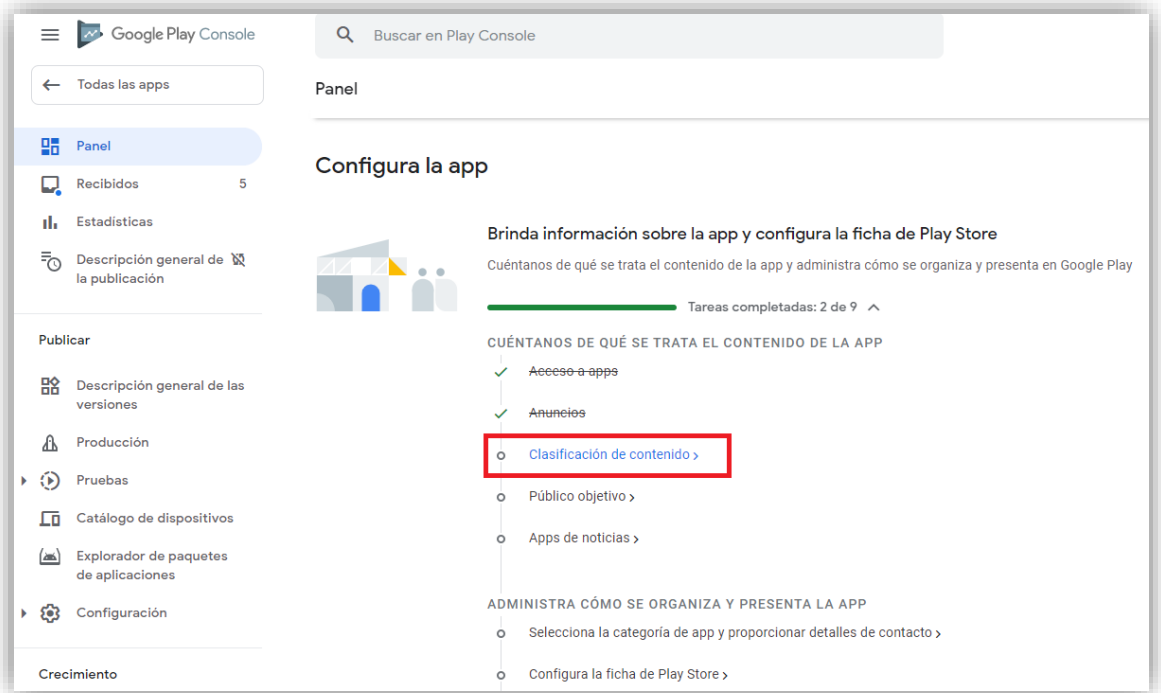
Infórmarnos si la app contiene anuncios. Se incluyen aquellos publicados por redes publicitarias de terceros. Asegúrate de que esta información sea correcta y esté actualizada. [Más información](#)

Anuncios

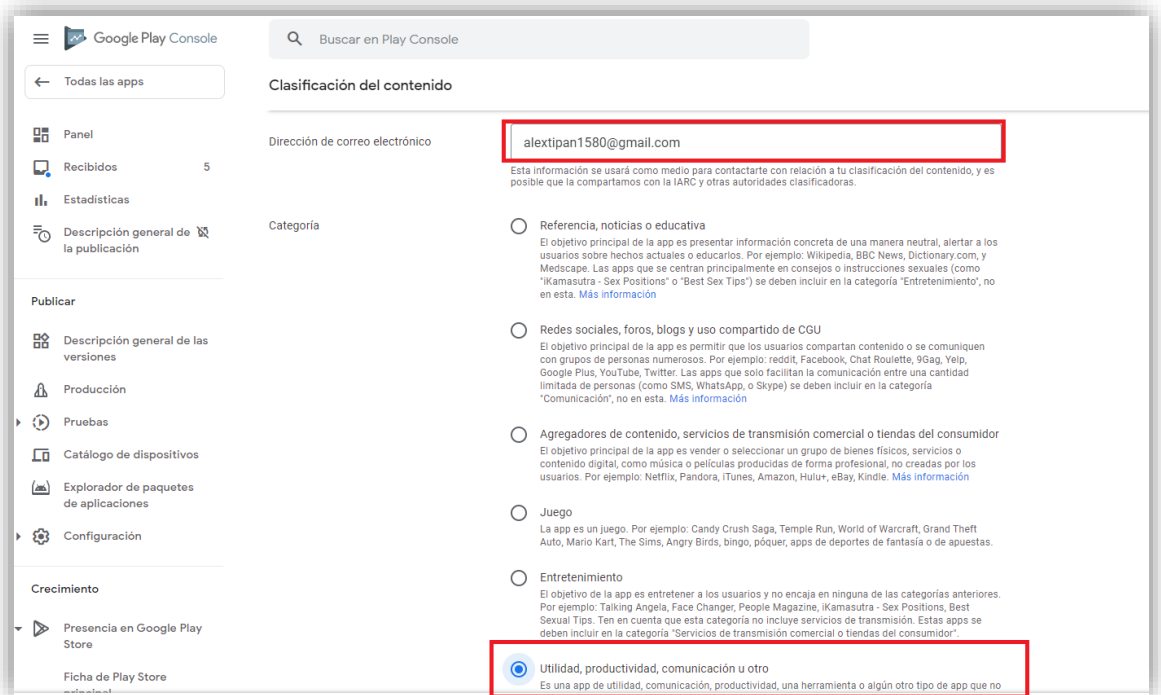
¿Tu app contiene anuncios? Consulta la [Política de Anuncios](#) para asegurarte de que tu app la cumpla.

- Sí, mi app contiene anuncios
Se mostrará la etiqueta "Contiene anuncios" junto a la app en Google Play. [Más información](#)
- No, mi app no contiene anuncios**

❖ Regresar al Panel y dar clic en Clasificación de Contenido



❖ Llenar el espacio vacío y colocar una categoría a la cual pertenece nuestra App



❖ Llenar el cuestionario que nos solicita a continuación

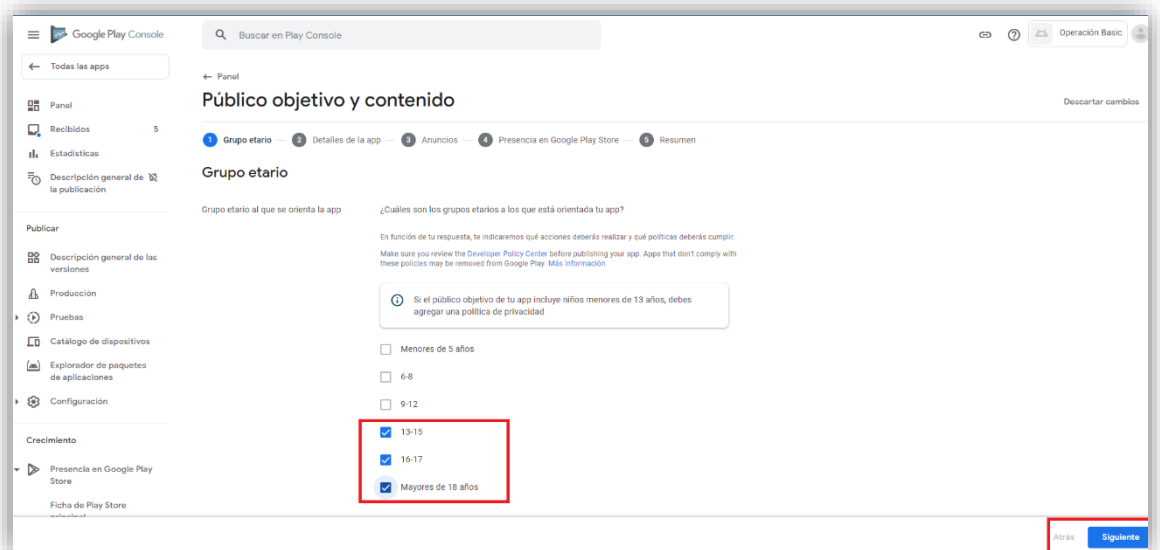
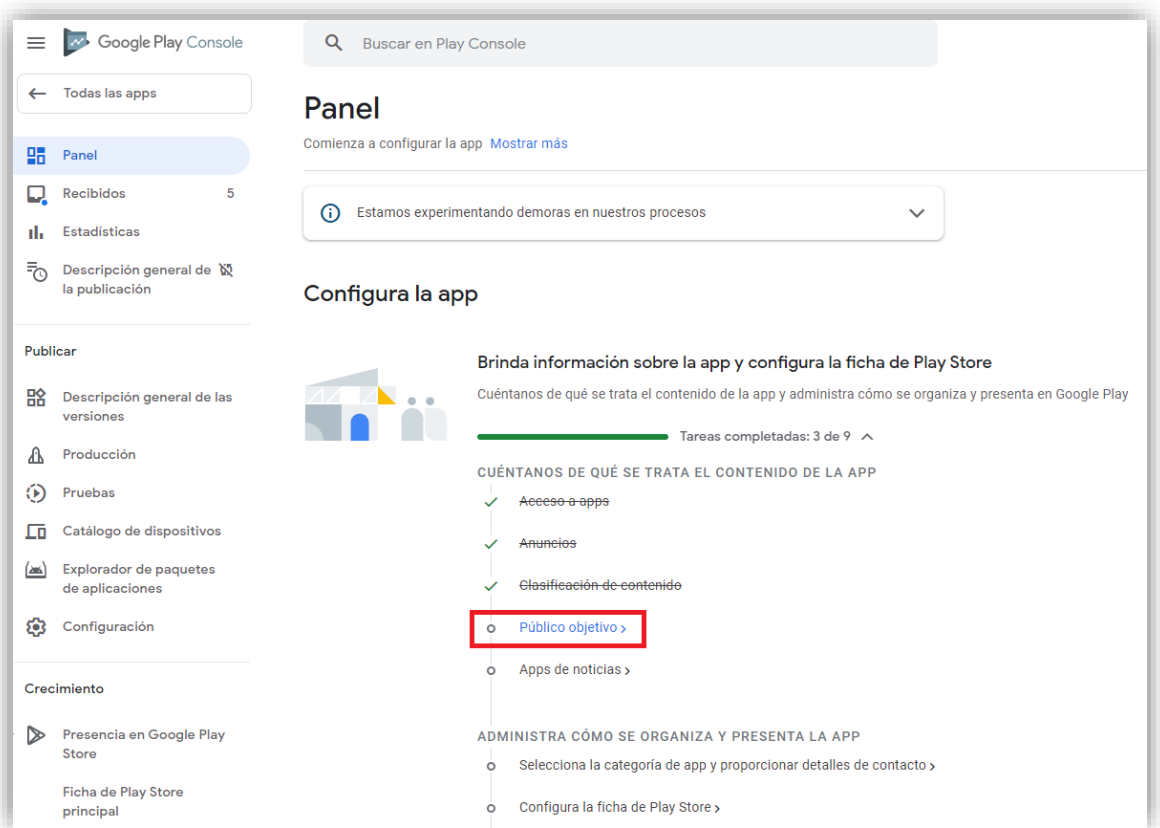
The screenshot shows the 'Clasificación del contenido' (Content Classification) page in the Google Play Console. The page is divided into three steps: 1. Categoría (Category), 2. Cuestionario (Questionnaire), and 3. Resumen (Summary). The 'Cuestionario' step is currently active. The questionnaire is titled 'Utilidad, productividad, comunicación u otro' (Usefulness, productivity, communication or other). It contains two questions, each with 'Sí' (Yes) and 'No' (No) radio button options. The first question is '¿La app incluye contenido violento?' (Does the app include violent content?) and the 'No' option is selected. The second question is '¿La app incluye contenido sexual o imágenes de desnudos (excepto en un contexto natural o científico)?' (Does the app include sexual content or images of nudity (except in a natural or scientific context?)) and the 'No' option is also selected. The 'Sí' and 'No' options for both questions are highlighted with red boxes.

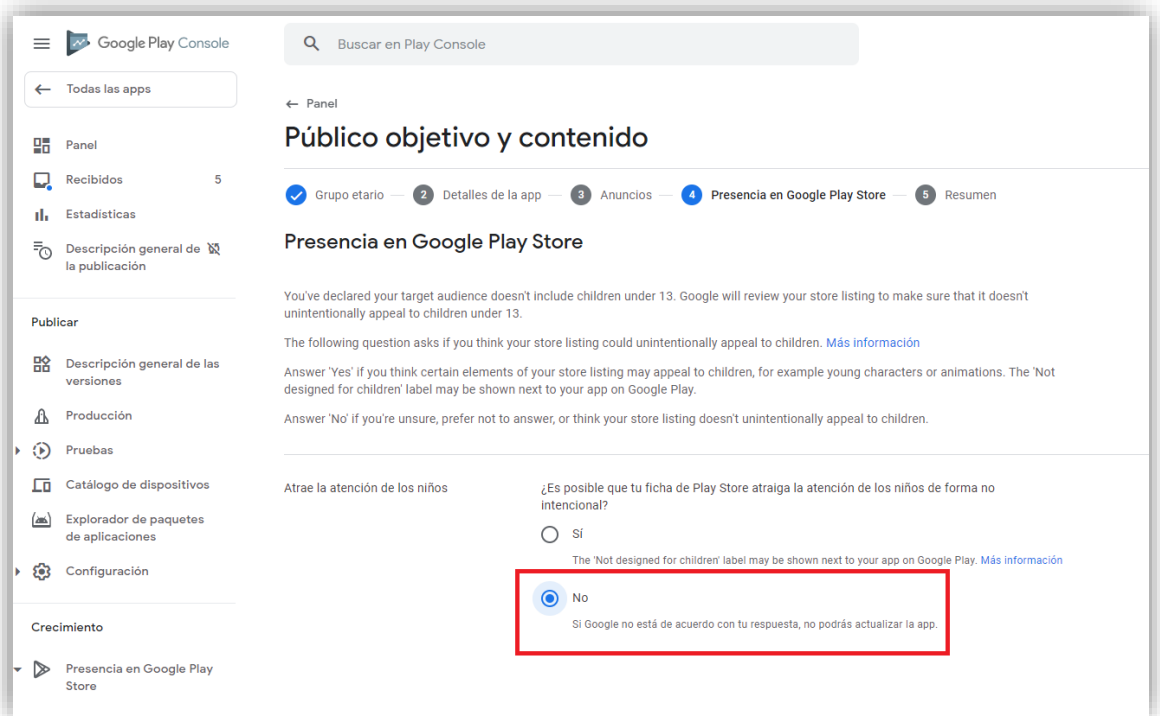
❖ A continuación, nos generara una calificación para nuestra App

The screenshot shows the 'Resumen' (Summary) page in the Google Play Console. The page is divided into three steps: 1. Categoría (Category), 2. Cuestionario (Questionnaire), and 3. Resumen (Summary). The 'Resumen' step is currently active. The page displays the following information:

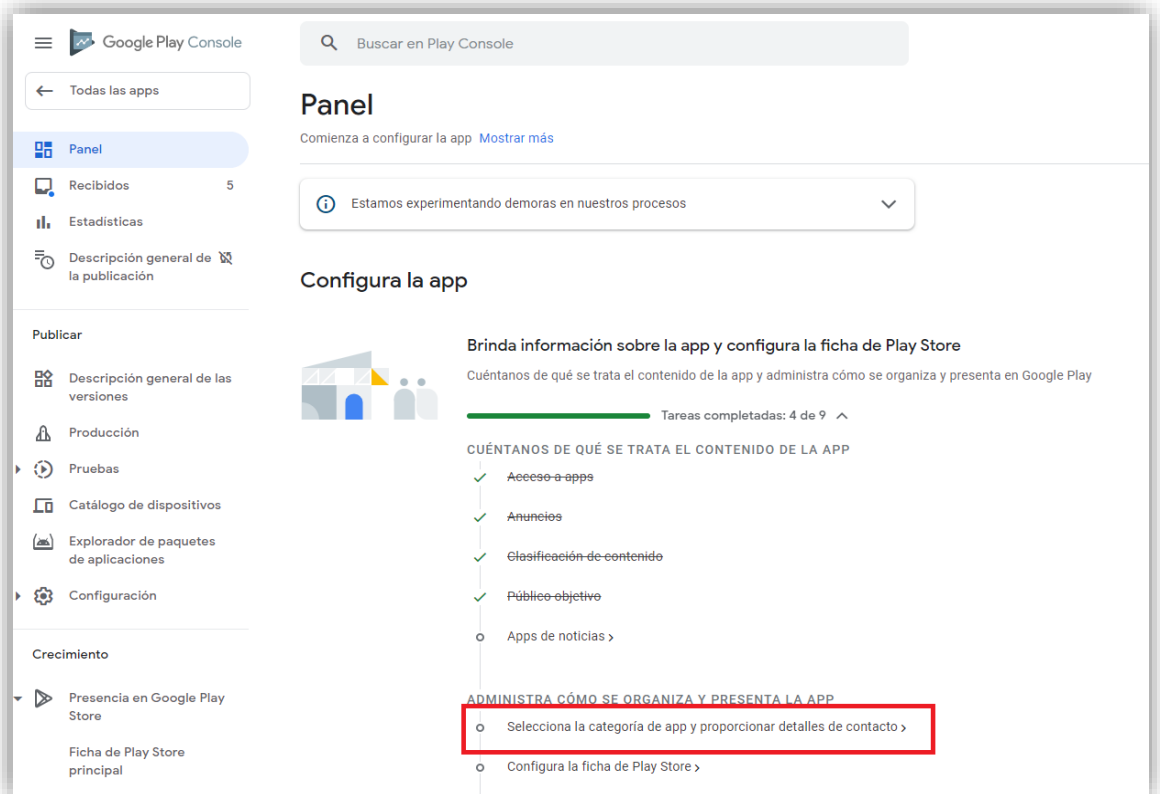
- Resumen:** Es posible que las clasificaciones que se muestran a continuación sean diferentes de las que ven los usuarios en Google Play.
- Notas:**
 - Google tiene derecho a rechazar tu app o actualización si identifica alguna respuesta que tergiverse la realidad con relación a su contenido.
 - Google puede usar tus respuestas al cuestionario para generar clasificaciones destinadas a territorios específicos según se requiera en conformidad con la ley local.
 - Las autoridades clasificadoras que participan en la IARC pueden cambiar la clasificación de tu app después de revisarla.
- Información:** Google y la IARC compartirán tu información de contacto, las respuestas que proporciones en el cuestionario, tu clasificación, las solicitudes de asistencia para desarrolladores y los detalles de la app con las autoridades clasificadoras. [Más información](#)
- Clasificación:** Categoría: Utilidad, productividad, comunicación u otro
- Tus clasificaciones:**

❖ Regresar al **Panel** y dar clic en **Público Objetivo**

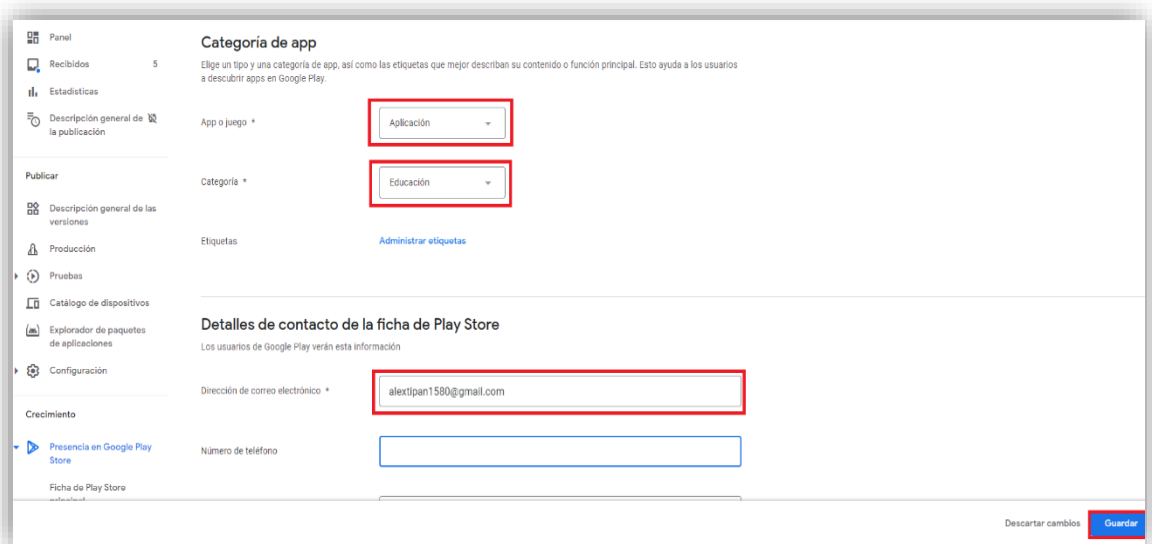




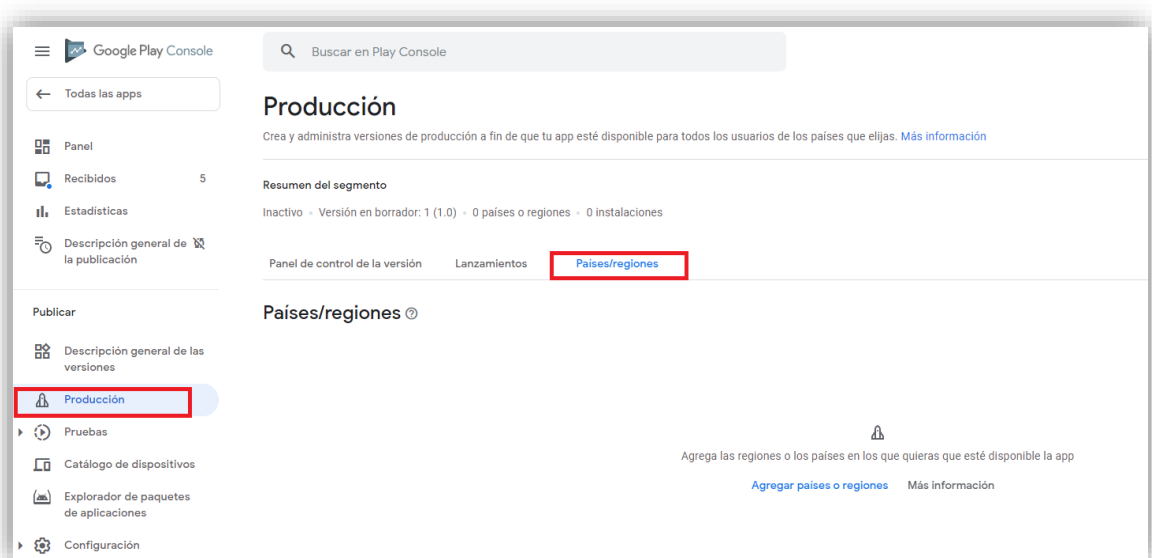
❖ Regresar al **Panel** y dar clic en **Categorías de App**

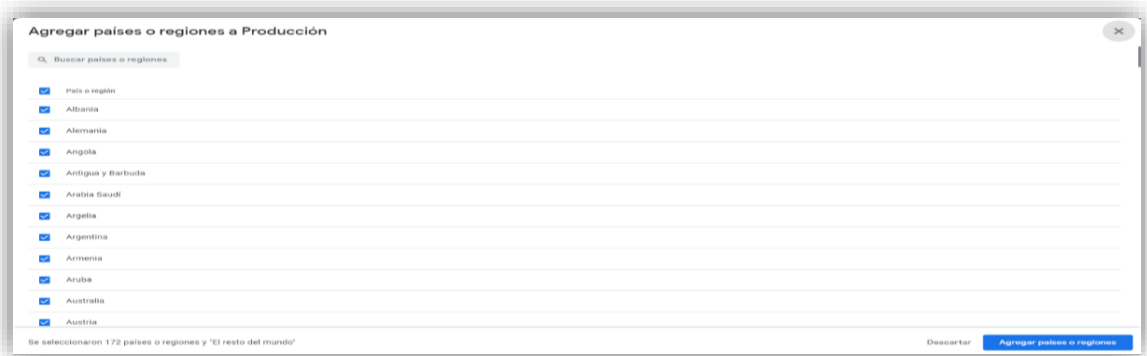


❖ Seleccionar la categoría para la App

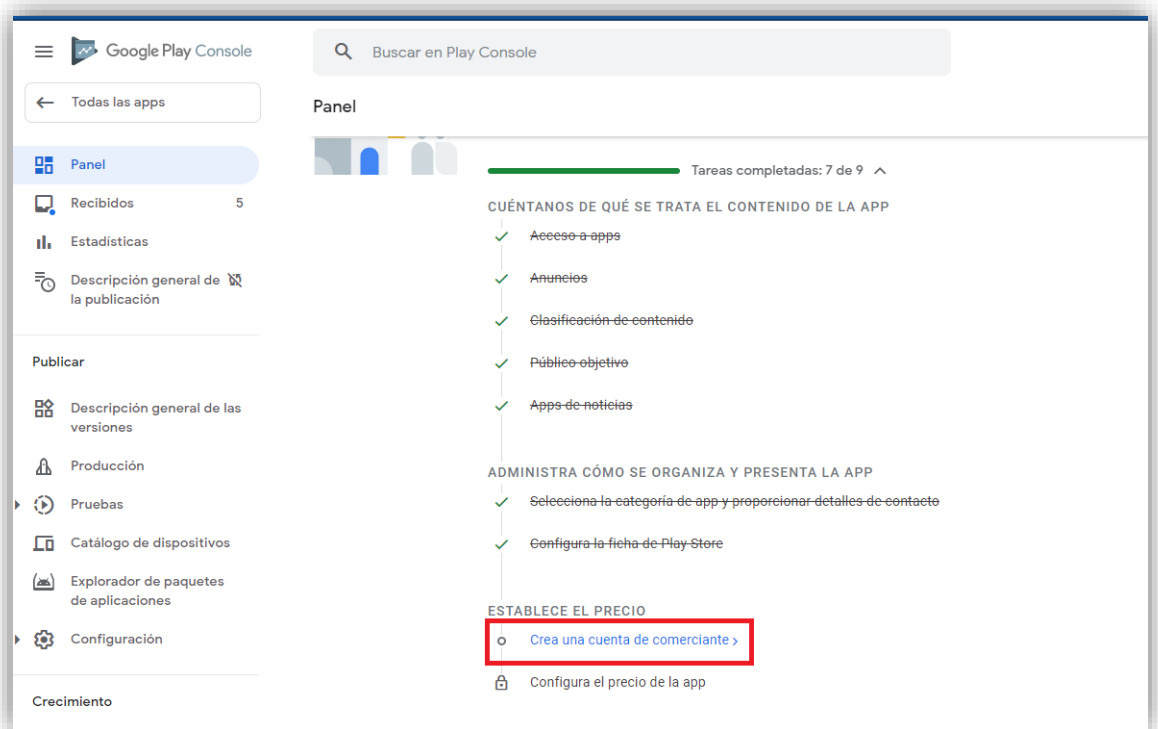


❖ Seleccionar los países y regiones en las que quisiéramos que este nuestra App

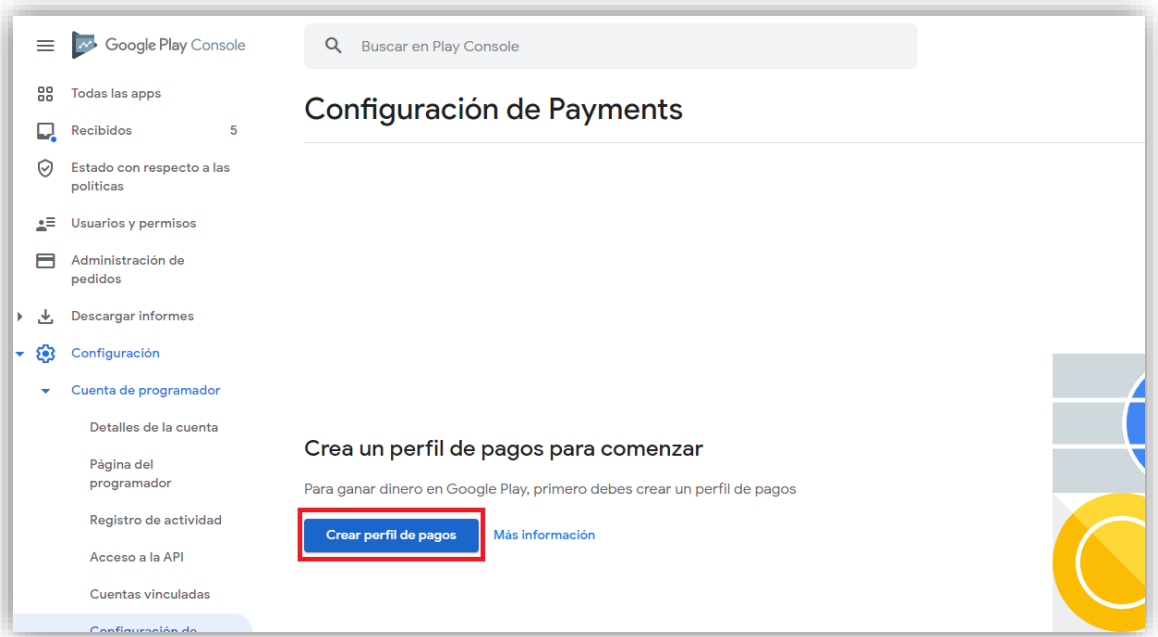




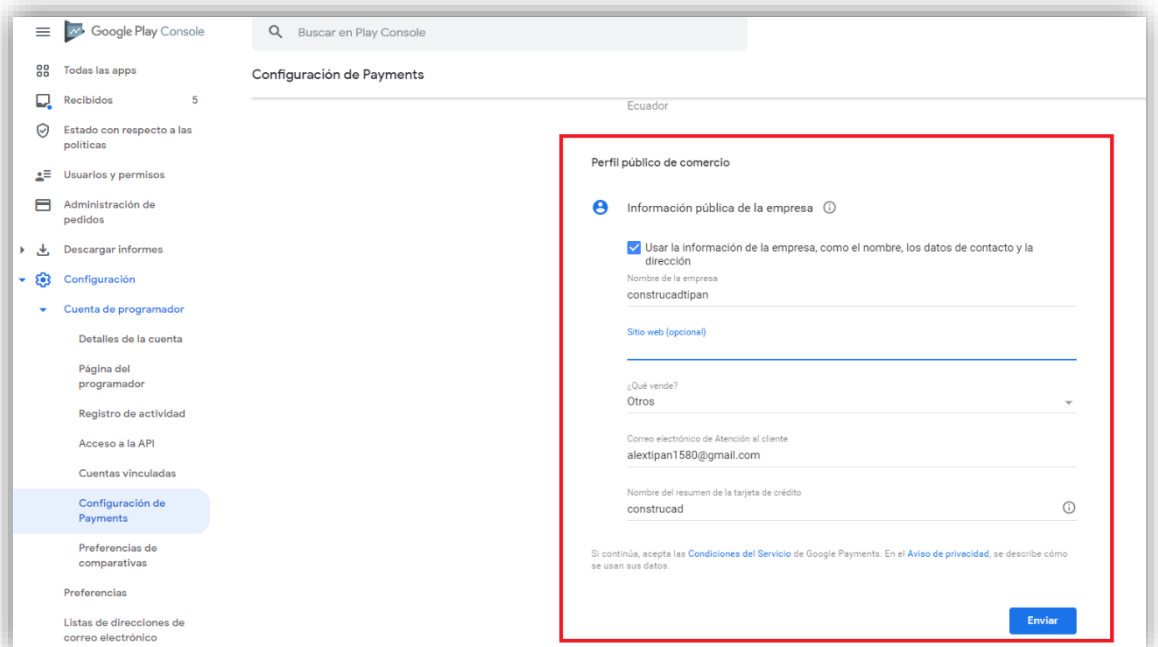
❖ **Crear una cuenta de comerciante** para que su aplicación sea de **Pago**.



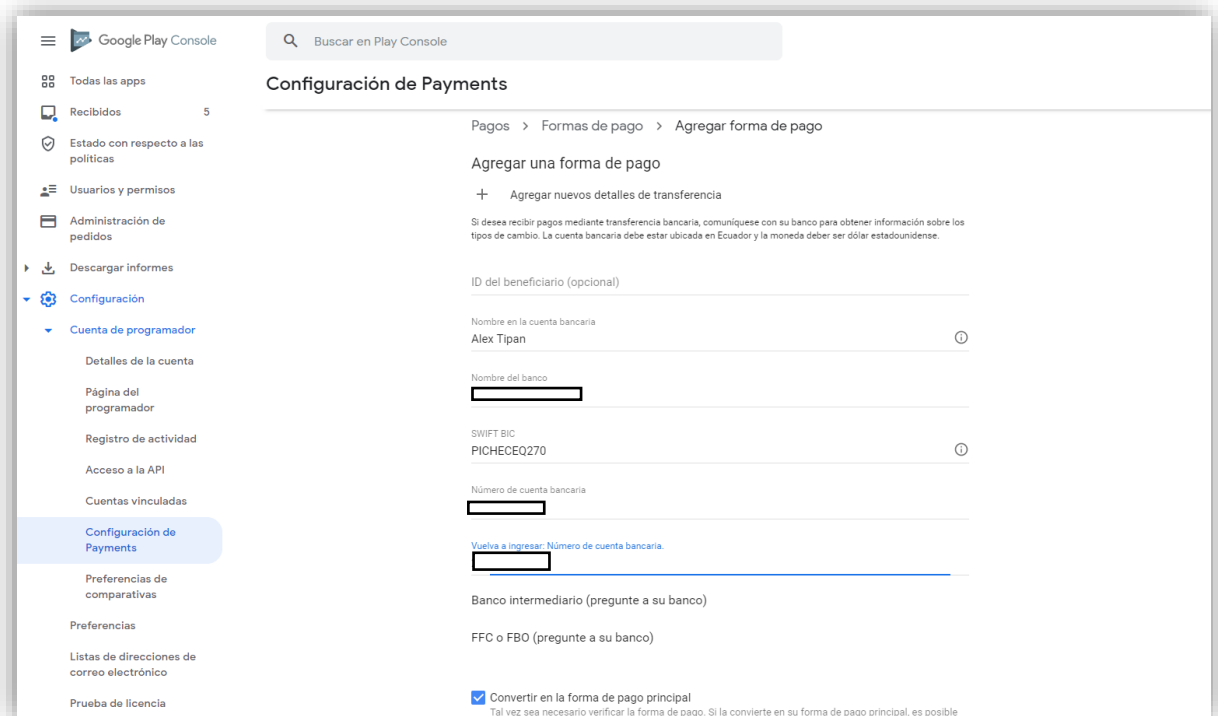
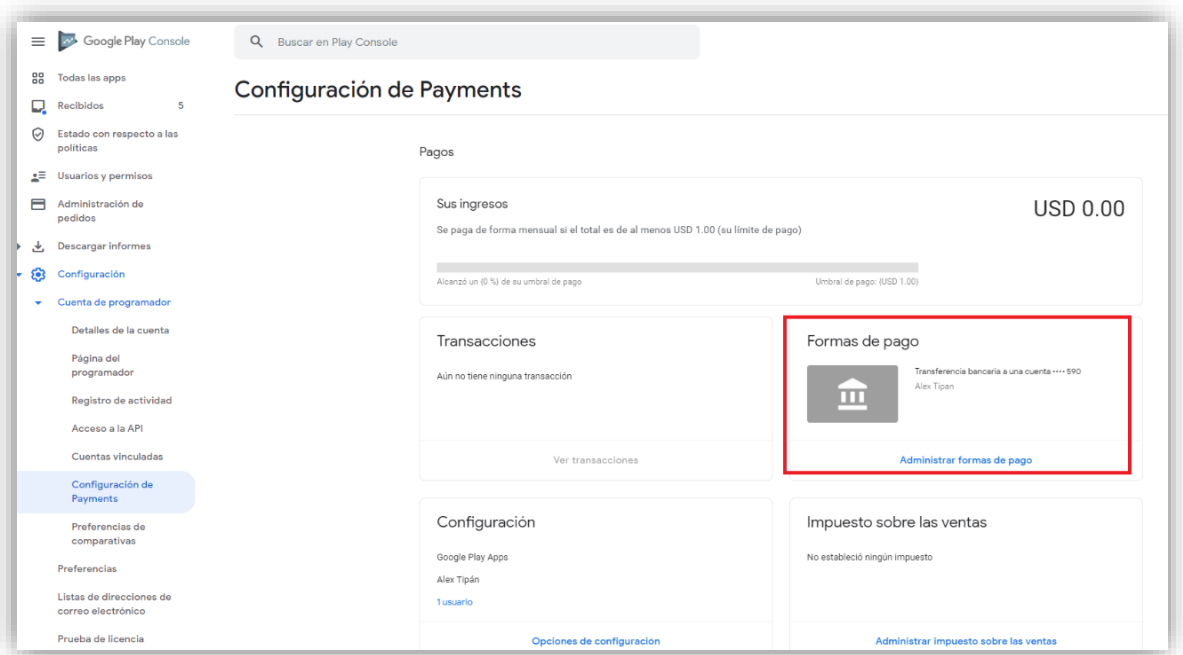
❖ **Crear un perfil de pago** para generar ingresos por las descargas de su App y dar clic en **Crear perfil de pagos**.



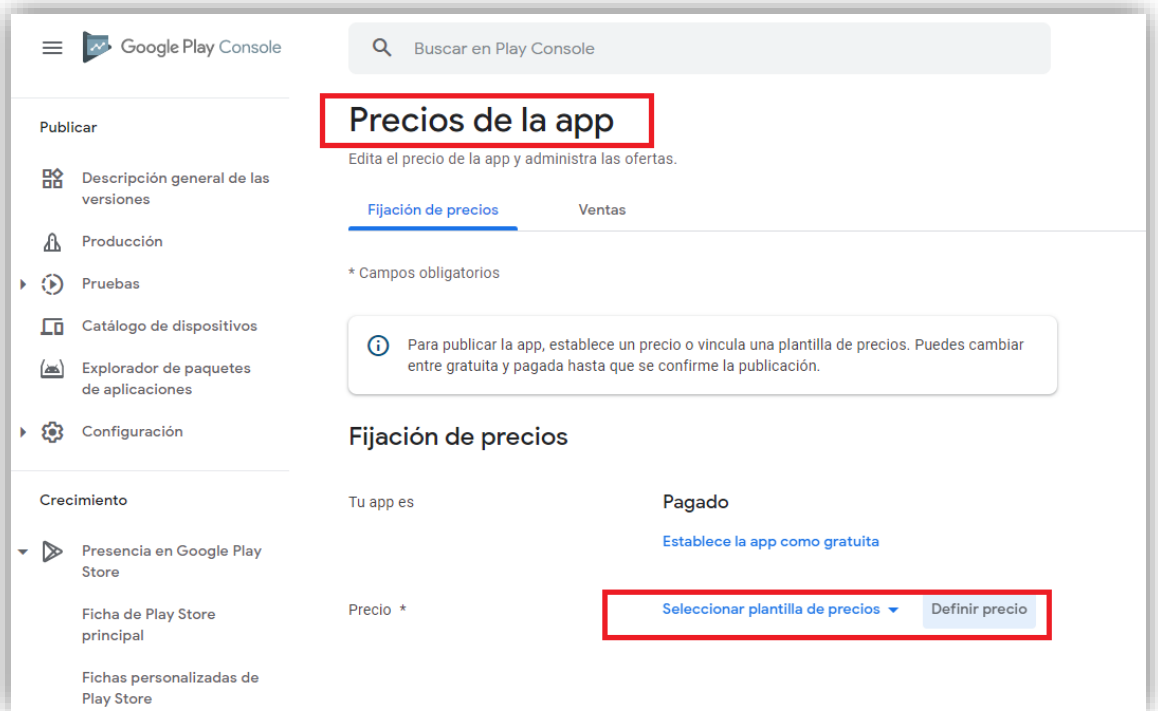
Ingresar la información básica para su perfil de pago como se muestra a continuación y dar clic en **Enviar**.



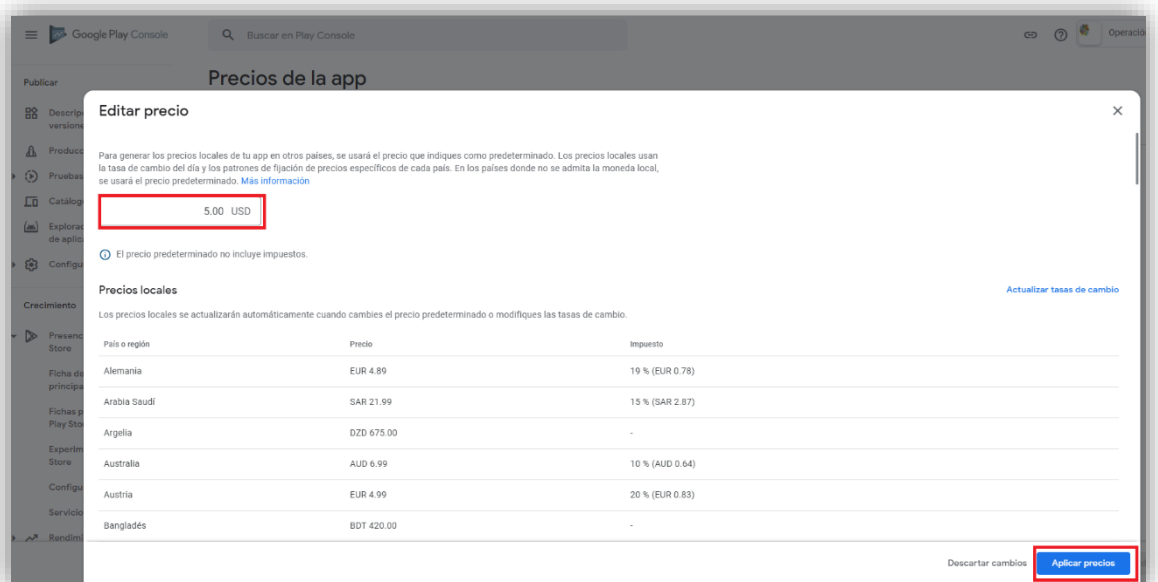
- ❖ Dar clic en **Formas de pago** para completar la información requerida, para que su dinero generado por las descargas sea dirigido a una cuenta bancaria.



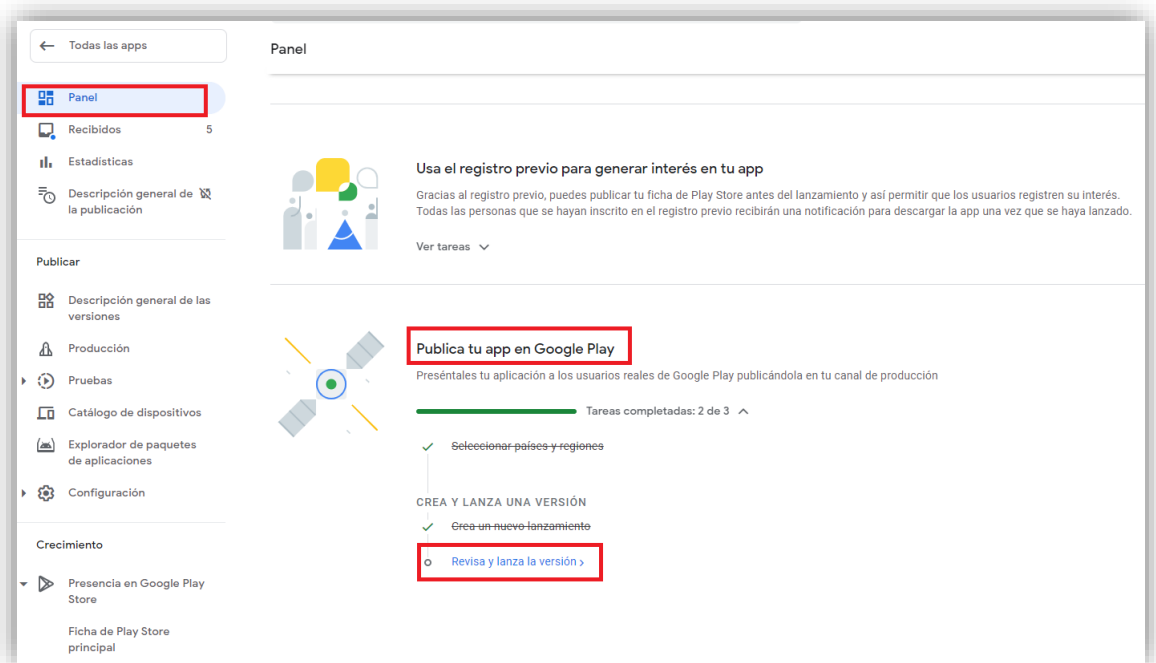
Establecer el precio a ser cobrado por cada descarga de su App dando clic en **Fijación de precios** y luego en **Seleccionar plantilla de precios** según la localidad definimos el precio de nuestra App.



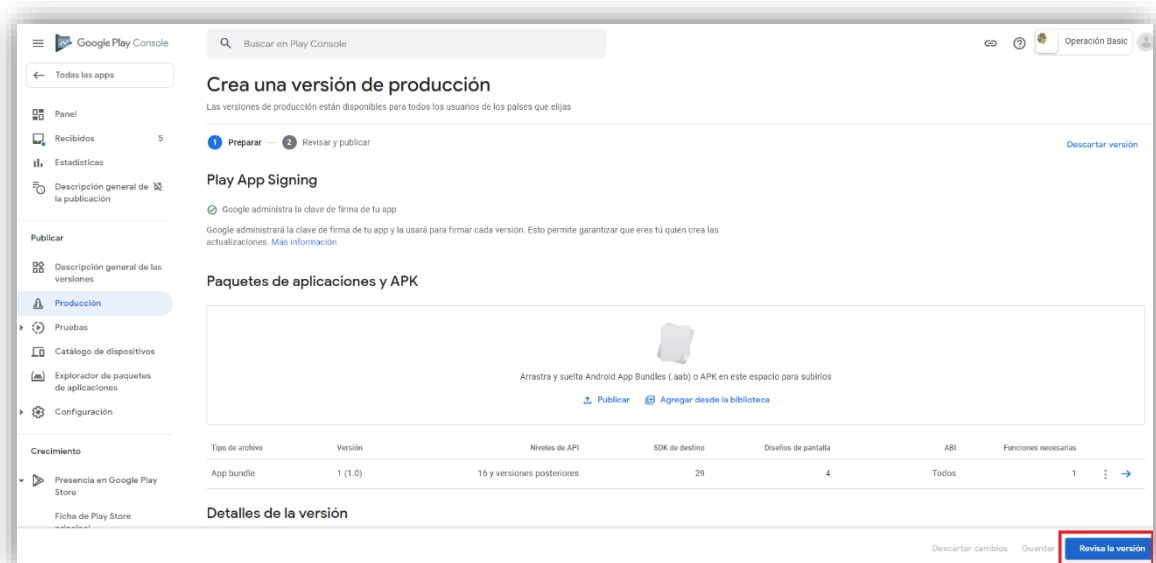
❖ **Digitar el precio de su aplicación en el siguiente recuadro y dar clic en **Aplicar precio**.**



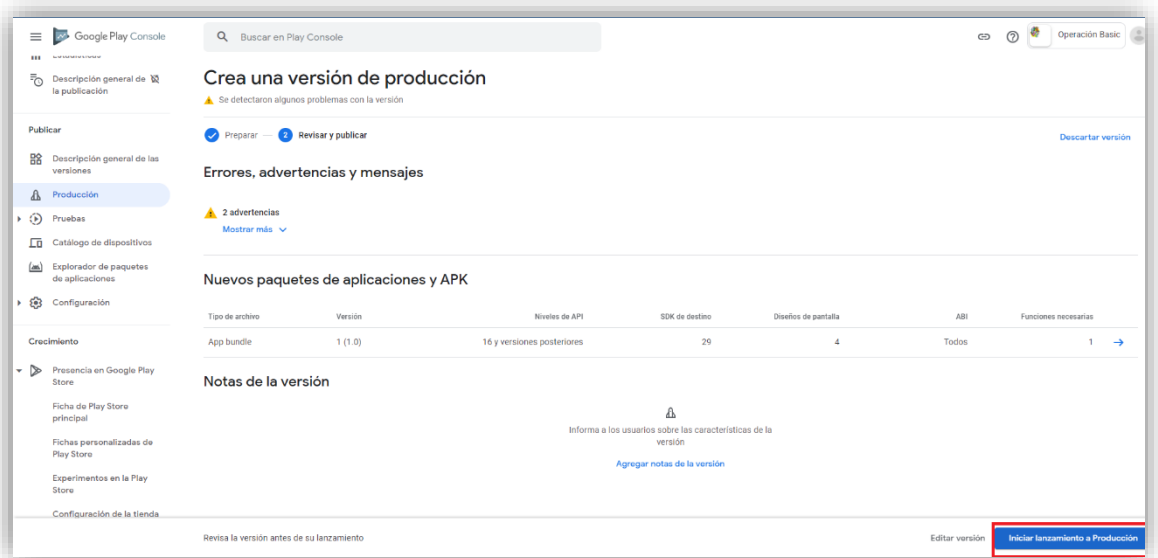
Una vez realizado todo el proceso de registro dar clic en **Panel** seguidamente de **Publica tu App en Google Play** y finalmente en **Revisa y lanza la versión**.



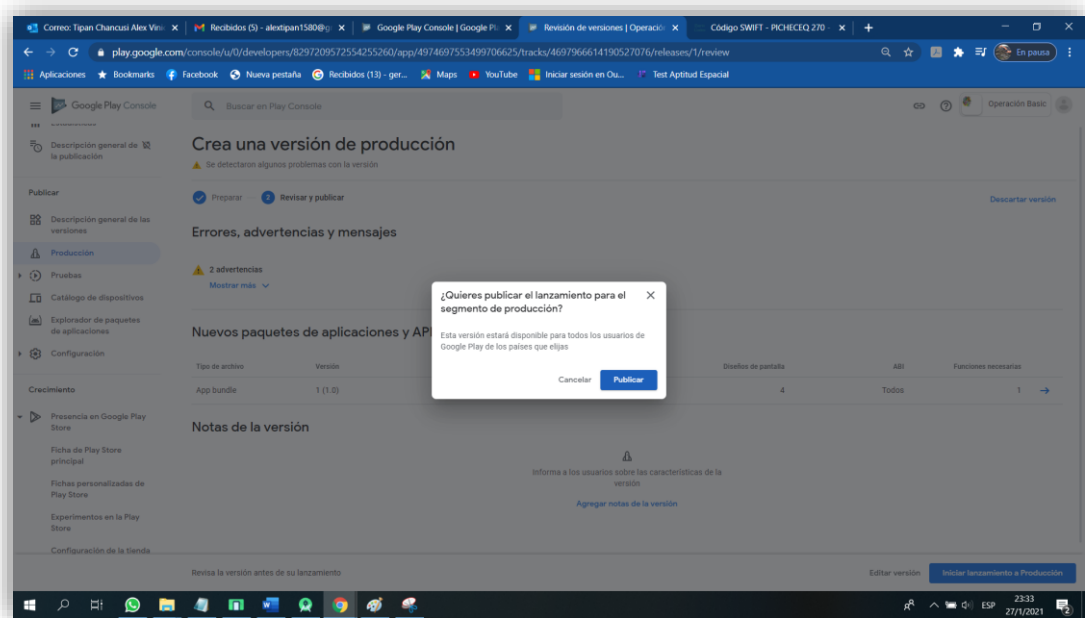
❖ Verificar si el paquete de tu aplicación (APK) se cargo correctamente y damos clic en **Revisar Versión**.



- ❖ Una vez que hayamos terminado con todo el proceso en la **Google Play Console** nos aparecerá el botón que diga **Iniciar lanzamiento a Producción**.

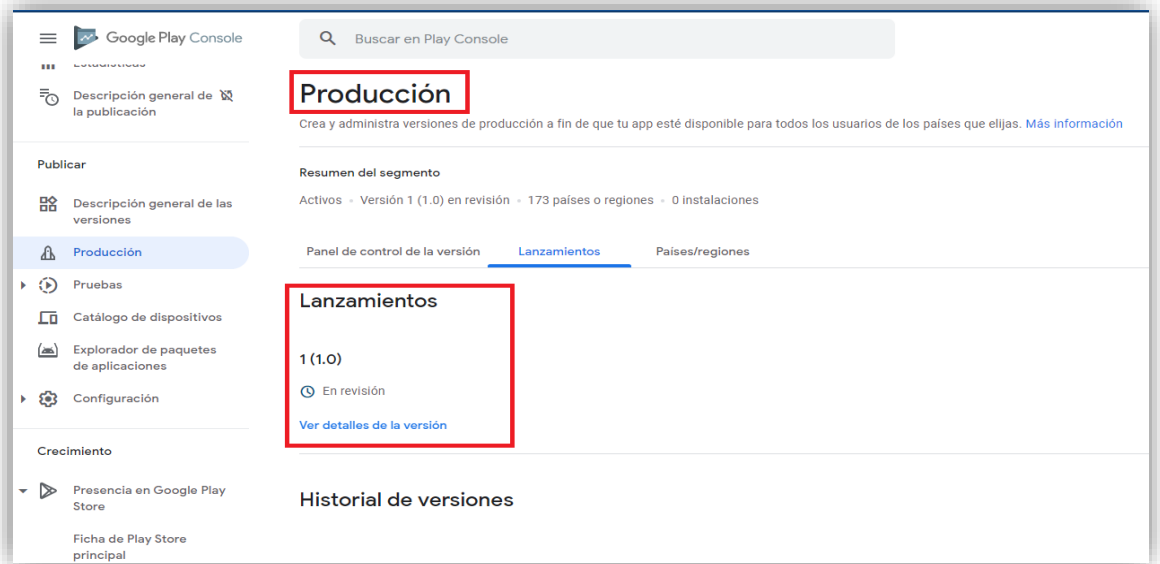


- ❖ A continuación aparecera un mensaje donde menciona si estamos o no seguros de hacer el lanzamiento de nuestra App.

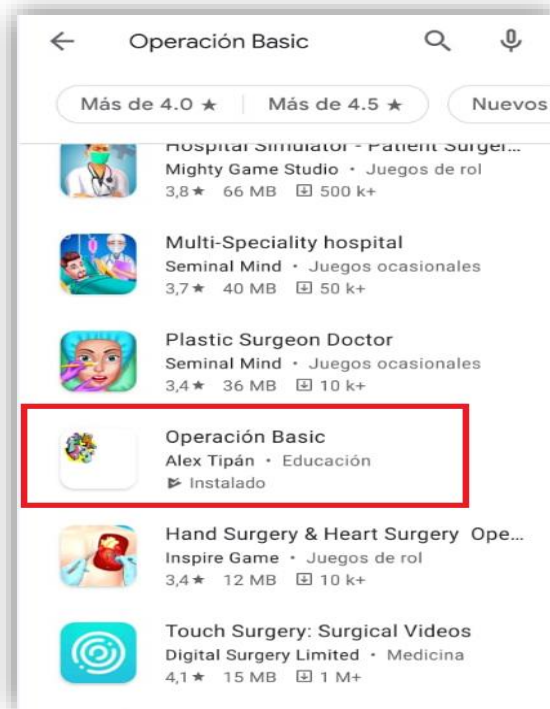


- ❖ En la parte donde dice **Producción** dar clic en **Lanzamientos** y verificamos que nuestra App este en revisión.

- ❖ Después de unos días o una semana se podrá verificar que nuestra App está dentro de los programas de la **Play Store**



- ❖ Transcurrido el tiempo necesario, podremos realizar la búsqueda de la aplicación en la Play Store como se muestra a continuación.



- ❖ Finalmente realizar la descarga de la App con el pago correspondiente que haga el usuario.

