



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS

TEMA:

**“SISTEMA INFORMÁTICO PARA DAR SEGUIMIENTO A DOCUMENTOS
Y TRÁMITES EN EL ILUSTRE MUNICIPIO DE CEVALLOS”**

Trabajo de Graduación Modalidad: TEMI (Trabajo Estructurado de Manera Independiente), presentada como requisito previo para la obtención del Título de Ingeniero en Sistemas Computacionales e Informáticos.

Autor: Bombón Ramos Víctor Alfonso

Tutor: Ing. Galo López

Ambato – Ecuador

2012

APROBACIÓN DEL TUTOR

En calidad de Tutor del Trabajo de Investigación sobre el tema: **“SISTEMA INFORMÁTICO PARA DAR SEGUIMIENTO A DOCUMENTOS Y TRÁMITES EN EL ILUSTRE MUNICIPIO DE CEVALLOS”**, de Víctor Alfonso Bombón Ramos, egresado de la Carrera de Ingeniería en Sistemas computacionales e informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, considero que dicho informe investigativo reúne los requisitos y méritos suficientes para ser sometidos a la evaluación del Tribunal de Grado, que el Honorable Consejo Directivo designe, para su correspondiente estudio y calificación.

Ambato, Abril 2012

Ing. Galo López

AUTORÍA

El presente trabajo de investigación titulado **“SISTEMA INFORMÁTICO PARA DAR SEGUIMIENTO A DOCUMENTOS Y TRÁMITES EN EL ILUSTRE MUNICIPIO DE CEVALLOS”**, es totalmente original, auténtico y personal, en esta virtud, el contenido, efectos legales y académicos que se desprenden del mismo son exclusiva responsabilidad del autor.

Ambato, Abril 2012

Víctor Alfonso Bombón Ramos

CC.: 180404269-3

APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Pilar Urrutia e Ing. Teresita Freire, reviso y aprobó el Informe Final del trabajo de graduación titulado: **“SISTEMA INFORMÁTICO PARA DAR SEGUIMIENTO A DOCUMENTOS Y TRÁMITES EN EL ILUSTRE MUNICIPIO DE CEVALLOS”**, presentado por el señor Víctor Alfonso Bombón Ramos de acuerdo al Art. 18 del Reglamento de Graduación para Obtener el Título Terminal de Tercer Nivel de la Universidad Técnica de Ambato.

Ing. Oswaldo Paredes

PRESIDENTE DEL TRIBUNAL

Ing. Pilar Urrutia

DOCENTE CALIFICADOR

Ing. Teresita Freire

DOCENTE CALIFICADOR

DEDICATORIA

Este trabajo va dedicado a todo el esfuerzo, sacrificio y constancia que he realizado para lograr esta anhelada meta en el campo profesional, gracias al apoyo y confianza que siempre me han brindado mis amados padres, en especial a mi mamá, que por más dura que ha sido la vida con ella siempre estuvo pendiente de mí brindándome todo su amor, confianza y apoyo siendo más que una madre mi mejor amiga. Así también a mi tutor, por su consideración a lo largo del presente trabajo.

Víctor Alfonso Bombón Ramos

AGRADECIMIENTO

A mi Virgencita de Agua Santa, a Jesucristo y a mi Papito Olmedo ya que son los que me protegen y me guían en mi camino día a día alcanzando las metas que me propongo.

A mis padres por su comprensión, amor y apoyo incondicional en mi accionar.

A mi amor Mónica, a mis familiares, a mis amigos y amigas que me dieron palabras de aliento para continuar con este trabajo sin rendirme.

Al Ing. Galo López, por brindarme parte de su tiempo en las inquietudes que he tenido a lo largo del desarrollo de este proyecto TEMI.

Victor Alfonso Bombón Ramos

ÍNDICE

ÍNDICE GENERAL

CONTENIDO	PÁGINA
APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
APROBACIÓN DE LA COMISIÓN CALIFICADORA	iv
DEDICATORIA	¡Error! Marcador no definido.
AGRADECIMIENTO	vi
ÍNDICE	vii
ÍNDICE GENERAL	viii
ÍNDICE DE TABLAS	xviii
RESUMEN EJECUTIVO	xx
INTRODUCCIÓN	xxi
CAPÍTULO I	1
EL PROBLEMA	1
1.1 Planteamiento del problema.....	1
1.1.1 Contextualización.....	1
1.1.2 Análisis crítico.....	2
1.1.3 Prognosis.....	2
1.2 Formulación del problema.....	3
1.2.1 Preguntas Directrices.....	3
1.2.2 Delimitación del problema.....	3
1.3 Justificación.....	3
1.4 Objetivos.....	4
1.4.1 Objetivo General.....	4
1.4.2 Objetivos Específicos.....	4
CAPÍTULO II	5
MARCO TEÓRICO	5
2.1 Antecedentes Investigativos.....	5
2.2 Fundamentos.....	6

2.2.1	Fundamentación Legal	6
2.2.2	Fundamentación Teórica	9
2.2.2.1	Sistema Informático.....	9
2.2.2.1.1	Software	9
2.2.2.1.2	Informática	10
2.2.2.1.3	Sistemas de Información	10
2.2.2.1.4	Arquitectura Informática	11
2.2.2.1.5	Arquitectura de Software	11
2.2.2.1.6	Lenguaje de Programación.....	12
2.2.2.1.7	SharpDevelop.....	13
2.2.2.1.8	Base de Datos	14
2.2.2.1.9	Microsoft SQL Server	15
2.2.2.1.10	SQL Server 2000	16
2.2.2.1.11	Funciones en SQL Server 2000	16
2.2.2.2	Control de la Documentación	17
2.2.2.2.1	Documento.....	17
2.2.2.2.2	Trámite	17
2.2.2.2.3	Automatización	18
2.2.2.2.4	DocFlow	18
2.3	Hipótesis	19
2.4	Variables.....	19
2.4.1	Variable Independiente.....	19
2.4.2	Variable Dependiente	19
CAPÍTULO III		20
METODOLOGÍA		20
3.1	Enfoque.....	20
3.2	Modalidad básica de la Investigación.....	20
3.2.1	Investigación de Campo	20
3.2.2	Investigación Documental – Bibliográfica.....	20
3.2.3	Proyecto Factible.....	21

3.3	Niveles o Tipos de Investigación.....	21
3.4	Población y Muestra	21
3.4.1	Población.....	21
3.4.2	Muestra.....	21
3.5	Operacionalización de las Variables.....	22
3.5.1	Variable Independiente.....	22
3.5.2	Variable Dependiente	26
3.6	Recolección de información	27
3.6.1	Plan de recolección de información	27
3.6.2	Procesamiento y análisis de información	27
3.7	Plan de análisis e interpretación de los resultados	27
CAPÍTULO IV	28
	ANÁLISIS E INTERPRETACIÓN DE RESULTADOS	28
4.1	Análisis de la necesidad.....	28
4.2	Análisis de Resultados	29
4.3	Interpretación de resultados	36
CAPÍTULO V	38
	CONCLUSIONES Y RECOMENDACIONES	38
5.1	Conclusiones.....	38
5.2	Recomendaciones	39
CAPÍTULO VI	40
	PROPUESTA	40
6.1	Tema	40
6.2	Datos Informativos	40
6.3	Antecedentes.....	40
6.4	Justificación	41
6.5	Objetivos.....	42
6.5.1	Objetivo General	42
6.5.2	Objetivos Específicos.....	42
6.6	Análisis de factibilidad	42

6.6.1 Factibilidad Operativa.....	42
6.6.2 Factibilidad Económica.....	43
6.6.3 Factibilidad Técnica.....	43
6.7 Fundamentación.....	44
6.8 Metodología.....	45
6.9 Modelo Operativo.....	46
6.9.1 Análisis del Sistema.....	46
6.9.1.1 Servicios Requeridos.....	46
6.9.1.2 Diagramas UML.....	47
6.9.1.2.1 Diagramas de Casos de Usos.....	47
6.9.1.2.1.1 Caso de uso Administración de cuentas de usuario.....	47
6.9.1.2.1.2 Caso de uso Parámetros.....	49
Figura 6.2 Parámetros Uso del Sistema.....	50
6.9.1.2.1.3 Caso de uso Movimientos.....	51
6.9.1.2.1.3 Caso de uso Consultas y Reportes.....	53
6.9.1.2.2 Diagrama de Secuencia.....	54
6.9.1.2.2.1 Creación de Usuarios.....	54
Actor(es): Administrador, Empleado.....	54
Descripción: Detalla en proceso de creación de Usuarios en el sistema.	54
Figura 6.5 Diagrama de Secuencia Creación de Usuarios.....	55
6.9.1.2.2.2 Manejo de la Base de Datos.....	55
6.9.1.2.2.3 Manejo y administración del Sistema.....	56
6.9.1.2.3 Diagrama de Clase.....	58
6.9.2 Diseño del Sistema.....	60
6.9.2.1 Diseño de la Base de Datos.....	60
6.9.2.2 Diccionario de datos.....	63
Tabla 6.25 Tabla TMP_TIPO.....	69
6.9.2.3 Diseño de la interfaz.....	69
6.9.2.4 Desarrollo del sistema.....	78

6.9.2.4.1 Extracto de código fuente	78
6.9.3 Pruebas	81
6.9.3.1 Pruebas Caja Blanca	81
6.9.3.2 Pruebas Caja Negra	82
6.9.3.3 Pruebas de verificación y validación	83
6.9.4 Implantación del Sistema	87
6.9.4.1 Cronograma de Implantación del Sistema.....	87
Tabla 6.26 Cronograma de Implantación del Sistema.....	88
6.9.4.2 Levantamiento del Sistema.....	88
6.10 Conclusiones y Recomendaciones.....	88
6.10.1 Conclusiones	88
6.10.2 Recomendaciones.....	89
6.11 Bibliografía	89
6.11.1 Información Bibliográfica de Libros.....	89
6.11.2 Información Bibliográfica de Internet.....	90
6.12 Glosario de términos.....	93
ANEXOS	95
ANEXO 1: MODELOS DE ENTREVISTAS PERSONALES	96
ANEXOS 2: MANUAL DE INSTALACIÓN DE SOFTWARE REQUERIDO .	101
Anexo 2.1: Manual de Instalación de SQL Server 2000.	101
Anexo 2.2: Instalación de SQL Server Management.	107
Anexo 1.3: Manual de Instalación de SharpDevelop	110
ANEXO 3: CREACIÓN DE TABLAS MEMBERSHIP.....	115
ANEXO 4: SECUENCIAS DE COMANDOS SQL.....	120
ANEXO 5: EXPLICACIÓN, ESTRUCTURA Y EJEMPLOS DE CÓDIGO	
FUENTE DE LA APLICACIÓN	125
1. Cliente.....	125
2. Servidor	144
ANEXO 6: MANUAL DE ADMINISTRACIÓN DEL SISTEMA	161
Manual de Administración	161

Manual de Usuarios.....	193
-------------------------	-----

ÍNDICE DE FIGURAS

CONTENIDO	PÁGINA
Figura 4.2.1 Gráfico pregunta 2.....	30
Figura 4.2.2 Gráfico pregunta 3.....	31
Figura 4.2.3 Gráfico pregunta 5.....	32
Figura 4.2.4 Gráfico pregunta 6.....	33
Figura 4.2.5 Gráfico pregunta 9.....	34
Figura 4.2.6 Gráfico pregunta 14.....	35
Figura 6.1 Diagrama Caso Uso Cuentas de Usuario.....	48
Figura 6.2 Parámetros Uso del Sistema.....	50
Figura 6.3 Crear y recibir documentos.....	51
Figura 6.4 Documentos ingresados al sistema.....	53
Figura 6.5 Diagrama de Secuencia Creación de Usuarios.....	55
Figura 6.5 Diagrama de Secuencia Manejo de BD.....	56
Figura 6.6 Diagrama de Secuencia Manejo y Administracion Sistema.....	57
Figura 6.7 Diagrama de Clases.....	59
Figura 6.8 Diagrama Relacional Principal.....	61
Figura 6.9 Diagrama Relacional Menú.....	61
Figura 6.10 Diagrama Relacional MemberShip.....	62
Figura 6.11 Diagrama Relacional General.....	62
Figura 6.12 Pantalla de Inicio.....	70
Figura 6.13 Pantalla Principal del Sistema.....	70
Figura 6.14 Creación de Usuarios.....	71
Figura 6.15 Departamentos – Sección y Responsable.....	73
Figura 6.16 Envío de Documentos.....	74
Figura 6.17 Hoja de Guía.....	77
Figura 6.18 Ingreso al Sistema.....	84
Figura 6.19 Creación de Usuarios.....	84

Figura 6.20 Ingreso de Trámites y Documentos.....	85
Figura 6.21 Envío de Documentos.....	85
Figura 6.22 Finalización de Documentos.	86
Figura 6.23 Consulta de Documentos.	86
Figura A1.1 Microsoft SQL Server 2000.....	101
Figura A1.2 Microsoft SQL Server 2000 Install.....	102
Figura A1.3 Microsoft SQL Server 2000 Name.....	102
Figura A1.4 Microsoft SQL Server 2000 Selection.....	103
Figura A1.5 Microsoft SQL Server 2000 Definition.	103
Figura A1.6 Microsoft SQL Server 2000 Name.....	104
Figura A1.7 Microsoft SQL Server 2000 Setup.	104
Figura A1.8 Microsoft SQL Server 2000 Accounts.	105
Figura A1.9 Microsoft SQL Server 2000 Mode.	105
Figura A1.10 Escritorio.....	106
Figura A1.11 Arrancar SQL Server.	106
Figura A2.1 Instalador Management.	107
Figura A2.2.	107
Figura A2.3.	108
Figura A2.4.	108
Figura A2.5.	108
Figura A2.6.	109
Figura A2.7.	109
Figura A2.8.	109
Figura A3.1 Microsoft .NET.....	111
Figura A3.2 SharpDevelop Setup.	112
Figura A3.3 SharpDevelop Setup Licencia.....	112
Figura A3.4 SharpDevelop Setup Direction.	113
Figura A3.5 SharpDevelop Setup Ready.	113
Figura A3.6 SharpDevelop Setup Ready.	113
Figura A3.7 SharpDevelop Setup Finish.	114

Figura A4.1 MembrerShip 1.....	115
Figura A4.2 MembrerShip 2.....	116
Figura A4.3 MembrerShip 3.....	117
Figura A4.4 MembrerShip 4.....	117
Figura A4.5 MembrerShip 5.....	117
Figura A4.6 MembrerShip 6.....	118
Figura A4.7 MembrerShip 7.....	118
Figura A4.8 MembrerShip Tablas..	118
Figura A4.9 MembrerShip Diagrama	119
Figura A5.1 Sistema Administración.....	161
Figura A5.2 Ingreso	161
Figura A5.3 Pantalla principal.	162
Figura A5.4 Cambio de Clave.....	163
Figura A5.5 Cambio de Clave en Lote.	164
Figura A5.5.1 Búsqueda de Usuarios.....	164
Figura A5.6 Creación de Usuarios.....	165
Figura A5.7 Permisos de Usuarios.....	166
Figura A5.8 Actualización y Eliminación de Usuarios.....	167
Figura A5.8.1 Búsqueda de Usuarios.....	167
Figura A5.9 Opciones del Usuario.....	168
Figura A5.9.1 Búsqueda de Usuarios.....	169
Figura A5.10 Departamentos	170
Figura A5.10.1 Usuarios Externos.....	170
Figura A5.11 Tipo de Documentos.....	171
Figura A5.12 Tipo de Trámites.....	172
Figura A5.12.1 Nuevo Trámite.....	173
Figura A5.13 Archivo Físico	174
Figura A5.13.1 Nuevo Archivo Físico.....	174
Figura A5.13.2 Editar Archivo Físico.....	175
Figura A5.14 Jefes / Directores.....	175

Figura A5.15 Envío de Documentos.....	176
Figura A5.15.1 Documentos Habilitantes.....	177
Figura A5.15.2 Buscar Contribuyentes Externos.	177
Figura A5.15.3 Anexos.	178
Figura A5.15.4 Tareas Internas.....	179
Figura A5.15.5 Botón de Envío	179
Figura A5.15.6 Selección de Direcciones.....	179
Figura A5.15.7 Opción Finalizar	180
Figura A5.15.8 Archivar.....	180
Figura A5.16 Documentos Recibidos	181
Figura A5.16.1 Documentos Recibidos 1	181
Figura A5.16.2 Documentos Recibidos 2	182
Figura A5.16.3 Documentos Recibidos 3	182
Figura A5.16.4 Botón Remitir	183
Figura A5.16.5 Remitir	183
Figura A5.17 Documentos Finalizados.....	183
Figura A5.18 Heredar Trámites	184
Figura A15.18.1 Consulta de Trámites	185
Figura A5.19 Archivo Documentos	185
Figura A5.20 Tareas Recibidas.....	186
Figura A5.20.1 Tareas Recibidas 1	186
Figura A5.21 Tareas Finalizadas	187
Figura A5.22 Hojas de Guía	187
Figura A5.23 Documentos Enviados	188
Figura A5.24 Envío General	188
Figura A5.25 Búsqueda Contribuyente Externo	189
Figura A5.26 Selección Departamentos.....	189
Figura A5.27 Consulta Control Remitente	190
Figura A5.28 Hoja de Guía Solicitada.....	190
Figura A5.29 Consulta General	191

Figura A5.30 Consulta de Trámites	191
Figura A5.31 Consulta General	192
Figura A6.1 Sistema Usuario	193
Figura A5.2 Conexión.....	193
Figura A5.3 Ventana Principal.....	194
Figura A6.4 Archivo Físico	195
Figura A6.5 Jefes / Directores.....	196
Figura A6.6 Envío de Documentos.....	196
Figura A6.7 Documentos Habilitantes.....	197
Figura A6.8 Buscar Contribuyente Externo.....	198
Figura A6.9 Anexar.....	199
Figura A6.10 Tareas Internas.....	199
Figura A6.11 Botón Envío	199
Figura A6.12 Selección de Direcciones.....	200
Figura A6.13 Opción Finalizar	200
Figura A6.14 Archivar	201
Figura A6.15 Documentos Recibidos	201
Figura A6.16 Documentos Recibidos 1	202
Figura A6.17 Documentos Recibidos 2	202
Figura A6.18 Documentos Recibidos	203
Figura A6.19 Botón Remitir	203
Figura A6.20 Remitir	204
Figura A6.21 Hoja de Guía	205
Figura A6.22 Documentos Enviados	205
Figura A6.23 Envío General	206
Figura A6.24 Buscar Contribuyente Externo.....	206
Figura A6.25 Búsqueda Destino	207
Figura A6.26 Consulta Control Remitente	207
Figura A6.27 Hoja de Guía Solicitada.....	208
Figura A6.28 Consulta General	208

Figura A6.29 Consulta por Trámites.....	209
Figura A6.30 Consulta General	209

ÍNDICE DE TABLAS

CONTENIDO	PÁGINA
Tabla 2.1 Microsoft SQL Server	15
Tabla 3.5.1 Operacionalización de Variable Independiente	22
Tabla 3.5.2 Operacionalización de Variable Dependiente.....	26
Tabla 4.2.1 Cuadro porcentual pregunta 2.....	30
Tabla 4.2.2 Cuadro porcentual pregunta 3.....	31
Tabla 4.2.3 Cuadro porcentual pregunta 5.....	32
Tabla 4.2.4 Cuadro porcentual pregunta 6.....	33
Tabla 4.2.5 Cuadro porcentual pregunta 9.....	34
Tabla 4.2.6 Cuadro porcentual pregunta 14.....	35
Tabla 6.1 Diagrama Caso Uso Cuentas de Usuario	48
Tabla 6.2 Parámetros Uso del Sistema.....	50
Tabla 6.3 Crear y recibir documentos	52
Tabla 6.4 Documentos ingresados al sistema.....	53
Tabla 6.5 Tabla aspnet_Membership.....	63
Tabla 6.6 Tabla aspnet_Roles	63
Tabla 6.7 Tabla aspnet_Users	63
Tabla 6.8 Tabla aspnet_UsersInRoles.....	64
Tabla 6.9 Tabla DEPARTAMENTOS.....	64
Tabla 6.10 Tabla FUNCIONARIOS.....	64
Tabla 6.11 Tabla DOCUMENTOS.....	65
Tabla 6.12 Tabla TRAMITES.....	65
Tabla 6.13 Tabla OFICIOS	65
Tabla 6.14 Tabla ANEXOS	66
Tabla 6.15 Tabla ARCHIVO	67
Tabla 6.16 Tabla DEPARTAMENTO_OFICIOS	67

Tabla 6.17	Tabla DOC_TRAMITES	67
Tabla 6.18	Tabla ARCHIVO_OFICIOS	67
Tabla 6.19	Tabla ESTADOFICIO	68
Tabla 6.20	Tabla EXTERNOS.....	68
Tabla 6.21	Tabla FLOW	68
Tabla 6.22	Tabla MENU	68
Tabla 6.23	Tabla FUNCIONARIO_MENU	69
Tabla 6.24	Tabla TMP_MOVIMIENTOS	69
Tabla 6.25	Tabla TMP_TIPO.....	69
Tabla 6.26	Cronograma de Implantación del Sistema	87

RESUMEN EJECUTIVO

La carencia de un sistema de control de documentos y trámites en el Municipio de Cevallos es evidente desde ya hace mucho tiempo pues al ingresar un documento no existe un respaldo de haberlo recibido así también si ya se está dando el tratamiento adecuado o está aún en circulación o lo que puede ser peor la pérdida del mismo.

La investigación se desarrolla dentro de la institución para revisar con qué tipo de infraestructura tanto en hardware y software así como de red de datos se cuenta.

El trabajo inicia con la investigación de la infraestructura hardware y software existente como es conocer qué tipo de servidor existe, con qué sistema operativo trabaja, si la institución tiene licencias de algún lenguaje de programación para el desarrollo del sistema o si se debe trabajar con un lenguaje de programación libre, con que motor de base de datos trabaja para el manejo de usuarios, como también de redes referente a componentes como medios y tecnología de transmisión de datos, sistema de cableado estructurado, aplicaciones soportadas por la red, acceso a Internet, entre otros, y básicamente lo referente a la infraestructura de red en lo que consiste a banda ancha; también se realizara un análisis del trafico de la red de datos para determinar la disponibilidad de ancho de banda para otros servicios de red y la factibilidad de implementar nuestro sistema que funcionara en red.

Esto es la base para aplicar una metodología de análisis y diseño de nuestro sistema en sí.

La incorrecta aplicación en lo referente al manejo de la documentación y de los trámites han impedido realizar un control exacto de los mismos provocando en algunos casos la amortización de cierto tramites importantes.

INTRODUCCIÓN

El tema del presente trabajo trata sobre el desarrollo e implementación de un Sistema Informático para dar seguimiento a documentos y trámites en el Ilustre Municipio de Cevallos.

Para lograr el objetivo principal de esta investigación, el presente trabajo está estructurado por seis capítulos: en el primer capítulo se plantea de forma clara y precisa el problema por el cual se decidió realizar el presente proyecto, se mencionan los objetivos que llevaron a la realización de la investigación, así como también una justificación con argumentos claros con los que se sustenta el porqué del presente trabajo.

Los antecedentes investigativos, las fundamentaciones legales, las categorías fundamentales que nos guiarán en la búsqueda de una posible solución al problema planteado: así como la definición de las variables dependientes e independientes, están contenidas en el segundo capítulo.

En el tercer capítulo, la modalidad y el nivel de la investigación, la población y muestra, recolección y procesamiento de la información son abarcados.

El análisis e interpretación de los resultados de las entrevistas realizadas a los usuarios de la Institución serán cubiertas en el capítulo cuatro.

El capítulo cinco se describe las conclusiones a las que se ha llegado y las recomendaciones que se realizarán al término de la investigación.

Finalmente en el sexto capítulo describe detalladamente la propuesta planteada al problema de investigación.

CAPÍTULO I

EL PROBLEMA

1.1 Planteamiento del problema

La necesidad de un sistema de seguimiento de trámites provoca un inadecuado control de documentos y trámites que se dan dentro del Municipio.

1.1.1 Contextualización

La mayoría de instituciones públicas del país cuentan con una infraestructura de red de comunicaciones de datos que no está bien utilizada. Un mayor desarrollo en la implementación de una red han logrado las instituciones que son cabezas cantonales como es el caso de Guayaquil, Quito, Ambato, etc.

Todas las alcaldías de los cantones de la provincia de Tungurahua tienen incorporada una adecuada infraestructura de red para su trabajo de intercambio de información entre dependencias.

En el Municipio de Cevallos existe una infraestructura de red adecuada para su trabajo como es: Comunicación entre departamentos también considerado dependencias como son; Cobros de Servicios municipales, Conexión a Internet entre otros.

El 70% de la información de las Empresas no está correctamente estructurada, lo que limita la posibilidad de utilizarla de forma ventajosa.

Los flujos de trabajo asociados a la gestión documental se administran de forma manual.

Generación y aprobación de órdenes de compra, autorizaciones de notas de pedidos, aprobaciones de créditos, procesos de reclamaciones, pago de facturas a proveedores, etc.

Hasta un 40% de su tiempo total dedican los profesionales en las empresas a tareas de gestión documental.

1.1.2 Análisis crítico

La administración manual de documentos y al no tener una constancia digital provoca pérdida de los mismos además demora en los trámites a los que hacen referencia.

Al no tener un respaldo digital de documentos y trámites puede generar la duplicación de los mismos por consiguiente esto provoca molestia a los usuarios ya que ellos van a ser los principales perjudicados y son quienes tendrán que asumir cualquier responsabilidad de ese error.

La institución no se ha preocupado en realizar la implementación de una herramienta que facilite la administración y el seguimiento correcto y oportuno a los documentos y trámites por lo que provoca pérdida de tiempo tanto a los empleados como a usuarios en general.

Las institución se verá beneficiada con el desarrollo y la implantación del sistema para seguimiento de documentos y trámites porque agilizará en los trámites que realizan y brindará un mejor y eficiente servicio a los usuarios quienes estarán conformes con la atención recibida.

1.1.3 Prognosis

De continuar con este problema en el Municipio seguirá existiendo un control inadecuado de la documentación, retraso en trámites y pérdida de información.

1.2 Formulación del problema

¿Qué beneficios tendrá la implementación de un Sistema Informático para dar seguimiento a documentos y trámites en el Ilustre Municipio de Cevallos?

1.2.1 Preguntas Directrices

¿Qué tipo de Sistema Operativo tiene el Servidor?

¿Qué lenguaje de programación es el más apropiado para nuestro trabajo?

¿La institución cuenta con licencias del lenguaje de programación que se va a utilizar?

¿Con qué Software de Base de Datos trabaja la institución?

¿La infraestructura presente es adecuada?

1.2.2 Delimitación del problema

El objeto de estudio es el área de sistemas y su campo de acción es un sistema de seguimiento de documentos y trámites.

El presente trabajo se desarrollará en el Ilustre Municipio de Cevallos ubicado en el centro de la ciudad del mismo nombre en las calles Felipa Real y Av. 24 de Mayo, el tiempo estimado para el desarrollo del proyecto es de seis meses.

1.3 Justificación

El estudio para el diseño del sistema de seguimiento de documentos y trámites se lo hace principalmente para tener un mayor control de los mismos además de evitar la pérdida de estos.

La alternativa más óptima es la implantación de un sistema de seguimiento de documentos y trámites en la actual infraestructura del Municipio de Cevallos.

Este convenio se lo ha realizado además para aplicar las bases teórico-práctico adquiridos en el transcurso de la vida universitaria, la misma que ayudara a fomentar la creatividad, iniciativa, una actitud positiva e innovadora y sobre todo buscando especializarse en el campo del desarrollo.

1.4 Objetivos

1.4.1 Objetivo General

- Diseñar e implantar un sistema informático para dar seguimiento a documentos y trámites para el control de los mismos en el Ilustre Municipio de Cevallos.

1.4.2 Objetivos Específicos

- Analizar la modalidad de dar seguimiento a los trámites en la actualidad en el Ilustre Municipio de Cevallos.
- Determinar el Hardware con las características especiales para el sistema.
- Diseñar la base de datos para la correcta utilización de la información existente.
- Automatizar los procesos de control de seguimiento a la documentación y trámites.
- Diseñar e implantar el sistema de manera correcta.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes Investigativos

Se debe manifestar que en el Ilustre Municipio de Cevallos, no se ha realizado temas relacionados con la implementación de un sistema de seguimiento de documentos y trámites, así como tampoco ha sido abordado por otros investigadores, constituyéndolo como un tema exclusivo, además en la Universidad Técnica de Ambato no se encontró temas similares de desarrollo de un sistema como el actual proyecto.

En la Municipalidad del Cantón Cevallos la secretaria general realiza un control manual de la documentación que ingresa, en un archivo de Excel normal para su propio beneficio y control.

Además cabe recalcar que en la actualidad existen muchos sistemas que realizan estas acciones como es el DocFlow, el mismo que servirá de apoyo para desarrollar el nuestro.

El presente trabajo se realizará en beneficio de la Municipalidad a través, del convenio de cooperación interinstitucional de pasantía entre la Ilustre Municipalidad de Cevallos y la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

“La Universidad Técnica de Ambato tiene como objetivo fundamental el de colaborar y participar con instituciones nacionales y extranjeras, previa suscripción de convenios para resolver problemas de interés común, sin menoscabo de su autonomía y soberanía nacional, de conformidad con el literal f) del Art.5 de su propio estatuto Universitario”

2.2 Fundamentos

2.2.1 Fundamentación Legal

Decreto N° 27

CONGRESO NACIONAL

EN PLENO DE LAS COMISIONES LEGISLATIVAS

Considerando

Que la parroquia Cevallos de la jurisdicción del Cantón Ambato, provincia de Tungurahua, ha alcanzado un alto nivel de desarrollo en lo urbanístico, poblacional, agrícola y comercial:

Que la citada parroquia cuenta con los recursos humanos, socio-económicos para ser erigida en Cantón; y por consiguiente, para administrarse por sí misma, lo que permitirá alcanzar el progreso eficiente en su beneficio y del país.

Que la Comisión Especial de Límites Internos de la República ha emitido el informe correspondiente al área territorial y límites del Cantón Cevallos y del H. Consejo provincial de Tungurahua ha informado favorablemente sobre su cantonización.

Que una vez que se ha efectuado un detenido estudio de la realidad geográfica, económica, social y política de la Parroquia Cevallos, se ha llegado a la conclusión de que es conveniente elevarla a la categoría de Cantón; y,

En uso de las atribuciones constitucionales de que se halla investido, expide la siguiente.

LEY DE CREACION DE CANTON CEVALLOS

Art 1.- Créase el Cantón Cevallos en la provincia del Tungurahua, cuya cabecera cantonal será la parroquia del mismo nombre.

Art 2.- La jurisdicción política-administrativa del nuevo Cantón Cevallos, comprenderá la cabecera cantonal y las parroquias que se creasen.

Art 3.- Los límites del Cantón Cevallos serán los siguientes:

POR EL NORTE.- desde la unión del Camino Real con la Quebrada Palagua con el Rio Pachanlica; por éste, aguas abajo, hasta la confluencia con el Rio Pachanlica.

POR EL ESTE.- Confluencia de la Quebrada Palagua con el Rio Pachanlica, por éste, aguas arriba, pasando el puente Quero hasta encontrar la recta del camino Quero-Mocha.

POR EL SUR.- Desde el punto más bajo del Rio Quero-Mocha por la línea imaginaria a encontrar el camino Quero-Mocha y por éste pasando por el ramo de la acequia Mocha-Tisaleo Cevallos, sigue por este camino avanzando la línea férrea, el mismo camino que divide las poblaciones San Pedro de Cevallos, El Mirador de Cevallos y Santo Domingo de Cevallos con las poblaciones de El Rosal de Mocha y Yanahurco de Mocha, sigue por este camino hasta empalmar con el camino del Rey (Real) en el punto denominado Cuatro Esquinas.

POR EL OESTE.- Desde el punto denominado Cuatro Esquinas hacia el Norte siguiendo el camino el Rey (Real) hasta empalmar con la Quebrada de Palagua.

Art 4.- Sin perjuicio de la asignación de los fondos provenientes de la explotación petrolera para el desarrollo de los organismos seccionales, dispuesto en el Decreto sin número del 19 de Noviembre de 1979, publicado en el Registro Oficial N° 113 del 24 de Enero de 1980, el Municipio de Cevallos percibirá además por esta sola vez la asignación de Treinta Millones de Suces con aplicación al Fondo Nacional de Participación.

DISPOSICIONES TRANSITORIAS

PRIMERA.- El Tribunal Supremo Electoral, convocara a elecciones de Concejales del Cantón Cevallos en el plazo de noventa días, contados a partir de la vigencia de la presente Ley. Los electos permanecerán en funciones hasta que se realicen las próximas elecciones generales de Presidente de consejo y Concejales; y estos se posesionaran legalmente en sus funciones.

SEGUNDA.- Todas las actividades y gestiones de carácter administrativas, fiscal y municipal del Cantón Cevallos, no serán suspendidas por falta de autoridades de elección popular o de designación gubernamental hasta que sean legalmente designadas; posesionadas o hayan entrado oficialmente en ejercicio de sus funciones. Hasta tal evento, dichas actividades y gestiones serán atendidas por las autoridades pertinentes de la Provincia del Tungurahua y el Cantón Ambato, en la forma en que se venía haciendo antes de la erección cantonal.

Art. FINAL.- La presente Ley entrará en vigencia a partir de su publicación en el Registro Oficial.

Dada en Quito en la Sala de Sesiones del Plenario de las Comisiones Legislativas, a los diecisiete días del mes de abril de mil novecientos ochenta y seis.

Firman: Bucaram Záceida, Presidente del Congreso Nacional, Wilson Córdova Loor, Prosecretario del Congreso Nacional.

Certifico que la “Ley de Creación del Cantón Cevallos” queda Sancionada de Hecho de conformidad con lo dispuesto por el Art. 68 de la Constitución Política.

Presidencia de la república

Ley de software libre

Mediante el decreto 1014 emitido el día jueves 10 de abril del 2008 por parte de la presidencia del Eco. Rafael Correa Delgado que promueve el uso de software libre en las instituciones públicas del Ecuador.

Art. 1: Establecer como política pública para las entidades de administración Pública Central la utilización del Software Libre en sus sistemas y equipamientos informáticos.

Art. 2: Se entiende por software libre, a los programas de computación que se pueden utilizar y distribuir sin restricción alguna, que permitan el acceso a los códigos fuentes y que sus aplicaciones puedan ser mejoradas.

Estos programas de computación tienen las siguientes libertades:

- Utilización de programa con cualquier propósito de uso común.
- Distribución de copias sin restricción alguna.
- Estudio y modificación de programa (Requisito: código fuente disponible).
- Publicación del programa mejorado (Requisito: código fuente disponible).

Art. 3: Las entidades de la administración pública central previa a la instalación del software libre en sus equipos, deberán verificar la existencia de capacidad técnica que brinde el soporte necesario para este tipo de software.

Art. 4: Se faculta la utilización de software propietario (no libre) únicamente cuando no exista una solución de software libre que supla las necesidades requeridas, o cuando este en riesgo de seguridad nacional, o cuando el proyecto informático se encuentre en un punto de no retorno.

2.2.2 Fundamentación Teórica

2.2.2.1 Sistema Informático

2.2.2.1.1 Software

Programas de computadoras. Son las instrucciones responsables de que el hardware (la máquina) realice su tarea. Como concepto general, el software puede dividirse en varias categorías basadas en el tipo de trabajo realizado. Las dos categorías primarias de software son los sistemas operativos (software del sistema), que controlan los trabajos del ordenador o computadora, y el software de aplicación, que dirige las distintas tareas para las que se utilizan las computadoras.

Por lo tanto, el software del sistema procesa tareas tan esenciales, aunque a menudo invisibles, como el mantenimiento de los archivos del disco y la administración de la pantalla, mientras que el software de aplicación lleva a cabo tareas de tratamiento de textos, gestión de bases de datos y similares. Constituyen dos categorías separadas el software de red, que permite comunicarse a grupos de usuarios, y el software de lenguaje utilizado para escribir programas (véase Lenguaje de programación).

(Microsoft ® Encarta ® 2009. © 1993-2008 Microsoft Corporation).

2.2.2.1.2 Informática

Conjunto de conocimientos científicos y de técnicas que hacen posible el tratamiento automático de la información por medio de computadoras. La informática combina los aspectos teóricos y prácticos de la ingeniería, electrónica, teoría de la información, matemáticas, lógica y comportamiento humano. Los aspectos de la informática cubren desde la programación y la arquitectura informática hasta la inteligencia artificial y la robótica.

(Microsoft ® Encarta ® 2009. © 1993-2008 Microsoft Corporation).

2.2.2.1.3 Sistemas de Información

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.

Actividades de los sistemas de información

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

- **Entrada de información:** Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas.
- **Almacenamiento de información:** El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya

que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior.

- **Procesamiento de información:** Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones.
- **Salida de información:** La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior.

(Sistema de Información (Manuel Peralta). Recuperado el 05 enero 2012, desde www.monografias.com/trabajos7/sisinf/sisinf.shtml).

2.2.2.1.4 Arquitectura Informática

Término general que se aplica a la estructura de un sistema informático o de una parte del mismo. El término se aplica también al diseño del software de sistema, por ejemplo, el sistema operativo, y a la combinación de hardware y *software* básico que comunica los aparatos de una red informática. La arquitectura de ordenadores se refiere a toda una estructura y a los detalles necesarios para que sea funcional, es decir, cubre sistemas informáticos, microprocesadores, circuitos y programas del sistema. Por lo general, el término no suele referirse a los programas de aplicación, como hojas de cálculo o procesadores de textos, que son necesarios para realizar una tarea pero no para que el sistema funcione.

(Microsoft ® Encarta ® 2009. © 1993-2008 Microsoft Corporation).

2.2.2.1.5 Arquitectura de Software

Componentes de software:

Se distinguen tres componentes básicos de software:

- **Presentación.-** Tiene que ver con la presentación al usuario de un conjunto de objetos visuales y llevar a cabo el procesamiento de los datos producidos por el mismo y los devueltos por el servidor.
- **Lógica de aplicación.-** Esta capa es la responsable del procesamiento de la información que tiene lugar en la aplicación.
- **Base de datos.-** Esta compuesta de los archivos que contienen los datos de la aplicación.

(Aplicaciones distribuidas. Recuperado 05 Enero 2012, desde <http://knol.google.com/k/aplicaciones-distribuidas#>).

2.2.2.1.6 Lenguaje de Programación

Es cualquier lenguaje artificial que puede utilizarse para definir una secuencia de instrucciones para su procesamiento por un ordenador o computadora. Es complicado definir qué es y qué no es un lenguaje de programación. Se asume generalmente que la traducción de las instrucciones a un código que comprende la computadora debe ser completamente sistemática. Normalmente es la computadora la que realiza la traducción.

Lenguajes de Bajo Nivel

Vistos a muy bajo nivel, los microprocesadores procesan exclusivamente señales electrónicas binarias. Dar una instrucción a un microprocesador supone en realidad enviar series de unos y ceros espaciadas en el tiempo de una forma determinada. Esta secuencia de señales se denomina código máquina. El código representa normalmente datos y números e instrucciones para manipularlos. Un modo más fácil de comprender el código máquina es dando a cada instrucción un mnemónico, como por ejemplo STORE, ADD o JUMP. Esta abstracción da como resultado el ensamblador, un lenguaje de muy bajo nivel que es específico de cada microprocesador.

Lenguajes de Alto Nivel

Por lo general se piensa que los ordenadores son máquinas que realizan tareas de cálculos o procesamiento de textos. La descripción anterior es sólo una forma

muy esquemática de ver una computadora. Hay un alto nivel de abstracción entre lo que se pide a la computadora y lo que realmente comprende. Existe también una relación compleja entre los lenguajes de alto nivel y el código máquina.

Intérpretes y Compiladores

La traducción de una serie de instrucciones en lenguaje ensamblador (el código fuente) a un código máquina (o código objeto) no es un proceso muy complicado y se realiza normalmente por un programa especial llamado compilador. La traducción de un código fuente de alto nivel a un código máquina también se realiza con un compilador, en este caso más complejo, o mediante un intérprete. Un compilador crea una lista de instrucciones de código máquina, el código objeto, basándose en un código fuente. El código objeto resultante es un programa rápido y listo para funcionar, pero que puede hacer que falle el ordenador si no está bien diseñado. Los intérpretes, por otro lado, son más lentos que los compiladores ya que no producen un código objeto, sino que recorren el código fuente una línea cada vez. Cada línea se traduce a código máquina y se ejecuta. Cuando la línea se lee por segunda vez, como en el caso de los programas en que se reutilizan partes del código, debe compilarse de nuevo. Aunque este proceso es más lento, es menos susceptible de provocar fallos en la computadora

(Microsoft ® Encarta ® 2009. © 1993-2008 Microsoft Corporation).

2.2.2.1.7 SharpDevelop

Es un entorno de desarrollo integrado libre para los lenguajes de programación C#, Visual Basic .NET y Boo.

Es usado típicamente por aquellos programadores de los citados lenguajes, que no desean o no pueden usar el entorno de desarrollo de Microsoft, el Microsoft Visual Studio. Hay disponible un port para Mono/Gtk#, llamado MonoDevelop, el cual funciona en otros sistemas operativos.

Para el completado automático de código, la aplicación incorpora sus propios parsers. La versión 1.1 de la aplicación puede importar proyectos de Visual Studio

.NET. La versión 2.0 ya es capaz de editarlos directamente. La versión 3.0 integra soporte para python y f#.

(Discusión:SharpDevelop . Recuperado 05 enero 2012, desde <http://es.wikipedia.org/w/index.php?title=Discusi%C3%B3n:SharpDevelop>)

2.2.2.1.8 Base de Datos

Hola conjunto de datos organizados para su almacenamiento en la memoria de un ordenador o computadora, diseñado para facilitar su mantenimiento y acceso de una forma estándar. La información se organiza en campos y registros. Un campo se refiere a un tipo o atributo de información, y un registro, a toda la información sobre un individuo. Por ejemplo, en una base de datos que almacene información de tipo agenda, un campo será el NOMBRE, otro el NIF, otro la DIRECCIÓN..., mientras que un registro viene a ser como la ficha en la que se recogen todos los valores de los distintos campos para un individuo, esto es, su nombre, NIF, dirección... Los datos pueden aparecer en forma de texto, números, gráficos, sonido o vídeo. Normalmente las bases de datos presentan la posibilidad de consultar datos, bien los de un registro o los de una serie de registros que cumplan una condición. También es frecuente que se puedan ordenar los datos o realizar operaciones sencillas, aunque para cálculos más elaborados haya que importar los datos en una hoja de cálculo. Para facilitar la introducción de los datos en la base se suelen utilizar formularios; también se pueden elaborar e imprimir informes sobre los datos almacenados.

Desde su aparición en la década de 1950, estas aplicaciones se han hecho imprescindibles para las sociedades industriales. La primera base de datos para PC data de 1980; era el DBase II, desarrollado por el ingeniero estadounidense Wayne Ratliff. Desde entonces, su evolución ha seguido paralela a la que ha experimentado el *software*, y hoy existen desde bases de datos para una utilización personal hasta bases de datos corporativas, soportadas por grandes sistemas informáticos.

Hay cuatro modelos principales de bases de datos: el modelo jerárquico, el modelo en red, el modelo relacional (el más extendido hoy día; los datos se

almacenan en tablas y se accede a ellos mediante consultas escritas en SQL) y el modelo de bases de datos deductivas. Otra línea de investigación en este campo son las bases de datos orientadas a objeto, o de objetos persistentes.

(Microsoft ® Encarta ® 2009. © 1993-2008 Microsoft Corporation).

2.2.2.1.9 Microsoft SQL Server

Es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL.

SQL Server Historia de Liberaciones			
Versión	Año	Nombre de la Liberación	Nombre Clave
1.0 (OS/2)	1989	SQL Server 1.0	-
4.21 (WinNT)	1993	SQL Server 4.21	-
6.0	1995	SQL Server 6.0	SQL95
6.5	1996	SQL Server 6.5	Hydra
7.0	1998	SQL Server 7.0	Sphinx
-	1999	SQL Server 7.0 OLAP Tools	Plato
8.0	2000	SQL Server 2000	Shiloh
8.0	2003	SQL Server 2000 64-bit Edition	Liberty
9.0	2005	SQL Server 2005	Yukon
10.0	2008	SQL Server 2008	Katmai
10.5	2010	SQL Server 2008 R2	Kilimanjaro

Tabla 2.1 Microsoft SQL Server

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son *Oracle*, *Sybase ASE*, *PostgreSQL*, *Interbase*, *Firebird* o *MySQL*.

Lo nuevo de SQL Server 2008

La nueva base de datos contiene mayor seguridad, integración con PowerShell, remueve La consola configuración del área expuesta (consola para configurar seguridad), tiene correctores de sintaxis del lenguaje Transact-SQL e

intellisense (una característica del visual studio que permite a la base de datos sugerir objetos existentes mientras uno escribe la mitad de la palabra). Así mismo incluye nuevos tipos de datos y funciones.

(Microsoft SQL Server (Wikipedia). Recuperado 05 enero 2012, desde http://es.wikipedia.org/w/index.php?title=Microsoft_SQL_Server).

2.2.2.1.10 SQL Server 2000

Es un potente motor de bases de datos de alto rendimiento capaz de soportar millones de registros por tabla con un interface intuitivo y con herramientas de desarrollo integradas como Visual Studio 6.0 o .NET, además incorpora un modelo de objetos totalmente programable (SQL-DMO) con el que podemos desarrollar cualquier aplicación que manipule componentes de SQL Server, es decir, hacer aplicación para crear bases de datos, tablas, DTS, backups, etc., todo lo que se puede hacer desde el administrador del SQL Server y podemos hacerlo no solo en Visual C++ sino también en Visual Basic, ASP y por supuesto en .NET.

Pero cuidado, que sea muy intuitivo en su administración o instalación no significa que sea fácil, una mala instalación, una base de datos mal creada o diseñada o una mala administración nos puede hacer la vida imposible y nuestras aplicaciones pueden tener un rendimiento malo, debemos tener cuidado y aprender a usarlo correctamente, como también es importante el hardware, lejos de los 64 MB mínimos que requiere el sistema es recomendable que tenga 256 o 512 para su buen funcionamiento y una cantidad suficiente de espacio en disco para que pueda trabajar con las bases de datos.

2.2.2.1.11 Funciones en SQL Server 2000

Microsoft agregó nuevas características a su producto SQL 2000, y lo más interesante para los programadores del SQL es la posibilidad de hacer funciones definidas por el usuario. La adición de funciones al lenguaje del SQL solucionara los problemas de reutilización del código y dará mayor flexibilidad al programar las consultas de SQL.

Este artículo se centra en la sintaxis, estructura, y la aplicación de funciones definidas por el usuario en SQL. El material presentado se basa en el beta 2 del servidor 2000 del SQL.

(Funciones en SQL Server 2000. Recuperado el 05 enero 2012, desde <http://www.sqlmax.com/func1.asp>, <http://www.sqlmax.com/func2.asp>)

2.2.2.2 Control de la Documentación

2.2.2.2.1 Documento

Un documento es el testimonio material de un hecho o acto realizado en el ejercicio de sus funciones por instituciones o personas físicas, jurídicas, públicas o privadas, registrado en una unidad de información en cualquier tipo de soporte (papel, cintas, discos magnéticos, películas, fotografías, etcétera) en lengua natural o convencional. Es el testimonio de una actividad humana fijada en un soporte.

Cuando los documentos se producen para atender a las competencias y actividades encomendadas a las organizaciones e instituciones, y se realizan a lo largo del tiempo, adquieren su carácter seriado y se denominarán series documentales (actas, libros de contabilidad, correspondencia, etcétera).

(Wikipedia. Documento. Recuperado 05 de enero 2012, desde <http://es.wikipedia.org/wiki/Documento>).

2.2.2.2.2 Trámite

Conjunto o serie de pasos o acciones reguladas por el Estado que deben efectuar los usuarios para adquirir un derecho o cumplir con una obligación prevista o autorizada en la Ley.

Un trámite es un proceso, diligencia, solicitud o entrega de información que las personas hacen frente a una dependencia, bien sea para cumplir obligaciones o para obtener beneficios o servicios. Por lo general los trámites producen documentos.

(www.jenesano-boyaca.gov.co/glosario.shtml)

2.2.2.2.3 Automatización

Sistema de fabricación diseñado con el fin de usar la capacidad de las máquinas para llevar a cabo determinadas tareas anteriormente efectuadas por seres humanos, y para controlar la secuencia de las operaciones sin intervención humana. El término automatización también se ha utilizado para describir sistemas no destinados a la fabricación en los que dispositivos programados o automáticos pueden funcionar de forma independiente o semiindependiente del control humano. En comunicaciones, aviación y astronáutica, dispositivos como los equipos automáticos de conmutación telefónica, los pilotos automáticos y los sistemas automatizados de guía y control se utilizan para efectuar diversas tareas con más rapidez o mejor de lo que podría hacerlo un ser humano.

(Microsoft ® Encarta ® 2009. © 1993-2008 Microsoft Corporation).

2.2.2.2.4 DocFlow

Es una herramienta personalizada por el usuario. La información relacionada a un tipo documental se define en lo que llamamos una “plantilla”. Una plantilla incluye un formulario (los valores que se quieren guardar de un documento), el flujo de aprobación, reportes, etc.

DocFlow le permite definir y administrar los flujos de aprobación de documentos electrónicos de su empresa con una mínima intervención del personal técnico.

Desde su navegador usted podrá definir fácilmente: formularios, flujos de aprobación y reportes.

DocFlow le permite a su empresa centralizar la información de sus flujos de aprobación con la seguridad necesaria para garantizar el acceso al personal que lo necesita.

(DocFlow. Junta de leones y Castilla. Recuperado 05 de enero 2012 desde, <http://www.giro.infor.uva.es/oldsite/docproy/wfs/presentacionDocFlow.htm>).

2.3 Hipótesis

La implantación de un sistema informático para dar seguimiento de trámites permitirá el mejor control de la documentación que se maneja dentro del I. Municipio de Cevallos.

2.4 Variables

2.4.1 Variable Independiente

Sistema Informático.

2.4.2 Variable Dependiente

Control de la Documentación.

CAPÍTULO III

METODOLOGÍA

3.1 Enfoque

El enfoque de la investigación es eminentemente cuantitativa ya que el investigador conoce, analiza y toma las decisiones más óptimas con las técnicas adecuadas para solucionar el problema y la población nos proporciona la información necesaria para alcanzar este objetivo. La investigación dentro de la institución será realizada por el técnico para solucionar el problema que será proporcionado por la población interesada en solucionarlo.

3.2 Modalidad básica de la Investigación

3.2.1 Investigación de Campo

Esta investigación permite el estudio sistemático de los hechos en el lugar en que se producen los acontecimientos, el investigador toma contacto en forma directa con la realidad, para tener informes de acuerdo con los objetivos del problema.

3.2.2 Investigación Documental – Bibliográfica

Esta modalidad permite conocer, comparar, ampliar, profundizar y deducir diferentes enfoques, teorías, conceptualizaciones y criterios de diversos autores para el diseño del presente programa para dar seguimiento a documentos y trámites, basándose en documentos (fuentes primarias), libros así también como

en Internet (fuentes secundarias) que se recomienda para estudios sociales, geográficos, históricos, geopolíticos, literarios, entre otros.

3.2.3 Proyecto Factible

Se realizara una propuesta en base al desarrollo de un sistema para dar seguimiento a documentos y trámites que es un modelo practico que permitirá solucionar los problemas detectados en el Municipio de Cevallos previo el diagnostico realizado con anterioridad al transcurso de la investigación y sustentación en el marco teórico.

3.3 Niveles o Tipos de Investigación

La investigación utilizará un nivel exploratorio para detectar las características del problema, determinar si es factible o no solucionarse; pasaremos al nivel descriptivo para conocer con mayor profundidad las circunstancias y la realidad en la que se desarrolla el problema; el nivel correlacional, facilita la comprensión, el análisis y el estudio del fenómeno dentro de un contexto determinado para la construcción de la base teórica de las variables.

3.4 Población y Muestra

3.4.1 Población

La población está conformada por 8 personas quienes son: Jefe Administrativo, Guarda Almacén, Jefe de Planificación, Servidor Público de Apoyo, Jefe de Avalúos y Catastros, Procurador Síndico, Técnico de Sistemas, Jefe de Unidad de Agua Potable.

3.4.2 Muestra

Como la población es pequeña todos pasan a constituir la muestra.

3.5 Operacionalización de las Variables

3.5.1 Variable Independiente

Conceptualización	Dimensión	Indicadores	Ítems	Tec-Inst
<p>Variable Independiente:</p> <p>Sistemas de Información.- Conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.</p>	Elementos que interactúan	Programa que facilitan la labor del usuario.	<p>¿La empresa dispone de equipos de cómputo adecuados para su trabajo? Si <input type="checkbox"/> No <input type="checkbox"/></p> <p>¿Los equipos de cómputo están interconectados a través de una red? Si <input type="checkbox"/> No <input type="checkbox"/></p> <p>¿Conoce la diferencia entre software Libre y Propietario? Si <input type="checkbox"/> No <input type="checkbox"/></p> <p>¿Conoce con que sistema operativo trabaja el servidor? SO Libre <input type="checkbox"/> SO Propietario <input type="checkbox"/></p> <p>¿Ha utilizado software Libre alguna vez? Si <input type="checkbox"/> No <input type="checkbox"/></p>	<p>Entrevista</p> <p>Entrevista</p> <p>Entrevista</p> <p>Entrevista</p> <p>Entrevista</p>

	Apoyo	Información para ser aplicada	¿De qué manera se maneja la información de los documentos y tramites en el Municipio de Cevallos? Manualmente <input type="checkbox"/> Excel <input type="checkbox"/> Otro: _____.	Entrevista
	Actividades	Manejo de Información	¿Qué tipo de información se maneja en la institución en lo que respecta a trámites y documentos? _____.	Entrevista
	Empresas y negocios	Medios de Transmisión	¿Qué medio de transmisión utiliza para la interconexión de Pc's? Cableado Inalámbrico Otro: _____.	Entrevista
		Servicios de Red	La red de comunicación se utiliza para: Datos Videos <input type="checkbox"/> Imágenes <input type="checkbox"/> Otros: _____.	Entrevista
			¿El servicio de red es óptimo para su trabajo?	Entrevista

			Si <input type="checkbox"/> No <input type="checkbox"/> Porque: _____.	
			¿Dispone de acceso a Internet? Si <input type="checkbox"/> No <input type="checkbox"/>	Entrevista
			¿Cuál es su proveedor del servicio de Internet? Punto Net <input type="checkbox"/> CNT <input type="checkbox"/> Speedy <input type="checkbox"/> InterActive <input type="checkbox"/> Portal <input type="checkbox"/> Otros: _____.	Entrevista
			¿Con que dispositivos IP trabaja la Empresa? Switches <input type="checkbox"/> Routers <input type="checkbox"/> Puentes <input type="checkbox"/> Encaminadores <input type="checkbox"/> Otro: _____.	Entrevista
	Arquitectura de Software	Arquitectura cliente-servidor	¿Conoce Ud. lo que es arquitectura n capas? Si <input type="checkbox"/> No <input type="checkbox"/>	Entrevista

			<p>¿Con qué arquitectura de software considera Ud. que se debería desarrollarse el sistema?</p> <p> Cliente-Servidor <input type="checkbox"/> Cliente-Servidor 2 Capas <input type="checkbox"/> Cliente-Servidor 3 Capas <input type="checkbox"/> Cliente-Servidor n Capas <input type="checkbox"/> </p>	Entrevista
	Lenguaje de Programación	Código fuente	<p>¿Con qué lenguaje de programación considera Ud. que se debería desarrollar el sistema?</p> <p>_____.</p>	Entrevista
	Base de Datos	Manejo de Datos	<p>¿Qué tipos de motor de Base de Datos se encuentra instalado en el Servidor?</p> <p>_____.</p>	Entrevista
			<p>¿La información de los usuarios se encuentra almacenada en alguna base de datos?</p> <p> Si <input type="checkbox"/> Parcialmente <input type="checkbox"/> No <input type="checkbox"/> </p>	Entrevista

3.5.2 Variable Dependiente

Conceptualización	Dimensión	Indicadores	Ítems	Tec-Inst	
Control de la Documentación.- Comprobación e inspección de todo tipo de documento y trámite que ingresa y circula en la institución.	Documento	Documentación	¿Qué tipo de documentos ingresan y circulan en la institución? _____	Entrevista	
	Trámite	Trámites	¿Qué tipo de Trámites ingresan y circulan en la institución? _____	Entrevista	
	Automatización	Tratamiento de la Información	¿Cómo se maneja la información actualmente en la institución? _____	Entrevista	
	DocFlow			¿Sabe que es DocFlow? Si <input type="checkbox"/> No <input type="checkbox"/>	Entrevista
				¿Conoce como funciona un DocFlow? Si <input type="checkbox"/> No <input type="checkbox"/>	Entrevista
				¿Conoce de software DocFlow existente en el mercado? Si <input type="checkbox"/> No <input type="checkbox"/>	Entrevista

3.6 Recolección de información

3.6.1 Plan de recolección de información

La recolección de Información se realizara mediante unas Entrevista realizadas a ocho miembros principales de la institución que son: Jefe Administrativo, Guarda Almacén, Jefe de Planificación, Servidor Público de Apoyo, Jefe de Avalúos y Catastros, Procurador Síndico, Técnico de Sistemas, Jefe de Unidad de Agua Potable.

3.6.2 Procesamiento y análisis de información

El análisis de la información se realizará mediante la interpretación de los datos recolectados, los cuales al ser procesados permitirá obtener un informe en base a sus resultados, el cual permitirá plantear conclusiones y recomendaciones para dar solución al problema trazado.

3.7 Plan de análisis e interpretación de los resultados

Los resultados obtenidos a través de las Entrevistas realizadas a los miembros del Municipio de Cevallos se los representará mediante la utilización de gráficos estadísticos, los cuales servirán para dar solución al problema planteado, por último se elaborará una síntesis general para la elaboración de las conclusiones y recomendaciones.

CAPÍTULO IV

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1 Análisis de la necesidad

El Ilustre Municipio de Cevallos al ser una Entidad Pública tiene la obligación de brindar un servicio eficiente en cuanto a los documentos y trámites que los usuarios hacen llegar día tras día a la Institución.

El Ilustre Municipio de Cevallos tiene la factibilidad de aceptar alumnos de los diferentes establecimientos educativos para realización de prácticas o proyectos de grado en los distintos departamentos de la Institución, se permite a un estudiante de la Universidad Técnica de Ambato de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial realice el trabajo de graduación y proceda a dar solución a los problemas de la Institución.

La carencia de un sistema de control de documentos y trámites en el Ilustre Municipio de Cevallos se evidencia pues al ingresar un documento no existe un respaldo de haberlo recibido, la necesidad de un sistema que apoye al desempeño de los trabajadores, de manera que ayude a entregar valor en sus procesos y de esta manera efectividad en la prestación de los servicios para los usuarios de la Institución.

4.2 Análisis de Resultados

La técnica que se utilizó fue la Entrevista con dos formatos distintos, el uno dirigido al Jefe Administrativo y el otro formato al personal municipal; que incluyó preguntas combinadas, las mismas que fueron tabuladas y analizadas para obtener criterios para la investigación.

Entrevista realizada al Jefe Administrativo

1. ¿Qué medio de transmisión utiliza para la interconexión de Pc's?

El Entrevistado respondió: Cableado.

2. ¿Cuál es su proveedor del servicio de Internet?

El Entrevistado respondió: CNT.

3. ¿Con que dispositivos IP trabaja la Empresa?

El Entrevistado respondió: Switches y Encaminadores.

4. ¿Conoce Ud. lo que es arquitectura n capas?

El Entrevistado respondió: NO.

5. ¿Con qué arquitectura de software considera Ud. que se debería desarrollar el sistema?

El Entrevistado respondió: Cliente-Servidor 3 Capas.

6. ¿Con qué lenguaje de programación considera Ud. que se debería desarrollar el sistema?

El Entrevistado respondió: Lenguaje Open Suse.

7. ¿Qué tipos de motor de Base de Datos se encuentra instalado en el Servidor?

El Entrevistado respondió: SQL Server 2000.

8. ¿La información de los usuarios se encuentra almacenada en alguna base de datos?

El Entrevistado respondió: SI

9. ¿Conoce como funciona un DocFlow?

El Entrevistado respondió: SI

Entrevista realizada al personal institucional del Ilustre Municipio de Cevallos. Cabe indicar que se han seleccionado para el análisis las preguntas más relevantes; las preguntas tabuladas son las siguientes:

2. ¿Los equipos de Cómputo están interconectados entre sí a través de una red?

Objetivo:

Conocer si están los equipos de Cómputo se conectan a través de una red.

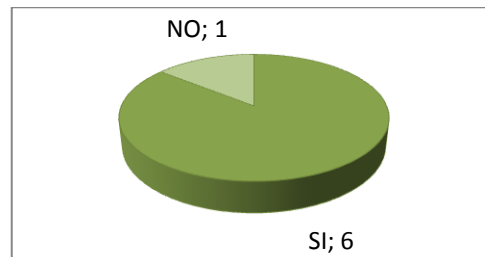


Figura 4.2.1 Gráfico pregunta 2

Respuesta	Cantidad	Porcentaje
SI	6	86%
NO	1	14%
Total	7	100%

Tabla 4.2.1 Cuadro porcentual pregunta 2

Interpretación:

Considerado el grafico se puede constatar que el 86% de las personas entrevistadas opinan que los equipos se interconectan a través de una red y un 14 % opina que no.

Análisis

Se demuestra que la mayor parte del personal conoce sobre la interconexión de equipos a través de una red.

3. ¿Conoce la diferencia entre software Libre y Propietario?

Objetivo:

Conocer si los funcionarios distinguen entre software Libre y software Propietario.

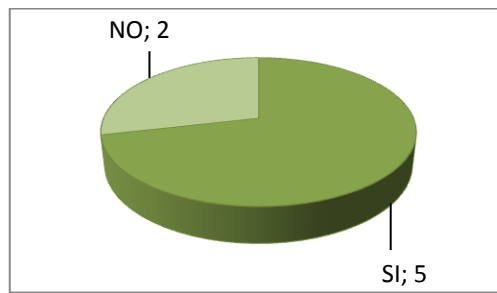


Figura 4.2.2 Gráfico pregunta 3

Respuesta	Cantidad	Porcentaje
SI	5	71%
NO	2	29%
Total	7	100%

Tabla 4.2.2 Cuadro porcentual pregunta 3

Interpretación:

Considerado el grafico se puede constatar que el 71% de las personas entrevistadas conocen la diferencia entre software Libre y propietario.

Análisis

Se demuestra que la mayor parte del personal conoce la diferencia entre software Libre y software propietario.

5. ¿Ha utilizado software Libre alguna vez?

Objetivo:

Conocer si el personal Institucional ha utilizado software libre.

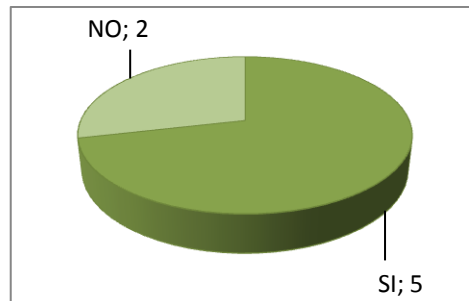


Figura 4.2.3 Gráfico pregunta 5

Respuesta	Cantidad	Porcentaje
SI	5	71%
NO	2	29%
Total	7	100%

Tabla 4.2.3 Cuadro porcentual pregunta 5

Interpretación:

Considerado el gráfico se puede constatar que el 71% de las personas entrevistadas han utilizado alguna vez Software Libre.

Análisis

Se demuestra que la mayor parte del personal a utilizado software Libre y sabe cómo funciona este software.

6. ¿De qué manera se maneja la información de los documentos y trámites en el Municipio de Cevallos?

Objetivo:

Determinar cómo se maneja los documentos y trámites en la Institución.

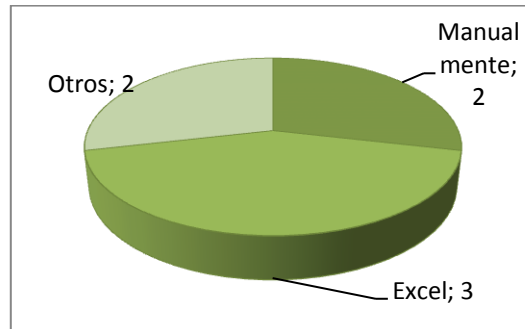


Figura 4.2.4 Gráfico pregunta 6

Respuesta	Cantidad	Porcentaje
Manualmente	2	29%
Excel	3	42%
Otros	2	29%
Total	7	100%

Tabla 4.2.4 Cuadro porcentual pregunta 6

Interpretación:

Considerado el gráfico se puede constatar que el 29% de las personas entrevistadas manejan los documentos y trámites manualmente, el 42% por medio de Excel y un 29% por otros medios.

Análisis

Se demuestra que la mayor parte del personal utiliza Excel para manejar y manipular los documentos y trámites que ingresan a la Institución.

9. ¿El servicio de red es óptimo para su trabajo?

Objetivo:

Conocer que opina el personal sobre el servicio de red de la Institución.

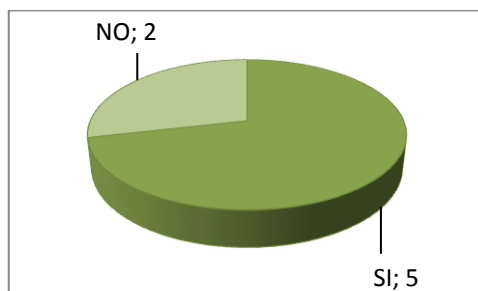


Figura 4.2.5 Gráfico pregunta 9

Respuesta	Cantidad	Porcentaje
SI	5	71%
NO	2	29%
Total	7	100%

Tabla 4.2.5 Cuadro porcentual pregunta 9

Interpretación:

Considerado el gráfico se puede constatar que el 71% de las personas entrevistadas opinan que el servicio de red es óptimo para el trabajo que ellos realizan.

Análisis

Se demuestra que la mayor parte del personal a considera que el servicio de red es óptimo y ayuda con su trabajo diario.

14. ¿Sabe que es DocFlow?

Objetivo:

Conocer si están los equipos de Cómputo se conectan a través de una red.

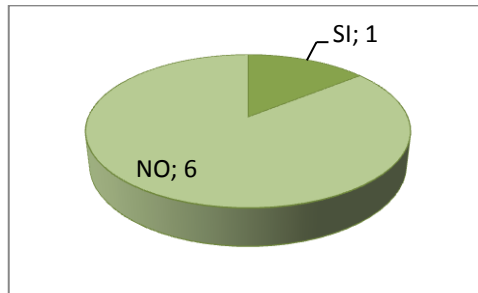


Figura 4.2.6 Gráfico pregunta 14

Respuesta	Cantidad	Porcentaje
SI	1	14%
NO	6	86%
Total	7	100%

Tabla 4.2.6 Cuadro porcentual pregunta 14

Interpretación:

Considerado el grafico se puede constatar que el 86% de las personas entrevistadas no conocen o no saben lo que es DocFlow.

Análisis

Se demuestra el personal en su mayoría no conoce el sistema DocFlow.

4.3 Interpretación de resultados

Los funcionarios elegidos para la entrevista proporcionaron información en la cual se obtuvo lo siguiente:

- En el Ilustre Municipio de Cevallos se manejan los documentos de manera manual, y en su mayoría en Excel sin garantizar ninguna seguridad en el tratamiento de esos documentos.
- El personal institucional conoce o al menos tienen idea de lo que es Software Libre y sabe manejarlo.
- El Ilustre Municipio de Cevallos no cuenta con un sistema que ayude al manejo de documentos, trámites, informes, solicitudes, proyectos, oficios, memos, etc.
- El sistema a desarrollarse deberá tener control de acceso con restricciones según los usuarios que ingresen al mismo.
- Deberá almacenar la información de todos los documentos que se manejan en la institución con códigos adecuados y de fácil comprensión.
- Tendrá la facilidad de consultas y reportes de los documentos por filtros; ya sea por fecha, enviado por, recibido por, etc. para su adecuada manipulación.

Se esperan que mediante la información recibida, el desarrollador del proyecto realice el sistema para cubrir las necesidades.

De manera general luego del análisis de las respuestas a cada pregunta de la entrevista se concluye que los funcionarios dan un manejo sedentario a los documentos que ingresan y salen de la institución por lo que se requiere de un sistema adecuado para proporcionar ayuda a los funcionarios y brindar una mejor atención a las personas que lo requieran.

El presente sistema se lo desarrollara tomando en cuenta que la Institución cuenta con licencias en lo que respecta a Base de Datos con **SQL Server 2000** el cual se encuentra trabajando actualmente.

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. **Microsoft SQL Server** constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son *Oracle*, *PostgreSQL* o *MySQL*.

En lo que tiene que ver con el lenguaje de programación se ha optado por C#, mediante el uso de un entorno de desarrollo integrado libre como lo es **SharpDevelop** para los lenguajes de programación C#, Visual Basic .NET y Boo.

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

En lo que concierne a seguridad del servidor en el cual se encontrara nuestra Base de Datos está a cargo del Jefe Administrativo encargado del manejo del servidor dentro de la institución.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- La Institución requiere de un sistema para la automatización el proceso de seguimiento a documentos y trámites con el fin de optimizar el tiempo y mejorar el servicio a los usuarios.
- El sistema debe contar con la facilidad de ingresos de administradores, usuarios, documentos, trámites, informes, solicitudes, proyectos, oficios, memos, etc. con la opción de consultar y obtener reportes de esos ingresos, con una interfaz fácil de manipular.
- El sistema a implantarse ayudará tanto a los funcionarios como a los usuarios de la Institución.
- Mediante el desarrollo e implementación de un sistema informático que trabaje con una base de datos, la información tiene mayor integridad y seguridad, los datos son más confiables y aportan con mayor eficiencia para recuperar la información.

- La información que se obtuvo de los empleados, especialmente del administrador, servirá de ayuda para satisfacer las necesidades de la Institución.
- Las herramientas de desarrollo elegidas para realizar el presente sistema son la apropiadas, con son el motor de Base de Datos (SQL Server) y la herramienta de desarrollo de programación (SharpDevelop) con un entorno de programación C#.

5.2 Recomendaciones

- Se recomienda analizar a profundidad los requerimientos de la institución antes del desarrollo del proyecto.
- Se recomienda realizar un estudio de alguna herramienta similar desarrollada, para usarla de referencia.
- Se recomienda estudiar las herramientas tanto de programación como de base de datos para optimizar el tiempo.
- Investigar sobre el Sistema que se desarrollará con el fin de conocer las funcionalidades, las ventajas y desventajas para el posterior desarrollo e implantación en el Ilustre Municipio de Cevallos.
- Al desarrollar e implementar el sistema en los equipos de la institución, es necesario realizar pruebas en cada una de las maquinas en las que se va a instalar el sistema para probar su correcto funcionamiento.
- Finalmente se recomienda implementar políticas de uso del sistema para su correcto y eficaz funcionamiento.

CAPÍTULO VI

PROPUESTA

6.1 Tema

SISTEMA INFORMÁTICO PARA DAR SEGUIMIENTO A DOCUMENTOS Y TRÁMITES EN EL ILUSTRE MUNICIPIO DE CEVALLOS

6.2 Datos Informativos

Institución: Gobierno Autónomo Descentralizado del Cantón Cevallos.

Ciudad: Cevallos

Dirección: Felipa Real y Av. 24 de Mayo.

Investigador: Víctor Alfonso Bombón Ramos.

Tutor: Ing. Galo López

6.3 Antecedentes

El Gobierno Autónomo Descentralizado del Cantón Cevallos es una Institución Pública encargada de velar por el avance y desarrollo del cantón y de sus habitantes, brindando servicios y resolviendo problemas.

La implantación de un sistema para dar seguimiento a los documentos surge por la necesidad de automatizar los manejos que en la actualidad se dan de manera manual de los trámites con los cuales la Institución trabaja diariamente; para lo cual se ha realizado previamente un análisis de los requerimientos y de las necesidades de los funcionarios y remitentes externos en general.

Basándose en la política de estado, se ha optado por realizar el proyecto con la utilización de software libre para el desarrollo e implantación del sistema de seguimiento de documentos y trámites.

Teniendo en cuenta la exigencia en cuanto a seguridad, fiabilidad, y veracidad de la información, además de que la institución cuenta con la licencia se decidió establecer como administrador de Base de Datos a SQLServer 2000 y para la parte de desarrollo del sistema el Lenguaje de Programación C# desarrollándolo en SharpDevelop.

6.4 Justificación

La implantación de un sistema para dar seguimiento a documentos y trámites se ha convertido en una necesidad en organizaciones que interactúan directamente con personas externas o ajenas a la institución las cuales realizan trámites constantes, para obtener un mejor control de los mismos.

Al utilizar en un entorno de desarrollo integrado libre como lo es SharpDevelop, nos permite desarrollar nuestro sistema en un lenguaje de programación C# sin la necesidad de utilizar Microsoft Visual Studio con el objetivo de utilizarlo libremente.

C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.

6.5 Objetivos

6.5.1 Objetivo General

Desarrollar e implantar un sistema informático utilizando lenguaje de programación C# con conexión a base de datos para dar seguimiento a documentos y trámites en el Gobierno Autónomo Descentralizado del Cantón Cevallos.

6.5.2 Objetivos Específicos.

- Analizar e interpretar los requerimientos facilitados por los miembros que laboran en la institución para el posterior desarrollo del sistema.
- Diseñar correctamente la estructura de la Base de Datos.
- Desarrollar el Sistema de seguimiento de documentos y trámites utilizando lenguaje de programación C# con conexión a base de datos con la seguridad necesaria para garantizar la correcta manipulación de la información.
- Realizar las pruebas necesarias para garantizar el funcionamiento óptimo del Sistema.
- Elaborar los manuales de instalación y uso, de usuario y de administrador.
- Capacitar u orientar a la persona que esté a cargo, acerca del manejo del sistema informático.

6.6 Análisis de factibilidad

6.6.1 Factibilidad Operativa

Durante el levantamiento de la información, se identificaron todas las actividades que son necesarias para alcanzar el objetivo principal, lo que generó en el desarrollo de un producto de software que satisface las necesidades de la institución, de fácil utilización, instalación y mantenimiento.

El sistema resulta operativo en todas las áreas especialmente en el área de información ya que a dicho lugar llegan la mayoría de documentos a ser procesados.

6.6.2 Factibilidad Económica

Es un proyecto factible desde el punto de vista económico ya que el Municipio de Cevallos cuenta con las herramientas necesarias tanto en hardware como software para el desarrollo del proyecto, puesto que el Sistema aportará significativamente a la Institución, ya que automatizará el tratamiento de los documentos y trámites.

El sistema cuenta con los siguientes tipos de usuarios:

- Administrador
- Secretaria
- Jefes / Directores

Administrador: Se les implementara el acceso para la persona que será la encargada de crear y habilitar los usuarios como Jefes / Directores y Secretaria, además de administrar todo el sistema ya que esta persona tendrá acceso total al mismo.

Secretaria: Se implementará el acceso a lo que es envío de documentos y ciertas tareas que serán analizadas en lo posterior.

Jefes / Directores: Se le implementara el acceso a todas las personas consideradas jefes de área, tendrá un acceso básico, con la posibilidad de incrementar según se crea conveniente.

6.6.3 Factibilidad Técnica

Para la creación del sistema informático se cuenta con las herramientas necesarias, otorgadas por el Ilustre Municipio de Cevallos, herramientas como un servidor de base de datos y equipos de cómputo para funcionarios.

Software

- Lenguaje de programación C#.
- Herramienta de programación SharpDevelop.
- SQLServer 2000 como motor de base de datos.

Hardware

El Ilustre Municipio de Cevallos al ser una institución del estado dedicada al mejoramiento de la calidad de vida de las personas del cantón, cuenta con toda la infraestructura adecuada para llevar a cabo el desarrollo del sistema.

6.7 Fundamentación

Lenguaje C# es un lenguaje orientado a objetos elegante y con seguridad de tipos que permite a los desarrolladores crear una amplia gama de aplicaciones sólidas y seguras que se ejecutan en .NET Framework.

Puede utilizar este lenguaje para crear aplicaciones cliente para Windows tradicionales, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos, y muchas tareas más. Microsoft Visual C# 2005 proporciona un editor de código avanzado, diseñadores de interfaz de usuario prácticos, un depurador integrado y muchas otras herramientas para facilitar un rápido desarrollo de la aplicación basado en la versión 2.0 del lenguaje C# y en .NET Framework.

WEB SITE, (2011). SharpDevelop. Extraído el Agosto 21 del 2011 desde <http://msdn.microsoft.com/es-es/library/z1zx9t92%28v=vs.80%29.aspx>

SharpDevelop es un entorno de desarrollo integrado libre para los lenguajes de programación C#, Visual Basic, .Net y Boo.

Es usado típicamente por aquellos programadores de los citados lenguajes, que no desean o no pueden usar el entorno de desarrollo de Microsoft Visual Studio. Hay disponible un port para Mono/GTH#, llamado MonoDevelop, el cual funciona en otro sistema operativo.

Para el completo automático de código, la aplicación incorpora sus propios analizadores sintácticos. La versión 1.1 de la aplicación puede importar proyectos de Visual Studio .NET. La versión 2.0 ya es capaz de editarlo directamente. La versión 3.0 integra soporte para los lenguajes de programación Python y F#.

WEB SITE, (2011). SharpDevelop. Extraído el Agosto 21 del 2011 desde <http://es.wikipedia.org/wiki/SharpDebelop>.

SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.

Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, PostgreSQL o MySQL.

WEB SITE, (2011). SharpDevelop. Extraído el Agosto 21 del 2011 desde http://es.wikipedia.org/wiki/Microsoft_SQL_Server

6.8 Metodología

En la actualidad, para desarrollar un producto de software es necesario contar con un plan bien estructurado. El cliente tiene que comprender que es lo que hará el investigador; además tiene que ser capaz de señalar cambios si no han sido claramente asimiladas sus necesidades.

Para el desarrollo de la propuesta se utilizo como base una metodología orientada a objetos basada en el lenguaje unificado de modelado UML, que es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. Posee la riqueza suficiente como para crear un modelo del sistema, pudiendo modelar

los procesos de negocios, funciones, esquemas de bases de datos, expresiones de lenguajes de programación, etc.

6.9 Modelo Operativo

6.9.1 Análisis del Sistema

6.9.1.1 Servicios Requeridos

El Ilustre Municipio de Cevallos carece de un sistema de seguimiento de documentos y trámites, por lo que en el desarrollo del sistema se toma como referencia los siguientes aspectos:

- Gestión de menús por roles.
- Creación de Departamentos, Usuarios.
- Asignación de roles a usuarios
- Gestión de menús por roles.
- Ingreso de los diferentes Tipos de Documentos y Tramites que se realizan en la institución.
- Ingreso de Remitentes Externos.
- Creación y Envió de documentos a los distintos departamentos.
- Revisión de documentos enviados y recibidos.
- Consulta de documentos en general.
- Reportes de documentos por remitentes externos.
- Publicación de noticias internas y externas.
- Descarga de archivos.

Con una interfaz fácil de manejar para los usuarios que ingresen al sistema informático para realizar los envíos y tratamientos de los tramites y documentos.

6.9.1.2 Diagramas UML

Considerado hoy en día, UML (Lenguaje unificado de modelado) está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo, se ha procedido a utilizarlo con el fin de establecer requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

6.9.1.2.1 Diagramas de Casos de Usos

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas para conseguir un objetivo específico.

6.9.1.2.1.1 Caso de uso Administración de cuentas de usuario

Actor(es): Administrador, Jefes/Directores, Secretaria.

Descripción: Es el proceso que lleva a cabo el Administrador en la creación de cuentas para que los Empleados puedan acceder al sistema.

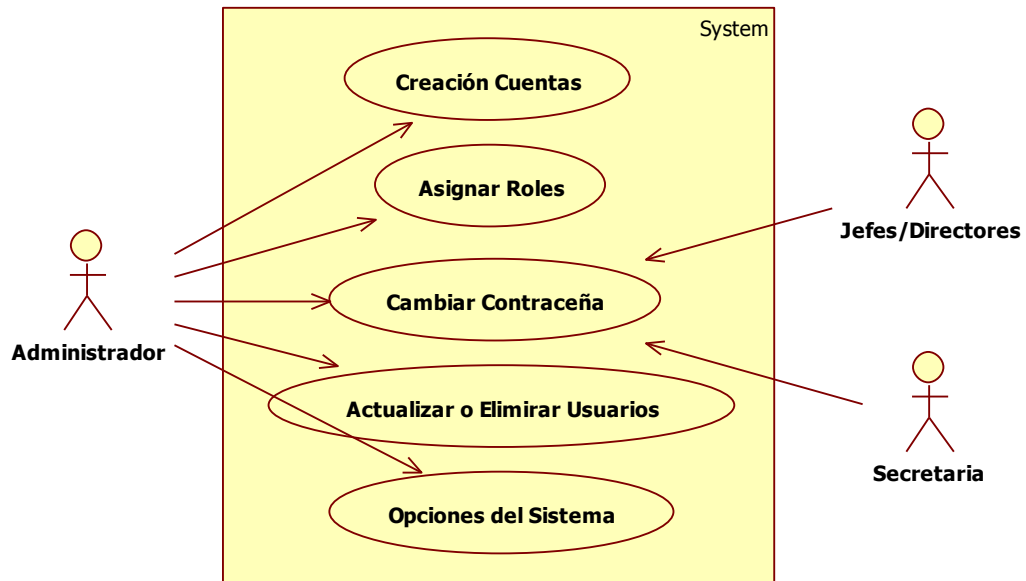


Figura 6.1 Diagrama Caso Uso Cuentas de Usuario

FLUJO DE EVENTOS	
CURSO NORMAL	ALTERNATIVAS
1. El Administrador crea cuentas de usuario para acceder al sistema.	
2. El Administrador asigna permisos a los usuarios creados, al hacer esto se asigna a un rol.	
3. El administrador le da a conocer su login y contraseña al usuario para acceder al sistema.	<p>3.1 El usuario accede al sistema por primera vez.</p> <p>3.2 El usuario acepta la contraseña proporcionada por el administrador.</p> <p>3.3 Fin del caso de uso, caso contrario</p>

	vamos al punto 4.
4. El usuario decide cambiar su contraseña.	<p>4.1 En un futuro el usuario olvida su contraseña.</p> <p>4.2 El empleado solicita cambio de contraseña al administrador.</p> <p>4.3 Retorna el caso de uso al punto 3.</p>
5. El Administrador actualiza o elimina usuarios.	<p>5.1 Para eliminar un usuario no debe estar asignado a un rol.</p> <p>5.2 Si lo está lo quitamos del mismo.</p>
6. El Administrador asigna nuevos parámetros de menú.	<p>6.1 Si un usuario en particular quiere un acceso a menú extra o si deseamos retirarlo lo hacemos aquí.</p> <p>6.2 Finaliza el caso de uso.</p>

Tabla 6.1 Diagrama Caso Uso Cuentas de Usuario

6.9.1.2.1.2 Caso de uso Parámetros.

Actor(es): Administrador, Jefes/Directores, Secretaria.

Descripción: Es el proceso que lleva a cabo el Administrador en la creación de parámetros para el uso del sistema.

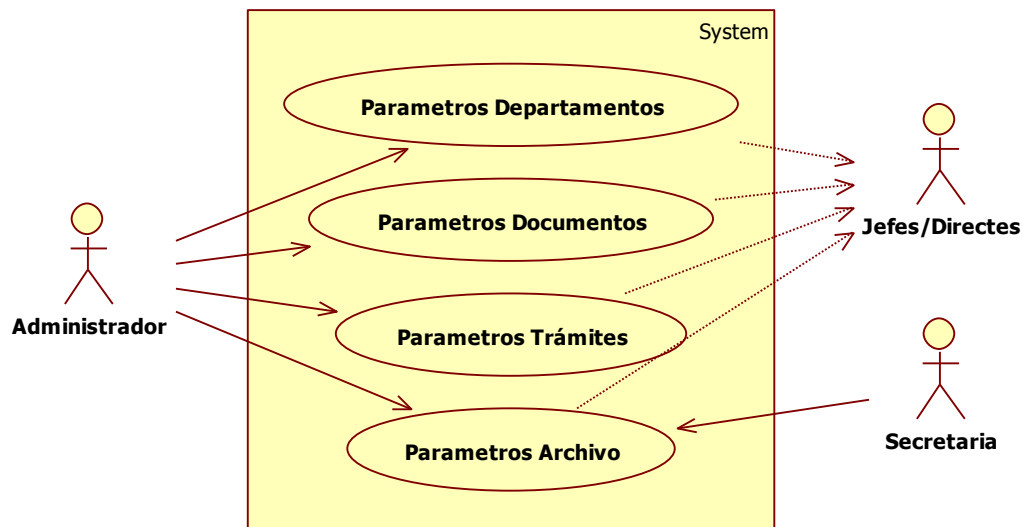


Figura 6.2 Parámetros Uso del Sistema

FLUJO DE EVENTOS	
CURSO NORMAL	ALTERNATIVAS
1. El Administrador crea departamentos.	1.1 Se puede crear un departamento, sección o unidad.
2. El Administrador asigna un jefe departamental.	2.1 El jefe a ser asignado debe ser un usuario ya ingresado con anterioridad. 2.2 El jefe departamental no necesariamente estar asignado al departamento en el cual labora.
3. El Administrador elimina o edita un Departamento.	3.1 Si se va a eliminar un departamento, no debe tener asignado ningún usuario.
4. El Administrador ingresa nuevos	4.1 Esta opción de menú se lo puede

documentos o edita los ya creados.	asignar a otros usuarios que no tengan rol de Administrador.
5. El Administrador crea nuevos trámites o edita los ya creados.	5.1 Este opción de menú puede tener cualquier otra persona que no sea administrador
6. El Administrador crea un nuevo archivo físico o se puede editar los ya creados.	6.1 El Archivo físico puede ser ingresado por el rol secretaria. 6.2 Finaliza el caso de uso.

Tabla 6.2 Parámetros Uso del Sistema

6.9.1.2.1.3 Caso de uso Movimientos.

Actor(es): Usuarios01, Usuarios02.

Descripción: Es el proceso que lleva a cabo cualquier usuario del sistema con un rol para crear, y recibir documentos en general.

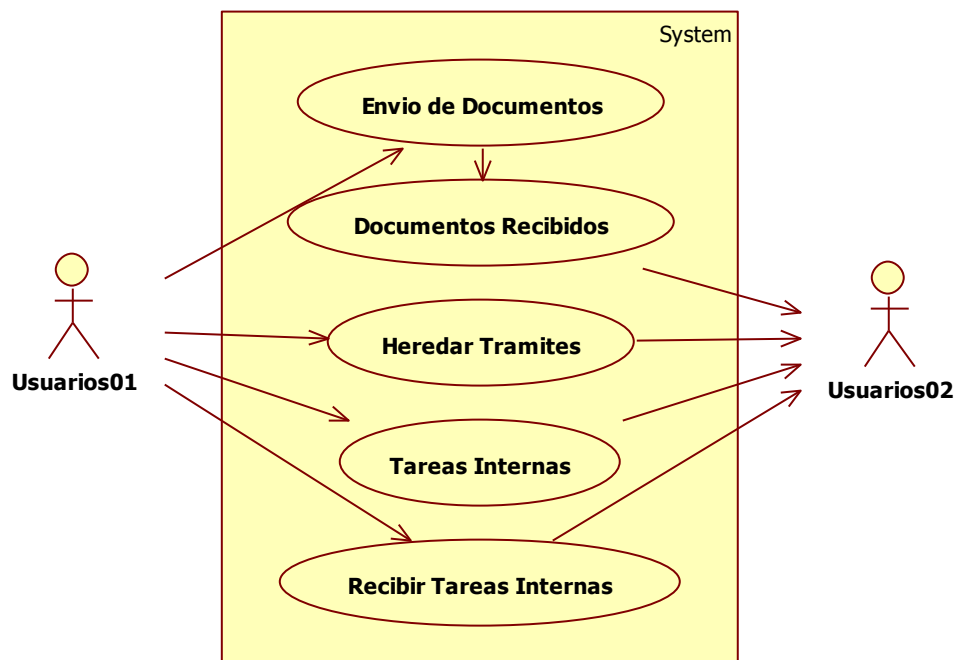


Figura 6.3 Crear y recibir documentos

FLUJO DE EVENTOS	
CURSO NORMAL	ALTERNATIVAS
<p>1. El Usuario01 realiza el ingreso y envió de documentos solicitados por remitentes externos.</p>	<p>1.1 Se puede enviar todo tipo de documentos.</p> <p>1.2 El documento ingresado puede ser editado mientras no sea recibido por el departamento al que se envió.</p>
<p>2. El Usuario02 puede recibir un documento, editarlo y finalizarlo según sea el caso.</p>	<p>2.1 El Usuario01 puede asignar tareas internas para los Usuarios02.</p>
<p>3. El Usuario01 puede realizar la edición de un documento enviado que tenga un previo envío.</p>	<p>3.1 El Usuario01 puede hacerlo solo de documentos que el mismo creó.</p> <p>3.2 El Usuario02 puede ser esta opción pero no puede editar.</p>
<p>4. El Usuario01 puede realizar el envío de Tareas Internas a Usuarios02.</p>	<p>4.1 El Usuario02 puede editarlo según sea el caso.</p>
<p>5. El Usuario02 puede recibir una tarea interna.</p>	<p>5.1 El Usuario01 lo puede procesar según dependa.</p>
<p>6. El Usuario01 puede dar tratamiento a todo trámite y cambiarlo de estado.</p> <ul style="list-style-type: none"> • Ingresado. • Enviado. 	<p>6.1 Finaliza el caso de uso.</p>

<ul style="list-style-type: none"> • Pendiente. • Finalizado. 	
---	--

Tabla 6.3 Crear y recibir documentos

6.9.1.2.1.3 Caso de uso Consultas y Reportes.

Actor(es): Usuarios01, Usuarios02.

Descripción: Es el proceso que lleva a cabo cualquier usuario del sistema con un rol para revisar documentos ingresados al sistema.

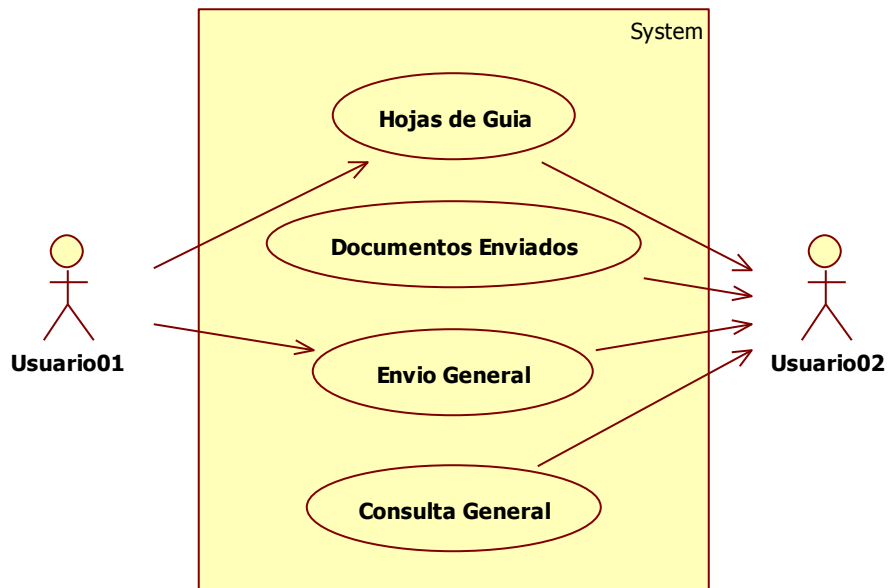


Figura 6.4 Documentos ingresados al sistema

FLUJO DE EVENTOS	
CURSO NORMAL	ALTERNATIVAS
<p>1. El Usuario01 realiza un ingreso de parámetros para ver la hoja de guía</p>	<p>1.1 El Usiario02 puede visualizar los documentos enviados por fechas.</p>

general enviada en una fecha determinada.	
2. El Usuario02 puede consultar los documentos enviados y el seguimiento de los mismos.	
3. El Usuario01 puede realizar la edición de un documento enviado que tenga un previo envío.	<p>3.1 El Usuario01 puede hacerlo solo de documentos que el mismo creo.</p> <p>3.2 El Usuario02 puede usar esta opción pero no puede editar.</p>
<p>4. El Usuario01 puede dar tratamiento a todo trámite y cambiarlo de estado.</p> <ul style="list-style-type: none"> • Ingresado. • Enviado. • Pendiente. • Finalizado. 	4.1 Finaliza el caso de uso.

Tabla 6.4 Documentos ingresados al sistema

6.9.1.2.2 Diagrama de Secuencia

Los diagramas de secuencia ilustran la interacción entre objetos y el orden secuencial en el que ocurren dichas interacciones, es decir cómo se comunican los objetos entre sí.

6.9.1.2.2.1 Creación de Usuarios

Actor(es): Administrador, Empleado

Descripción: Detalla en proceso de creación de Usuarios en el sistema.

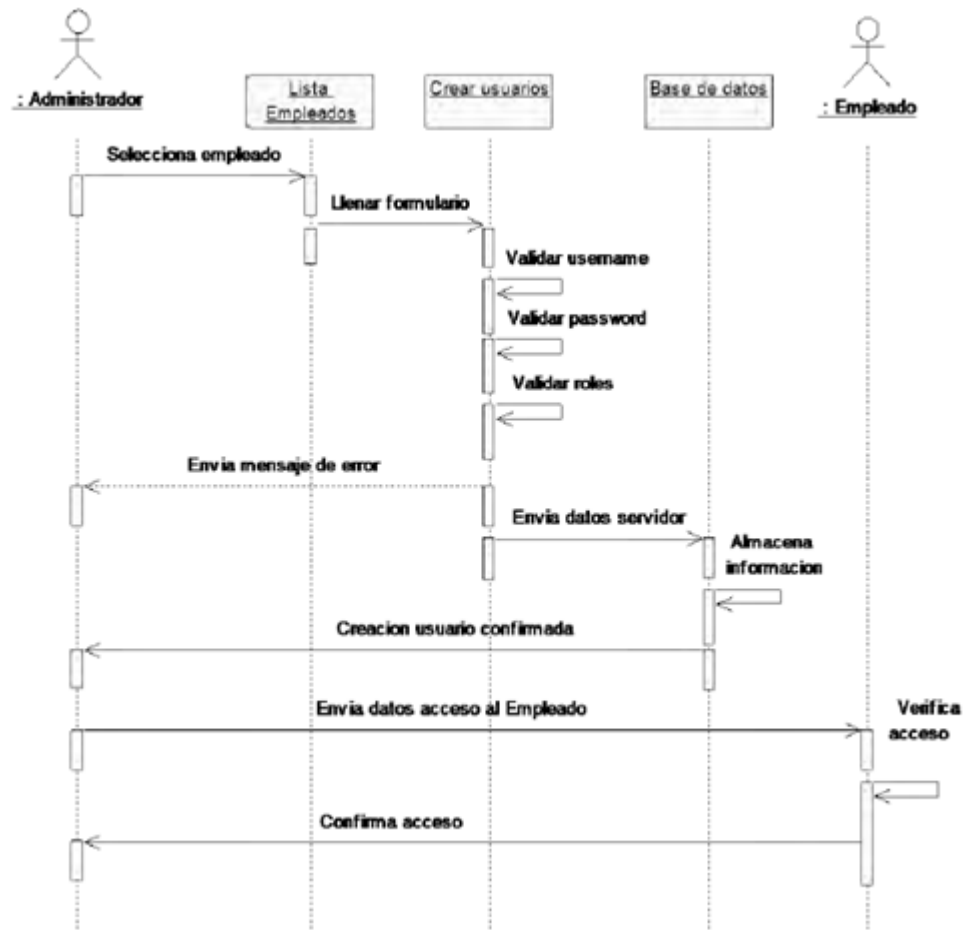


Figura 6.5 Diagrama de Secuencia Creación de Usuarios

6.9.1.2.2.2 Manejo de la Base de Datos

Actor(es): Administrador

Descripción: Detalla la administración de la base de datos.

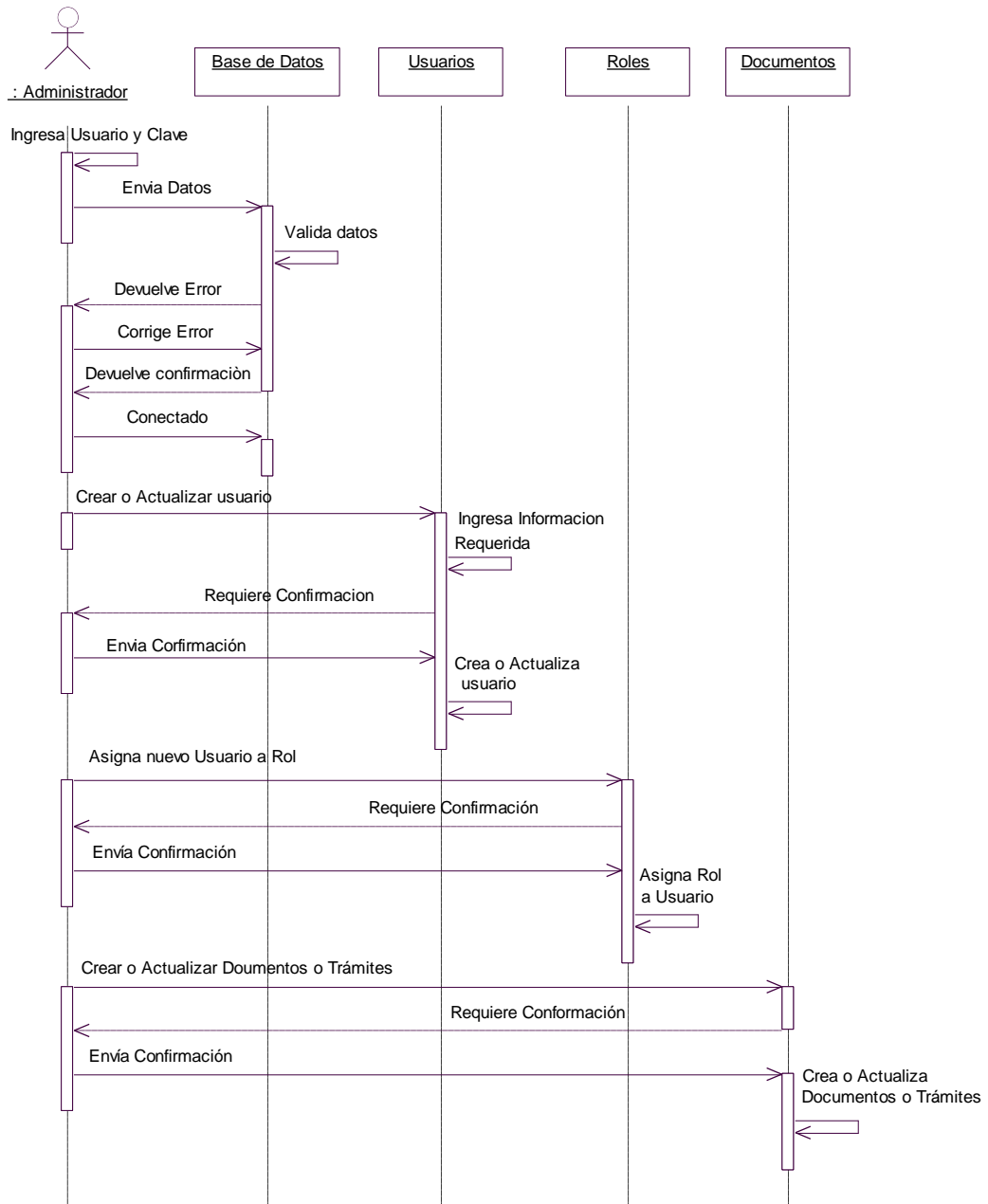


Figura 6.6 Diagrama de Secuencia Manejo de BD

6.9.1.2.2.3 Manejo y administración del Sistema

Actor(es): Usuario

Descripción: Detalla el funcionamiento del sistema.

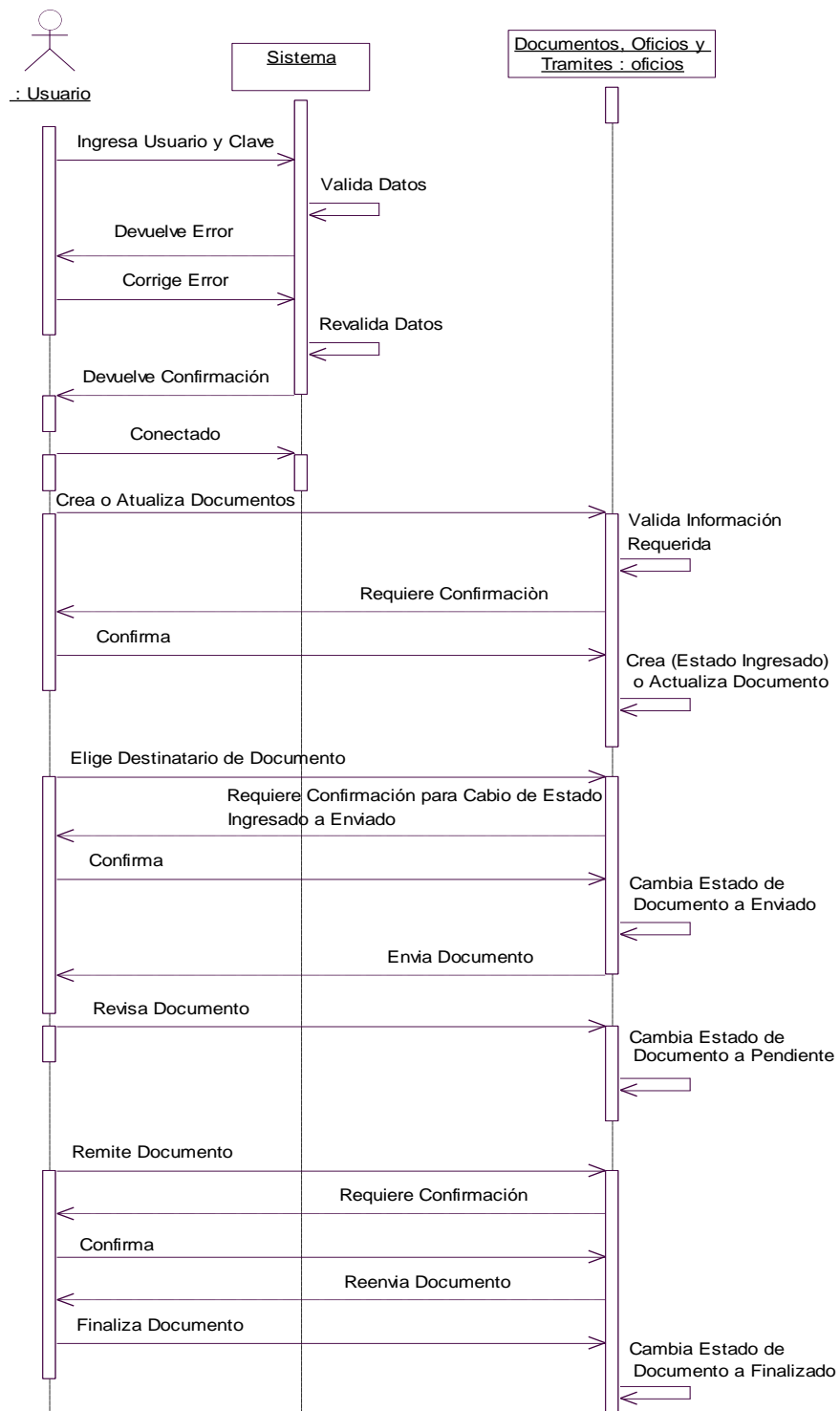


Figura 6.7 Diagrama de Secuencia Manejo y Administración Sistema

6.9.1.2.3 Diagrama de Clase

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

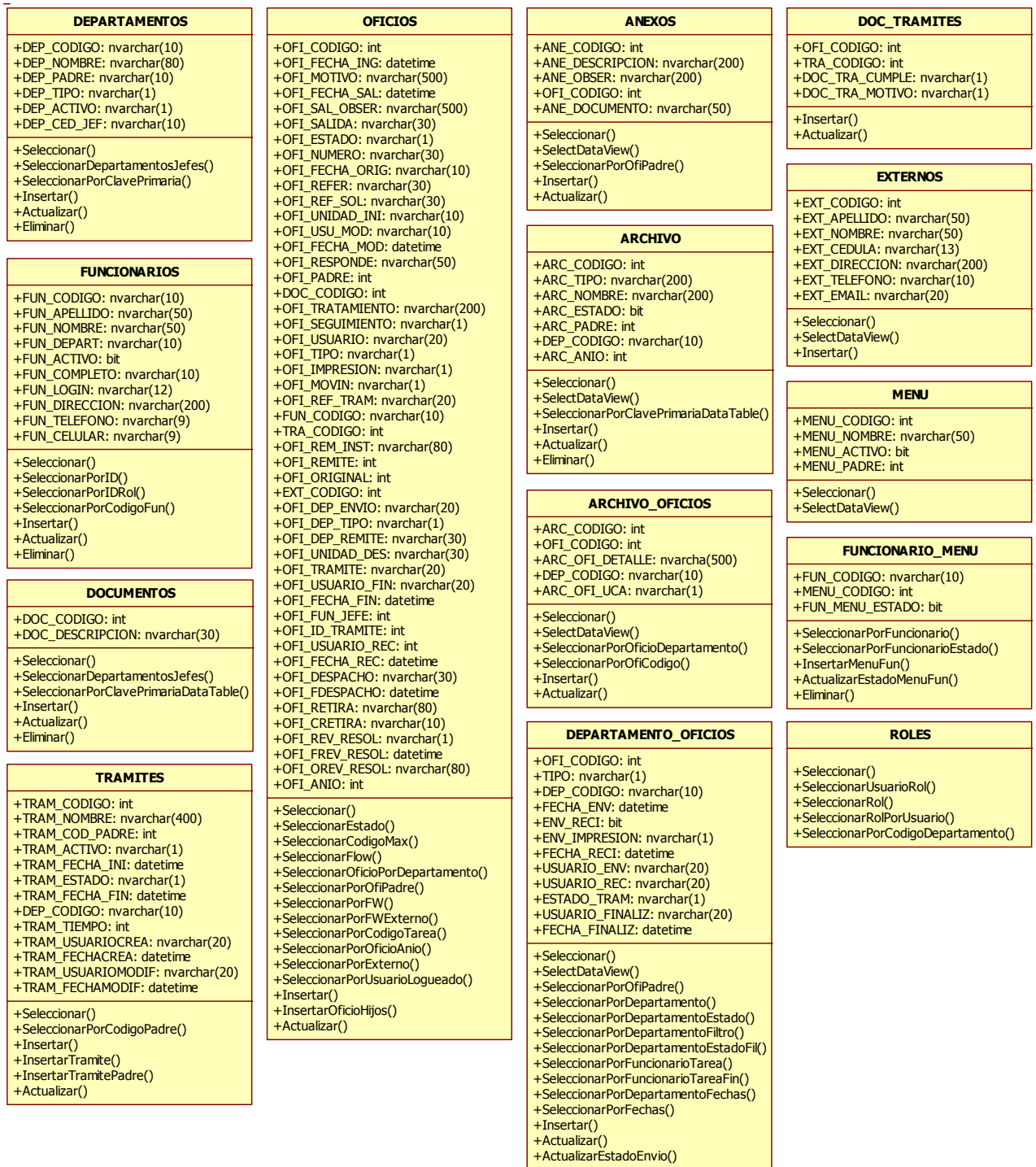


Figura 6.7 Diagrama de Clases

6.9.2 Diseño del Sistema

6.9.2.1 Diseño de la Base de Datos

Hoy en día las empresas manejan una gran cantidad de datos. Cualquier empresa que se precie debe tener almacenados todos estos datos en una base de datos para poder realizarlos mediante una aplicación profesional; sin esta funcionalidad resultaría imposible tratar y manejar en su totalidad los datos que lleva a cabo la empresa y se perdería un tiempo y un dinero muy valiosos.

Uno de los pasos cruciales en la construcción de una aplicación que maneje una base de datos, es sin duda, el diseño de la base de datos.

Si las tablas no son definidas apropiadamente, podemos tener muchos dolores de cabeza al momento de ejecutar consultas a la base de datos para tratar de obtener algún tipo de información.

Previa la creación de las tabla para poder trabajar en la Base de Datos es necesario la creación le la tablas de MemberShip y Role Provider en SQL Server.

Luego de analizar minuciosamente de información se ha modelado la base de datos de la siguiente manera.

Modelo Relacional Lógico

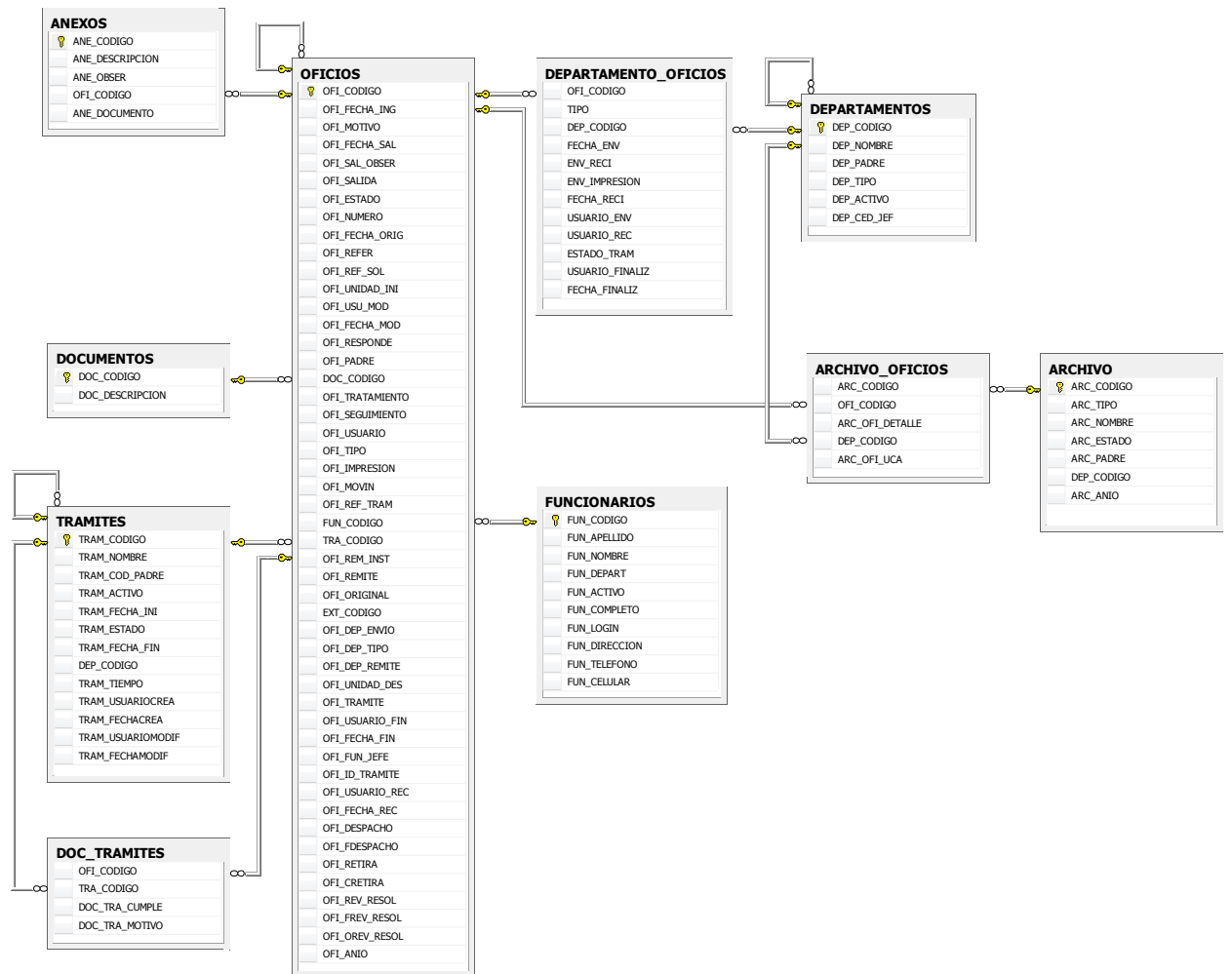


Figura 6.8 Diagrama Relacional Principal

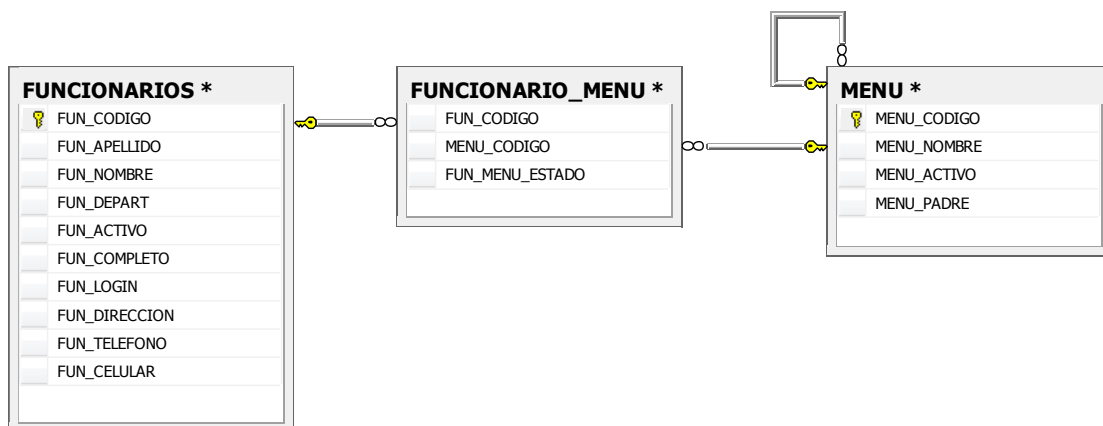


Figura 6.9 Diagrama Relacional Menú

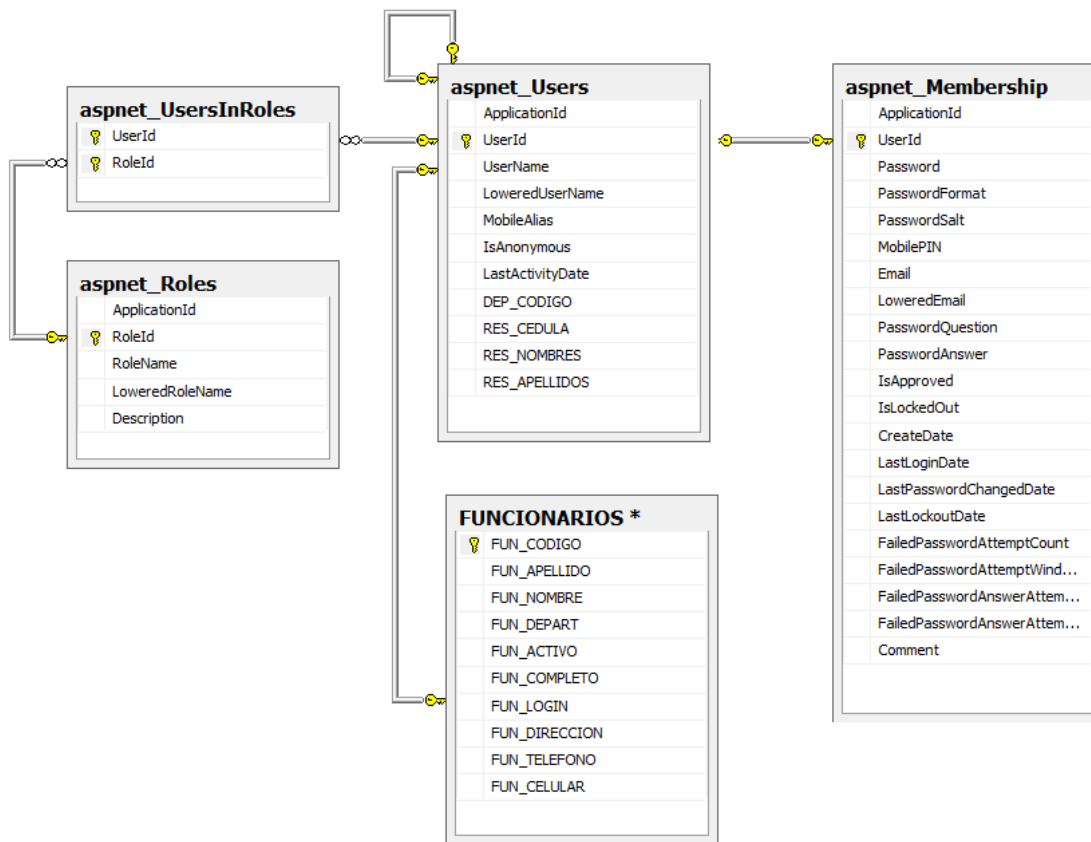


Figura 6.10 Diagrama Relacional Membership



Figura 6.11 Diagrama Relacional General

6.9.2.2 Diccionario de datos

Tablas Creadas Automáticamente por el MemberShip.

Tabla		
Nombre:	aspnet_Membership	
Descripción:		
Campo	Tipo	Descripción
ApplicationId	uniqueidentifier	
UserId	uniqueidentifier	
Password	nvarchar(128)	
PasswordFormat	int	
PasswordSalt	nvarchar(128)	
MobilePIN	nvarchar(16)	
Email	nvarchar(256)	
LoweredEmail	nvarchar(256)	
PasswordQuestion	nvarchar(256)	
PasswordAnswer	nvarchar(128)	
IsApproved	bit	
IsLockedOut	bit	
CreateDate	datetime	
LastLoginDate	datetime	
LastPasswordChangedDate	datetime	
LastLockoutDate	datetime	
FailedPasswordAttemptCount	int	
FailedPasswordAttemptWindowStart	datetime	
FailedPasswordAnswerAttemptCount	int	
FailedPasswordAnswerAttemptWindowStart	datetime	
Comment	ntext	

Tabla 6.5 Tabla aspnet_Membership

Tabla		
Nombre:	aspnet_Roles	
Descripción:		
Campo	Tipo	Descripción
ApplicationId	Uniqueidentifier	
RoleId	Uniqueidentifier	
RoleName	nvarchar(256)	
LoweredRoleName	nvarchar(256)	
Description	nvarchar(256)	

Tabla 6.6 Tabla aspnet_Roles

Tabla		
Nombre:	aspnet_Users	
Descripción:		
Campo	Tipo	Descripción
ApplicationId	Uniqueidentifier	

UserId	Uniqueidentifier	
UserName	nvarchar(256)	
LoweredUserName	nvarchar(256)	
MobileAlias	nvarchar(16)	
IsAnonymous	bit	
LastActivityDate	datetime	

Tabla 6.7 Tabla aspnet_Users

Tabla		
Nombre:	aspnet_UsersInRoles	
Descripción:		
Campo	Tipo	Descripción
UserId	uniqueidentifier	
RoleId	uniqueidentifier	

Tabla 6.8 Tabla aspnet_UsersInRoles

Tablas Creadas para la administración y funcionamiento del sistema.

Tabla		
Nombre:	DEPARTAMENTO	
Descripción:	Contiene la información de un departamento del municipio de Cevallos.	
Campo	Tipo	Descripción
DEP_CODIGO	nvarchar(10)	Código de Departamento.
DEP_NOMBRE	nvarchar(80)	Nombre del Departamento.
DEP_PADRE	nvarchar(10)	Secuencial del código padre.
DEP_TIPO	nvarchar(1)	Tipo de departamento. Departamento, Sección o Unidad.
DEP_ACTIVADO	nvarchar(1)	Si esta activo o no 1 ó 0
DEP_CED_JEF	nvarchar(10)	Cedula del Jefe Departamental.

Tabla 6.9 Tabla DEPARTAMENTOS

Tabla		
Nombre:	FUNCIONARIOS	
Descripción:	Contiene campos con información la información de funcionarios.	
Campo	Tipo	Descripción
FUN_CODIGO	nvarchar(10)	Cedula de identidad del funcionario.
FUN_APELLIDO	nvarchar(50)	Apellidos completos
FUN_NOMBRE	nvarchar(50)	Nombres completos
FUN_DEPART	nvarchar(10)	Código del Departamento
FUN_ACTIVADO	bit	Esta activo, True o False
FUN_COMPLETO	nvarchar(100)	Apellidos y Nombres unidos.
FUN_LOGIN	nvarchar(12)	UserName de la tabla aspnet_Users
FUN_DIRECCION	nvarchar(200)	Dirección del Funcionario
FUN_TELEFONO	nvarchar(9)	Teléfono del Funcionario
FUN_CELULAR	nvarchar(9)	Celular del Funcionario

Tabla 6.10 Tabla FUNCIONARIOS

Tabla		
Nombre:	DOCUMENTOS	
Descripción:	Contiene campos con información de documentos propios del municipio	
Campo	Tipo	Descripción
DOC_CODIGO	int	Secuencial del documento
DOC_DESCRIPCION	nvarchar(30)	Descripción del Documento

Tabla 6.11 Tabla DOCUMENTOS

Tabla		
Nombre:	TRAMITES	
Descripción:	Contiene campos con información de trámites propios que maneja el municipio.	
Campo	Tipo	Descripción
TRAM_CODIGO	int	Secuencial de trámite
TRAM_NOMBRE	nvarchar(400)	Nombre el tramite que se realiza
TRAM_COD_PADRE	int	Código secuencial padre de trámite
TRAM_ACTIVADO	nvarchar(1)	Si el trámite esta activo S y N
TRAM_FECHA_INI	datetime	
TRAM_ESTADO	nvarchar(1)	
TRAM_FECHA_FIN	datetime	
DEP_CODIGO	nvarchar(10)	
TRAM_TIEMPO	int	Tiempo máximo para realizar el trámite.
TRAM_USUARIOCREA	nvarchar(20)	Login del usuario que creó el tram.
TRAM_FECHACREA	datetime	Fecha que se ingreso el tramite
TRAM_USUARIOMODIF	nvarchar(20)	
TRAM_FECHAMODIF	datetime	

Tabla 6.12 Tabla TRAMITES

Tabla		
Nombre:	OFICIOS	
Descripción:	Esta es la tabla principal de nuestro sistema.	
Campo	Tipo	Descripción
OFI_CODIGO	int	Código secuencial de la tabla
OFI_FECHA_ING	datetime	Fecha de ingreso del documento
OFI_MOTIVO	nvarchar(500)	Asunto
OFI_FECHA_SAL	datetime	Fecha de salida de tarea
OFI_SAL_OBSER	nvarchar(500)	Observación de Tarea
OFI_SALIDA	nvarchar(30)	Núm. Oficio de tarea
OFI_ESTADO	nvarchar(1)	Estado del documento
OFI_NUMERO	nvarchar(30)	Núm. Oficio
OFI_FECHA_ORIG	nvarchar(10)	Fecha origen del tramite
OFI_REFER	nvarchar(30)	Referencia oficio
OFI_REF_SOL	nvarchar(30)	Referencia solicitud
OFI_UNIDAD_INI	nvarchar(10)	Departamento de Inicio
OFI_USU_MOD	nvarchar(10)	Usuario de modifica

OFI_FECHA_MOD	datetime	Fecha de modificación
OFI_RESPONDE	nvarchar(50)	Oficio de respuesta de tarea
OFI_PADRE	int	Oficio padre
DOC_CODIGO	int	Tipo de documento
OFI_TRATAMIENTO	nvarchar(200)	Tratamiento que se da al tramite
OFI_SEGUIMIENTO	nvarchar(1)	
OFI_USUARIO	nvarchar(20)	Usuario que ingresa
OFI_TIPO	nvarchar(1)	Interno I, Externo E
OFI_IMPRESION	nvarchar(1)	
OFI_MOVIN	nvarchar(1)	
OFI_REF_TRAM	nvarchar(20)	Referencia a tramites
FUN_CODIGO	nvarchar(10)	Código del funcionario
TRA_CODIGO	int	Código del tramite
OFI_REM_INST	nvarchar(80)	
OFI_REMITE	int	1 si es tabla externa
OFI_ORIGINAL	int	
EXT_CODIGO	int	Código del Remitente Externo
OFI_DEP_ENVIO	nvarchar(20)	Departamento que envía
OFI_DEP_TIPO	nvarchar(1)	Para P Copia C
OFI_DEP_REMITE	nvarchar(30)	
OFI_UNIDAD_DES	nvarchar(30)	RE(respuesta) RM(remitido)
OFI_TRAMITE	nvarchar(20)	1 si es tarea
OFI_USUARIO_FIN	nvarchar(20)	Usuario de finaliza el tramite
OFI_FECHA_FIN	datetime	Fecha que finaliza
OFI_FUN_JEFE	int	
OFI_ID_TRAMITE	int	Id del tramite
OFI_USUARIO_REC	nvarchar(20)	Usuario que recibe
OFI_FECHA_REC	datetime	Fecha que recibe
OFI_DESPACHO	nvarchar(30)	
OFI_FDESPACHO	datetime	
OFI_RETIRA	nvarchar(80)	
OFI_CRETIRA	nvarchar(10)	
OFI_REV_RESOL	nvarchar(1)	
OFI_FREV_RESOL	datetime	
OFI_OREV_RESOL	nvarchar(80)	
OFI_ANIO	int	

Tabla 6.13 Tabla OFICIOS

Tabla		
Nombre:	ANEXOS	
Descripción:	Tabla que almacena la información de anexos adjuntados al documento.	
Campo	Tipo	Descripción
ANE_CODIGO	int	Secuencial de anexos
ANE_DESCRIPCION	nvarchar(200)	Descripción del Anexo
ANE_OBSER	nvarchar(200)	Observación del Anexo
OFI_CODIGO	int	Secuencial de oficio
ANE_DOCUMENTO	nvarchar(50)	Número del Documento Anexado

Tabla 6.14 Tabla ANEXOS

Tabla		
Nombre:	ARCHIVO	
Descripción:	Campos con información de archivo físico	
Campo	Tipo	Descripción
ARC_CODIGO	int	Secuencial de archivo
ARC_TIPO	nvarchar(200)	Abreviación de nombre de Archivo
ARC_NOMBRE	nvarchar(200)	Nombre del Archivo Físico
ARC_ESTADO	bit	True, False
ARC_PADRE	int	Secuencial Padre de archivo
DEP_CODIGO	nvarchar(10)	Código de departamento
ARC_ANIO	numeric(4, 0)	Año de creación

Tabla 6.15 Tabla ARCHIVO

Tabla		
Nombre:	DEPARTAMENTO_OFICIOS	
Descripción:	Tabla intermedia relacional entre Departamentos y Oficios	
Campo	Tipo	Descripción
OFI_CODIGO	int	
TIPO	nvarchar(1)	
DEP_CODIGO	nvarchar(10)	
FECHA_ENV	datetime	
ENV_RECIBI	bit	
ENV_IMPRESION	nvarchar(1)	
FECHA_RECIBI	datetime	
USUARIO_ENV	nvarchar(20)	
USUARIO_RECIBI	nvarchar(20)	
ESTADO_TRAM	nvarchar(1)	
USUARIO_FINALIZ	nvarchar(20)	
FECHA_FINALIZ	datetime	

Tabla 6.16 Tabla DEPARTAMENTO_OFICIOS

Tabla		
Nombre:	DOC_TRAMITES	
Descripción:	Tabla intermedia relacional entre Documentos y Tramites	
Campo	Tipo	Descripción
OFI_CODIGO	int	
TRA_CODIGO	int	
DOC_TRA_CUMPLE	nvarchar(1)	
DOC_TRA_MOTIVO	nvarchar(1)	

Tabla 6.17 Tabla DOC_TRAMITES

Tabla		
Nombre:	ARCHIVO_OFICIOS	
Descripción:	Tabla intermedia relacional entre Archivo y Oficio	
Campo	Tipo	Descripción

ARC_CODIGO	int	Secuencial de Archivo
OFI_CODIGO	int	Secuencial de Oficio
ARC_OFI_DETALLE	nvarchar(500)	Detalle de la forma de archivo
DEP_CODIGO	nvarchar(10)	Secuencial de Departamento
ARC_OFI_UCA	nvarchar(1)	Unidad Central Archivo (Null)

Tabla 6.18 Tabla ARCHIVO_OFICIOS

Tabla		
Nombre:	ESTADOFICIO	
Descripción:	Tabla general con campos del estado del oficio	
Campo	Tipo	Descripción
EST_CODIGO	int	Secuencial del estado de Oficio
EST_OFICIO	nvarchar(1)	Nombre de estado del oficio
EXT_DESCRIPCION	nvarchar(10)	Descripción de estado oficio

Tabla 6.19 Tabla ESTADOFICIO

Tabla		
Nombre:	EXTERNOS	
Descripción:	Tabla en la que guarda la información de los remitentes externos que están generado o solicitando un documento de envió.	
Campo	Tipo	Descripción
EXT_CODIGO	int	Secuencial del remitente externo
EXT_APELLIDO	nvarchar(50)	Apellidos del remitente externo
EXT_NOMBRE	nvarchar(50)	Nombres del remitente externo
EXT_CEDULA	nvarchar(13)	Numero de cedula del remitente externo
EXT_DIRECCION	nvarchar(200)	Dirección del remitente externo
EXT_TELEFONO	nvarchar(10)	Teléfono del remitente externo
EXT_EMAIL	nvarchar(50)	Email del remitente si lo tuviera.

Tabla 6.20 Tabla EXTERNOS

Tabla		
Nombre:	FLOW	
Descripción:	Tabla que servirá de auxiliar de la tabla Oficios. Tabla de control de flujo.	
Campo	Tipo	Descripción
FLOW_CODIGO	int	Secuencial de flujo.
FLOW_ANIO	int	Año en el que se produce el flujo
FLOW_OFI_CODIGO	int	Secuencial de Oficio.

Tabla 6.21 Tabla FLOW

Tabla		
Nombre:	MENU	
Descripción:	Tabla con campos de información de menú del sistema	
Campo	Tipo	Descripción
MENU_CODIGO	int	Secuencial de código de menú
MENU_NOMBRE	nvarchar(50)	Nombre del menú
MENU_ACTIVIVO	bit	Menú si esta activo true o false si no lo esta

MENU_PADRE	int	Secuencial de código padre
------------	-----	----------------------------

Tabla 6.22 Tabla MENU

Tabla		
Nombre:	FUNCIONARIO_MENU	
Descripción:	Tabla intermedia entre Funcionario y Menú, para saber que menús se presentaran a un funcionario en particular.	
Campo	Tipo	Descripción
FUN_CODIGO	nvarchar(10)	Código del Funcionario
MENU_CODIGO	int	Secuencial del Menú
FUN_MENU_ESTADO	bit	Menú para el usuario si es true o false

Tabla 6.23 Tabla FUNCIONARIO_MENU

Tabla		
Nombre:	TMP_MOVIMIENTO	
Descripción:	Tabla temporal con información de los movimientos en envío de oficios	
Campo	Tipo	Descripción
MOV_CODIGO	Int	Secuencial de movimiento
ENV_RECI	nvarchar(5)	Estado True si esta recibido false si no lo esta
MOV_NOMBRE	nvarchar(10)	Nombre del movimiento sea: Recibido o Pendiente

Tabla 6.24 Tabla TMP_MOVIMIENTOS

Tabla		
Nombre:	TMP_TIPO	
Descripción:	Tabla temporal con información de los tipos de envíos de oficios	
Campo	Tipo	Descripción
TIPO_CODIGO	int	Secuencial de tipo
TIPO	nvarchar(5)	P cuando es para y C cuando es copia
TIPO_NOMBRE	nvarchar(10)	Nombre del tipo Para o Copia

Tabla 6.25 Tabla TMP_TIPO

6.9.2.3 Diseño de la interfaz

Para el diseño de la interfaz del sistema informático para dar seguimiento a documentos y trámites en el Municipio de Cevallos se ha tomado como referencia los requerimientos proporcionados por la institución.

Pantalla de Ingreso

El sistema debe tener una pantalla de ingreso al mismo, en la cual deberemos ingresar un usuario y contraseña obtenidos previamente en su registro, estos datos deberán ser validados para permitir el acceso al sistema.

		X
Usuario:	<input type="text"/>	1
Contraseña:	<input type="text"/>	2
Base de Datos:	<input type="text"/>	3
	<input type="button" value="Conectar"/>	<input type="button" value="Cancelar"/> 5

Figura 6.12 Pantalla de Inicio

1. Caja de texto que permite el ingreso del nombre de usuario.
2. Caja de texto que permite el ingreso de la contraseña.
3. Caja de texto la cual tiene cargado el nombre de la base de datos.
4. Botón Conectar: Valida los parámetros de ingreso tanto usuario como contraseña. Si los parámetros son correctos ingresa a la pantalla principal del sistema, caso contrario genera un mensaje de error.
5. Botón Cancelar: Cierra la ventana de validación.

Pantalla Principal del Sistema

Cuando el usuario ingresa al sistema se presenta el siguiente menú de contenidos.

1							_ = X
	Archivo	Seguridad	Parámetros	Movimientos	Consultas y Reportes	Ayuda	2
3	Salir	Cambio Clave	Departamentos	Envió	Hoja de Guía	Acerca de...	
		Cambio Clave en Lote	Tipo Documentos	Documentos Recibidos	Documentos Enviados		
		Crear Usuarios	Tipo Trámites	Documentos Finalizados	Envió General		

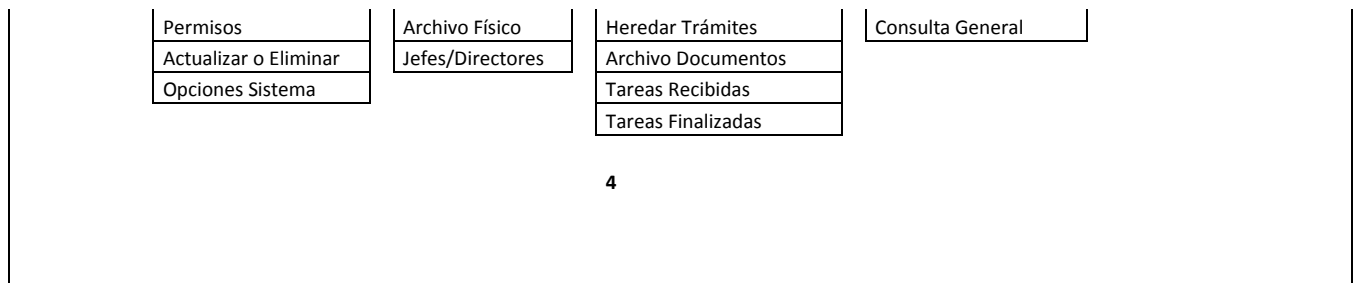


Figura 6.13 Pantalla Principal del Sistema

1. Sistema Principal: Esta es la ventana principal de nuestro sistema.
2. Menú Principal: Se podrá escoger 6 opciones como: Actualizar, Seguridad, Parámetros, Movimientos, Consultas y Reportes y Ayuda.
3. Submenú: Al ingresar en cualquiera de las opciones del menú principal se desplegara un submenú dependiendo de la acción a realizar en el sistema.
4. Contenedor de Formas: Es este espacio se cargaran las formas o ventanas dependiendo de la opción de submenú seleccionada.

Menú de Seguridad.

En esta opción de menú se encuentran varios submenús los cuales sirven para la administración de seguridad de nuestro sistema, ya sea para la creación de usuarios del sistema como para otorgar permisos a cada uno de ello, en la misma que podemos realizar cambios de claves tanto del propio usuario logueado como el administrador de cualquier usuario.

Ejemplo: Creación de Usuarios.

Guardar	Cancelar	1
2	3	
	Login:	<input style="width: 150px;" type="text"/> 4
	Contraseña:	<input style="width: 150px;" type="text"/> 5
	Confirmación:	<input style="width: 150px;" type="text"/> 6
	Cedula:	<input style="width: 150px;" type="text"/> 7
	Apellidos:	<input style="width: 250px;" type="text"/> 8

Nombres:	<input type="text"/>	9			
Email:	<input type="text"/>	10			
Dirección:	<input type="text"/>	11			
Teléfono:	<input type="text"/>	12	Celular:	<input type="text"/>	13
Departamento:	<input type="text"/>	V	14		

Figura 6.14 Creación de Usuarios

1. Barra de Botones: En esta sección se encuentra los botones de esta ventana.
2. Botón Guardar: Guarda el nuevo usuario creado.
3. Botón Cancelar: Cierra esta ventana.
4. Caja de Texto de ingreso de nombre de usuario.
5. Caja de Texto de ingreso de contraseña.
6. Caja de texto de ingreso de confirmación de contraseña.
7. Caja de texto de ingreso de cedula del usuario.
8. Caja de texto de ingreso de los apellidos de los usuarios creados.
9. Caja de texto de ingreso de los nombres de los usuarios creados.
10. Caja de texto de ingreso del mail del usuario.
11. Caja de texto de ingreso de la dirección.
12. Caja de texto de ingreso de número de teléfono.
13. Caja de texto de ingreso de número de celular.

14. Caja combo box de selección del departamento en el cual trabaja el usuario a crear.

Menú de Parámetros.

Es esta opción de menú se encuentran varios submenús los cuales sirven paramentrizar los distintos componente con los que trabajara nuestro sistema como: creación y administración de Departamentos, Ingreso y administración de documentos, Ingreso y administración de trámites, creación y administración de Archivos, y vista de Jefes/Directores por departamentos.

Ejemplo: Departamentos – Sección y Responsables

The screenshot shows a web form for managing departments. It includes a top button bar (1), a data table (2), and several input fields: ID (3), parent code (4), name (5), type (6), active status (9), and director/manager (10). There is also a checkbox (8) and a table for responsible parties (11).

Código	Nombre

ID: 3 Cód. Padre: 4

Nombre: 5

Tipo: 6 8

Activo: 9 Director / Jefe: 10

Responsable:

Login	Cedula	Funcionario	Rol

Figura 6.15 Departamentos – Sección y Responsable

1. Barra de Botones: En esta sección se encuentra los botones de esta ventana.
2. Caja de Datos: Carga todos los departamentos creados en el sistema.
3. Caja de texto de información de la identificación de los departamentos, esta casilla puede ser de carga o de ingreso según corresponda.
4. Caja de texto de información del código padre al que pertenece un departamento. Puede ser de carga o de ingreso.

5. Caja de texto de información del nombre del departamento, esta casilla puede ser de ingreso o carga de datos.
6. Caja combo box con la información del tipo de departamento, este puede ser de carga o permite escoger el tipo cuando se está creando uno nuevo.
7. Caja de texto de carga automática con la cedula luego de seleccionar un usuario creado al hacer clic en el botón de carga.
8. Botón de carga de Funcionario: Al hacer clic en este botón abre una nueva ventana con la información de todos los usuarios creados en el sistema para poder escoger uno que será seleccionado como jefe de departamento.
9. Check que señala si en departamento está activo o no.
10. Caja de texto de carga automática funciona de la misma forma que la caja de texto 7 pero esta carga el nombre del usuario seleccionado.
11. Caja de datos que carga la información de todos los usuarios que trabajan en un determinado departamento.

Menú de Movimientos:

En esta opción del menú se despliegan varios ítems de submenú los cuales sirven para realizar el ingreso, envío y tratamiento de los documentos y tramites que ingresen a la institución.

Ejemplo: Envío de Documentos.

1										
	AÑO:	<input type="text"/>	2	FW:	<input type="text"/>	3	Tramite:	<input type="text"/>	4	
	Tipo Documentos:	<input type="text"/>	5		Fecha Ingreso:	<input type="text"/>	6	Estado:	<input type="text"/>	
		<input type="checkbox"/>	V			<input type="checkbox"/>	V	7	Finalizar	25
	Número:	<input type="text"/>	8		Origen:	<input type="text"/>	9		Archivar	26
		<input type="checkbox"/>	V			<input type="checkbox"/>	V			
	Nombre:	<input type="text"/>	10	Búsqueda	11	<input type="text"/>	12	Nuevo	13	

Dirección:	<input type="text"/>	14	Institución:	<input type="text"/>	15
FW Origen:	<input type="text"/>	16	Oficio Refer:	<input type="text"/>	17
			Refer. Solic:	<input type="text"/>	18
Tratamiento:	<input type="text"/>				19
Asunto:	<input type="text"/>				20
	<input type="button" value="Documentos"/>	21	<input type="button" value="Anexos"/>	22	<input type="button" value="Tareas Inten."/>
					<input type="button" value="Enviar"/>
					24

Figura 6.16 Envío de Documentos

1. Barra de Botones: En esta sección se encuentra los botones de esta ventana.
2. Caja de texto en la cual no es editable, carga automáticamente el año en el cual nos encontramos.
3. Caja de texto no editable la cual al guardar un documento se carga de manera automática según el secuencial.
4. Caja combo box la que carga todos los trámites que se hayan ingresado en la base.
5. Caja combo box la que carga todos los documentos que se hayan ingresado en la base.
6. Caja de texto no editable la cual contiene la fecha y hora actual del sistema operativo.
7. Caja combo box no seleccionable que tiene los distintos tipos estados del documento sean estos: Ingresado, Enviado, Pendiente o Finalizado.
8. Caja de texto en el que se ingresara el número de documento que se está creado, si lo tiene.
9. Caja combo box en la que se encuentra el tipo de origen del tramite sea este Interno o Externo.

10. Caja de texto no editable la cual cargara el secuencial del remitente externo que este solicitando el presente documento.
11. Botón de búsqueda que al hacer clic carga una ventana con todos los nombres de los remitentes externos que se han creado en la base.
12. Caja de texto no editable, funciona de la misma forma que la caja de texto 10.
13. Botón de creación de nuevos remitentes externos en la base de datos.
14. Caja de texto no editable, funciona de la misma forma que la caja de texto 10.
15. Caja de texto la cual permite el ingreso de la institución a la cual se hará el presente documento.
16. Caja de texto no editable.
17. Caja de texto la cual permite el ingreso de información de un oficio referencia si existiera.
18. Caja de texto la cual permite el ingreso de información de solicitud de referencia si existiera.
19. Caja de texto que permite el ingreso del tratamiento que se dará al documento, esto debe res puntual.
20. Caja de texto que permite el ingreso del asunto, aquí se podrá hacer un resumen de lo que se pide en el presente documento.
21. Botón Documentos: Este botón funciona de manera conjunta con la caja 4. Al hacer clic se verán todos los documentos que habilitaran un trámite.
22. Botón Anexo: Al hacer clic nos abrirá una nueva ventana para realizar el ingreso de documentos que se anexan el documento que se está ingresando.

- 23. Botón Tareas Internas: Al hacer clic nos abre una nueva ventana para poder asignar a un funcionario de la institución una tarea que depende del presente documento que estamos ingresando.
- 24. Botón Enviar: Al hacer clic se abre una ventana con el nombre de los departamentos a los cuales podemos enviar el presente documento.
- 25. Botón Finalizar: Al hacer clic sobre este botón podemos dar como finalizado el documento que se está ingresando.
- 26. Botón Archivar: Al hacer clic en este botón se abre una ventada con el nombre de todos los archivos físicos a los cuales podemos asignar para archivar nuestro documento.

Menú de Consultas y Reportes:

En esta opción del menú se despliegan varios ítems de submenú los cuales sirven para realizar consultas y generar reportes de los documentos ingresados al sistema.

Ejemplo: Hoja de Guía

1						
Titulo del Reporte						2
Título del Reporte						
Tipo de Guía		Período				
Tarea: <input type="radio"/>	3	Departamento: <input type="radio"/>	4	Desde: <input type="text"/>	5	Hasta: <input type="text"/>
Obtención guía de documentos enviados a:						7
Obtención guía de documentos enviados a:						
Vista:						8
Tramite	Oficio	Motivo	Destino	Fecha	Remitente	

Figura 6.17 Hoja de Guía

1. Barra de Botones: En esta sección se encuentra los botones de esta ventana.
2. Caja de texto que permite ingresar el titulo que se dará al reporte a generar.
3. Radio Botón: Si este está seleccionado nos generara una reporte generar.
4. Radio Botón: Si este está seleccionado se habilitara la caja combo box 7 para seleccionar un departamento específico a donde se envió el documento.
5. Caja de texto que permite ingresar la fecha de ingreso del documento en un periodo desde.
6. Caja de texto que permite ingresar la fecha de ingreso del documento en un periodo hasta.
7. Caja combo box, estará desbloqueado cuando este seleccionado el radio botón 4, es esta caja se despliega el nombre de los departamentos destino.
8. Caja de Datos: En esta se cargara los datos de los documentos enviados según los parámetros seleccionado.

6.9.2.4 Desarrollo del sistema

El presente sistema será desarrollado en código C# mediante el uso de SharpDevelop como herramienta de programación.

6.9.2.4.1 Extracto de código fuente

Conexión a la base de datos mediante App.config utilizado para conectar usuarios creados en forms.

App.config
<pre> <?xml version="1.0" encoding="utf-8" ?> <configuration> <system.web> <membership defaultProvider="SqlProvider"> <providers> <clear /> <add name="SqlProvider" type="System.Web.Security.SqlMembershipProvider" connectionStringName="MiConnectionString" applicationName="Test" </pre>

```

        enablePasswordRetrieval="false" enablePasswordReset="true"
        requiresQuestionAndAnswer="false"
        requiresUniqueEmail="true" passwordFormat="Hashed"
        maxInvalidPasswordAttempts="3" passwordAttemptWindow="10"
minRequiredPasswordLength="1" minRequiredNonalphanumericCharacters="0" />
    </providers>
</membership>
    <roleManager enabled="true" defaultProvider="SqlRoleProvider">
        <providers>
        <clear/>
        <add name="SqlRoleProvider"
            type="System.Web.Security.SqlRoleProvider"
            connectionStringName="MiConnectionString"
            applicationName="Test"/>
        </providers>
    </roleManager>
</system.web>
<connectionStrings>
    <add name="MiConnectionString"
        connectionString="Database=MuniDoc;Server=VICTORB-PC\SQLEXPRESS;User=sa;Pwd=sa;"
        providerName="System.Data.SqlClient" />
</connectionStrings>
</configuration>

```

Código de validación de Usuarios.

```

ControlUsuario.cs
using System;
using System.Drawing;
using System.Windows.Forms;
//Para conectar
using DTBServer;
using System.Web.Security;
using System.Security.Principal;
using System.Data.SqlClient;

namespace DTBServer
{
    /// <summary>
    /// Description of ControlUsuario.
    /// </summary>
    public partial class ControlUsuario : Form
    {
        public ControlUsuario()
        {
            //
            // The InitializeComponent() call is required for Windows Forms
designer support.
            //
            InitializeComponent();

            //
            // TODO: Add constructor code after the InitializeComponent() call.
            //if (System
.Deployment.Application.ApplicationDeployment.IsNetworkDeployed)
            //{
                this.txtUsuario.Text = "";
                this.txtContrasena.Text = "";
            //}
            //
        }

        void BtnConectarClick(object sender, EventArgs e)
        {
            SqlConnection cnNorthwind = new SqlConnection();

```

```

        try
        {
            if (Membership.ValidateUser(txtUsuario.Text,
txtContrasena.Text))
            {
                //Parámetros de conexión
                ConectarBD.BaseDatos = "MuniDoc";
                ConectarBD.Contraseña = "sa";
                ConectarBD.Servidor = "VICTORB-PC\\SQLEXPRESS";
                ConectarBD.Usuario = "sa";
                ConectarBD.CadenaDeConexion = "Data Source=VICTORB-PC\\SQLEXPRESS;Initial
Catalog=MuniDoc;User ID=sa; Password=sa ";
                this.Hide();

                ClaseEstatica.usuariologueado = txtUsuario.Text ;
                ClaseEstatica.usuariopass = txtContrasena.Text;
                //Control
                SistemaSeguimiento ventana = new SistemaSeguimiento ();
                ventana.Show();
            }
            else
            {
                MessageBox.Show("Error de Acceso: Por favor, compruebe su nombre de
usuario y contraseña y vuelva a intentar", "Error al iniciar Sesión");
                LimpiarCampos ();
            }
        }
        catch (System .Exception ex)
        {
        }
        finally
        {
            this .Cursor = Cursors .Default ;
        }
    }

    void ControlUsuarioLoad(object sender, EventArgs e)
    {
        //Cuando ejecute
        txtBase .Text = "MuniDoc";
        txtBase .Enabled = false ;
    }

    void BtnCancelarClick(object sender, EventArgs e)
    {
        Application.ExitThread();
    }

    void Timer1Tick(object sender, EventArgs e)
    {
        lbReloj .Text = DateTime .Now .ToLongTimeString ();
    }

    private void AutenticarUsuario()
    {
        string unUsuario = txtUsuario .Text .Trim ();
        string unaClave = txtContrasena .Text .Trim ();
    }

    private void LimpiarCampos()
    {
        this.txtUsuario .Text = "";
        this .txtContrasena .Text = "";
        this.txtUsuario .Focus ();
    }

```

```

    }
    void TxtUsuarioEnter(object sender, EventArgs e)
    {
        this.txtUsuario.SelectAll ();
    }
    void TxtContrasenaEnter(object sender, EventArgs e)
    {
        this .txtContrasena .SelectAll ();
    }
}

```

6.9.3 Pruebas

Una de las últimas fases del ciclo de vida antes de entregar un programa para su explotación, es la fase de Pruebas.

La fase de pruebas añade valores al producto que se maneja: todos los programas tienen errores y la fase de pruebas los descubre: ese es el valor que añade. El objetivo específico de la fase de pruebas es encontrar todos los errores posibles generados y no controlados en el desarrollo del sistema.

Una vez concluido con el desarrollo del sistema se procedió a realizar las pruebas pertinentes.

6.9.3.1 Pruebas Caja Blanca

Las pruebas de caja blanca permitieron examinar la estructura interna y el funcionamiento del sistema en la lógica del negocio con respecto a la aplicación final, para lo cual fue necesario evaluar la lógica en todos los niveles y módulos que el sistema presenta, es decir:

- Se evaluaron los caminos independientes de cada uno de los módulos como:
 - Seguridad; ingresando un nuevo empleado, asignándole a un rol, cambiándolo de clave de acceso y asignándole roles adicionales.

- Parámetros; Ingresando nuevos departamentos, documentos y trámites.
 - Movimientos; Ingresando un nuevo ingreso de documentos externos con sus respectivos pasos y estados.
 - Consultas y Reportes; realizando la consulta del documento ingresado anteriormente.
- Se evaluaron las decisiones lógicas realizando ingresos erróneos en cada uno de los módulos para verificar su correcto funcionamiento.
 - Se ejecutaron las estructuras de datos internos realizando un ingreso de un documento desde cero para probar su comportamiento.
 - Se evaluaron las respuestas de los posibles errores a cometer como ejemplo se ingreso un usuario que ya existe en el sistema probando que no puede ser ingresado por duplicidad en la identificación como la cédula.
 - Se evaluaron las validaciones existentes en el control de ingreso de usuarios al sistema, realizando ingresos con usuarios y contraseñas falsa probando que no ingresa si no es validada la cadena de conexión e ingreso.
 - Se evaluaron todos los bucles, esto en lo que respecta al seguimiento que se le da a un documento ingresado al sistema.

De esta manera se intentó encontrar la mayoría de errores posibles que se pudieron provocar por una inadecuada programación y falta de control para darle una solución óptima.

6.9.3.2 Pruebas Caja Negra

Las pruebas de caja negra, se realiza sobre la interfaz fina, por lo tanto estas pruebas son completamente indiferentes del comportamiento interno y la estructura de las capas lógicas del sistema.

Con estas pruebas se pudo demostrar que.

- Las funciones del sistema son completamente operativas o factibles ya que un usuario puede realizar un ingreso desde la etapa cero y darle seguimiento hasta el final sin tener problemas en el manejo.
- El ingreso de usuario al sistema se realiza de forma adecuada y sencilla mediante una ventana amigable con el usuario.
- Los datos y servicios del Sistema presenta la información de forma adecuada y sencilla.
- Se mantiene la integridad del sistema a lo largo de su utilización, la cual fue probada en un periodo de pruebas.

Estas pruebas demostraron que las funciones del sistema son completamente operativas y factibles, registros correctos, obtención de información de forma adecuada e integridad de la información.

6.9.3.3 Pruebas de verificación y validación

Las pruebas de verificación y validación se realizaron con el fin de verificar que el sistema cumpla con las especificaciones planteadas. La validación permitió comprobar lo que se ha especificado, es decir lo que el usuario realmente necesita.

Verificación: Este proceso determinó que el sistema satisface las condiciones impuestas al comienzo del proyecto. Como:

1. Ingreso al sistema por validación de usuarios y contraseñas.

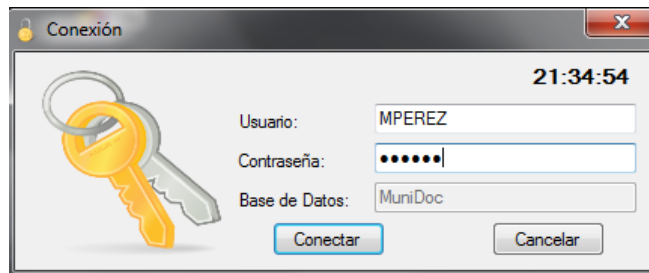


Figura 6.18 Ingreso al Sistema

2. Creación de Usuarios con validaciones de datos: login, contraseña, cedula, nombres, apellidos, correo, etc.

Registro de Usuarios al Sistema

Usuario

Login: JRAMOS

Contraseña: ***** Confirmación: *****

Cedula: 1804042693

Apellidos: RAMOS ESPINOZA

Nombres: JORGE WASHINGTON

Email: jorge_ramos@mail.com

Dirección: CEVALLOS

Teléfono: Celular: 098765432

Departamento

Selección Departamento: ADMINISTRATIVO

Figura 6.19 Creación de Usuarios

3. Ingreso de Trámites y Documentos.

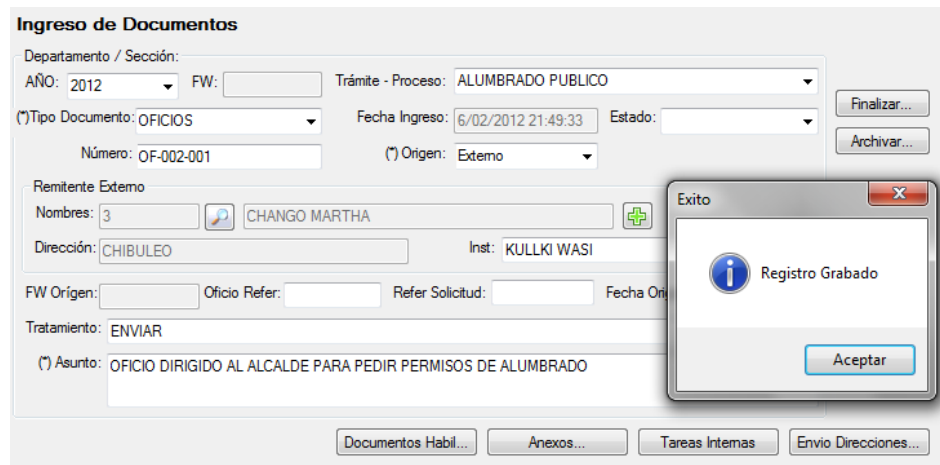


Figura 6.20 Ingreso de Trámites y Documentos

4. Envío de Documentos a dependencias municipales.

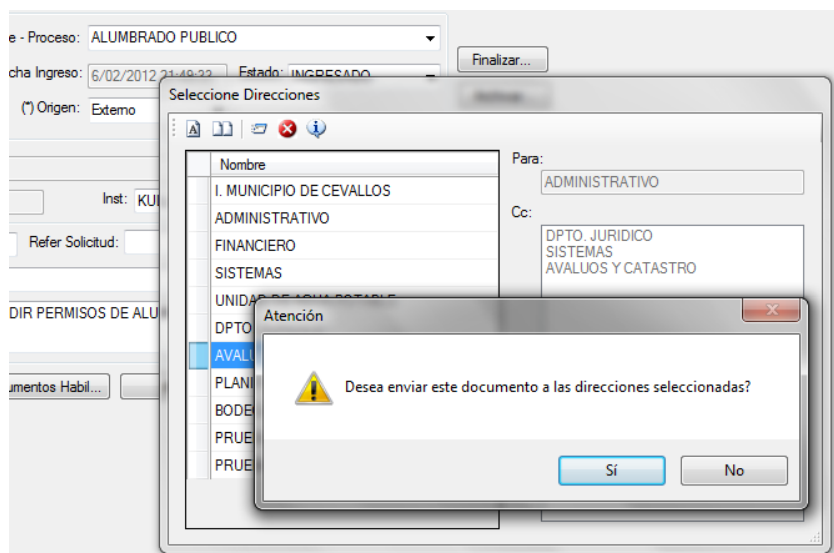


Figura 6.21 Envío de Documentos

5. Finalización de Documentos enviados.

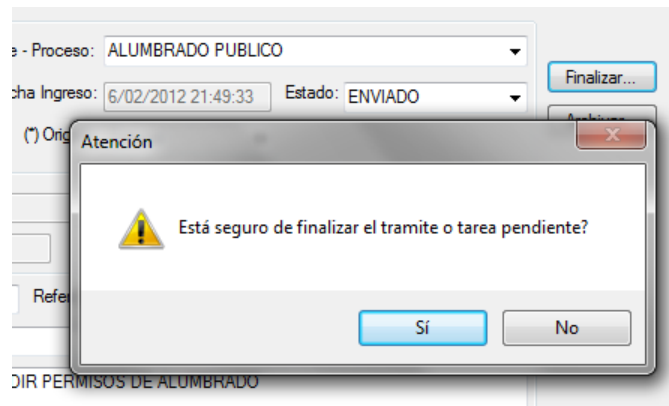


Figura 6.22 Finalización de Documentos

6. Consulta para el seguimiento de los distintos Documentos ingresados a la municipalidad.

Figura 6.23 Consulta de Documentos

Validación: Esta prueba verificó si se cumple con las expectativas del cliente.

- Pruebas de aceptación y validación que fueron desarrolladas por el administrador.

- Pruebas alfa que fueron realizadas por los usuarios que ingresaron y manipularán la información del sistema con el desarrollador como observador.
- Pruebas beta que fueron realizadas por los usuarios que ingresaron y manipularan la información del sistema de manera individual.

Todas estas pruebas fueron realizadas por los funcionarios de la institución de manera concreta en un entorno de trabajo real obteniendo los resultados deseados por ellos.

6.9.4 Implantación del Sistema

Luego de que el sistema informático para el seguimiento de documentos y trámites en el Municipio de Cevallos pasó satisfactoriamente las pruebas se procedió a su implantación.

Primer Paso: Base de Datos

Crear la base de datos junto a las tablas que necesita el Sistema Informático para su funcionamiento. Para lo cual se contó con la ayuda del Técnico encargado de Sistemas de la Institución.

Segundo Paso: Instalación del Sistema en la Institución

Se procedió a la instalación del sistema en todas las máquinas de la institución realizando primero la instalación del Framework 3.5 necesario para el funcionamiento del sistema.

Mediante los pasos anteriormente mencionados se concluyó que el método utilizado fue la Implantación Directa.

6.9.4.1 Cronograma de Implantación del Sistema.

Nombre de Tarea	Duración
Implantación	
Instalación de la Bases de datos	1 Días
Ejecución código y creación de la Base de Datos	1 Días

Creación de las tablas MembesShip	1 Días
Verificación de operaciones de la BD	1 Días
Acceso al sistema, e ingreso de datos a la base	1 Días
Verificación de todos los módulos del sistema	3 Días
Pruebas	
Pruebas de caja negra, blanca y validación	15 Días
Capacitación	
Capacitación al funcionario encargado del sistema	2 Días

Tabla 6.26 Cronograma de Implantación del Sistema

6.9.4.2 Levantamiento del Sistema

Para la implantación del Sistema Informático, se iniciará con la instalación del software requerido como son:

- Instalación del Motor de Base de Datos SQL Server 2000
- Instalación del manejador de Base de Datos Microsoft SQL Server Management Studio.
- Instalación de NET Framework 3.5 y 4.
- Instalación del sistema.

Finalmente nuestro sistema está listo para ser usado.

6.10 Conclusiones y Recomendaciones

6.10.1 Conclusiones

- La implantación de un sistema informático para dar seguimiento a Documentos y Trámites en el Municipio de Cevallos permitirá tener un mejor control de la documentación solicitada por cualquier persona ajena a la municipalidad.

- El sistema reducirá el tiempo en consultar manualmente que tipo de tratamiento se le está proporcionando a un determinado trámite solicitado.
- Los funcionarios de la institución ahora cuentan con una fuente muy impórtate la cual ayudara a agilizar los tratamientos que cada uno de ellos tengan que dar a un determinado tramite que ingrese a la municipalidad.
- El departamento de información reducirá tiempo de consulta telefónica que realiza alguna persona acerca de un determinado trámite solicitado.
- El sistema ayudara tanto a los funcionarios que trabajan en la institución como también a personas ajenas a la misma las cuales acuden a solicitar un determinado trámite y necesitan su pronta atención y solución.

6.10.2 Recomendaciones

- Se recomienda realizar pruebas minuciosas de funcionamiento y productividad del Sistema en todas las maquinas en las que va a estas instalado el sistema para verificar su correcto funcionamiento.
- Realizar una capacitación a todo el personal que utilizara el sistema para evitar un mal uso del mismo.
- Realizar respaldos periódicos de la base de datos para salvaguardar la información de la institución de posible mala manipulación de la base.
- Es importante que los usuarios le den en buen uso al sistema y que se realice las acciones correspondientes a lo solicitado por el sistema.

6.11Bibliografía

6.11.1 Información Bibliográfica de Libros

- CERVO, Amado (1997) *Metodología de la Investigación*. McGrawHill.

- CAZAR, Héctor (2001) *Compendio de Computación*. Librería Española.
- MEDINA, Washington (2008) *Guía para el desarrollo de trabajos de graduación*.
- PAZMAY, Galo (2004) *Guía práctica para la elaboración de tesis y trabajos de investigación*. Editorial Freire.

6.11.2 Información Bibliográfica de Internet

- Microsoft ® Encarta ® 2009. © 1993-2008 Microsoft Corporation
- DocFlow. Recuperado el 05 enero 2012, desde <http://www.giro.infor.uva.es/oldsite/docproy/wfs/presentacionDocFlow.htm>
- DocFlow. Gestión Documental. Recuperado el 05 enero 2012 desde <http://www.slideshare.net/tys2009/docflow>
- DocFlow. Document Portal. Recuperado el 05 enero 2012 desde <http://docflow.codeplex.com>
- DocFlow. Recuperado el 05 enero 2012 desde http://www.labri.fr/perso/anca/docflow/main_files/description_scientifique.pdf
- DOCFLOW. Recuperado el 05 enero del 2012 desde <http://www.markosweb.com/www/docflow.ru/>
- ANR Docflow. Recuperado el 05 enero del 2012 desde <http://ralyx.inria.fr/2008/Raweb/distribcom/uid37.html>
- El hipertexto documental como solución a la crisis conceptual del hipertexto. Recuperado el 05 enero 2012 desde <http://www.ucm.es/info/multidoc/multidoc/revista/cuadern4/hiperdoc.htm>
- DOC FLOW SOCIEDAD ANONIMA CERRADA - DOC FLOW S.A.C. Recuperado el 05 enero 2012 desde <http://www.universidadperu.com/empresas/doc-flow-sociedad-anonima-cerrada-doc-flow.php>

- SQL Server 2000. Recuperado el 05 enero del 2012 desde http://www.netveloper.com/contenido2.aspx?IDC=64_0
- SharpDevelop. Recuperado el 05 enero del 2012 desde <http://sharpdevelop.malavida.com/d3784-descargar-windows>
- SQL Server 2000. Recuperado el 05 enero 2012 desde <http://www.abcdatos.com/tutoriales/tutorial/11637.html>
- Wikipedia la enciclopedia libre. Microsoft SQL Server. Recuperado 05 enero 2012 desde http://es.wikipedia.org/wiki/Microsoft_SQL_Server
- Wikipedia la enciclopedia libre. SharpDevelop. Recuperado 05 enero 2012 desde <http://es.wikipedia.org/wiki/SharpDevelop>
- The Open Source Development Environment for .NET. Recuperado el 05 enero del 2012 desde <http://www.icsharpcode.net/OpenSource/SD/>
- Funciones en SQL Server 2000. Recuperado el 05 de enero 2012 desde <http://www.sqlmax.com/func1.asp> y <http://www.sqlmax.com/func2.asp>
- Wikipedia la enciclopedia libre. Microsoft SQL Server. Recuperado el 05 enero 2012 desde http://es.wikipedia.org/w/index.php?title=Microsoft_SQL_Server
- JASoft.org. Creación de tablas de proveedores ASP.NET desde línea de comandos. Recuperado el 07 enero 2012 desde <http://www.jasoft.org/Blog/post/Permalink.aspxguid=2689b369-03ed-4c52-9878-98a249f.aspx>
- Msdn. Membership Class. Recuperado del 07 enero 2012 desde <http://msdn.microsoft.com/en-us/library/system.web.security.membership.aspx>
- DotNetJalps. Custom Membership and Role Provider In ASP.NET. Recuperado el 07 enero del 2012 desde <http://www.dotnetjalps.com/2006/07/custom-membership-and-role-provider-in.html>

- Using the ASP.NET membership provider in a Windows forms application. Recuperado el 07 enero del 2012 desde <http://www.theproblemsolver.nl/usingthemembershipproviderinwinforms.htm>
- ASP.NET Membership - Retrieve Password and PasswordSalt from Membership Table - Hash UserID. Recuperado el 07 enero del 2012 desde <http://stackoverflow.com/questions/2741917/asp-net-membership-retrieve-password-and-passwordsalt-from-membership-table>
- Shining Star Services LLC. Creating a Custom Membership Provider and Membership User utilizing a DataSet Table Adapter. Recuperado el 07 enero 2012 desde http://www.shiningstar.net/ASPNet_Articles/CustomProvider/CustomProvider.aspx
- AJDev.net. ASP.net: Creación de una Aplicación con Membership y Roles. Recuperado el 07 de enero de 2011 desde <http://ajdev.net/Publicacion/Articulo/67.aspx> y <http://ajdev.net/Publicacion/Articulo/68>
- Crear Usuarios con Membership Provider. Recuperado del 07 enero de 2012 desde <http://www.omdsite.com/blog/post/Crear-Usuarios-con-Membership-Provider.aspx>
- Cómo crear las tablas del Membership y Role Provider en SQL Server. Recuperado el 07 enero 2012 desde <http://www.viale.mx/2008/02/22/como-crear-las-tablas-del-membership-y-role-provider-en-sql-server/>
- TARINGA. Como conectar SQL y dataGridView con C# Recuperado el 07 enero 2012 desde http://www.taringa.net/posts/taringa/2384002/Como-conectar-SQL-y-dataGridView-con-C_.html
- C# TreeView Tutorial. Recuperado el 07 de enero de 2012 desde <http://www.dotnetperls.com/treeview>
- DotNetcr.com. Llenar un Control TreeView dinámicamente desde una Base de Datos, con ASP.NET 2005. Recuperado el 07 de enero de 2012 desde

<http://www.dotnetcr.com/recurso/Llenar-un-Control-TreeView-dinamicamente-desde-una-Base-de-Datos-con-ASPNET-2005>

- Llenar un TreeView con las bases de datos de MySql en C#. Recuperado el 07 de enero de 2012 desde <http://www.ubicuos.com/2010/08/06/llenar-un-treeview-con-las-bases-de-datos-de-mysql-en-c/>
- Msdn. TreeView (Clase). Recuperado el 07 de enero de 2012 desde <http://msdn.microsoft.com/es-es/library/system.web.ui.webcontrols.treeview.aspx>
- Utilizando el proveedor de suscripciones ASP.Net en un 4 Windows forma parte de la aplicación. Recuperado el 07 de enero de 2012 desde <http://msmvps.com/blogs/theproblemsolver/archive/2006/01/15/81140.aspx>
- Pdf a través de datagridview C #. Recuperado el 07 de enero de 2012 desde <http://www.dotnetspider.com/forum/247558-Pdf-through-datagridview-C.aspx>
- DataGridView Exportación a Excel. Recuperado el 07 de enero de 2012 desde <http://www.kodefuguru.com/post/2010/06/02/DataGridView-Export-to-Excel.aspx>
- GridView.NET. Exportación DataGridView a Excel en C #. Recuperado el 10 de enero 2012 desde <http://www.gridview.net/export-datagridview-to-excel-in-c/>

6.12 Glosario de términos.

Base de Datos.- Es un conjunto de información almacenada en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos.

Software Libre: programas o aplicaciones informáticas que pueden ser usadas, adquiridas, copiadas y distribuidas de manera libre y de manera gratuita.

Framework: Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

Interfaz.- Permite la circulación correcta y sencilla de información entre varias aplicaciones y el usuario.

Login.- Autenticación en la seguridad del ordenador, la autenticación es el proceso de verificar la identidad digital del remitente de una comunicación como una petición para conectarse.

IDE: Integrated Development Environment o Entorno integrado de desarrollo, aplicación compuesta por un conjunto de herramientas útiles para un programador.

Open Source: Código abierto es el término de con el que se conoce al software distribuido y desarrollado libremente.

GPL: General Public License o Licencia Pública General, licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software libre.

UML: Unified Modeling Language o Lenguaje Unificado de Modelado, es un lenguaje gráfico para especificar, visualizar, construir y documentar los sistemas de software, representa un conjunto de las mejores prácticas que han probado ser exitosas en el modelado de sistemas grandes y complejos.

ANEXOS

ANEXO 1: MODELOS DE ENTREVISTAS PERSONALES

UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL
CARRERA DE INGENIERÍA EN SISTEMAS

NOTA: Modelo de Entrevista orientada al Jefe Administrativo.

1. ¿Qué medio de transmisión utiliza para la interconexión de Pc's?

Cableado

Inalámbrico

Otro:.....

2. ¿Cuál es su proveedor del servicio de Internet?

Punto Net

CNT

Speedy

InterActive

Portal

Otros:.....

3. ¿Con que dispositivos IP trabaja la Empresa?

Switches

Routers

Puentes

Encaminadores

Otro:.....

4. ¿Conoce Ud. lo que es arquitectura n capas?

Si

No

5. ¿Con qué arquitectura de software considera Ud. que se debería desarrollar el sistema?

Cliente-Servidor

Cliente-Servidor 2 Capas

Cliente-Servidor 3 Capas

Cliente-Servidor n Capas

6. ¿Con qué lenguaje de programación considera Ud. que se debería desarrollar el sistema?

.....

7. ¿Qué tipos de motor de Base de Datos se encuentra instalado en el Servidor?

.....

8. ¿La información de los usuarios se encuentra almacenada en alguna base de datos?

Si

Parcialmente

No

9. ¿Conoce como funciona un DocFlow?

Si

No

.....

UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL
CARRERA DE INGENIERÍA EN SISTEMAS

NOTA: Entrevista dirigida al personal institucional del Ilustre Municipio de Cevallos.

1. ¿La empresa dispone de equipos de cómputo adecuados para su trabajo?
Si
No
.....
2. ¿Los equipos de Cómputo están interconectados entre sí a través de una red?
Si
No
.....
3. ¿Conoce la diferencia entre software Libre y Propietario?
Si
No
4. ¿Conoce con que sistema operativo trabaja el servidor?
SO Libre
SO Propietario
5. ¿Ha utilizado software Libre alguna vez?
Si
No
6. ¿De qué manera se maneja la información de los documentos y tramites en el Municipio de Cevallos?

Manualmente

Excel

Otro:.....

7. ¿Qué tipo de información se maneja en la institución en lo que respecta a trámites y documentos?

.....

8. La red de comunicación se utiliza para:

Datos

Videos

Imágenes

Otros:.....

9. ¿El servicio de red es óptimo para su trabajo?

Si

No

Porque:.....

10. ¿Dispone de acceso a Internet?

Si

No

11. ¿Qué tipo de documentos ingresan y circulan en la institución?

.....

12. ¿Qué tipo de Trámites ingresan y circulan en la institución?

.....

13. ¿Cómo se maneja la información actualmente en la institución?

.....

14. ¿Sabe que es DocFlow?

Si

No

15. ¿Conoce de software DocFlow existente en el mercado?

Si

No

ANEXOS 2: MANUAL DE INSTALACIÓN DE SOFTWARE REQUERIDO

Anexo 2.1: Manual de Instalación de SQL Server 2000.

A continuación se presenta gráficamente el proceso de instalación del SQL Server 2000.

Requisitos:

- Sistemas operativos compatibles: Windows 2000 Advanced Server, Windows 2000 Professional Edition, Windows 2000 Server, Windows Server 2003, Windows XP Professional Edition o superiores.
- 300 Mbytes de espacio libre como mínimo.

Proceso de instalación:

Ingresar el disco de instalación y abrirlo.

Pulsar doble clic sobre el fichero “AUTORUN” que se encuentra en el disco, mostrándole la pantalla de la Figura A1.1. Pulsamos entonces sobre la opción “SQL Server 2000 Componentes”, lo que le llevará a la pantalla de la Figura A1.2. En esta seleccionará la opción “Install database server”.

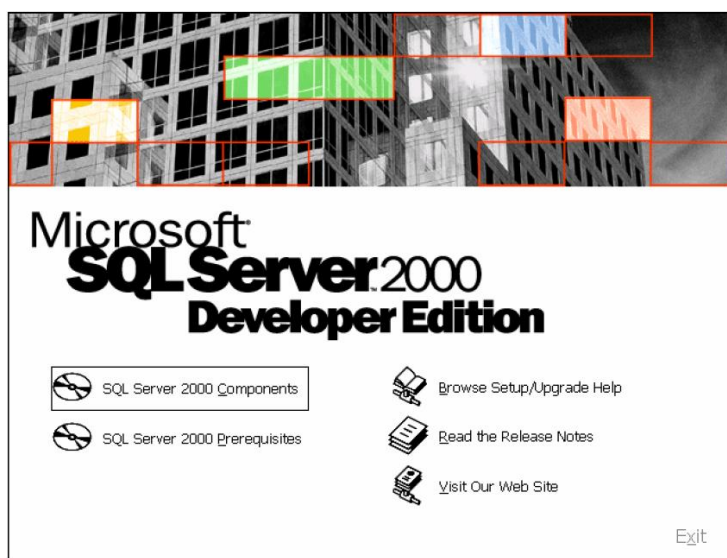


Figura A1.1 Microsoft SQL Server 2000

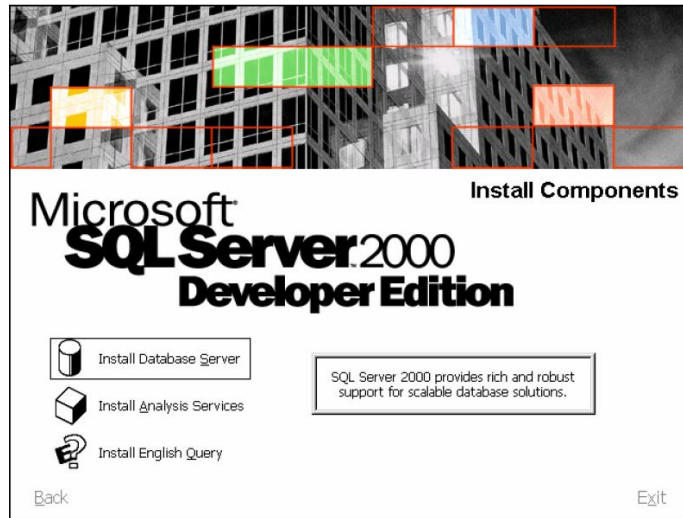


Figura A1.2 Microsoft SQL Server 2000 Install

A continuación se le irán presentando diferentes ventanas, como la de acuerdos de licencia que irá aceptando hasta llegar a la pantalla de la Figura A1.3, donde seleccionará la opción “Local Computer”. Esto significa que se instalará el gestor en la propia máquina.

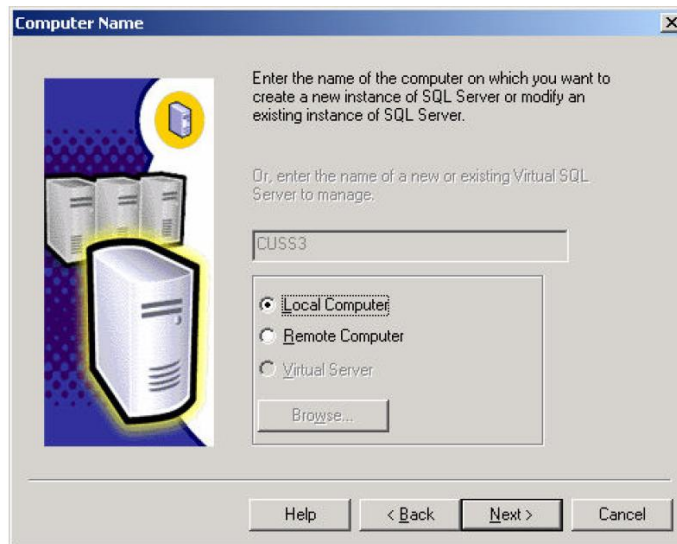


Figura A1.3 Microsoft SQL Server 2000 Name

A continuación se seleccionará la opción “Create a new...” Como indicamos en la Figura 1.4, posteriormente seleccionamos la opción “Server and clients tools” con la

que se instalará el gestor y herramientas cliente como son Enterprise Manager y SQL Analyzar.

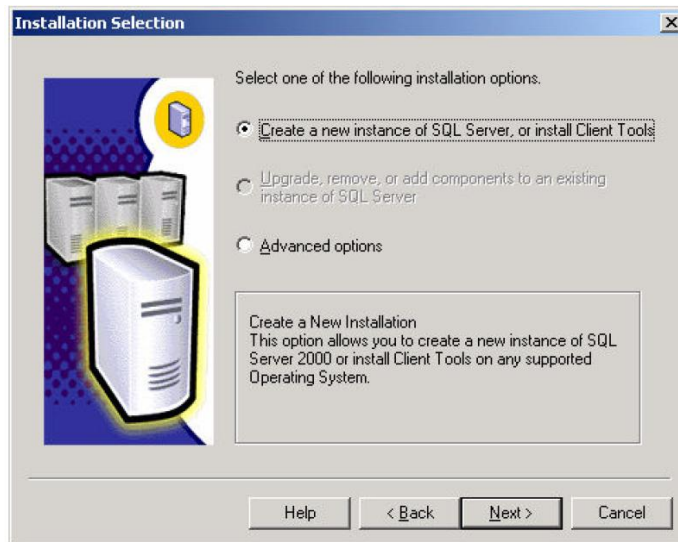


Figura A1.4 Microsoft SQL Server 2000 Selection

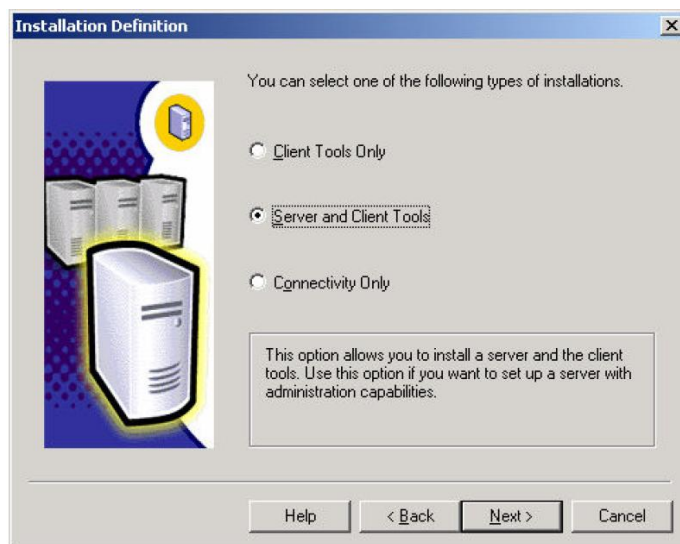


Figura A1.5 Microsoft SQL Server 2000 Definition

Se pulsará Next y aparecerá la ventana de la Figura A1.6. Se tomará la opción Default.

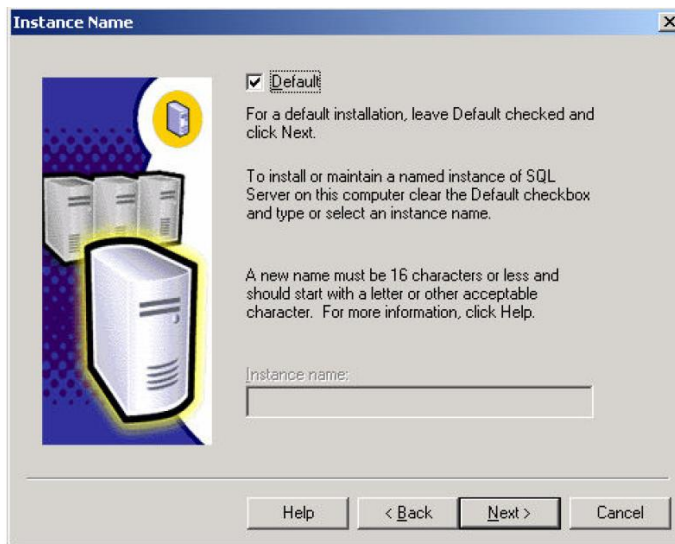


Figura A1.6 Microsoft SQL Server 2000 Name

Aparecerá la ventana de la Figura A1.7 donde se establecerá la ubicación física donde se instalarán los archivos de programa y de los datos.

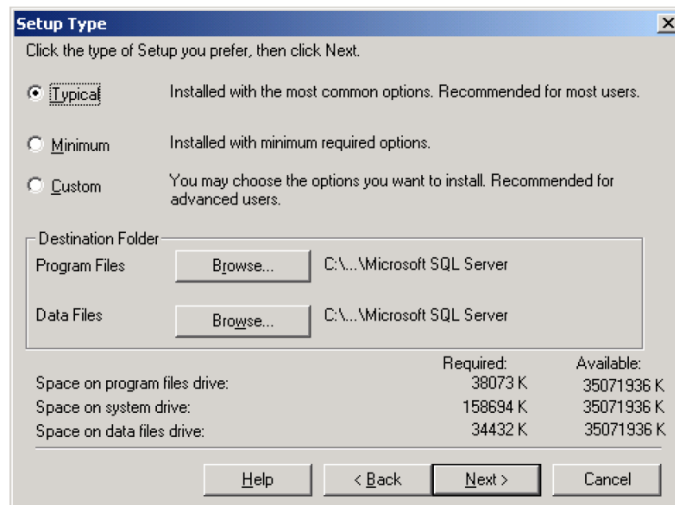


Figura A1.7 Microsoft SQL Server 2000 Setup

El SQL Server se instalara en 2000 como un servicio y, por tanto, es necesario asignar un usuario que inicie ese servicio y para que el SQL Server trabaje en ese contexto de seguridad. En nuestro caso como se trata de una instalación local, tomaremos la opción señalada en la Figura A1.8.

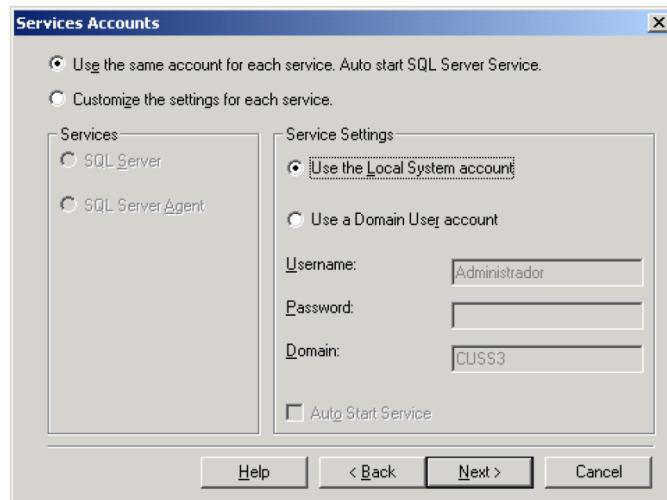


Figura A1.8 Microsoft SQL Server 2000 Accounts

Se tomara el método de autenticación de Windows.

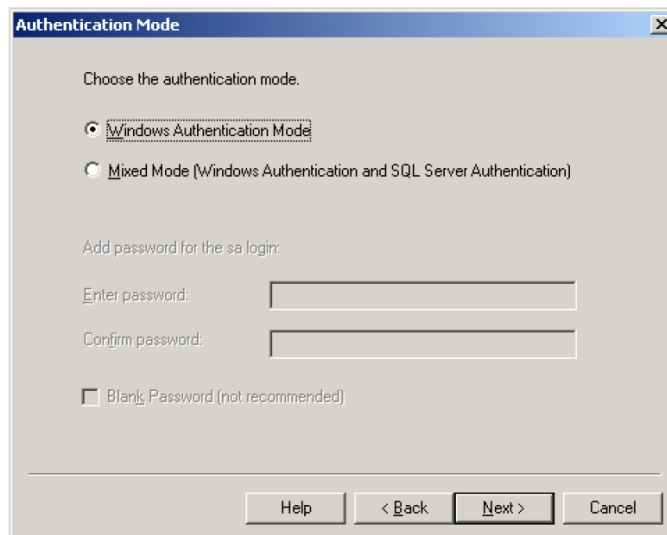


Figura A1.9 Microsoft SQL Server 2000 Mode

Finalizado el proceso de instalación, se debe arrancar el servicio. Comprobaremos que se han instalado los programas que se muestran en la Figura A1.10.

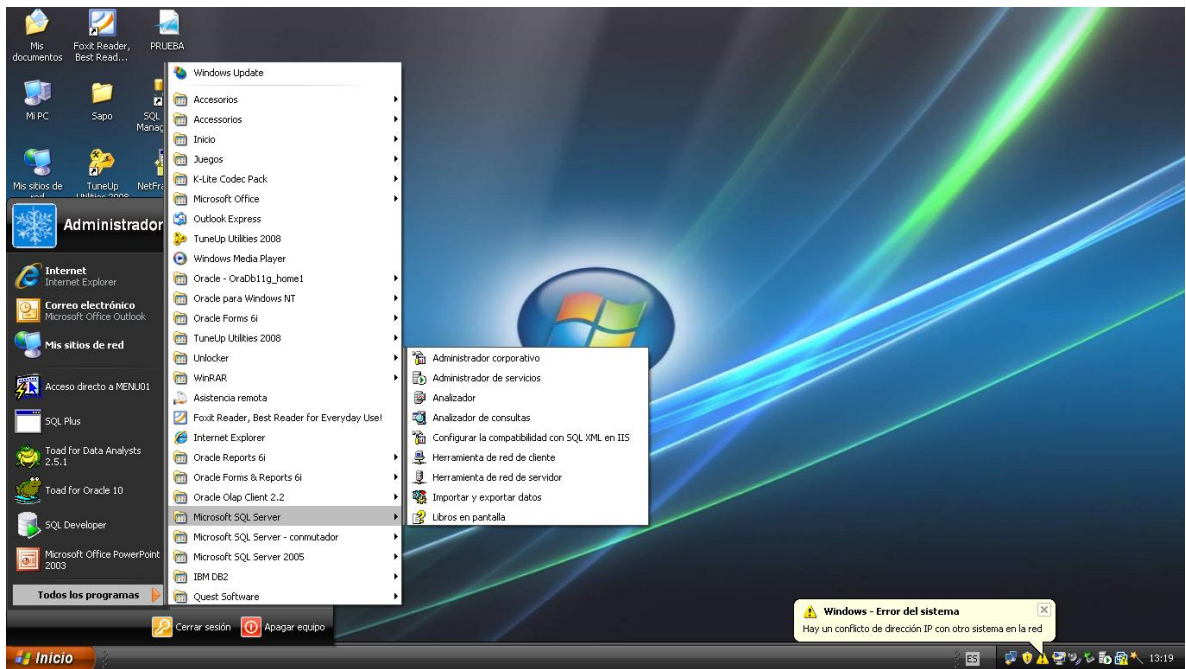


Figura A1.10 Escritorio

Se tomará el programa Service Manager que permite arrancar y parar los servicios que incluye el motor: SQL Server, SQL Server Agent, Coordinador de transacciones y MSSQLServerOLAPService. Se arrancará SQL Server exclusivamente.



Figura A1.11 Arrancar SQL Server

Anexo 2.2: Instalación de SQL Server Management.

Para poder utilizar la Base de datos debemos instalar Microsoft SQL Server Management Studio.

Hacemos doble clic en el programa de instalación que es la Figura A2.1.



Figura A2.1 Instalador Management

Se nos presenta la siguiente pantalla como en la Figura A2.2 y presionamos el botón Siguiente.



Figura A2.2

Aceptamos la licencia, ingresamos la información de registro, miramos que se va a instalar el Manager y presionamos siguiente, por ultimo instalar.

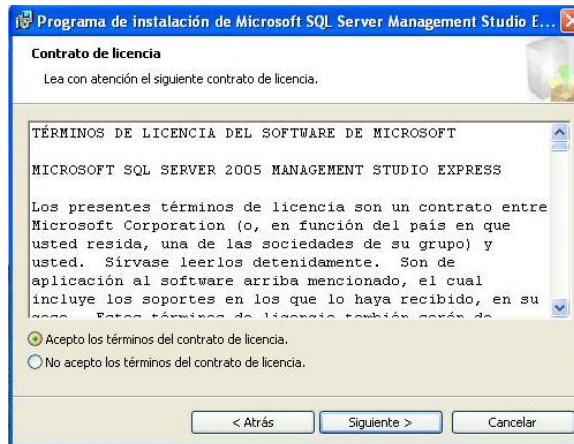


Figura A2.3

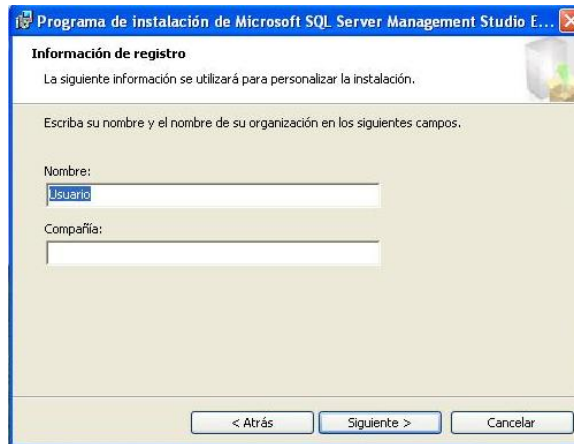


Figura A2.4

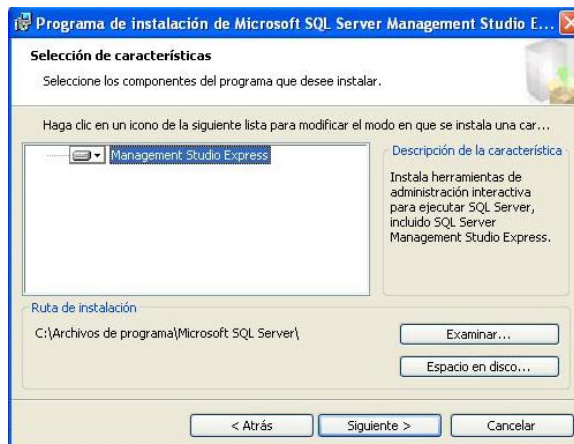


Figura A2.5

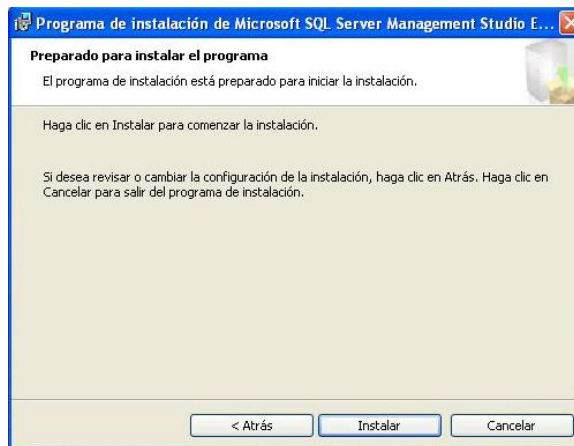


Figura A2.6



Figura A2.7

Una vez concluido el proceso de instalación finalizamos y listo, ya tenemos instalado el manejador para poder manipular la base de datos.



Figura A2.8

Anexo 1.3: Manual de Instalación de SharpDevelop

A continuación se presenta gráficamente el proceso de instalación:

#develop como es llamado también, es un **IDE libre** para C# y proyectos de VB.NET sobre la plataforma Microsoft .NET. También es **open-source**, es decir de código abierto y como sabemos esto significa que podemos descargar las fuentes y modificarlas a nuestro gusto.

Con #develop podremos desarrollar proyectos fácilmente, gracias a su soporte para plantillas de diseño y para diversos lenguajes de programación.

#develop incluye:

- Diseñador de Formularios
- Completado de Código
- Auto insertado de Código
- Conversor de Código C# a VB.Net y viceversa
- Importar/Exportar Soluciones VS.NET a Visual Studio .NET
- Plegado de Código ("Folding")
- Visor gráfico para realizar pruebas con NUnit
- Analizador del Código Ensamblador
- Vista previa de Documentación en XML
- También incluye sintaxis coloreada, paréntesis inteligentes, bookmarks, plantillas, herramientas para expresiones regulares, asistentes, exportación HTML, visor de clases, integración con NDoc, integración con Nprof, etc.

Requerimientos:

- Windows XP SP2 y posteriores (por ejemplo, Windows Server 2008 R2 y Windows 7).
- 20 MB aprox. de espacio en disco.
- Súper VGA (800 x 600) o superior con 256 colores.

Para poder instalar el programa, primero debemos descargarlo de:
<http://www.icsharpcode.net/OpenSource/SD/>

Antes de instalarlo debemos tener instalados los Framework:

- . NET 4.0 tiempo de ejecución total.
<http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=17718>
- . NET 3.5 SP1 en tiempo de ejecución.
<http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=22>



Figura A3.1 Microsoft .NET

Una vez descargados e instalados los Frameworks y habiendo descargado ya el instalador de (SharpDevelop) debemos **ejecutar** el programa de instalación, veremos entonces la **pantalla de bienvenida** a lo cual hacemos clic en **siguiente**.

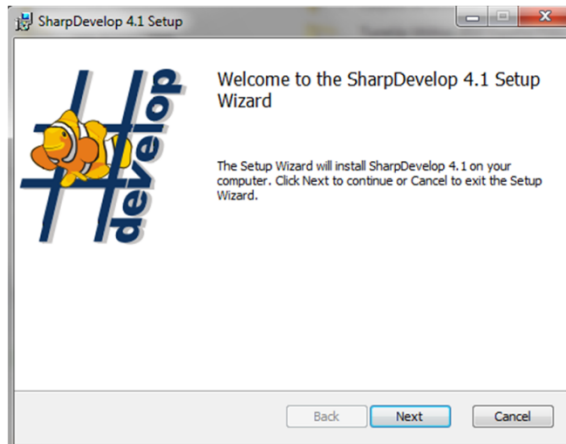


Figura A3.2 SharpDevelop Setup

En esta pantalla debemos **aceptar** los términos de **licencia**.

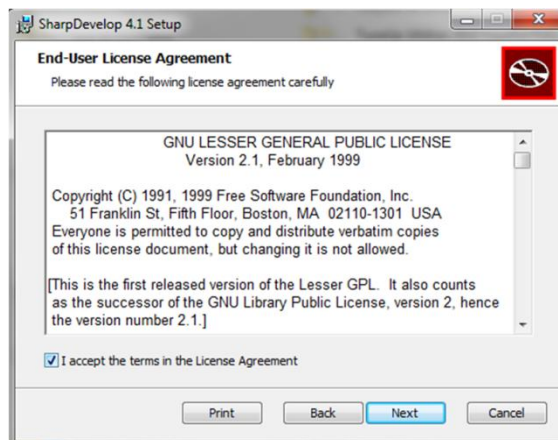


Figura A3.3 SharpDevelop Setup Licencia

Aquí podremos **cambiar** o no el **directorio de instalación**, hecho esto hacer clic en **instalar**.

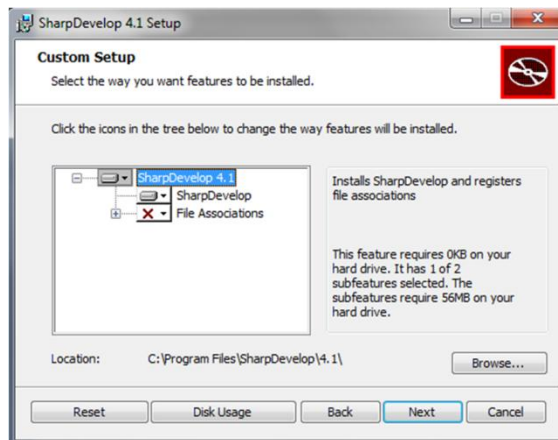


Figura A3.4 SharpDevelop Setup Dirección

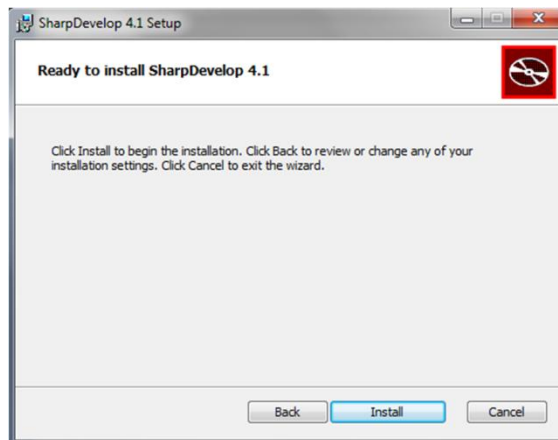


Figura A3.5 SharpDevelop Setup Ready

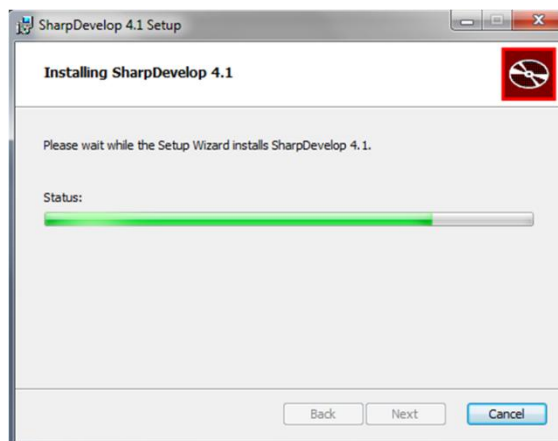


Figura A3.6 SharpDevelop Setup Ready

Después de culminada la **instalación** que por cierto es muy rápida veremos en el **prompt** algunos **componentes** que ya han sido instalados correctamente y por ultimo debemos hacer clic en **finalizar**.

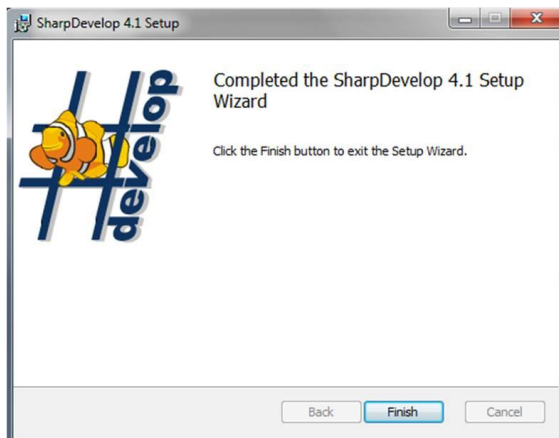


Figura A3.7 SharpDevelop Setup Finish

ANEXO 3: CREACIÓN DE TABLAS MEMBERSHIP

Cómo crear las tablas del Membership y Role Provider en SQL Server

Si vas a implementar el Membership Provider de ASP.NET con SQL Server, seguramente encontrarás mucha información en Internet de cómo hacerlo. La parte que no todo mundo explica es cómo crear las tablas que contendrán toda la información de nombres de usuario, contraseñas, roles, etc.

Las tablas se crean mediante la herramienta `aspnet_regsql.exe` que se encuentra en el directorio de la versión del Framework que estés usando (ejemplo: `C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727`). Si usas la versión 3.5 tienes que usar la herramienta que viene en el fólder de la versión 2.0

Abre una ventana de MS-DOS y cámbiate a dicho fólder:

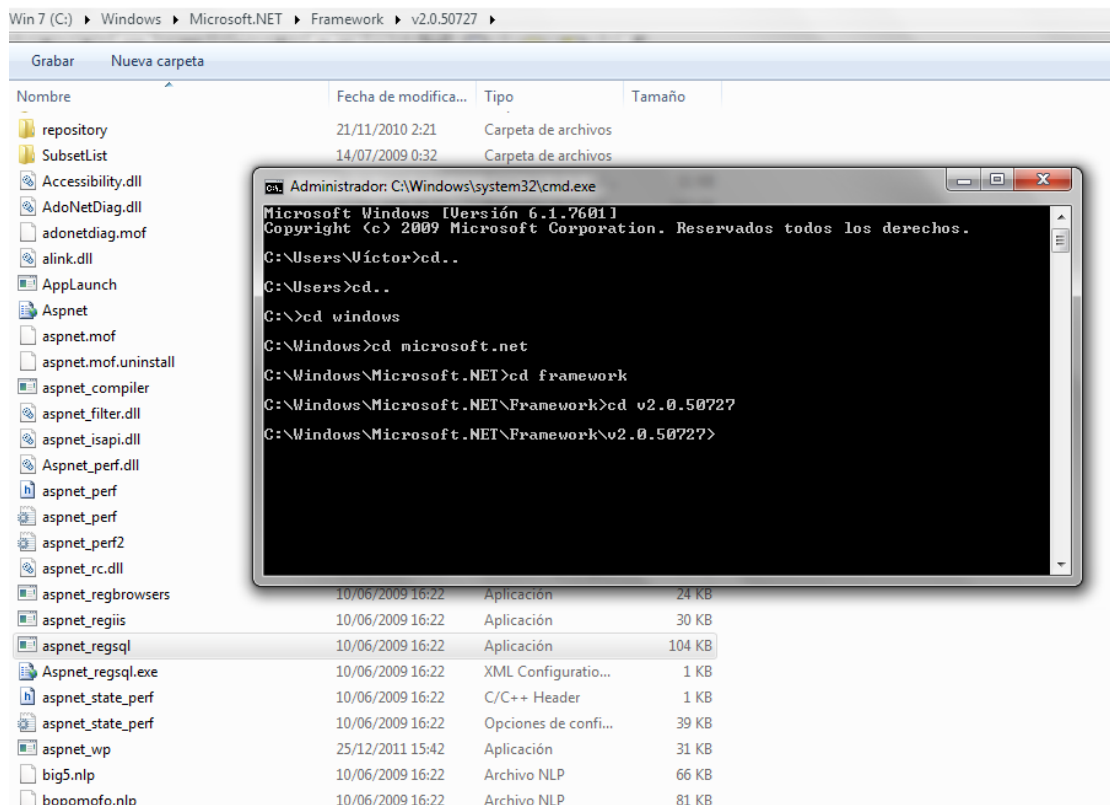


Figura A4.1 MembresShip 1

En un caso sencillo, solamente necesitamos habilitar el Membership provider y el Role provider, por lo que nuestra sintaxis sería:

```
aspnet_regsql -S servidor\sqlexpress -U usuario -P contraseña -A mr -d nombredelabasededatos
```

Donde:

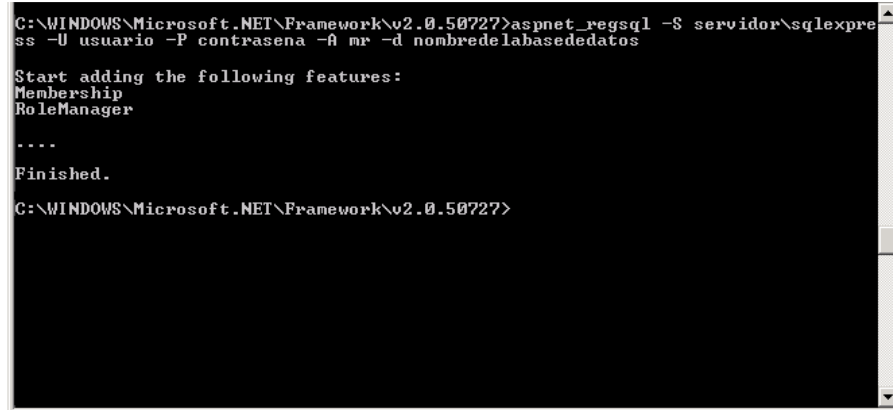
-S va seguido del nombre de tu servidor, ya sea en tu red local o en Internet (usa la IP)

-U va seguido de tu nombre de usuario

-P va seguido de tu contraseña

-A va seguido de "mr" ya que queremos habilitar el Membership y Role.

-d va seguido del nombre de tu base de datos



```
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727>aspnet_regsql -S servidor\sqlexpre
ss -U usuario -P contraseña -A mr -d nombredelabasededatos

Start adding the following features:
Membership
RoleManager
....
Finished.
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727>
```

Figura A4.2 MembresShip 2

Listo, ahora solamente verifica que se crearon las tablas correctamente:

De manera visual hacemos doble clic en el aspnet_regsql.exe de C:\Windows\Microsoft.NET\Framework\v2.0.50727 y se visualizara la siguiente ventana.

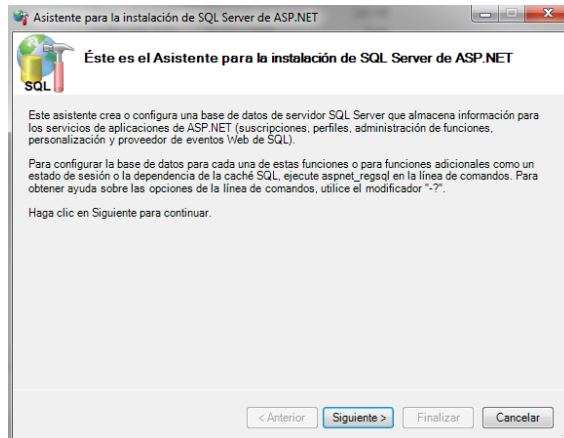


Figura A4.3 MemberShip 3

Presionamos el botón de siguiente.

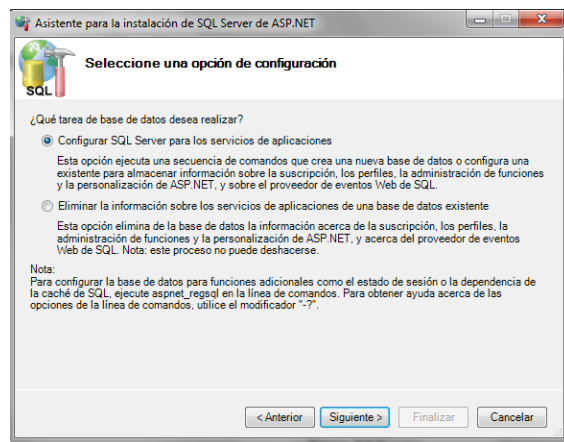


Figura A4.4 MemberShip 4

Seleccionamos el tipo de instalación que queremos realizar y clic en siguiente.

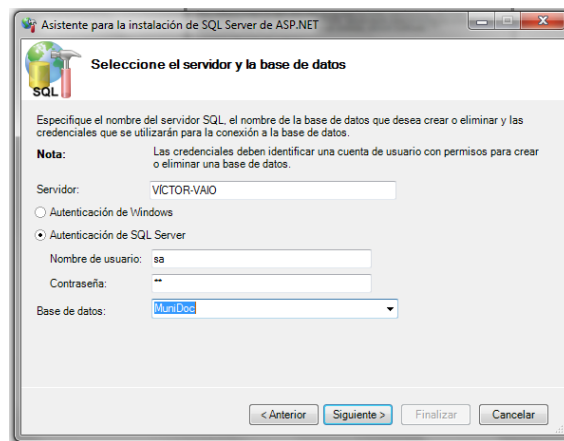


Figura A4.5 MemberShip 5

Seleccionamos el servidor y la base de datos y siguiente.

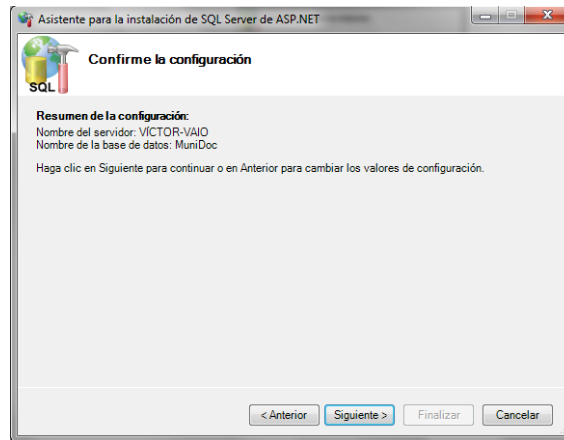


Figura A4.6 MemberShip 6

Nos pide confirmación para la creación y hacemos clic en siguiente.

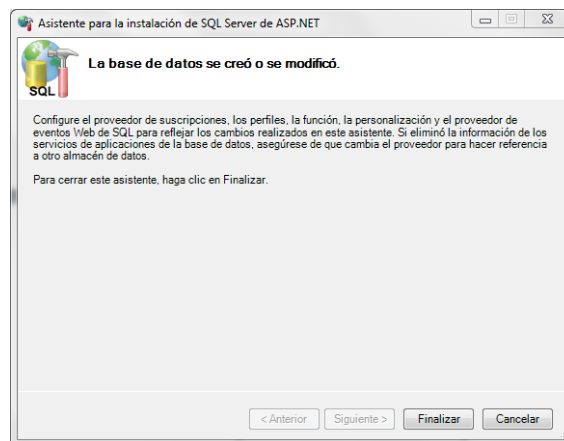


Figura A4.7 MemberShip 7

Se presenta la ventana de finalización que indica que se creó las tablas correctamente. Listo ahora solo verificamos que están creadas correctamente.

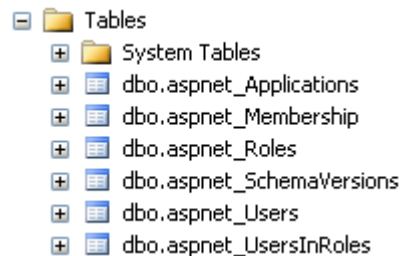


Figura A4.8 MemberShip Tablas

Para poder entender cómo se encuentra la relación entre las tablas creadas podemos ver el siguiente diagrama:

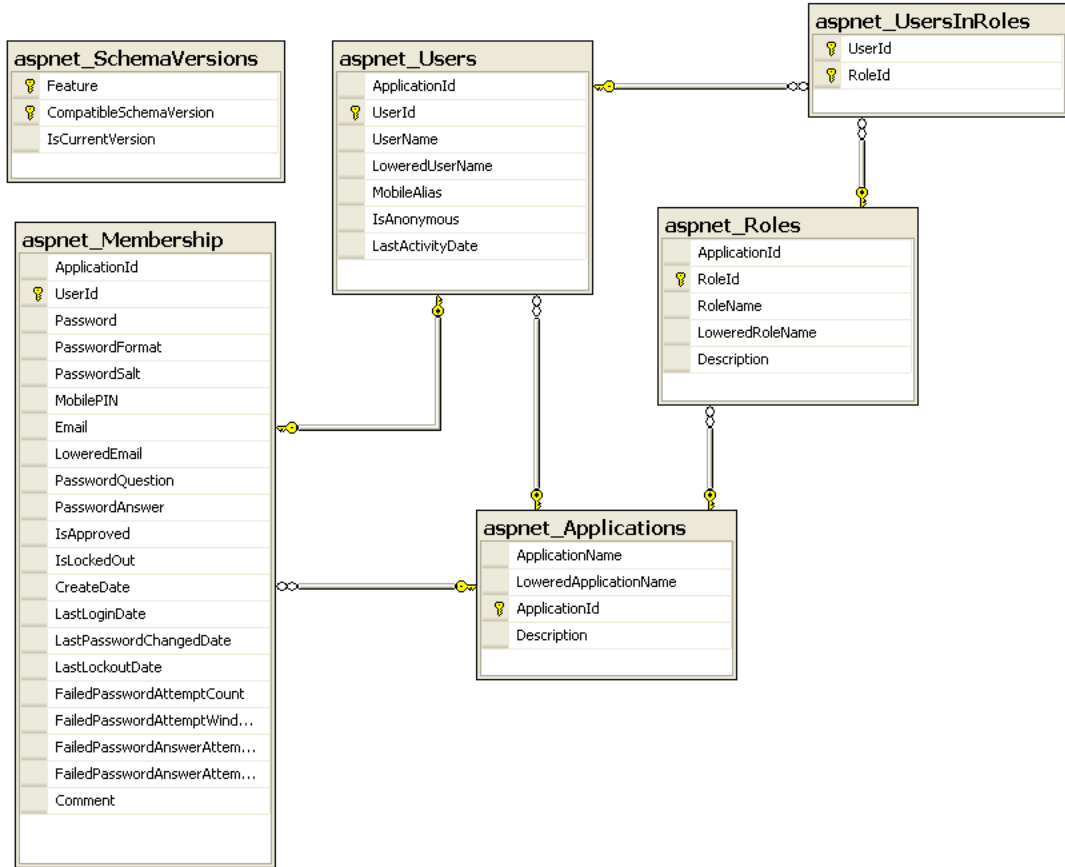


Figura A4.9 Membreship Diagrama

ANEXO 4: SECUENCIAS DE COMANDOS SQL

A continuación se cita cada uno de los comandos SQL necesarios para la creación de algunas tablas.

CREATE TABLE DEPARTAMENTOS

```
(
    DEP_CODIGO          NVARCHAR(10)      NOT NULL,
    DEP_NOMBRE          NVARCHAR(80),
    DEP_PADRE           NVARCHAR(10),
    DEP_TIPO            NVARCHAR(1),
    DEP_ACTIVO          NVARCHAR(1),
    DEP_CED_JEF         NVARCHAR(10)
);
```

ALTER TABLE DEPARTAMENTOS

```
ADD CONSTRAINT "DEPARTAMENTO_PK1" PRIMARY KEY ("DEP_CODIGO");
```

CREATE TABLE FUNCIONARIOS

```
(
    FUN_CODIGO          INTEGER           NOT NULL,
    FUN_APELLIDO        NVARCHAR(50),
    FUN_NOMBRE          NVARCHAR(50),
    FUN_DEPART          NVARCHAR(10),
    FUN_ACTIVO          NVARCHAR(1),
    FUN_COMPLETO        NVARCHAR(100),
    FUN_LOGIN           NVARCHAR(8)
);
```

ALTER TABLE FUNCIONARIOS

```
ADD CONSTRAINT "FUNCIONARIOS_PK1" PRIMARY KEY ("FUN_CODIGO");
```

CREATE TABLE DOCUMENTOS

```
(
    DOC_CODIGO          INTEGER           NOT NULL,
    DOC_DESCRIPCION     NVARCHAR(30)      NOT NULL
);
```

ALTER TABLE DOCUMENTOS

```
ADD CONSTRAINT "DOCUMENTOS_PK1" PRIMARY KEY ("DOC_CODIGO");
```

CREATE TABLE TRAMITES

```
(
    TRAM_CODIGO         INTEGER           NOT NULL,
    TRAM_NOMBRE         NVARCHAR(400),
    TRAM_COD_PADRE      INTEGER,

```

```

TRAM_ACTIVON      VARCHAR(1),
TRAM_FECHA_INI   DATETIME,
TRAM_ESTADO      NVARCHAR(1),
TRAM_FECHA_FIN   DATETIME,
DEP_CODIGO       NVARCHAR(10),
TRAM_TIEMPO      INTEGER,
TRAM_USUARIOCREA NVARCHAR(20),
TRAM_FECHACREA   DATETIME,
TRAM_USUARIOMODIF NVARCHAR(20),
TRAM_FECHAMODIF  DATETIME
);
ALTER TABLE TRAMITES
ADD CONSTRAINT "TRAMITES_PK1" PRIMARY KEY ("TRAM_CODIGO");

```

CREATE TABLE OFICIOS

```

(
    OFI_CODIGO      INTEGER          NOT NULL,
    OFI_FECHA_ING   DATETIME,
    OFI_MOTIVO      NVARCHAR(500),
    OFI_FECHA_SAL   DATETIME,
    OFI_SAL_OBSER   NVARCHAR(500),
    OFI_SALIDA      NVARCHAR(30),
    OFI_ESTADO      NVARCHAR(1),
    OFI_NUMERO      NVARCHAR(30),
    OFI_FECHA_ORIG  NVARCHAR(10),
    OFI_REFER       NVARCHAR(30),
    OFI_REF_SOL     NVARCHAR(30),
    OFI_UNIDAD_INI  NVARCHAR(10),
    OFI_USU_MOD     NVARCHAR(10),
    OFI_FECHA_MOD   DATETIME,
    OFI_RESPONDE    NVARCHAR(50),
    OFI_PADRE       NVARCHAR(12),
    DOC_CODIGO      INTEGER          NOT NULL,
    OFI_TRATAMIENTO NVARCHAR(200),
    OFI_SEGUIMIENTO NVARCHAR(1),
    OFI_USUARIO     NVARCHAR(20),
    OFI_TIPO        NVARCHAR(1),
    OFI_IMPRESION   NVARCHAR(1),
    OFI_MOVIN       NVARCHAR(1),
    OFI_REF_TRAM    NVARCHAR(20),
    FUN_CODIGO      INTEGER,
    TRA_CODIGO      INTEGER,
    OFI_REM_INST    NVARCHAR(80),
    OFI_REMITE      INTEGER
)

```

```

        OFI_ORIGINAL          INTEGER
        EXT_CODIGO           INTEGER
        OFI_DEP_ENVIO        NVARCHAR(20),
        OFI_DEP_TIPO         NVARCHAR(1),
        OFI_DEP_REMITE       NVARCHAR(30),
        OFI_UNIDAD_DES       NVARCHAR(30),
        OFI_TRAMITE          NVARCHAR(20),
        OFI_USUARIO_FIN      NVARCHAR(20),
        OFI_FECHA_FIN        DATETIME,
        OFI_FUN_JEFE         INTEGER,
        OFI_ID_TRAMITE       INTEGER,
        OFI_USUARIO_REC      NVARCHAR(20),
        OFI_FECHA_REC        DATETIME,
        OFI_DESPACHO         NVARCHAR(30),
        OFI_FDESPACHO        DATETIME,
        OFI_RETIRA           NVARCHAR(80),
        OFI_CRETIRA          NVARCHAR(10),
        OFI_REV_RESOL        NVARCHAR(1),
        OFI_FREV_RESOL       DATETIME,
        OFI_OREV_RESOL       NVARCHAR(80),
        OFI_ANIO             INTEGER
    );
    ALTER TABLE OFICIOS
        ADD CONSTRAINT "OFICIOS_PK1" PRIMARY KEY ("OFI_CODIGO");

```

CREATE TABLE ANEXOS

```

(
    ANE_CODIGO          INTEGER          NOT NULL,
    ANE_DESCRIPCION     NVARCHAR(200),
    ANE_OBSER           NVARCHAR(200),
    OFI_CODIGO          INTEGER,
    ANE_DOCUMENTO       NTEXT
);
    ALTER TABLE ANEXOS
        ADD CONSTRAINT "ANEXOS_PK1" PRIMARY KEY ("ANE_CODIGO");

```

CREATE TABLE ARCHIVO

```

(
    ARC_CODIGO          INTEGER          NOT NULL,
    ARC_TIPO            NVARCHAR(200),
    ARC_NOMBRE          NVARCHAR(200),
    ARC_ESTADO          NVARCHAR(1),
    ARC_PADRE           INTEGER,
    DEP_CODIGO          NVARCHAR(10)    NOT NULL,

```

```

        ARC_ANIO          INTEGER
    );
ALTER TABLE ARCHIVO
    ADD CONSTRAINT "ARCHIVO_PK1" PRIMARY KEY ("ARC_CODIGO");

```

CREATE TABLE RESPONSABLES

```

(
    RES_CODIGO          INTEGER          NOT NULL,
    RES_NOMBRE          NVARCHAR(100),
    RES_LOGIN           NVARCHAR(10),
    RES_ROL             NVARCHAR(20),
    DEP_CODIGO          NVARCHAR(10)     NOT NULL,
    RES_PERMUCA         NVARCHAR(1)
);
ALTER TABLE RESPONSABLES
    ADD CONSTRAINT "RESPONSABLES_PK1" PRIMARY KEY ("RES_CODIGO");

```

CREATE TABLE DEPARTAMENTO_OFICIOS

```

(
    OFI_CODIGO          INTEGER          NOT NULL,
    TIPO                NVARCHAR(1)
    DEP_CODIGO          NVARCHAR(10)     NOT NULL,
    FECHA_ENV           DATETIME,
    ENV_RECI            NVARCHAR(1),
    ENV_IMPRESION       NVARCHAR(1),
    FECHA_RECIBI        DATETIME,
    USUARIO_ENV         NVARCHAR(20),
    USUARIO_REC         NVARCHAR(20),
    ESTADO_TRAM        NVARCHAR(1),
    USUARIO_FINALIZ    NVARCHAR(20),
    FECHA_FINALIZ      DATETIME
);

```

CREATE TABLE DOC_TRAMITES

```

(
    OFI_CODIGO          INTEGER          NOT NULL,
    TRA_CODIGO          INTEGER          NOT NULL,
    DOC_TRA_CUMPLE     NVARCHAR,
    DOC_TRA_MOTIVO     NVARCHAR
);

```

CREATE TABLE ARCHIVO_OFICIOS

```

(
    ARC_CODIGO          INTEGER          NOT NULL,

```



```
    OFI_CODIGO          INTEGER          NOT NULL,  
    ARC_OFI_DETALLE     NVARCHAR(500),  
    DEP_CODIGO          NVARCHAR(10)      NOT NULL,  
    ARC_OFI_UCA         NVARCHAR(1)  
);
```

La creación de Relaciones entre las tablas se lo realizo de manara manual en el diagrama del Management.

ANEXO 5: EXPLICACIÓN, ESTRUCTURA Y EJEMPLOS DE CÓDIGO

FUENTE DE LA APLICACIÓN

El sistema esta creado en dos capas: presenta varios módulos dentro de su desarrollo por lo tanto a continuación se presentará la codificación de algunos de los más importantes.

1. Cliente

El siguiente código fuente representa una de las clases llamado CIOficios, en donde se muestra los datos, propiedades y métodos necesarios.

```
Clase Oficios
using System;
using System.Collections.Generic;
using System.Text;
using System.Linq;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace DTVBServidor
{
    /// <summary>
    /// Description of CIOficios.
    /// </summary>
    public class CIOficios
    {
        #region Datos

        private int _ofi_codigo;
        private DateTime _ofi_fecha_ing;
        private string _ofi_motivo;
        private DateTime _ofi_fecha_sal;
        private string _ofi_sal_obser;
        private string _ofi_salida;
        private string _ofi_estado;
        private string _ofi_numero;
        private string _ofi_fecha_orig;
        private string _ofi_refer;
        private string _ofi_ref_sol;
        private string _ofi_unidad_ini;
        private string _ofi_usu_mod;
        private DateTime _ofi_fecha_mod;
        private string _ofi_responde;
        private int _ofi_padre;
        private int _doc_codigo;
        private string _ofi_tratamiento;
        private string _ofi_seguimiento;
        private string _ofi_usuario;
        private string _ofi_tipo;
        private string _ofi_impresion;
        private string _ofi_movin;
        private string _ofi_ref_tram;
```

```

private string _fun_codigo;
private int _tra_codigo;
private string _ofi_rem_inst;
private int _ofi_remite;
private int _ofi_original;
private int _ext_codigo;
private string _ofi_dep_envio;
private string _ofi_dep_tipo;
private string _ofi_dep_remite;
private string _ofi_unidad_des;
private string _ofi_tramite;
private string _ofi_usuario_fin;
private DateTime _ofi_fecha_fin;
private int _ofi_fun_jefe;
private int _ofi_id_tramite;
private string _ofi_usuario_rec;
private DateTime _ofi_fecha_rec;
private string _ofi_despacho;
private DateTime _ofi_fdespacho;
private string _ofi_retira;
private string _ofi_cretira;
private string _ofi_rev_resol;
private DateTime _ofi_frev_resol;
private string _ofi_orev_resol;
private int _ofi_anio;
private int _flow_anio;
private int _flow_ofi_codigo;

```

```
#endregion;
```

```
#region Propiedades
```

```

public int OfiCodigo
{
    get { return _ofi_codigo; }
    set { _ofi_codigo = value; }
}

public DateTime OfiFechaIng
{
    get { return _ofi_fecha_ing; }
    set { _ofi_fecha_ing = value; }
}

public string OfiMotivo
{
    get { return _ofi_motivo; }
    set { _ofi_motivo = value; }
}

public DateTime OfiFechaSal
{
    get { return _ofi_fecha_sal; }
    set { _ofi_fecha_sal = value; }
}

public string OfiSalObser
{
    get { return _ofi_sal_obser; }
    set { _ofi_sal_obser = value; }
}

public string OfiSalida
{
    get { return _ofi_salida; }
    set { _ofi_salida = value; }
}

public string OfiEstado

```

```

{
    get { return _ofi_estado; }
    set { _ofi_estado = value; }
}
public string OfiNumero
{
    get { return _ofi_numero; }
    set { _ofi_numero = value; }
}
public string OfiFechaOrig
{
    get { return _ofi_fecha_orig; }
    set { _ofi_fecha_orig = value; }
}
public string OfiRefer
{
    get { return _ofi_refer; }
    set { _ofi_refer = value; }
}
public string OfiRefSol
{
    get { return _ofi_ref_sol; }
    set { _ofi_ref_sol = value; }
}
public string OfiUnidadIni
{
    get { return _ofi_unidad_ini; }
    set { _ofi_unidad_ini = value; }
}
public string OfiUsuMod
{
    get { return _ofi_usu_mod; }
    set { _ofi_usu_mod = value; }
}
public DateTime OfiFechaMod
{
    get { return _ofi_fecha_mod; }
    set { _ofi_fecha_mod = value; }
}
public string OfiResponde
{
    get { return _ofi_responde; }
    set { _ofi_responde = value; }
}
public int OfiPadre
{
    get { return _ofi_padre; }
    set { _ofi_padre = value; }
}
public int DocCodigo
{
    get { return _doc_codigo; }
    set { _doc_codigo = value; }
}
public string OfiTratamiento
{
    get { return _ofi_tratamiento; }
    set { _ofi_tratamiento = value; }
}
public string OfiSeguimiento
{
    get { return _ofi_seguimiento; }
    set { _ofi_seguimiento = value; }
}

```

```

}
public string OfiUsuario
{
    get { return _ofi_usuario; }
    set { _ofi_usuario = value; }
}
public string OfiTipo
{
    get { return _ofi_tipo; }
    set { _ofi_tipo = value; }
}
public string OfiImpresion
{
    get { return _ofi_impresion; }
    set { _ofi_impresion = value; }
}
public string OfiMovin
{
    get { return _ofi_movin; }
    set { _ofi_movin = value; }
}
public string OfiRefTram
{
    get { return _ofi_ref_tram; }
    set { _ofi_ref_tram = value; }
}
public string FunCodigo
{
    get { return _fun_codigo; }
    set { _fun_codigo = value; }
}
public int TraCodigo
{
    get { return _tra_codigo; }
    set { _tra_codigo = value; }
}
public string OfiRemInst
{
    get { return _ofi_rem_inst; }
    set { _ofi_rem_inst = value; }
}
public int OfiRemite
{
    get { return _ofi_remite; }
    set { _ofi_remite = value; }
}
public int OfiOriginal
{
    get { return _ofi_original; }
    set { _ofi_original = value; }
}
public int ExtCodigo
{
    get { return _ext_codigo; }
    set { _ext_codigo = value; }
}
public string OfiDepEnvio
{
    get { return _ofi_dep_envio; }
    set { _ofi_dep_envio = value; }
}
public string OfiDepTipo
{

```

```

    get { return _ofi_dep_tipo; }
    set { _ofi_dep_tipo = value; }
}
public string OfiDepRemite
{
    get { return _ofi_dep_remite; }
    set { _ofi_dep_remite = value; }
}
public string OfiUnidadDes
{
    get { return _ofi_unidad_des; }
    set { _ofi_unidad_des = value; }
}
public string OfiTramite
{
    get { return _ofi_tramite; }
    set { _ofi_tramite = value; }
}
public string OfiUsuarioFin
{
    get { return _ofi_usuario_fin; }
    set { _ofi_usuario_fin = value; }
}
public DateTime OfiFechaFin
{
    get { return _ofi_fecha_fin; }
    set { _ofi_fecha_fin = value; }
}
public int OfiFunJefe
{
    get { return _ofi_fun_jefe; }
    set { _ofi_fun_jefe = value; }
}
public int OfidTramite
{
    get { return _ofi_id_tramite; }
    set { _ofi_id_tramite = value; }
}
public string OfiUsuarioRec
{
    get { return _ofi_usuario_rec; }
    set { _ofi_usuario_rec = value; }
}
public DateTime OfiFechaRec
{
    get { return _ofi_fecha_rec; }
    set { _ofi_fecha_rec = value; }
}
public string OfiDespacho
{
    get { return _ofi_despacho; }
    set { _ofi_despacho = value; }
}
public DateTime OfiFDespacho
{
    get { return _ofi_fdespacho; }
    set { _ofi_fdespacho = value; }
}
public string OfiRetira
{
    get { return _ofi_retira; }
    set { _ofi_retira = value; }
}
}

```

```

public string OfiCretira
{
    get { return _ofi_cretira; }
    set { _ofi_cretira = value; }
}
public string OfiRevResol
{
    get { return _ofi_rev_resol; }
    set { _ofi_rev_resol = value; }
}
public DateTime OfiFRevResol
{
    get { return _ofi_frev_resol; }
    set { _ofi_frev_resol = value; }
}
public string OfiORevResol
{
    get { return _ofi_orev_resol; }
    set { _ofi_orev_resol = value; }
}
public int OfiAnio
{
    get { return _ofi_anio; }
    set { _ofi_anio = value; }
}
public int FlowAnio
{
    get { return _flow_anio; }
    set { _flow_anio = value; }
}
public int FlowOfiCodigo
{
    get { return _flow_ofi_codigo; }
    set { _flow_ofi_codigo = value; }
}
#endregion

#region Metodos

public DataSet Seleccionar()
{
    SqlConnection cnConexion = new SqlConnection();
    try
    {
        cnConexion.ConnectionString = ConectarBD.CadenaDeConexion;

        string comandoSelect = "SELECT OFI_CODIGO, OFI_FECHA_ING, OFI_MOTIVO, OFI_FECHA_SAL, OFI_SAL_OBSER,
OFI_SALIDA, OFI_ESTADO, OFI_NUMERO, OFI_FECHA_ORIG, OFI_REFER, OFI_REF_SOL, OFI_UNIDAD_INI, OFI_USU_MOD,
OFI_FECHA_MOD, OFI_RESPONDE, OFI_PADRE, DOC_CODIGO, OFI_TRATAMIENTO, OFI_SEGUIMIENTO, OFI_USUARIO,
OFI_TIPO, OFI_IMPRESION, OFI_MOVIN, OFI_REF_TRAM, FUN_CODIGO, TRA_CODIGO, OFI_REM_INST, OFI_REMITE,
OFI_ORIGINAL, EXT_CODIGO, OFI_DEP_ENVIO, OFI_DEP_TIPO, OFI_DEP_REMITE, OFI_UNIDAD_DES, OFI_TRAMITE,
OFI_USUARIO_FIN, OFI_FECHA_FIN, OFI_FUN_JEFE, OFI_ID_TRAMITE, OFI_USUARIO_REC, OFI_FECHA_REC, OFI_DESPACHO,
OFI_FDESPACHO, OFI_RETIRA, OFI_CRETIRA, OFI_REV_RESOL, OFI_FREV_RESOL, OFI_OREV_RESOL, OFI_ANIO " +
"FROM OFICIOS";

        SqlDataAdapter daDepartamentos = new SqlDataAdapter(comandoSelect, cnConexion);
        DataSet dsTablasYDatos = new DataSet();
        daDepartamentos.Fill(dsTablasYDatos, "OFICIOS");
        return dsTablasYDatos;
    }
    catch (SqlException error)
    {
        throw error;
    }
}

```

```

    }
    catch (Exception error)
    {
        throw error;
    }
    finally
    {
        cnConexion.Dispose();
    }
}

public DataSet SeleccionarEstado()
{
    SqlConnection cnConexion = new SqlConnection();
    try
    {
        cnConexion.ConnectionString = ConectarBD.CadenaDeConexion;

        string comandoSelect = "SELECT EST_CODIGO, EST_OFICIO, EXT_DESCRIPCION " +
            "FROM ESTADOFICIO";

        SqlDataAdapter daDepartamentos = new SqlDataAdapter(comandoSelect, cnConexion);
        DataSet dsTablasYDatos = new DataSet();
        daDepartamentos.Fill(dsTablasYDatos, "ESTADOFICIO");
        return dsTablasYDatos;
    }
    catch (SqlException error)
    {
        throw error;
    }
    catch (Exception error)
    {
        throw error;
    }
    finally
    {
        cnConexion.Dispose();//Destructor.
    }
}

public DataSet SeleccionarCodigoMax()
{
    SqlConnection cnConexion = new SqlConnection();
    try
    {
        cnConexion.ConnectionString = ConectarBD.CadenaDeConexion;

        string sentenciaSelect = "SELECT max(TRAM_CODIGO) " +
            "FROM TRAMITES ";

        SqlDataAdapter daDepartamentos = new SqlDataAdapter(sentenciaSelect, cnConexion);
        DataSet dsTablasYDatos = new DataSet();
        daDepartamentos.Fill(dsTablasYDatos, "TRAMITES");
        MessageBox.Show(sentenciaSelect.ToString());
        MessageBox.Show(daDepartamentos.ToString());
        return dsTablasYDatos;
    }
    catch (SqlException error)
    {
        throw error;
    }
    catch (Exception error)
    {

```



```

        throw error;
    }
    finally
    {
        cnConexion.Dispose();
    }
}

public DataSet SeleccionarFlow()
{
    SqlConnection cnConexion = new SqlConnection();
    try
    {
        cnConexion.ConnectionString = ConectarBD.CadenaDeConexion;

        string comandoSelect = "SELECT FLOW_CODIGO, FLOW_ANIO " +
            "FROM FLOW";

        SqlDataAdapter daDepartamentos = new SqlDataAdapter(comandoSelect, cnConexion);
        DataSet dsTablasYDatos = new DataSet();
        daDepartamentos.Fill(dsTablasYDatos, "FLOW");
        return dsTablasYDatos;
    }
    catch (SqlException error)
    {
        throw error;
    }
    catch (Exception error)
    {
        throw error;
    }
    finally
    {
        cnConexion.Dispose();//Destructor.
    }
}

public DataTable SeleccionarOficioPorDepartamento(string nombreDepartamento)
{
    SqlConnection cnBaseDatos2 = new SqlConnection();

    try
    {
        cnBaseDatos2.ConnectionString = ConectarBD.CadenaDeConexion;

        string sentenciaSelect = "SELECT OFI_CODIGO, OFI_FECHA_ING, OFI_MOTIVO, OFI_FECHA_SAL, OFI_SAL_OBSER,
OFI_SALIDA, OFI_ESTADO, OFI_NUMERO " +
            "FROM OFICIOS" +
            "WHERE OFI_CODIGO=@nombreDepartamento ";

        SqlCommand cmdSelect = new SqlCommand(sentenciaSelect, cnBaseDatos2);
        cmdSelect.CommandType = CommandType.Text;
        cmdSelect.Parameters.Add("@nombreDepartamento", SqlDbType.NVarChar, 256).Value = nombreDepartamento;
        SqlDataAdapter daProductos = new SqlDataAdapter();
        daProductos.SelectCommand = cmdSelect;
        DataTable dtProductos = new DataTable();
        daProductos.Fill(dtProductos);
        return dtProductos;
    }
    catch (SqlException error)
    {
        throw error;
    }
}

```

```

catch (Exception error)
{
    throw error;
}
}

public DataTable SeleccionarPorOfiPadre(string ofiPadre)
{
    SqlConnection cnBaseDatos2 = new SqlConnection();

    try
    {
        cnBaseDatos2.ConnectionString = ConectarBD.CadenaDeConexion;

        string sentenciaSelect = "SELECT D.\"DOC_DESCRIPCION\", O.\"OFI_NUMERO\", O.\"OFI_FECHA_ING\",
O.\"OFI_MOTIVO\", F.\"FUN_COMPLETO\", O.\"OFI_SALIDA\", O.\"OFI_SAL_OBSER\", O.\"OFI_FECHA_SAL\",
E.\"EXT_DESCRIPCION\", O.\"OFI_CODIGO\" " +
        "FROM \"OFICIOS\" O, \"DOCUMENTOS\" D, \"FUNCIONARIOS\" F, \"ESTADOFICIO\" E " +
        "WHERE O.\"OFI_PADRE\"=@ofiPadre and D.\"DOC_CODIGO\" = O.\"DOC_CODIGO\" and
F.\"FUN_CODIGO\" = O.\"FUN_CODIGO\" and E.\"EST_OFICIO\" = O.\"OFI_ESTADO\" ";

        SqlCommand cmdSelect = new SqlCommand(sentenciaSelect, cnBaseDatos2);
        cmdSelect.CommandType = CommandType.Text;
        cmdSelect.Parameters.Add("@ofiPadre", SqlDbType.Int).Value = ofiPadre;
        SqlDataAdapter daProductos = new SqlDataAdapter();
        daProductos.SelectCommand = cmdSelect;
        DataTable dtProductos = new DataTable();
        daProductos.Fill(dtProductos);
        return dtProductos;
    }
    catch (SqlException error)
    {
        throw error;
    }
    catch (Exception error)
    {
        throw error;
    }
}

public DataSet SeleccionarFlow1()
{
    SqlConnection cnConexion = new SqlConnection();

    try
    {
        cnConexion.ConnectionString = ConectarBD.CadenaDeConexion;

        string comandoSelect = "SELECT FLOW_CODIGO, FLOW_OFI_CODIGO " +
        "FROM FLOW ORDER BY FLOW_OFI_CODIGO";

        SqlDataAdapter daDocumentos = new SqlDataAdapter(comandoSelect, cnConexion);
        DataSet dsTablasYDatos = new DataSet();
        daDocumentos.Fill(dsTablasYDatos, "FLOW");
        return dsTablasYDatos;
    }
    catch (SqlException error)
    {
        throw error;
    }
    catch (Exception error)
    {
        throw error;
    }
}

```

```

    }
    finally
    {
        cnConexion.Dispose();
    }
}

public DataTable SeleccionarPorFW(string ofIDTramite)
{
    SqlConnection cnBaseDatos2 = new SqlConnection();

    try
    {
        cnBaseDatos2.ConnectionString = ConectarBD.CadenaDeConexion;

        string sentenciaSelect = "SELECT O.\"OFI_CODIGO\", O.\"OFI_ANIO\", O.\"OFI_MOTIVO\", O.\"TRA_CODIGO\",
O.\"DOC_CODIGO\", O.\"OFI_FECHA_ING\", O.\"OFI_ESTADO\", O.\"OFI_NUMERO\", D.\"DEP_NOMBRE\",
O.\"EXT_CODIGO\", O.\"OFI_REFER\", O.\"OFI_REF_SOL\", O.\"OFI_FECHA_ORIG\", O.\"OFI_TRATAMIENTO\" " +
        "FROM \"OFICIOS\" O, \"DEPARTAMENTOS\" D " +
        "WHERE O.\"OFI_ID_TRAMITE\"=@ofIDTramite AND O.\"OFI_UNIDAD_INI\" = D.\"DEP_CODIGO\" ";

        SqlCommand cmdSelect = new SqlCommand(sentenciaSelect, cnBaseDatos2);
        cmdSelect.CommandType = CommandType.Text;
        cmdSelect.Parameters.Add("@ofIDTramite", SqlDbType.Int).Value = ofIDTramite;
        SqlDataAdapter daProductos = new SqlDataAdapter();
        daProductos.SelectCommand = cmdSelect;
        DataTable dtProductos = new DataTable();
        daProductos.Fill(dtProductos);
        return dtProductos;
    }
    catch (SqlException error)
    {
        throw error;
    }
    catch (Exception error)
    {
        throw error;
    }
}

public DataTable SeleccionarPorFWExterno(string ofIDTramite)
{
    SqlConnection cnBaseDatos2 = new SqlConnection();

    try
    {
        cnBaseDatos2.ConnectionString = ConectarBD.CadenaDeConexion;

        string sentenciaSelect = "SELECT O.\"OFI_CODIGO\", O.\"OFI_ANIO\", O.\"OFI_MOTIVO\", O.\"TRA_CODIGO\",
O.\"DOC_CODIGO\", O.\"OFI_FECHA_ING\", O.\"OFI_ESTADO\", O.\"OFI_NUMERO\", D.\"DEP_NOMBRE\",
O.\"EXT_CODIGO\", E.\"EXT_APELLIDO\", E.\"EXT_NOMBRE\", O.\"OFI_REFER\", O.\"OFI_REF_SOL\", O.\"OFI_FECHA_ORIG\",
O.\"OFI_TRATAMIENTO\" " +
        "FROM \"OFICIOS\" O, \"DEPARTAMENTOS\" D, \"EXTERNOS\" E " +
        "WHERE O.\"OFI_ID_TRAMITE\"=@ofIDTramite AND O.\"OFI_UNIDAD_INI\" = D.\"DEP_CODIGO\" and
E.\"EXT_CODIGO\" = O.\"EXT_CODIGO\" ";

        SqlCommand cmdSelect = new SqlCommand(sentenciaSelect, cnBaseDatos2);
        cmdSelect.CommandType = CommandType.Text;
        cmdSelect.Parameters.Add("@ofIDTramite", SqlDbType.Int).Value = ofIDTramite;
        SqlDataAdapter daProductos = new SqlDataAdapter();
        daProductos.SelectCommand = cmdSelect;
        DataTable dtProductos = new DataTable();
        daProductos.Fill(dtProductos);
    }
}

```

```

        return dtProductos;
    }
    catch (SQLException error)
    {
        throw error;
    }
    catch (Exception error)
    {
        throw error;
    }
}

public DataTable SeleccionarPorCodigoTarea(string ofiCodigo)
{
    SqlConnection cnBaseDatos2 = new SqlConnection();

    try
    {
        cnBaseDatos2.ConnectionString = ConectarBD.CadenaDeConexion;

        string sentenciaSelect = "SELECT O.\"OFI_CODIGO\", DO.\"DOC_DESCRIPCION\", O.\"OFI_NUMERO\",
O.\"OFI_FECHA_ING\", O.\"OFI_MOTIVO\", F.\"FUN_COMPLETO\", O.\"OFI_SALIDA\", O.\"OFI_SAL_OBSER\",
O.\"OFI_FECHA_SAL\", EO.\"EXT_DESCRIPCION\" " +
        "FROM \"OFICIOS\" O, \"DOCUMENTOS\" DO, \"FUNCIONARIOS\" F, \"ESTADOFICIO\" EO " +
        "WHERE O.\"OFI_CODIGO\"=@ofiCodigo AND O.\"DOC_CODIGO\" = DO.\"DOC_CODIGO\" AND
F.\"FUN_CODIGO\" = O.\"FUN_CODIGO\" AND EO.\"EST_OFICIO\" = O.\"OFI_ESTADO\" ";

        SqlCommand cmdSelect = new SqlCommand(sentenciaSelect, cnBaseDatos2);
        cmdSelect.CommandType = CommandType.Text;
        cmdSelect.Parameters.Add("@ofiCodigo", SqlDbType.Int).Value = ofiCodigo;
        SqlDataAdapter daProductos = new SqlDataAdapter();
        daProductos.SelectCommand = cmdSelect;
        DataTable dtProductos = new DataTable();
        daProductos.Fill(dtProductos);
        return dtProductos;
    }
    catch (SQLException error)
    {
        throw error;
    }
    catch (Exception error)
    {
        throw error;
    }
}

public DataTable SeleccionarPorOficioAnio(string ofiCodigo, string ofiAnio)
{
    SqlConnection cnBaseDatos2 = new SqlConnection();

    try
    {
        cnBaseDatos2.ConnectionString = ConectarBD.CadenaDeConexion;

        string sentenciaSelect = "SELECT O.\"OFI_CODIGO\", O.\"OFI_ID_TRAMITE\", O.\"OFI_ANIO\",
D.\"DOC_DESCRIPCION\", O.\"OFI_FECHA_ING\", O.\"OFI_ESTADO\", O.\"OFI_NUMERO\", O.\"OFI_TIPO\",
E.\"EXT_CODIGO\", E.\"EXT_APELLIDO\", E.\"EXT_NOMBRE\", E.\"EXT_DIRECCION\", O.\"OFI_REM_INST\", O.\"OFI_REFER\",
O.\"OFI_REF_SOL\", O.\"OFI_FECHA_ORIG\", O.\"OFI_TRATAMIENTO\", O.\"OFI_MOTIVO\", T.\"TRAM_NOMBRE\",
O.\"OFI_ORIGINAL\" " +
        "FROM \"OFICIOS\" O, \"DOCUMENTOS\" D, \"EXTERNOS\" E, \"TRAMITES\" T " +
        "WHERE O.\"OFI_ID_TRAMITE\"=@ofiCodigo and O.\"OFI_ANIO\"=@ofiAnio and O.\"DOC_CODIGO\" =
D.\"DOC_CODIGO\" and O.\"EXT_CODIGO\" = E.\"EXT_CODIGO\" and O.\"TRA_CODIGO\" = T.\"TRAM_CODIGO\" ";

```

```

SqlCommand cmdSelect = new SqlCommand(sentenciaSelect, cnBaseDatos2);
cmdSelect.CommandType = CommandType.Text;
cmdSelect.Parameters.Add("@ofCodigo", SqlDbType.Int).Value = ofCodigo;
cmdSelect.Parameters.Add("@ofAnio", SqlDbType.Int).Value = ofAnio;
SqlDataAdapter daProductos = new SqlDataAdapter();
daProductos.SelectCommand = cmdSelect;
DataTable dtProductos = new DataTable();
daProductos.Fill(dtProductos);
return dtProductos;
}
catch (SqlException error)
{
    throw error;
}
catch (Exception error)
{
    throw error;
}
}

public DataTable SeleccionarPorOficioAnio1(string ofCodigo, string ofAnio)
{
    SqlConnection cnBaseDatos2 = new SqlConnection();

    try
    {
        cnBaseDatos2.ConnectionString = ConectarBD.CadenaDeConexion;

        string sentenciaSelect = "SELECT O.\"OFI_CODIGO\", O.\"OFI_ID_TRAMITE\", O.\"OFI_ANIO\",
D.\"DOC_DESCRIPCION\", O.\"OFI_FECHA_ING\", O.\"OFI_ESTADO\", O.\"OFI_NUMERO\", O.\"OFI_TIPO\",
O.\"OFI_REM_INST\", O.\"OFI_REFER\", O.\"OFI_REF_SOL\", O.\"OFI_FECHA_ORIG\", O.\"OFI_TRATAMIENTO\",
O.\"OFI_MOTIVO\", T.\"TRAM_NOMBRE\", O.\"OFI_ORIGINAL\" " +
        "FROM \"OFICIOS\" O, \"DOCUMENTOS\" D, \"TRAMITES\" T " +
        "WHERE O.\"OFI_ID_TRAMITE\"=@ofCodigo and O.\"OFI_ANIO\"=@ofAnio and O.\"DOC_CODIGO\" =
D.\"DOC_CODIGO\" and O.\"TRA_CODIGO\" = T.\"TRAM_CODIGO\" ";

        SqlCommand cmdSelect = new SqlCommand(sentenciaSelect, cnBaseDatos2);
cmdSelect.CommandType = CommandType.Text;
cmdSelect.Parameters.Add("@ofCodigo", SqlDbType.Int).Value = ofCodigo;
cmdSelect.Parameters.Add("@ofAnio", SqlDbType.Int).Value = ofAnio;
SqlDataAdapter daProductos = new SqlDataAdapter();
daProductos.SelectCommand = cmdSelect;
DataTable dtProductos = new DataTable();
daProductos.Fill(dtProductos);
return dtProductos;
}
catch (SqlException error)
{
    throw error;
}
catch (Exception error)
{
    throw error;
}
}

public DataTable SeleccionarPorCodigo(string ofCodigo)
{
    SqlConnection cnBaseDatos2 = new SqlConnection();

    try
    {
        cnBaseDatos2.ConnectionString = ConectarBD.CadenaDeConexion;

```

```

        string sentenciaSelect = "SELECT O.\"OFI_ID_TRAMITE\", O.\"OFI_ANIO\", D.\"DOC_DESCRIPCION\",
O.\"OFI_FECHA_ING\", O.\"OFI_ESTADO\", O.\"OFI_NUMERO\", O.\"OFI_TIPO\", O.\"OFI_REM_INST\", O.\"OFI_REFER\",
O.\"OFI_REF_SOL\", O.\"OFI_FECHA_ORIG\", O.\"OFI_TRATAMIENTO\", O.\"OFI_MOTIVO\", T.\"TRAM_NOMBRE\",
F.\"FUN_COMPLETO\", DE.\"DEP_NOMBRE\", O.\"OFI_ORIGINAL\", O.\"OFI_SAL_OBSER\", DO.\"USUARIO_ENV\",
DO.\"USUARIO_REC\", DO.\"FECHA_REC\", O.\"OFI_ORIGINAL\" " +
        "FROM \"OFICIOS\" O, \"DOCUMENTOS\" D, \"TRAMITES\" T, \"FUNCIONARIOS\" F,
\"DEPARTAMENTOS\" DE, \"DEPARTAMENTO_OFICIOS\" DO " +
        "WHERE O.\"OFI_CODIGO\"=@ofiCodigo and O.\"DOC_CODIGO\" = D.\"DOC_CODIGO\" and
O.\"TRA_CODIGO\" = T.\"TRAM_CODIGO\" and F.\"FUN_LOGIN\" = O.\"OFI_USUARIO\" and DE.\"DEP_CODIGO\" =
O.\"OFI_UNIDAD_INI\" and O.\"OFI_CODIGO\" = DO.\"OFI_CODIGO\" ";

```

```

        SqlCommand cmdSelect = new SqlCommand(sentenciaSelect, cnBaseDatos2);
        cmdSelect.CommandType = CommandType.Text;
        cmdSelect.Parameters.Add("@ofiCodigo", SqlDbType.Int).Value = ofiCodigo;
        SqlDataAdapter daProductos = new SqlDataAdapter();
        daProductos.SelectCommand = cmdSelect;
        DataTable dtProductos = new DataTable();
        daProductos.Fill(dtProductos);
        return dtProductos;
    }
    catch (SqlException error)
    {
        throw error;
    }
    catch (Exception error)
    {
        throw error;
    }
}

```

```

public DataTable SeleccionarPorExterno(string ofiCodigo)

```

```

{
    SqlConnection cnBaseDatos2 = new SqlConnection();

    try
    {
        cnBaseDatos2.ConnectionString = ConectarBD.CadenaDeConexion;

        string sentenciaSelect = "SELECT O.\"OFI_CODIGO\", O.\"EXT_CODIGO\", O.\"OFI_ANIO\" " +
            "FROM \"OFICIOS\" O " +
            "WHERE O.\"OFI_ID_TRAMITE\"=@ofiCodigo ";
    }
}

```

```

        SqlCommand cmdSelect = new SqlCommand(sentenciaSelect, cnBaseDatos2);
        cmdSelect.CommandType = CommandType.Text;
        cmdSelect.Parameters.Add("@ofiCodigo", SqlDbType.Int).Value = ofiCodigo;
        SqlDataAdapter daProductos = new SqlDataAdapter();
        daProductos.SelectCommand = cmdSelect;
        DataTable dtProductos = new DataTable();
        daProductos.Fill(dtProductos);
        return dtProductos;
    }
    catch (SqlException error)
    {
        throw error;
    }
    catch (Exception error)
    {
        throw error;
    }
}

```

```

public DataTable SeleccionarPorExterno1(string ofiCodigo)

```

```

{
    SqlConnection cnBaseDatos2 = new SqlConnection();

    try
    {
        cnBaseDatos2.ConnectionString = ConectarBD.CadenaDeConexion;

        string sentenciaSelect = "SELECT O.\"OFI_CODIGO\", O.\"EXT_CODIGO\" " +
            "FROM \"OFICIOS\" O " +
            "WHERE O.\"OFI_CODIGO\"=@ofiCodigo ";

        SqlCommand cmdSelect = new SqlCommand(sentenciaSelect, cnBaseDatos2);
        cmdSelect.CommandType = CommandType.Text;
        cmdSelect.Parameters.Add("@ofiCodigo", SqlDbType.Int).Value = ofiCodigo;
        SqlDataAdapter daProductos = new SqlDataAdapter();
        daProductos.SelectCommand = cmdSelect;
        DataTable dtProductos = new DataTable();
        daProductos.Fill(dtProductos);
        return dtProductos;
    }
    catch (SqlException error)
    {
        throw error;
    }
    catch (Exception error)
    {
        throw error;
    }
}

public DataTable SeleccionarPorUsuarioLogueado(string ofiCodigo)
{
    SqlConnection cnBaseDatos2 = new SqlConnection();
    try
    {
        cnBaseDatos2.ConnectionString = ConectarBD.CadenaDeConexion;
        string sentenciaSelect = "SELECT O.\"OFI_CODIGO\", O.\"OFI_USUARIO\" " +
            "FROM \"OFICIOS\" O " +
            "WHERE O.\"OFI_ID_TRAMITE\"=@ofiCodigo ";

        SqlCommand cmdSelect = new SqlCommand(sentenciaSelect, cnBaseDatos2);
        cmdSelect.CommandType = CommandType.Text;
        cmdSelect.Parameters.Add("@ofiCodigo", SqlDbType.Int).Value = ofiCodigo;
        SqlDataAdapter daProductos = new SqlDataAdapter();
        daProductos.SelectCommand = cmdSelect;
        DataTable dtProductos = new DataTable();
        daProductos.Fill(dtProductos);
        return dtProductos;
    }
    catch (SqlException error)
    {
        throw error;
    }
    catch (Exception error)
    {
        throw error;
    }
}

public bool InsertarOficio(CIOficios oficios)
{
    bool insertoRegistro = false;
    SqlConnection cnConexion = new SqlConnection();

```

```

try
{
    cnConexion.ConnectionString = ConectarBD.CadenaDeConexion;

    StringBuilder sentencialInsert = new StringBuilder();
    sentencialInsert.Append("INSERT INTO OFICIOS ");
    sentencialInsert.Append("(OFI_FECHA_ING, OFI_MOTIVO, OFI_ESTADO, OFI_NUMERO, OFI_FECHA_ORIG, OFI_REFER,
OFI_REF_SOL, OFI_UNIDAD_INI, DOC_CODIGO, OFI_TRATAMIENTO, OFI_USUARIO, OFI_TIPO, OFI_MOVIN, TRA_CODIGO,
OFI_REM_INST, EXT_CODIGO, OFI_ANIO) ");
    sentencialInsert.Append("VALUES(@ofiFechaIng, @ofiMotivo, @ofiEstado, @ofiNumero, @ofiFechaOrig, @ofiRefer,
@ofiRefSol, @ofiUnidadIni, @docCodigo, @ofiTratamiento, @ofiUsuario, @ofiTipo, @ofiMovin, @traCodigo, @ofiRemInst,
@extCodigo, @ofiAnio)");

    SqlCommand cmdInsert = new SqlCommand();
    cmdInsert.CommandText = sentencialInsert.ToString();
    cmdInsert.CommandType = CommandType.Text;
    cmdInsert.Connection = cnConexion;
    cmdInsert.Parameters.Add("@ofiFechaIng", SqlDbType.DateTime);
    cmdInsert.Parameters.Add("@ofiMotivo", SqlDbType.NVarChar, 500);
    cmdInsert.Parameters.Add("@ofiEstado", SqlDbType.NVarChar, 1);
    cmdInsert.Parameters.Add("@ofiNumero", SqlDbType.NVarChar, 30);
    cmdInsert.Parameters.Add("@ofiFechaOrig", SqlDbType.NVarChar, 10);
    cmdInsert.Parameters.Add("@ofiRefer", SqlDbType.NVarChar, 30);
    cmdInsert.Parameters.Add("@ofiRefSol", SqlDbType.NVarChar, 30);
    cmdInsert.Parameters.Add("@ofiUnidadIni", SqlDbType.NVarChar, 10);
    cmdInsert.Parameters.Add("@docCodigo", SqlDbType.Int);
    cmdInsert.Parameters.Add("@ofiTratamiento", SqlDbType.NVarChar, 200);
    cmdInsert.Parameters.Add("@ofiUsuario", SqlDbType.NVarChar, 20);
    cmdInsert.Parameters.Add("@ofiTipo", SqlDbType.NVarChar, 1);
    cmdInsert.Parameters.Add("@ofiMovin", SqlDbType.NVarChar, 1);
    cmdInsert.Parameters.Add("@traCodigo", SqlDbType.Int);
    cmdInsert.Parameters.Add("@ofiRemInst", SqlDbType.NVarChar, 80);
    cmdInsert.Parameters.Add("@extCodigo", SqlDbType.Int);
    cmdInsert.Parameters.Add("@ofiAnio", SqlDbType.Int);
    cmdInsert.Parameters[0].Value = oficios.OfiFechaIng;
    cmdInsert.Parameters[1].Value = oficios.OfiMotivo;
    cmdInsert.Parameters[2].Value = oficios.OfiEstado;
    cmdInsert.Parameters[3].Value = oficios.OfiNumero;
    cmdInsert.Parameters[4].Value = oficios.OfiFechaOrig;
    cmdInsert.Parameters[5].Value = oficios.OfiRefer;
    cmdInsert.Parameters[6].Value = oficios.OfiRefSol;
    cmdInsert.Parameters[7].Value = oficios.OfiUnidadIni;
    cmdInsert.Parameters[8].Value = oficios.DocCodigo;
    cmdInsert.Parameters[9].Value = oficios.OfiTratamiento;
    cmdInsert.Parameters[10].Value = oficios.OfiUsuario;
    cmdInsert.Parameters[11].Value = oficios.OfiTipo;
    cmdInsert.Parameters[12].Value = oficios.OfiMovin;
    cmdInsert.Parameters[13].Value = oficios.TraCodigo;
    cmdInsert.Parameters[14].Value = oficios.OfiRemInst;
    cmdInsert.Parameters[15].Value = oficios.ExtCodigo;
    cmdInsert.Parameters[16].Value = oficios.OfiAnio;
    cnConexion.Open();
    int filasAfectas = cmdInsert.ExecuteNonQuery();
    if (filasAfectas == 1)
    {
        insertoRegistro = true;
    }
    return insertoRegistro;
}
catch (SqlException errorSqlServer)
{
    throw errorSqlServer;
}

```



```

catch (Exception errorCSharp)
{
    throw errorCSharp;
}
}

public bool InsertarOficioHijos(CIOficios oficios)
{
    bool insertoRegistro = false;
    SqlConnection cnConexion = new SqlConnection();
    try
    {
        cnConexion.ConnectionString = ConectarBD.CadenaDeConexion;
        StringBuilder sentencialInsert = new StringBuilder();
        sentencialInsert.Append("INSERT INTO OFICIOS ");
        sentencialInsert.Append("(OFI_FECHA_ING, OFI_MOTIVO, OFI_FECHA_SAL, OFI_SAL_OBSER, OFI_SALIDA,
OFI_ESTADO, OFI_NUMERO, OFI_UNIDAD_INI, OFI_PADRE, DOC_CODIGO, OFI_USUARIO, FUN_CODIGO, OFI_TRAMITE,
OFI_ANIO) ");
        sentencialInsert.Append("VALUES(@ofiFechaIng, @ofiMotivo, @ofiFechaSal, @ofiSalObser,@ofiSalida, @ofiEstado,
@ofiNumero, @ofiUnidadIni, @ofiPadre, @docCodigo, @ofiUsuario, @funCodigo, @ofiTramite, @ofiAnio)");

        SqlCommand cmdInsert = new SqlCommand();
        cmdInsert.CommandText = sentencialInsert.ToString();
        cmdInsert.CommandType = CommandType.Text;
        cmdInsert.Connection = cnConexion;
        cmdInsert.Parameters.Add("@ofiFechaIng", SqlDbType.DateTime);
        cmdInsert.Parameters.Add("@ofiMotivo", SqlDbType.NVarChar, 500);
        cmdInsert.Parameters.Add("@ofiFechaSal", SqlDbType.DateTime);
        cmdInsert.Parameters.Add("@ofiSalObser", SqlDbType.NVarChar, 500);
        cmdInsert.Parameters.Add("@ofiSalida", SqlDbType.NVarChar, 30);
        cmdInsert.Parameters.Add("@ofiEstado", SqlDbType.NVarChar, 1);
        cmdInsert.Parameters.Add("@ofiNumero", SqlDbType.NVarChar, 30);
        cmdInsert.Parameters.Add("@ofiUnidadIni", SqlDbType.NVarChar, 10);
        cmdInsert.Parameters.Add("@ofiPadre", SqlDbType.Int);
        cmdInsert.Parameters.Add("@docCodigo", SqlDbType.Int);
        cmdInsert.Parameters.Add("@ofiUsuario", SqlDbType.NVarChar, 20);
        cmdInsert.Parameters.Add("@funCodigo", SqlDbType.NVarChar, 10);
        cmdInsert.Parameters.Add("@ofiTramite", SqlDbType.NVarChar, 20);
        cmdInsert.Parameters.Add("@ofiAnio", SqlDbType.Int);
        cmdInsert.Parameters[0].Value = oficios.OfiFechaIng;
        cmdInsert.Parameters[1].Value = oficios.OfiMotivo;
        cmdInsert.Parameters[2].Value = oficios.OfiFechaSal;
        cmdInsert.Parameters[3].Value = oficios.OfiSalObser;
        cmdInsert.Parameters[4].Value = oficios.OfiSalida;
        cmdInsert.Parameters[5].Value = oficios.OfiEstado;
        cmdInsert.Parameters[6].Value = oficios.OfiNumero;
        cmdInsert.Parameters[7].Value = oficios.OfiUnidadIni;
        cmdInsert.Parameters[8].Value = oficios.OfiPadre;
        cmdInsert.Parameters[9].Value = oficios.DocCodigo;
        cmdInsert.Parameters[10].Value = oficios.OfiUsuario;
        cmdInsert.Parameters[11].Value = oficios.FunCodigo;
        cmdInsert.Parameters[12].Value = oficios.OfiTramite;
        cmdInsert.Parameters[13].Value = oficios.OfiAnio;
        cnConexion.Open();
        int filasAfectas = cmdInsert.ExecuteNonQuery();
        if (filasAfectas == 1)
        {
            insertoRegistro = true;
        }
        return insertoRegistro;
    }
    catch (SqlException errorSqlServer)
    {

```

```

        throw errorSqlServer;
    }
    catch (Exception errorCSharp)
    {
        throw errorCSharp;
    }
}

public bool InsertarFlow(CIOficios flow)
{
    bool insertoRegistro = false;

    SqlConnection cnConexion = new SqlConnection();
    try
    {
        cnConexion.ConnectionString = ConectarBD.CadenaDeConexion;
        StringBuilder sentencialInsert = new StringBuilder();
        sentencialInsert.Append("INSERT INTO FLOW ");
        sentencialInsert.Append("(FLOW_ANIO, FLOW_OFI_CODIGO) ");
        sentencialInsert.Append("VALUES(@flow_anio, @flow_ofi_codigo)");

        SqlCommand cmdInsert = new SqlCommand();
        cmdInsert.CommandText = sentencialInsert.ToString();
        cmdInsert.CommandType = CommandType.Text;
        cmdInsert.Connection = cnConexion;
        cmdInsert.Parameters.Add("@flow_anio", SqlDbType.Int);
        cmdInsert.Parameters.Add("@flow_ofi_codigo", SqlDbType.Int);
        cmdInsert.Parameters[0].Value = flow.FlowAnio;
        cmdInsert.Parameters[1].Value = flow.FlowOfiCodigo;
        cnConexion.Open();
        int filasAfectas = cmdInsert.ExecuteNonQuery();
        if (filasAfectas == 1)
        {
            insertoRegistro = true;
        }
        return insertoRegistro;
    }
    catch (SqlException errorSqlServer)
    {
        throw errorSqlServer;
    }
    catch (Exception errorCSharp)
    {
        throw errorCSharp;
    }
}

public void ActualizarFlow(CIOficios flow)
{
    SqlConnection cnConexion = null;
    try
    {
        cnConexion = new SqlConnection(ConectarBD.CadenaDeConexion);
        StringBuilder cadenaUpdate = new StringBuilder();
        cadenaUpdate.Append("UPDATE OFICIOS ");
        cadenaUpdate.Append("SET OFI_ID_TRAMITE = @ofi_id_tramite ");
        cadenaUpdate.Append("WHERE OFI_CODIGO = '"+ flow.OfiCodigo + "'");

        SqlCommand cmdUpdate = new SqlCommand();
        cmdUpdate.CommandText = cadenaUpdate.ToString();
        cmdUpdate.CommandType = CommandType.Text;
        cmdUpdate.Connection = cnConexion;
    }
}

```

```

cmdUpdate.Parameters.Add("@ofi_id_tramite", SqlDbType.Int).Value = flow.OfidTramite;
cnConexion.Open();
cmdUpdate.ExecuteNonQuery();
}
catch (SqlException error)
{
    throw error;
}
catch (Exception error)
{
    throw error;
}
finally
{
    cnConexion.Close();
}
}

public void Actualizar(CIOficios oficios)
{
    SqlConnection cnConexion = null;
    try
    {
        cnConexion = new SqlConnection(ConectarBD.CadenaDeConexion);
        StringBuilder cadenaUpdate = new StringBuilder();
        cadenaUpdate.Append("UPDATE OFICIOS ");
        cadenaUpdate.Append("SET OFI_MOTIVO = @ofiMotivo, ");
        cadenaUpdate.Append("OFI_NUMERO = @ofiNumero, ");
        cadenaUpdate.Append("OFI_FECHA_ORIG = @ofiFechaOrig, ");
        cadenaUpdate.Append("OFI_REFER = @ofiRefer, ");
        cadenaUpdate.Append("OFI_REF_SOL = @ofiRefSol, ");
        cadenaUpdate.Append("DOC_CODIGO = @docCodigo, ");
        cadenaUpdate.Append("OFI_TRATAMIENTO = @ofiTratamiento, ");
        cadenaUpdate.Append("OFI_TIPO = @ofiTipo, ");
        cadenaUpdate.Append("TRA_CODIGO = @traCodigo, ");
        cadenaUpdate.Append("OFI_REM_INST = @ofiRemInst, ");
        cadenaUpdate.Append("EXT_CODIGO = @extCodigo ");
        cadenaUpdate.Append("WHERE OFI_CODIGO = '"+ oficios.OfiCodigo + "'");
        SqlCommand cmdUpdate = new SqlCommand();
        cmdUpdate.CommandText = cadenaUpdate.ToString();
        cmdUpdate.CommandType = CommandType.Text;
        cmdUpdate.Connection = cnConexion;
        cmdUpdate.Parameters.Add("@ofiMotivo", SqlDbType.VarChar, 500).Value = oficios.OfiMotivo;
        cmdUpdate.Parameters.Add("@ofiNumero", SqlDbType.VarChar, 30).Value = oficios.OfiNumero;
        cmdUpdate.Parameters.Add("@ofiFechaOrig", SqlDbType.VarChar, 10).Value = oficios.OfiFechaOrig;
        cmdUpdate.Parameters.Add("@ofiRefer", SqlDbType.VarChar, 30).Value = oficios.OfiRefer;
        cmdUpdate.Parameters.Add("@ofiRefSol", SqlDbType.VarChar, 30).Value = oficios.OfiRefSol;
        cmdUpdate.Parameters.Add("@docCodigo", SqlDbType.Int).Value = oficios.DocCodigo;
        cmdUpdate.Parameters.Add("@ofiTratamiento", SqlDbType.VarChar, 200).Value = oficios.OfiTratamiento;
        cmdUpdate.Parameters.Add("@ofiTipo", SqlDbType.VarChar, 1).Value = oficios.OfiTipo;
        cmdUpdate.Parameters.Add("@traCodigo", SqlDbType.Int).Value = oficios.TraCodigo;
        cmdUpdate.Parameters.Add("@ofiRemInst", SqlDbType.VarChar, 80).Value = oficios.OfiRemInst;
        cmdUpdate.Parameters.Add("@extCodigo", SqlDbType.Int).Value = oficios.ExtCodigo;
        cnConexion.Open();
        cmdUpdate.ExecuteNonQuery();
    }
    catch (SqlException error)
    {
        throw error;
    }
    catch (Exception error)
    {
        throw error;
    }
}

```

```

    }
    finally
    {
        cnConexion.Close();
    }
}

public void ActualizarEstado(CIOficios oficios)
{
    SqlConnection cnConexion = null;
    try
    {
        cnConexion = new SqlConnection(ConectarBD.CadenaDeConexion);
        StringBuilder cadenaUpdate = new StringBuilder();
        cadenaUpdate.Append("UPDATE OFICIOS ");
        cadenaUpdate.Append("SET OFI_ESTADO = @ofiEstado ");
        cadenaUpdate.Append("WHERE OFI_ESTADO <> 'F' AND OFI_CODIGO = '"+ oficios.OfiCodigo + "'");
        SqlCommand cmdUpdate = new SqlCommand();
        cmdUpdate.CommandText = cadenaUpdate.ToString();
        cmdUpdate.CommandType = CommandType.Text;
        cmdUpdate.Connection = cnConexion;
        cmdUpdate.Parameters.Add("@ofiEstado", SqlDbType.NVarChar, 1).Value = oficios.OfiEstado;
        cnConexion.Open();
        cmdUpdate.ExecuteNonQuery();
    }
    catch (SqlException error)
    {
        throw error;
    }
    catch (Exception error)
    {
        throw error;
    }
    finally
    {
        cnConexion.Close();
    }
}

public void ActualizarEstadoFinalizado(CIOficios oficios)
{
    SqlConnection cnConexion = null;
    try
    {
        cnConexion = new SqlConnection(ConectarBD.CadenaDeConexion);
        StringBuilder cadenaUpdate = new StringBuilder();
        cadenaUpdate.Append("UPDATE OFICIOS ");
        cadenaUpdate.Append("SET OFI_ESTADO = @ofiEstado ");
        cadenaUpdate.Append("WHERE OFI_ESTADO = 'F' AND OFI_CODIGO = '"+ oficios.OfiCodigo + "'");
        SqlCommand cmdUpdate = new SqlCommand();
        cmdUpdate.CommandText = cadenaUpdate.ToString();
        cmdUpdate.CommandType = CommandType.Text;
        cmdUpdate.Connection = cnConexion;
        cmdUpdate.Parameters.Add("@ofiEstado", SqlDbType.NVarChar, 1).Value = oficios.OfiEstado;
        cnConexion.Open();
        cmdUpdate.ExecuteNonQuery();
    }
    catch (SqlException error)
    {
        throw error;
    }
    catch (Exception error)
    {

```

```

        throw error;
    }
    finally
    {
        cnConexion.Close();
    }
}

public void ActualizarOficioOriginal(CIOficios oficios)
{
    SqlConnection cnConexion = null;
    try
    {
        cnConexion = new SqlConnection(ConectarBD.CadenaDeConexion);
        StringBuilder cadenaUpdate = new StringBuilder();
        cadenaUpdate.Append("UPDATE OFICIOS ");
        cadenaUpdate.Append("SET OFI_ORIGINAL = @ofiOriginal ");
        cadenaUpdate.Append("WHERE OFI_CODIGO = '"+ oficios.OfiCodigo + "'");

        SqlCommand cmdUpdate = new SqlCommand();
        cmdUpdate.CommandText = cadenaUpdate.ToString();
        cmdUpdate.CommandType = CommandType.Text;
        cmdUpdate.Connection = cnConexion;

        cmdUpdate.Parameters.Add("@ofiOriginal", SqlDbType.Int).Value = oficios.OfiOriginal;

        cnConexion.Open();

        cmdUpdate.ExecuteNonQuery();
    }
    catch (SqlException error)
    {
        throw error;
    }
    catch (Exception error)
    {
        throw error;
    }
    finally
    {
        cnConexion.Close();
    }
}
#endregion
}
}

```

2. Servidor

El siguiente código fuente representa la creación de un documento mediante la Forma EnviodeDocumentos.

Forma EnviodeDocumentos
<pre> using System; using System.Drawing; using System.Windows.Forms; using Microsoft.Practices.CompositeUI.Commands; using DTVBServidor; using System.Data; </pre>

```

using System.Data.SqlClient;
using System.Security.Principal;
using System.IO;

namespace DTVBCevallos.Movimientos
{
    /// <summary>
    /// Description of EnviodeDocumentos.
    /// </summary>
    public partial class EnviodeDocumentos : Form
    {
        #region Datos
        //Para capturar la pantalla
        [System.Runtime.InteropServices.DllImport("gdi32.dll")]
        public static extern long BitBlt(IntPtr hdcDest, int nXDest, int nYDest, int nWidth, int nHeight, IntPtr
hdcSrc, int nXSrc, int nYSrc, int dwRop);
        private Bitmap memoryImage;
        private void CaptureScreen()
        {
            Graphics mygraphics = this.CreateGraphics();
            Size s = this.Size;
            memoryImage = new Bitmap(s.Width, s.Height, mygraphics);
            Graphics memoryGraphics = Graphics.FromImage(memoryImage);
            IntPtr dc1 = mygraphics.GetHdc();
            IntPtr dc2 = memoryGraphics.GetHdc();
            BitBlt(dc2, 0, 0, this.ClientRectangle.Width, this.ClientRectangle.Height, dc1, 0, 0, 13369376);
            mygraphics.ReleaseHdc(dc1);
            memoryGraphics.ReleaseHdc(dc2);
        }
        //Ingreso de Variables
        private DataSet dsDatosXML; //Almacena las tablas y los datos recuperados.
        private DataTable dtDatosXML;
        private int registroActual = 0;
        private int original; //Almacena el total de registros
        private string banderaEstado = "";
        // Instanciamos un objeto de BuscarContribuyenteExterno
        EnviodeDocumentos.BuscarContribuyenteExterno formulario = new EnviodeDocumentos.BuscarContribuyenteExterno();
        EnviodeDocumentos.SeleccionDirecciones envio = new EnviodeDocumentos.SeleccionDirecciones();
        #endregion

        public EnviodeDocumentos()
        {
            //
            // The InitializeComponent() call is required for Windows Forms designer support.
            //
            InitializeComponent();

            dtDatosXML = new DataTable();
            //
            // TODO: Add constructor code after the InitializeComponent() call.
            //
        }

        #region Funciones

        public void MostrarDatosComboBoxTipoDocumento()
        {
            TipoDocumentos tipoDocumentos = new TipoDocumentos();
            try
            {
                cboTipoDocumentos.DataSource = tipoDocumentos.Seleccionar().Tables["DOCUMENTOS"];
                cboDocumentoCodigo.DataSource = tipoDocumentos.Seleccionar().Tables["DOCUMENTOS"];
                cboTipoDocumentos.DisplayMember = "DOC_DESCRIPCION";
            }
            catch { }
        }
    }
}

```

```

        cboDocumentoCodigo.DisplayMember = "DOC_CODIGO";
        cboTipoDocumentos.ValueMember = "DOC_CODIGO";
        cboDocumentoCodigo.ValueMember = "DOC_CODIGO";
    }
    catch (SqlException error)
    {
        Excepciones.TratarErroresWindows(error);
    }
    catch (Exception error)
    {
        Excepciones.TratarErroresWindows(error);
    }
}

public void MostrarDatosComboBoxTramite()
{
    CITramite tramite = new CITramite();
    try
    {
        cboTramites.DataSource = tramite.Seleccionar().Tables["TRAMITES"];
        cboTramiteCodigo.DataSource = tramite.Seleccionar().Tables["TRAMITES"];
        cboTramites.DisplayMember = "TRAM_NOMBRE";
        cboTramiteCodigo.DisplayMember = "TRAM_CODIGO";
        cboTramites.ValueMember = "TRAM_CODIGO";
        cboTramiteCodigo.ValueMember = "TRAM_CODIGO";
    }
    catch (SqlException error)
    {
        Excepciones.TratarErroresWindows(error);
    }
    catch (Exception error)
    {
        Excepciones.TratarErroresWindows(error);
    }
}

public void MostrarDatosComboBoxEstado()
{
    CIOficios estado = new CIOficios();
    try
    {
        cboEstado.DataSource = estado.SeleccionarEstado().Tables["ESTADOFICIO"];
        cboEstadoOfi.DataSource = estado.SeleccionarEstado().Tables["ESTADOFICIO"];
        cboEstado.DisplayMember = "EXT_DESCRIPCION";
        cboEstadoOfi.DisplayMember = "EST_OFICIO";
        cboEstado.ValueMember = "EST_OFICIO";
        cboEstadoOfi.ValueMember = "EST_OFICIO";
    }
    catch (SqlException error)
    {
        Excepciones.TratarErroresWindows(error);
    }
    catch (Exception error)
    {
        Excepciones.TratarErroresWindows(error);
    }
}

public void MostrarDatosComboFW1()
{
    CIOficios oficios = new CIOficios();
    try
    {

```

```

        dsDatosXML = oficios.Seleccionar();
        this.BindingContext[dsDatosXML, "OFICIOS"].Position = this.BindingContext[dsDatosXML, "OFICIOS"].Count;
        registroActual = this.BindingContext[dsDatosXML, "OFICIOS"].Position;
        txtFW.Text = dsDatosXML.Tables["OFICIOS"].Rows[registroActual]["OFI_CODIGO"].ToString();
    }
    catch (SQLException error)
    {
        Excepciones.TratarErroresWindows(error);
    }
    catch (Exception error)
    {
        Excepciones.TratarErroresWindows(error);
    }
}

public void MostrarDatosComboFW2()
{
    CIOficios flow = new CIOficios();
    try
    {
        dsDatosXML = flow.SeleccionarFlow();
        //Mover el puntero al ultimo registro del DataSet.
        this.BindingContext[dsDatosXML, "FLOW"].Position = this.BindingContext[dsDatosXML, "FLOW"].Count;
        registroActual = this.BindingContext[dsDatosXML, "FLOW"].Position; //Siempre el puntero esta en el primer registro
(0).
        txtFW.Text = dsDatosXML.Tables["FLOW"].Rows[registroActual]["FLOW_CODIGO"].ToString();
    }
    catch (SQLException error)
    {
        Excepciones.TratarErroresWindows(error);
    }
    catch (Exception error)
    {
        Excepciones.TratarErroresWindows(error);
    }
}

public void MostrarDatosEnControles()
{
    try
    {
        CIOficios oficios = new CIOficios();
        dtDatosXML = oficios.SeleccionarPorFW(txtFW.Origin.Text);

        cboTramiteCodigo.Text = dtDatosXML.Rows[0]["TRA_CODIGO"].ToString();
        cboDocumentoCodigo.Text = dtDatosXML.Rows[0]["DOC_CODIGO"].ToString();
        txtOfiRefer.Text = dtDatosXML.Rows[0]["OFI_NUMERO"].ToString();
        txtFechaOriginal.Text = dtDatosXML.Rows[0]["OFI_FECHA_ING"].ToString();
        original = Convert.ToInt32(dtDatosXML.Rows[0]["OFI_CODIGO"].ToString());
    }
    catch (SQLException error)
    {
        Excepciones.TratarErroresWindows(error);
    }
    catch (Exception error)
    {
        Excepciones.TratarErroresWindows(error);
    }
}

public void GuardarNuevoRegistro()
{
    try

```



```

{
    COficios oficios = new COficios();
    oficios.OfiFechaIng = Convert.ToDateTime(txtFechaHora.Text);
    oficios.OfiMotivo = txtAsunto.Text.Trim();
    oficios.OfiEstado = cboEstadoOfi.Text.Trim();
    oficios.OfiNumero = txtNumero.Text.Trim();
    oficios.OfiFechaOrig = txtFechaOriginal.Text.Trim();
    oficios.OfiRefer = txtOfiRefer.Text.Trim();
    oficios.OfiRefSol = txtOfiRefSol.Text.Trim();
    oficios.OfiUnidadIni = ClaseEstatica.usuariodepartamentocodigo.Trim();
    oficios.DocCodigo = System.Convert.ToInt32(cboDocumentoCodigo.Text.Trim());
    oficios.OfiTratamiento = txtTratamiento.Text.Trim();
    oficios.OfiUsuario = ClaseEstatica.usuariologueado.Trim();
    if(cboOrigen.Text == "Interno")
        oficios.OfiTipo = "I";
    else
        oficios.OfiTipo = "E";
    oficios.OfiMovin = "E";
    oficios.TraCodigo = System.Convert.ToInt32(cboTramiteCodigo.Text.Trim());
    oficios.OfiRemInst = txtInstituto.Text.Trim();
    if(txtCedula.Text == "")
        oficios.ExtCodigo = 0;
    else
        oficios.ExtCodigo = System.Convert.ToInt32(txtCedula.Text.Trim());
    oficios.OfiAnio = System.Convert.ToInt32(cboAnio.Text.Trim());

    oficios.InsertarOficio(oficios);

    MessageBox.Show("Registro Grabado", "Exito", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
catch (SQLException error)
{
    Excepciones.TratarErroresWindows(error);
}
catch (Exception error)
{
    Excepciones.TratarErroresWindows(error);
}
}

public void GuardarNuevoRegistroTablaFlow()
{
    try
    {
        COficios flow = new COficios();
        flow.FlowAnio = System.Convert.ToInt32(cboAnio.Text.Trim());
        flow.FlowOfiCodigo = System.Convert.ToInt32(txtFW.Text.Trim());
        flow.InsertarFlow(flow);
    }
    catch (SQLException error)
    {
        Excepciones.TratarErroresWindows(error);
    }
    catch (Exception error)
    {
        Excepciones.TratarErroresWindows(error);
    }
}

private void ActualizarRegistroFlow()
{
    try

```

```

    {
        CIOficios flow = new CIOficios();
        flow.OfiCodigo = Convert.ToInt32(cboFW.Text.Trim());
        flow.OfiIdTramite = Convert.ToInt32(txtFW.Text.Trim());
        flow.ActualizarFlow(flow);

        //MessageBox.Show("Actualizado con exito", "Información", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (SqlException error)
    {
        Excepciones.TratarErroresWindows(error);
    }
    catch (Exception error)
    {
        Excepciones.TratarErroresWindows(error);
    }
}

private void ActualizarRegistro()
{
    try
    {
        CIOficios oficios = new CIOficios();
        oficios.OfiCodigo = Convert.ToInt32(cboFW.Text.Trim());
        oficios.OfiMotivo = txtAsunto.Text.Trim();
        oficios.OfiNumero = txtNumero.Text.Trim();
        oficios.OfiFechaOrig = txtFechaOriginal.Text.Trim();
        oficios.OfiRefer = txtOfiRefer.Text.Trim();
        oficios.OfiRefSol = txtOfiRefSol.Text.Trim();
        oficios.DocCodigo = System.Convert.ToInt32(cboDocumentoCodigo.Text.Trim());
        oficios.OfiTratamiento = txtTratamiento.Text.Trim();
        if(cboOrigen.Text == "Interno")
            oficios.OfiTipo = "I";
        else
            oficios.OfiTipo = "E";
        //oficios.OfiMovin = "E";
        oficios.TraCodigo = System.Convert.ToInt32(cboTramiteCodigo.Text.Trim());
        oficios.OfiRemInst = txtInstituto.Text.Trim();
        if(txtCedula.Text == "")
            oficios.ExtCodigo = 0;
        else
            oficios.ExtCodigo = System.Convert.ToInt32(txtCedula.Text.Trim());

        oficios.Actualizar(oficios);

        MessageBox.Show("Actualizado con exito", "Información", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (SqlException error)
    {
        Excepciones.TratarErroresWindows(error);
    }
    catch (Exception error)
    {
        Excepciones.TratarErroresWindows(error);
    }
}

private void LimpiarComponentes()
{
    txtFW.Text = "";
    cboEstadoOfi.Text = "";
    cboTramiteCodigo.Text = "1";
    cboTramites.Text = "";
}

```

```

cboDocumentoCodigo.Text = "1";
cboTipoDocumentos.Text = "";
txtFechaHora.Text = DateTime.Now.ToString("d/MM/yyyy HH:mm:ss");
banderaEstado = "";
cboEstado.Text = "";
txtNumero.Text = "";
cboFW.Text = "";
txtCedula.Text = "";
txtNombresExternos.Text = "";
txtDireccion.Text = "";
txtInstituto.Text = "";
txtFWOrigen.Text = "";
txtOfiRefer.Text = "";
txtOfiRefSol.Text = "";
txtFechaOriginal.Text = "";
txtTratamiento.Text = "";
txtAsunto.Text = "";
}

private void AbilitarComponentes()
{
    cboTramites.Enabled = true;
    cboTipoDocumentos.Enabled = true;
    txtFechaHora.Enabled = true;
    cboEstado.Enabled = true;
    txtNumero.Enabled = true;
    cboOrigen.Enabled = true;
    txtInstituto.Enabled = true;
    txtOfiRefer.Enabled = true;
    txtOfiRefSol.Enabled = true;
    txtFechaOriginal.Enabled = true;
    txtTratamiento.Enabled = true;
    txtAsunto.Enabled = true;
}

private void DesabilitarComponentes()
{
    cboTramites.Enabled = false;
    cboTipoDocumentos.Enabled = false;
    txtFechaHora.Enabled = false;
    cboEstado.Enabled = false;
    txtNumero.Enabled = false;
    cboOrigen.Enabled = false;
    txtInstituto.Enabled = false;
    txtOfiRefer.Enabled = false;
    txtOfiRefSol.Enabled = false;
    txtFechaOriginal.Enabled = false;
    txtTratamiento.Enabled = false;
    txtAsunto.Enabled = false;
}

public void MostrarDatosEnVentana()
{
    string tipo;
    string estado;
    try
    {
        CIOficios oficios = new CIOficios();
        dtDatosXML = oficios.SeleccionarPorOficioAnio1(txtFW.Text, cboAnio.Text);

        if(this.BindingContext[dtDatosXML].Count != 0)
        {
            cboFW.Text = dtDatosXML.Rows[0]["OFI_CODIGO"].ToString();
        }
    }
}

```

```

cboTipoDocumentos.Text = dtDatosXML.Rows[0]["DOC_DESCRIPCION"].ToString();
cboTramites.Text = dtDatosXML.Rows[0]["TRAM_NOMBRE"].ToString();
txtFechaHora.Text = dtDatosXML.Rows[0]["OFI_FECHA_ING"].ToString();
txtNumero.Text = dtDatosXML.Rows[0]["OFI_NUMERO"].ToString();
tipo = dtDatosXML.Rows[0]["OFI_TIPO"].ToString();
if(tipo == "I")
    cboOrigen.Text = "Interno";
if(tipo == "E")
    cboOrigen.Text = "Extermo";
estado = dtDatosXML.Rows[0]["OFI_ESTADO"].ToString();
if(estado != "I")
{
    DesabilitarComponentes();
}
else
{
    AbilitarComponentes();
}
if(estado == "I")
{
    banderaEstado = "INGRESADO";
    cboEstado.Text = "INGRESADO";
}
if(estado == "E")
{
    banderaEstado = "ENVIADO";
    cboEstado.Text = "ENVIADO";
}
if(estado == "P")
{
    banderaEstado = "PENDIENTE";
    cboEstado.Text = "PENDIENTE";
}
if(estado == "F")
{
    banderaEstado = "FINALIZADO";
    cboEstado.Text = "FINALIZADO";
}
txtAsunto.Text = dtDatosXML.Rows[0]["OFI_MOTIVO"].ToString();
txtTratamiento.Text = dtDatosXML.Rows[0]["OFI_TRATAMIENTO"].ToString();
txtFWOrigen.Text = dtDatosXML.Rows[0]["OFI_ORIGINAL"].ToString();
txtOfiRefer.Text = dtDatosXML.Rows[0]["OFI_REFER"].ToString();
txtOfiRefSol.Text = dtDatosXML.Rows[0]["OFI_REF_SOL"].ToString();
txtFechaOriginal.Text = dtDatosXML.Rows[0]["OFI_FECHA_ORIG"].ToString();
}
else
{
    MessageBox.Show("Documento no existente. Ingrese el ID
correcto.", "Información", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}
catch (SqlException error)
{
    Excepciones.TratarErroresWindows(error);
}
catch (Exception error)
{
    Excepciones.TratarErroresWindows(error);
}
}

public void MostrarDatosEnVentanaExterno()
{

```

```

string tipo;
string estado;
try
{
    CIOficios oficios = new CIOficios();
    dtDatosXML = oficios.SeleccionarPorOficioAnio(txtFW.Text, cboAnio.Text);

    if(this.BindingContext[dtDatosXML].Count != 0)
    {
        cboFW.Text = dtDatosXML.Rows[0]["OFI_CODIGO"].ToString();
        cboTipoDocumentos.Text = dtDatosXML.Rows[0]["DOC_DESCRIPCION"].ToString();
        cboTramites.Text = dtDatosXML.Rows[0]["TRAM_NOMBRE"].ToString();
        txtFechaHora.Text = dtDatosXML.Rows[0]["OFI_FECHA_ING"].ToString();
        txtNumero.Text = dtDatosXML.Rows[0]["OFI_NUMERO"].ToString();
        tipo = dtDatosXML.Rows[0]["OFI_TIPO"].ToString();
        if(tipo == "I")
            cboOrigen.Text = "Interno";
        if(tipo == "E")
            cboOrigen.Text = "Externo";
        estado = dtDatosXML.Rows[0]["OFI_ESTADO"].ToString();
        if(estado != "I")
        {
            DesabilitarComponentes();
        }
        else
        {
            HabilitarComponentes();
        }
        if(estado == "I")
        {
            banderaEstado = "INGRESADO";
            cboEstado.Text = "INGRESADO";
        }
        if(estado == "E")
        {
            banderaEstado = "ENVIADO";
            cboEstado.Text = "ENVIADO";
        }
        if(estado == "P")
        {
            banderaEstado = "PENDIENTE";
            cboEstado.Text = "PENDIENTE";
        }
        if(estado == "F")
        {
            banderaEstado = "FINALIZADO";
            cboEstado.Text = "FINALIZADO";
        }
        txtCedula.Text = dtDatosXML.Rows[0]["EXT_CODIGO"].ToString();
        txtNombresExternos.Text = dtDatosXML.Rows[0]["EXT_APELLIDO"].ToString() + "
+ dtDatosXML.Rows[0]["EXT_NOMBRE"].ToString();
        txtDireccion.Text = dtDatosXML.Rows[0]["EXT_DIRECCION"].ToString();
        txtInstituto.Text = dtDatosXML.Rows[0]["OFI_REM_INST"].ToString();
        txtAsunto.Text = dtDatosXML.Rows[0]["OFI_MOTIVO"].ToString();
        txtTratamiento.Text = dtDatosXML.Rows[0]["OFI_TRATAMIENTO"].ToString();
        txtFWOrigen.Text = dtDatosXML.Rows[0]["OFI_ORIGINAL"].ToString();
        txtOfiRefer.Text = dtDatosXML.Rows[0]["OFI_REFER"].ToString();
        txtOfiRefSol.Text = dtDatosXML.Rows[0]["OFI_REF_SOL"].ToString();
        txtFechaOriginal.Text = dtDatosXML.Rows[0]["OFI_FECHA_ORIG"].ToString();
    }
}
else
{

```

```

        MessageBox.Show("Documento no existente. Ingrese el ID
correcto.", "Información", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
catch (SqlException error)
{
    Excepciones.TratarErroresWindows(error);
}
catch (Exception error)
{
    Excepciones.TratarErroresWindows(error);
}
}
}

#endregion

void EnviodeDocumentosLoad(object sender, EventArgs e)
{
    formulario.MiEvento += new EnvioDocumentos.BuscarContribuyenteExterno.DelegadoTitulo(PonerTitulo);
    envio.MiEvento += new EnvioDocumentos.SeleccioneDirecciones.DelegadoTitulo(PonerEstado);
    ControlBox=false;
    if(ClaseEstatica.ControlRemite == 0)
    {
        this.WindowState = FormWindowState.Maximized;
    }
    else
    {
        txtTitulo.Text = "Se encuentra: REMITIENDO";
        txtFWOrigen.Text = ClaseEstatica.fwpadreenvio;
    }
    MostrarDatosComboBoxTipoDocumento();
    MostrarDatosComboBoxTramite();
    MostrarDatosComboBoxEstado();

    txtFechaHora.Text = DateTime.Now.ToString("d/MM/yyyy HH:mm:ss");
    cboAnio.Text = DateTime.Now.ToString("yyy");
    cboTipoDocumentos.Text="";
    cboTramites.Text="";
    cboTramiteCodigo.Text="199";
    cboDocumentoCodigo.Text = "1";
    cboEstado.Text = "";
    cboEstadoOfi.Text = "";
    if(txtFWOrigen.Text != "")
    {
        MostrarDatosEnControles();
    }
}

//Metodo de Form1 que pondrá el titulo al Form
void PonerTitulo(string mensaje)
{
    txtCedula.Text = mensaje;
}

void PonerEstado(string mensaje)
{
    banderaEstado = mensaje;
    cboEstado.Text = mensaje;
}

void Button1Click(object sender, EventArgs e)
{

```

```

    formulario.ShowDialog();
}

void Button2Click(object sender, EventArgs e)
{
    EnvioDocumentos.UsuariosExternos ventana = new EnvioDocumentos.UsuariosExternos();
    ventana.ShowDialog();
}

void TsbCancelarClick(object sender, EventArgs e)
{
    ClaseEstatica.ControlRemite = 0;
    if(txtFW.Text != "" && txtFWOrigen.Text != "")
    {
        if(cboEstado.Text != "ENVIADO")
        {
            MessageBox.Show("El Documento no a sido enviado. No se puede cerrar la
ventana", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        else
        {
            this.Close();
        }
    }
    else
    {
        this.Close();
    }
}

void TsbInformacionClick(object sender, EventArgs e)
{
    MessageBox.Show("Cree un nuevo documento. Los campos con * con campos
requeridos.", "Información", MessageBoxButtons.OK, MessageBoxIcon.Information);
}

void TsbGuardarClick(object sender, EventArgs e)
{
    if (cboTipoDocumentos.Text == "")
    {
        MessageBox.Show("Ingrese el Tipo de
Documento", "Información", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        if (cboOrigen.Text == "")
        {
            MessageBox.Show("Ingrese el Origen del
Documento", "Información", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            if (txtAsunto.Text == "")
            {
                MessageBox.Show("Ingrese el Asunto del
Documento", "Información", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
            else
            {
                if(txtFW.Text == "")
                {
                    GuardarNuevoRegistro();
                    MostrarDatosComboFW1();
                }
            }
        }
    }
}

```

```

        cboFW.Text = txtFW.Text;
        GuardarNuevoRegistroTablaFlow();
        MostrarDatosComboFW2();
        ActualizarRegistroFlow();
        banderaEstado = "INGRESADO";
        cboEstado.Text = "INGRESADO";
    }
    else
    {
        ActualizarRegistro();
    }
    if(txtFWOrigen.Text != "")
    {
        CIOficios oficios = new CIOficios();
        oficios.OfiCodigo = Convert.ToInt32(cboFW.Text.Trim());
        oficios.OfiOriginal = original;
        oficios.ActualizarOficioOriginal(oficios);
    }
}
}
}

void TxtCedulaTextChanged(object sender, EventArgs e)
{
    txtNombresExternos.Text = ClaseEstatica.usuarioexternonombre;
    txtDireccion.Text = ClaseEstatica.usuarioexternodireccion;
}

void Button10Click(object sender, EventArgs e)
{
    if(cboEstado.Text == "INGRESADO")
    {
        ClaseEstatica.fwpadre = txtFW.Text ;
        envio.ShowDialog();
    }
    else
    {
        if(cboEstado.Text == "ENVIADO")
        {
            MessageBox.Show("Este documento ya fue
enviado.", "Nota", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            ClaseEstatica.fwpadre = txtFW.Text;
            // MessageBox.Show(ClaseEstatica.fwpadre);
            EnvioDocumentos.DocumentosEnviadosa ventana = new EnvioDocumentos.DocumentosEnviadosa();
            ventana.ShowDialog();
        }
        else
        {
            if(cboEstado.Text == "FINALIZADO")
            MessageBox.Show("El documento ya se a
finalizado.", "Información", MessageBoxButtons.OK, MessageBoxIcon.Information);
            else
            MessageBox.Show("El documento no se a
ingresado.", "Información", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}

void Button3Click(object sender, EventArgs e)
{
    if(cboEstado.Text != "")
    {

```



```

        if (DialogResult.Yes == MessageBox.Show("Está seguro de finalizar el tramite o tarea
pendiente?", "Atención", MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation))
        {
            banderaEstado = "FINALIZADO";
            cboEstado.Text = "FINALIZADO";
        }
    }
}

void Button4Click(object sender, EventArgs e)
{
    ClaseEstatica.fwpadre = txtFW.Text;
    if(txtFW.Text != "")
    {
        EnvioDocumentos.ArchivarEn ventana = new EnvioDocumentos.ArchivarEn();
        ventana.ShowDialog();
    }
}

void Button7Click(object sender, EventArgs e)
{
    if(cboTramites.Text != "")
    {
        ClaseEstatica.tramiteproceso = cboTramites.Text;
        ClaseEstatica.TramiteProcesoCodigo = cboTramiteCodigo.Text;
        EnvioDocumentos.DocumentosHabilitantes ventana = new EnvioDocumentos.DocumentosHabilitantes();
        ventana.ShowDialog();
    }
    else
    {
        MessageBox.Show("No definió ningún proceso... No puede habilitar
documentos", "Atención", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

void CboTramitesSelectedIndexChanged(object sender, EventArgs e)
{
    if (cboTramites.SelectedValue.ToString() != "System.Data.DataRowView")
    {
        try
        {
            CITramite tramites = new CITramite();
            cboTramiteCodigo.SelectedIndex = cboTramites.SelectedIndex;
        }
        catch (SqlException error)
        {
            Excepciones.TratarErroresWindows(error);
        }
        catch (Exception error)
        {
            Excepciones.TratarErroresWindows(error);
        }
    }
}

void Button8Click(object sender, EventArgs e)
{
    ClaseEstatica.fwpadre = txtFW.Text;
    EnvioDocumentos.Anexar ventana = new EnvioDocumentos.Anexar();
    ventana.ShowDialog();
}

void Button9Click(object sender, EventArgs e)

```

```

{
    ClaseEstatica.fwpadre = txtFW.Text ;
    EnvioDocumentos.TareasInternas ventana = new EnvioDocumentos.TareasInternas();
    ventana.ShowDialog();
}

void TxtCedulaModifiedChanged(object sender, EventArgs e)
{
}

void CboTipoDocumentosSelectedIndexChanged(object sender, EventArgs e)
{
    if (cboTipoDocumentos.SelectedValue.ToString() != "System.Data.DataRowView")
    {
        try
        {
            TipoDocumentos documento = new TipoDocumentos();
            cboDocumentoCodigo.SelectedIndex = cboTipoDocumentos.SelectedIndex;
        }
        catch (SqlException error)
        {
            Excepciones.TratarErroresWindows(error);
        }
        catch (Exception error)
        {
            Excepciones.TratarErroresWindows(error);
        }
    }
}

void CboEstadoSelectedIndexChanged(object sender, EventArgs e)
{
    if (banderaEstado == "")
        cboEstado.Text = "";
    if (banderaEstado == "INGRESADO")
        cboEstado.Text = "INGRESADO";
    if (banderaEstado == "ENVIADO")
        cboEstado.Text = "ENVIADO";
    if (banderaEstado == "PENDIENTE")
        cboEstado.Text = "PENDIENTE";
    if (banderaEstado == "FINALIZADO")
        cboEstado.Text = "FINALIZADO";
    if (cboEstado.SelectedValue.ToString() != "System.Data.DataRowView")
    {
        try
        {
            cboEstadoOfi.SelectedIndex = cboEstado.SelectedIndex;
        }
        catch (SqlException error)
        {
            Excepciones.TratarErroresWindows(error);
        }
        catch (Exception error)
        {
            Excepciones.TratarErroresWindows(error);
        }
    }
    if (cboEstado.Text == "ENVIADO")
    {
    }
    if (cboEstado.Text == "FINALIZADO")

```

```

    {
        MessageBox.Show("Esta finalizado");
    }
}

void CboAnioSelectedIndexChanged(object sender, EventArgs e)
{
    cboAnio.Text = DateTime.Now.ToString("yyy");
}

void TsblImprimirClick(object sender, EventArgs e)
{
    if(txtFW.Text != "")
    {
        if (DialogResult.Yes == MessageBox.Show("Establesca la impresora a la que desea enviar la impresion como
predeterminada y presione en Aceptar.", "Atención", MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation))
        {
            CaptureScreen();
            printPantalla.Print();
        }
    }
}

void PrintPantallaPrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
{
    e.Graphics.DrawImage(memoryImage, 0, 0);
}

void TsblInsertarNuevoClick(object sender, EventArgs e)
{
    if(txtFW.Text != "")
    {
        LimpiarComponentes();
    }
    else
    {
        //MessageBox.Show("", "Mensaje", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

void CboDocumentoCodigoSelectedIndexChanged(object sender, EventArgs e)
{
    if(txtFW.Text != "")
    {
        if (cboDocumentoCodigo.SelectedValue.ToString() != "System.Data.DataRowView")
        {
            try
            {
                cboTipoDocumentos.SelectedIndex = cboDocumentoCodigo.SelectedIndex;
            }
            catch (SqlException error)
            {
                Excepciones.TratarErroresWindows(error);
            }
            catch (Exception error)
            {
                Excepciones.TratarErroresWindows(error);
            }
        }
    }
}

void CboEstadoOfiSelectedIndexChanged(object sender, EventArgs e)

```

```

{
    try
    {
        if(txtFW.Text != "")
        {
            CIOficios oficios = new CIOficios();
            oficios.OfiCodigo = Convert.ToInt32(cboFW.Text.Trim());
            oficios.OfiEstado = cboEstadoOfi.Text.Trim();
            oficios.ActualizarEstado(oficios);
        }
    }
    catch (SQLException error)
    {
        Excepciones.TratarErroresWindows(error);
    }
    catch (Exception error)
    {
        Excepciones.TratarErroresWindows(error);
    }
}

void CboTramiteCodigoSelectedIndexChanged(object sender, EventArgs e)
{
    if (cboTramiteCodigo.SelectedValue.ToString() != "System.Data.DataRowView")
    {
        try
        {
            cboTramites.SelectedIndex = cboTramiteCodigo.SelectedIndex;
        }
        catch (SQLException error)
        {
            Excepciones.TratarErroresWindows(error);
        }
        catch (Exception error)
        {
            Excepciones.TratarErroresWindows(error);
        }
    }
}

void TsbBuscarClick(object sender, EventArgs e)
{
    txtFW.Enabled = true;
    DesabilitarComponentes();
    LimpiarComponentes();
    txtFW.Focus();
    MessageBox.Show("Escriba el ID a buscar y presione
enter.", "Información", MessageBoxButtons.OK, MessageBoxIcon.Information);
}

void TxtFWEnter(object sender, EventArgs e)
{
    //MessageBox.Show("Enter.", "Información", MessageBoxButtons.OK, MessageBoxIcon.Information);
}

void TxtFWKeyPress(object sender, KeyPressEventArgs e)
{
    string externo;
    if (e.KeyChar == (char)(Keys.Enter))
    {
        txtFW.Enabled = false;
        CIOficios oficios = new CIOficios();
        dtDatosXML = oficios.SeleccionarPorExterno(txtFW.Text);
    }
}

```

```
if(this.BindingContext[dtDatosXML].Count != 0)
{
    externo = dtDatosXML.Rows[0]["EXT_CODIGO"].ToString();
    //MessageBox.Show(externo);
    if(externo == "0")
    {
        MostrarDatosEnVentana();
    }
    else
    {
        MostrarDatosEnVentanaExterno();
    }
}
}
```

ANEXO 6: MANUAL DE ADMINISTRACIÓN DEL SISTEMA

Manual de Administración

Esta documentación tiene como finalidad brindar al administrador una guía práctica, facilitando su comprensión, manejo y agilizando sus procesos.

Es recomendable que el digitador o usuario deba tener conocimientos básicos de computación, además de tener las precauciones necesarias para el uso del sistema, por lo cual se recomienda que antes de empezar a operar el sistema leer este manual.

INGRESO AL SISTEMA



Figura A5.1 Sistema Administración

Para ingresarnos al sistema Doc-Tram, en el escritorio de nuestro computador tenemos ícono indicado.

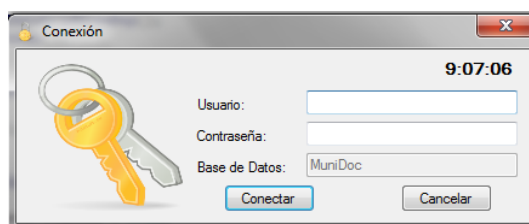


Figura A5.2 Ingreso

Luego de dar doble clic, ingresamos a una pantalla de identificación del usuario, donde debemos ingresar nuestro usuario, la clave de acceso y la cadena de la base de datos que esta proporcionada por defecto.

Luego de ingresar estos datos damos clic en Conectar y nos aparecerá la pantalla de menú principal del Sistema.

MENU PRINCIPAL

El menú principal está compuesto de cuatro grupos de opciones:

Seguridad, Parámetros, Movimientos y Consultas y Reportes, además de los básicos como Archivo y Ayuda.

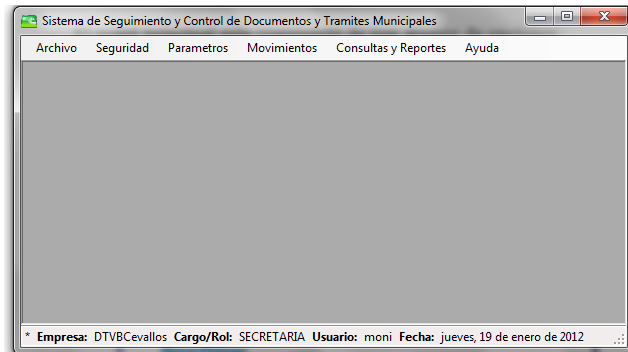














Figura A5.3 Pantalla principal

ICONOS DE BARRA DE HERRAMIENTAS

Para una mejor y rápida aplicación de las opciones, debemos utilizar los iconos establecidos en la Barra de Herramientas ubicados en la parte superior de cada una de las pantallas que conforman el sistema.

Cada Icono realiza una función diferente dependiendo de la pantalla en la que nos encontremos, los más destacados son:

-  Para salir de una ventana.
-  Nuevo registro
-  Cancelar
-  Procesar
-  Imprimir Ventana
-  Información de Ventana
-  Grabar
-  Refrescar

-  Buscar
-  Cerrar
-  Eliminar
-  Editar
- Entre otros que al situarse en ellos podrá leer su función.

SEGURIDAD

Opción del sistema que permite al usuario administrador hacer un control de usuarios, roles y accesos al sistema.

Cambio de Clave

Nos permite realizar el cambio de clave de manera muy fácil según el usuario que este logueado.



Figura A5.4 Cambio de Clave

1. Digitar en la caja de texto la clave anterior del usuario logueado.
2. Digitar en la caja de texto la clave nueva y en la caja de texto siguiente confirmar la clave nueva digitada.
3. Hacer clic en el botón guardar de la barra de herramientas.

Cambio de Clave en Lote

El usuario administrador tiene acceso a esta opción para poder realizar el cambio de clave a cualquier usuario que lo solicite ya sea por olvido o cambio involuntario.

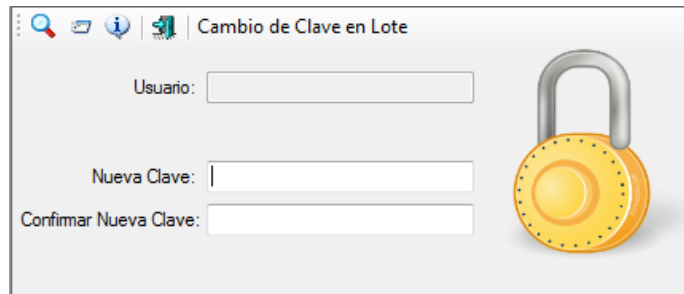


Figura A5.5 Cambio de Clave en Lote

1. Buscamos al usuario al cual queremos realizar el cambio de clave para lo cual hacemos clic el botón buscar ubicado en la barra de herramientas.
 - a. Se nos visualizara la siguiente ventana.

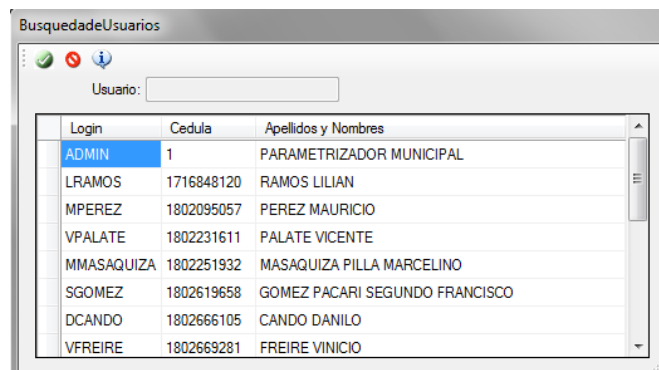


Figura A5.5.1 Búsqueda de Usuarios

- b. Aquí debemos ubicar el usuario al cual deseamos cambiar la clave
 - c. Una vez ubicado hacemos clic en el botón Aceptar de la barra de herramientas.
2. Ingresamos la nueva clave y la confirmamos en las cajas de texto correspondientes.

3. Hacer clic en el botón Procesar ubicado en la barra de herramientas.

Creación de Usuarios

Aquí se podrá crear a nuevos usuarios para que puedan utilizar el sistema.

Creación de Usuarios

Registro de Usuarios al Sistema

Usuario

Login: MSRE2012

Contraseña: ***** Confirmación: *****

Cedula: 1804042694

Apellidos: RAMOS ESPINOZA

Nombres: MARIA SOLEDAD

Email: maria@mail.com

Dirección: SANTO DOMINGO DE CEVALLOS

Teléfono: Celular: 091674422

Departamento

Selección Departamento: DPTO. JURIDICO

- I. MUNICIPIO DE CEVALLOS ADMINISTRATIVO
- FINANCIERO
- SISTEMAS
- UNIDAD DE AGUA POTABLE
- DPTO. JURIDICO
- AVALUOS Y CATASTRO
- PLANIFICACION
- BODEGA
- PRUEBA 1
- PRUEBA DE DEPARTAMENTO

Figura A5.6 Creación de Usuarios

1. Debemos ingresar toda la información necesaria de nuestro usuario.
2. Cuando se presenta un icono de color rojo al costado de una caja de texto indica que existe información incorrecta ingresada.
3. El numero de cedula debe de ser valido ya que el sistema verifica dicho campo.
4. Para buscar el departamento en el cual trabaja un usuario debemos seleccionarlo de la caja combo Selección Departamentos.

5. Una vez ingresada toda la información de manera correcta la guardamos haciendo clic en el botón Guardar de la barra de herramientas.

Permisos de Usuarios

En esta opción se podrá asignar un rol a un usuario creado o cambiar de rol a uno ya asignado.

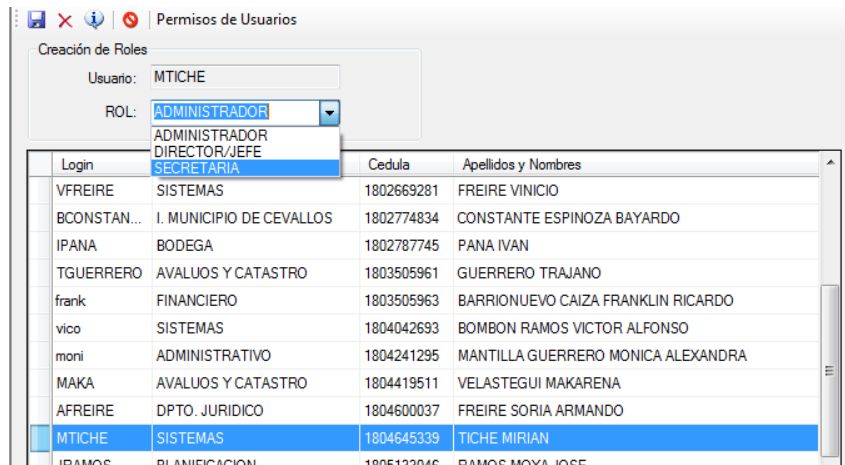


Figura A5.7 Permisos de Usuarios

1. Ubicamos en la tabla a un usuario al que se debes asignar a un rol.
2. Seleccionamos el rol de la caja combo.
3. En la parte superior nos ubicamos en el botón guardar y listo.
4. Si queremos quitar un rol a un usuario, seleccionamos el usuario y hacemos clic en el Botón Quitar Rol de Usuario Seleccionado.

Actualizar o Eliminar

Para actualizar información de un usuario o eliminarlo ingresamos en esta opción.

Usuarios

Datos del Usuario

Login: LRAMOS

Cedula: 1716848120

Nombres: LILIAN

Apellidos: RAMOS

Email: lilian_ramos@cevallos.com

Dirección: CEVALLOS

Teléfono: 099887655 Celular: 0

Departamentos: ADMINISTRATIVO

Figura A5.8 Actualización y Eliminación de Usuarios

1. Para poder editar o eliminar a un usuario lo debemos buscar mediante el botón Buscar.
 - a. Se nos visualizara la siguiente ventana.

Busqueda de Usuarios

Usuario:

Login	Cedula	Apellidos y Nombres
ADMIN	1	PARAMETRIZADOR MUNICIPAL
LRAMOS	1716848120	RAMOS LILIAN
MPEREZ	1802095057	PEREZ MAURICIO
VPALATE	1802231611	PALATE VICENTE
MMAQUIZA	1802251932	MASAQUIZA PILLA MARCELINO
SGOMEZ	1802619658	GOMEZ PACARI SEGUNDO FRANCISCO
DCANDO	1802666105	CANDO DANILO
VFREIRE	1802669281	FREIRE VINICIO

Figura A5.8.1 Búsqueda de Usuarios

- b. Aquí debemos ubicar el usuario al cual deseamos cambiar la clave
 - c. Una vez ubicado hacemos clic en el botón Aceptar de la barra de herramientas.

2. Si queremos editar el usuario seleccionado hacemos clic en el botón Editar y se desbloquearan ciertas casillas de texto que pueden ser actualizadas como son: Email, Dirección, Teléfonos y Departamento en el cual trabaja.
3. Luego de lo cual podemos hacer clic en el botón Guardar para almacenar los cambios.
4. Si queremos eliminar un usuario, primero lo debemos buscar y posteriormente hacemos clic el botón Eliminar.

Opciones del Usuario

En esta opción podemos seleccionar o asignar opciones adicionales a usuarios creados con rol o quitarle ciertas opciones si este es el caso.

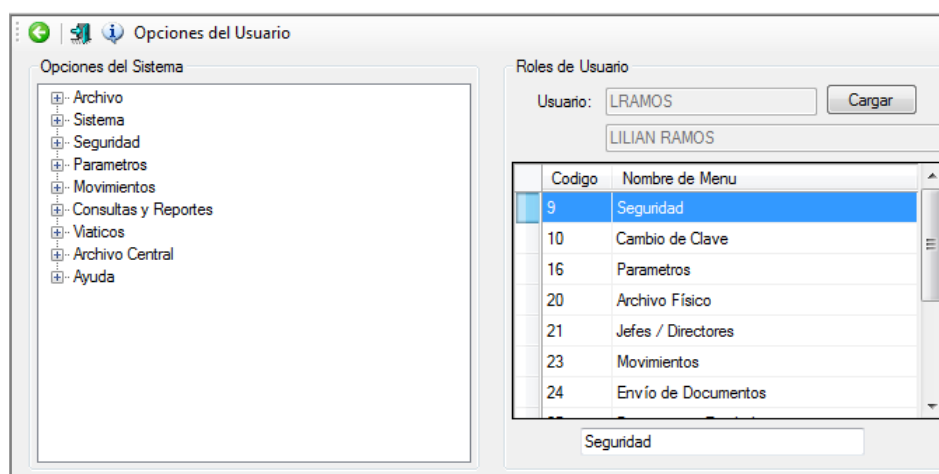


Figura A5.9 Opciones del Usuario

1. Para poder asignar o quitar opciones de menú primero debemos seleccionar un usuario, haciendo clic en el botón Cargar.
 - a. Se nos visualizara la siguiente ventana.

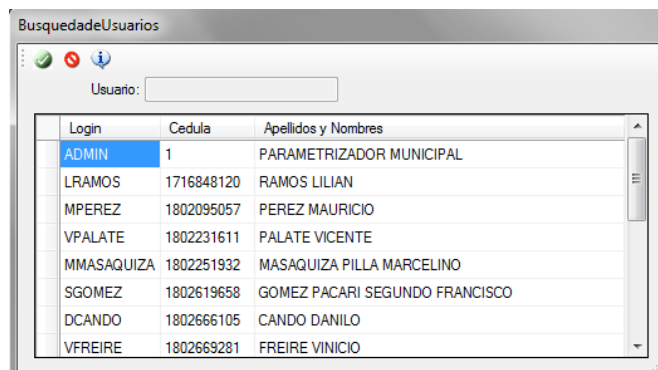


Figura A5.9.1 Búsqueda de Usuarios

- b. Aquí debemos ubicar el usuario al cual deseamos cambiar la clave
 - c. Una vez ubicado hacemos clic en el botón Aceptar de la barra de herramientas.
2. Una vez seleccionado el usuario, se visualizara una tabla con la información de las opciones de menú que tenga el usuario.
 3. Si queremos quitar uno, debemos seleccionarlo de la tabla y hacer clic en el botón Quitar Opción de la barra de herramientas.
 4. Si queremos asignar una nueva opción lo hacemos del árbol de opciones, haciendo doble clic en la opción a asignar.

PARAMETROS


Opción del sistema que permite parametrizar campos o valores que posteriormente serán utilizados conforme se va alimentando de datos las aplicaciones.

Departamentos

En esta opción podemos crear y editar un Departamento además de poder ver los funcionarios asignados a mismo.



Figura A5.10 Departamentos

1. Para crear un nuevo Departamento hacemos clic en el botón Nuevo de la barra de herramientas y se desbloquean las cajas de texto para poder ingresar la información de nuestro nuevo departamento.
2. Para asignar un Jefe o Director hacer clic en el botón .
 - a. Se visualiza la siguiente ventana.

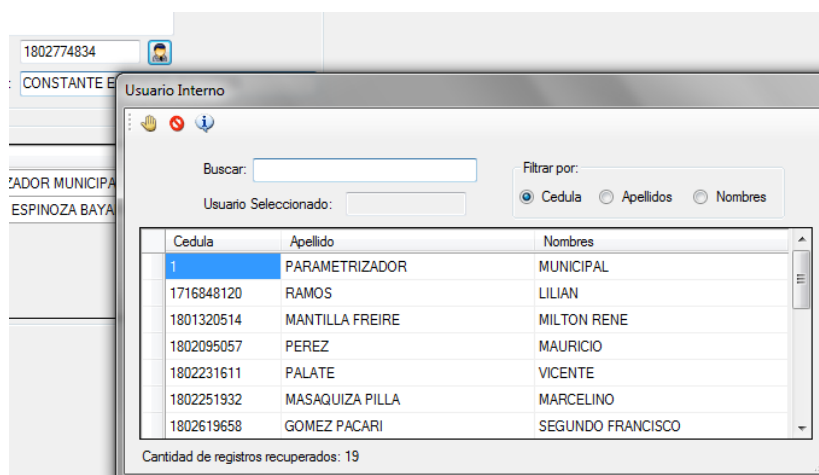
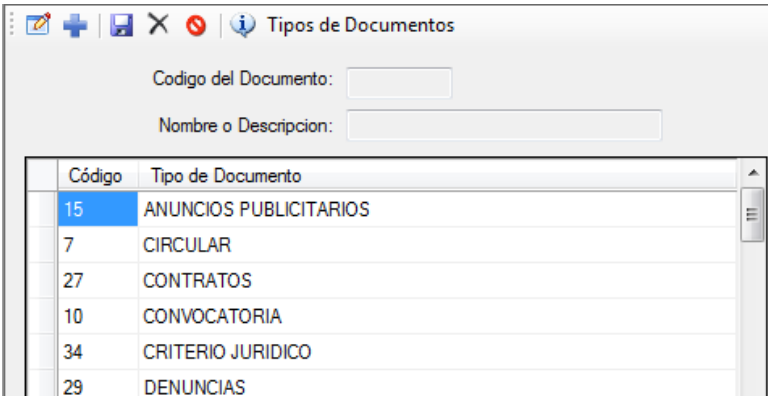


Figura A5.10.1 Usuarios Externos

- b. Aquí podemos ubicar al usuario que queremos asignarlo como jefe departamental.
 - c. Podemos filtrar la búsqueda por cedula, apellido o nombre.
 - d. Una vez seleccionado un usuario hacemos clic en aceptar y listo.
3. Una vez ingresado todos los datos para el nuevo departamento procedemos a grabarlo haciendo clic en el botón Guardar la de barra de herramientas.
 4. Para editar un departamento lo seleccionamos de la tabla y hacemos clic en el botón Editar.
 5. Una vez editado lo podemos grabar haciendo clic en el botón Guardar.
 6. Para eliminar un departamento primero no debe tener asignado ningún usuario, y podemos hacer clic en el botón Eliminar.

Tipos de Documentos

En esta opción podemos ver todos los tipos de documentos con los que trabaja en Municipio, podemos ingresar nuevos o editar los existentes.



Código	Tipo de Documento
15	ANUNCIOS PUBLICITARIOS
7	CIRCULAR
27	CONTRATOS
10	CONVOCATORIA
34	CRITERIO JURIDICO
29	DENUNCIAS

Figura A5.11 Tipo de Documentos

1. Para ingresar un nuevo Documento institucional hacemos clic en el botón Nuevo.
2. Se desbloquea la caja de texto en la cual podemos ingresar el nombre del documento institucional que estamos ingresando, el código se genera automáticamente.
3. Para grabarlo, hacemos clic en el botón de Guardar.
4. Si queremos Editar el nombre de uno ya creado, lo seleccionamos y hacemos clic en el botón editar.
5. Si queremos eliminar uno ya creado lo hacemos primero seleccionándolo de la lista y luego hacemos clic en el botón de eliminar.

Tipo de Trámites

En esta opción podemos ubicar e ingresar nuevos Trámites institucionales así como los documentos que lo habilitan.

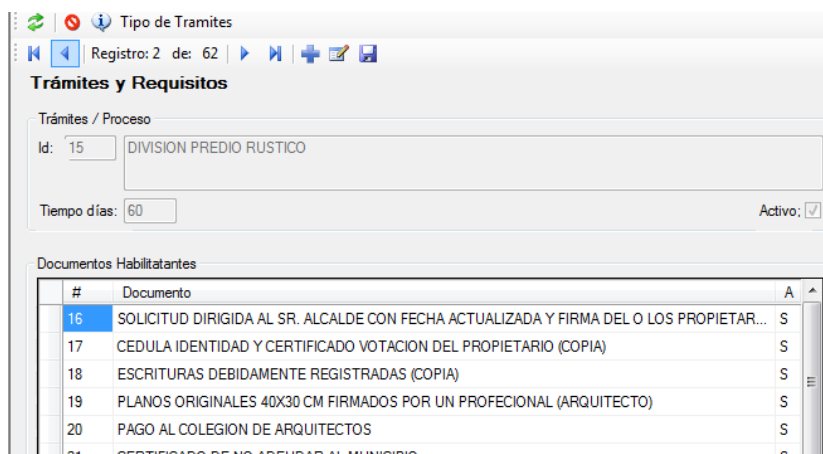


Figura A5.12 Tipo de Trámites

1. Para crear un nuevo Trámite hacemos clic en el botón nuevo de la barra de herramientas.

- a. Se visualiza una nueva ventada.

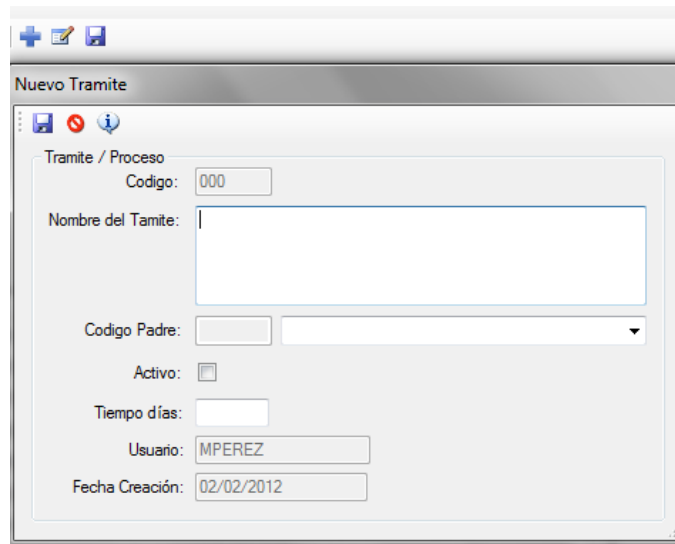


Figura A5.12.1 Nuevo Trámite

- b. Llenamos todos los campos y lo procedemos a guardar en el botón Grabar de la barra de herramientas.
 - c. Hay que tener muy en cuenta si se trata de un Trámite padre y depende de uno ya creado.
2. Para poder revisar los distintos trámites padres lo hacemos mediante los botones de navegación de la barra de herramientas.
 3. Para editar algún trámite debemos hacer clic en el botón Editar, esta desbloquea las cajas de texto y además para editar los trámites hijos debemos hacer clic en el trámite de la tabla.

Archivo Físico

Nos permite registrar la ubicación física, en donde se van a almacenar los documentos, para que su localización sea más ágil en caso de necesitarla.

Parámetros de archivado
Departamento / Sección: ADMINISTRATIVO

Tipo	Nombre	Estado	Año
ADM-1	EXPROPIACION	<input checked="" type="checkbox"/>	2004
ADM-10	AGRADECIMIENTOS	<input checked="" type="checkbox"/>	2010
ADM-1-1	ACUERDOS MINISTERIALES	<input checked="" type="checkbox"/>	2011
ADM-1-2	ACUERDOS MUNICIPALES	<input checked="" type="checkbox"/>	2011
ADM-4	GENERAL	<input checked="" type="checkbox"/>	2011
ADM-6	PRUEBAS	<input checked="" type="checkbox"/>	2011
ADM-1	EXPROPIACION	<input checked="" type="checkbox"/>	2004
ADM-2	GENERAL M	<input checked="" type="checkbox"/>	2004
		<input type="checkbox"/>	

Figura A5.13 Archivo Físico

1. Para ingresar una nueva ubicación física para archivar los documentos hacemos clic en el botón Nuevo.
 - a. Se visualiza la siguiente ventana en donde podemos ingresar toda la información de la nueva ubicación física o archivo.

Nuevo Archivo Físico

Ingreso de Nuevo Archivo Físico

Código:

Tipo:

Nombre:

Estado:

Archivo Padre:

Departamento:

Año:

Figura A5.13.1 Nuevo Archivo Físico

- b. El código no es necesario ya que se genera automáticamente.
2. Para editar debemos hacer clic en el botón editar y luego seleccionar de la lista en archivo.
 - a. Se visualiza la siguiente ventana con la información del archivo.

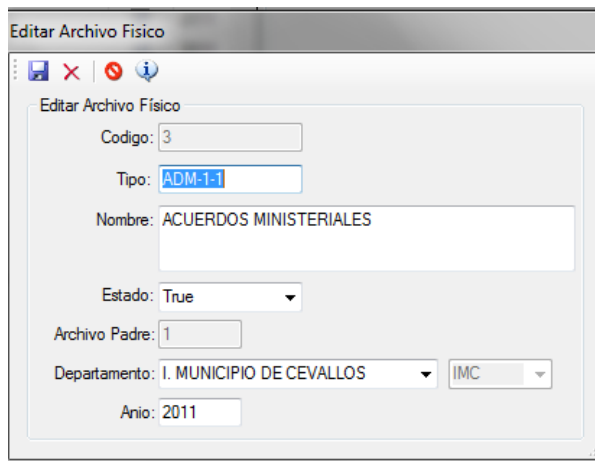


Figura 5.13.2 Editar Archivo Físico

- b. Una vez hecho esto guardamos los cambios.
- c. En esta ventana también podemos eliminar archivos mediante el botón Eliminar.

Jefes / Directores

Nos permite conocer a los diferentes jefes departamentales.

Departamento / Sección	Jefe / Director	Activo
I. MUNICIPIO DE CEVALLOS	CONSTANTE ESPINOZA BAYARDO	<input checked="" type="checkbox"/>
ADMINISTRATIVO	MANTILLA GUERRERO MONICA ALEXANDRA	<input checked="" type="checkbox"/>
FINANCIERO	BARRIONUEVO CAIZA FRANKLIN RICARDO	<input checked="" type="checkbox"/>
SISTEMAS	FREIRE VINICIO	<input checked="" type="checkbox"/>
UNIDAD DE AGUA POTABLE	PALATE VICENTE	<input checked="" type="checkbox"/>
DPTO. JURIDICO	FREIRE SORIA ARMANDO	<input checked="" type="checkbox"/>
AVALUOS Y CATASTRO	GUERRERO TRAJANO	<input checked="" type="checkbox"/>
PLANIFICACION	RAMOS MOYA JOSE	<input checked="" type="checkbox"/>
BODEGA	PANA IVAN	<input checked="" type="checkbox"/>
PRUEBA 1	RAMOS JULIAN	<input checked="" type="checkbox"/>

Figura A5.14 Jefes / Directores

MOVIMIENTOS

Opciones que nos permiten dar el mantenimiento a la base de datos, es decir alimentar nuestro sistema con información relacionada a la institución.

Envío de Documentos

Lo cual nos permitirá ingresarnos a la pantalla siguiente:

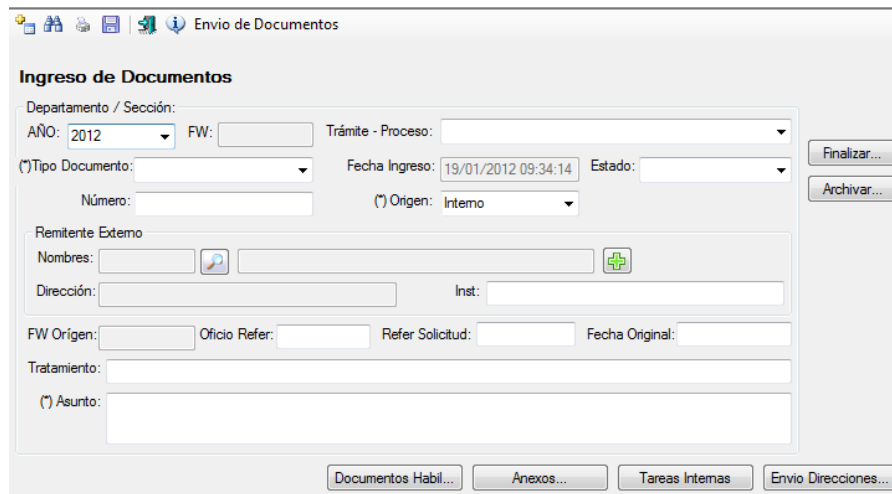


Figura A5.15 Envío de Documentos

En esta pantalla debemos ingresar los datos referentes a nuestro oficio, tomando en cuenta las siguientes consideraciones:

- (*) son campos obligatorios, es decir que deben ser llenados, caso contrario no podremos guardar el registro,
- **FW**: número de identificación del trámite, el sistema pone automáticamente.
- **Fecha Ingreso**, de igual manera no es ingresado por el usuario, coge la fecha y hora del sistema,
- **Estado**, el sistema lo interpreta dependiendo del tipo de movimiento que realiza el usuario,
- **Trámite-Proceso**, está directamente relacionado con la opción **Documentos Habilitantes** de la parte inferior de la pantalla, al escogemos un tipo de proceso, se habilita esta opción, y podemos revisar los documentos parametrizados para este tipo de proceso de la siguiente manera:

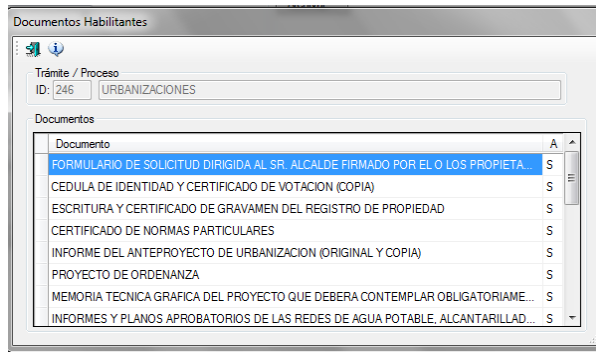


Figura A5.15.1 Documentos Habilitantes

- Remitente Externo**, cuando se trata de una solicitud externa que ingresa a una dependencia municipal, es necesario que ingresemos en nombre del remitente, con la finalidad de establecer quién es el responsable de la solicitud, para eso debemos dar un clic en el botón con una lupa para buscar el nombre del remitente, luego de lo cual nos aparece otra pantalla en donde podemos buscar al remitente ya sea por cedula, apellido o nombre y posteriormente dar clic en el botón buscar.

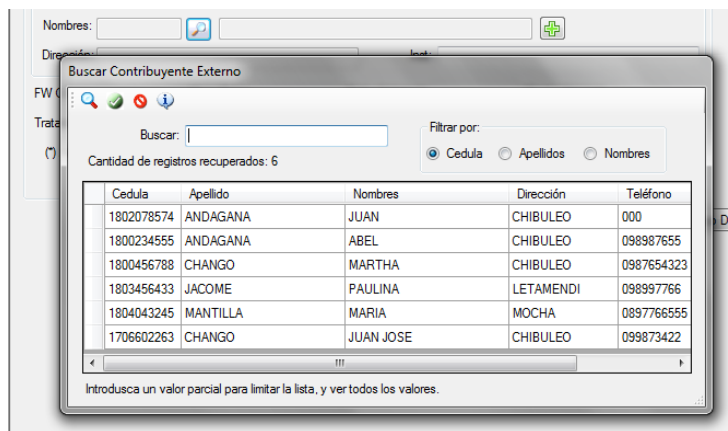


Figura A5.15.2 Buscar Contribuyentes Externos

Aquí debemos ubicar el remitente y presionar aceptar, para continuar con el ingreso de nuestro documento.

- **Campos de Referencia**, permiten hacer una referencia a un trámite o solicitud anterior.
- **Tratamiento**, es utilizado solamente para casos puntuales tales como: Favorable, Negado, Pendiente, etc. Aunque puede ser llenado según lo crea conveniente la persona que ingresa el documento.
- **Asunto**, en este campo debemos ingresar un breve resumen del contenido del oficio, tratando de ser lo más concreto posible.

Opción de Anexos

Al dar clic en la opción de anexos, aparece otra pantalla en donde debemos especificar los documentos que van anexados al trámite, en caso de poseerlos.

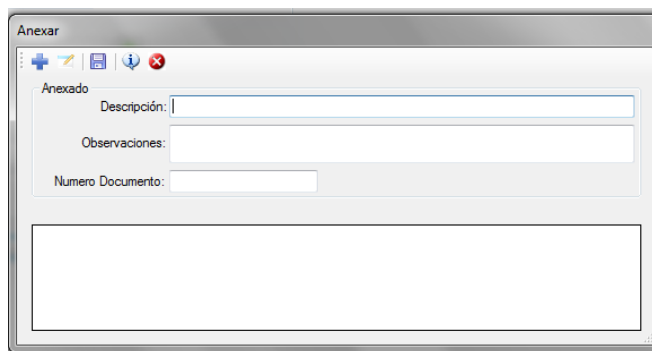


Figura A5.15.3 Anexos

Opción de Tareas Internas

Al dar clic en esta opción aparece la siguiente pantalla, en donde debemos especificar las tareas que están asignadas a cada uno de los funcionarios dentro del departamento o de la jefatura.

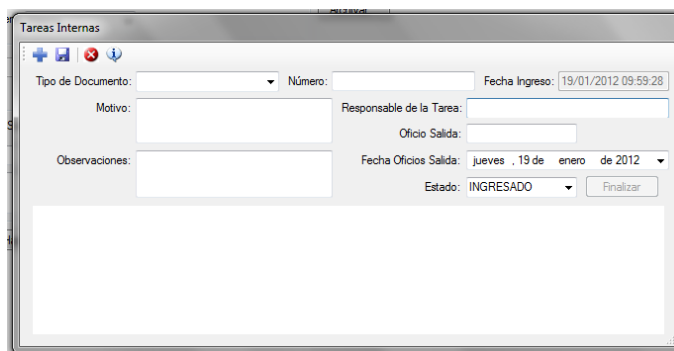


Figura A5.15.4 Tareas Internas

Opción Envío Direcciones

Para el envío de documentos a otras dependencias municipales damos clic en la opción.

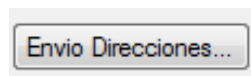


Figura A5.15.5 Botón de Envío

A continuación aparecerá la siguiente pantalla, en donde podemos seleccionar el departamento a donde vamos a enviar el documento, podemos indicar si es original o copia conjunta si es el caso, luego presionamos la opción enviar ubicado en la parte inferior de la pantalla.

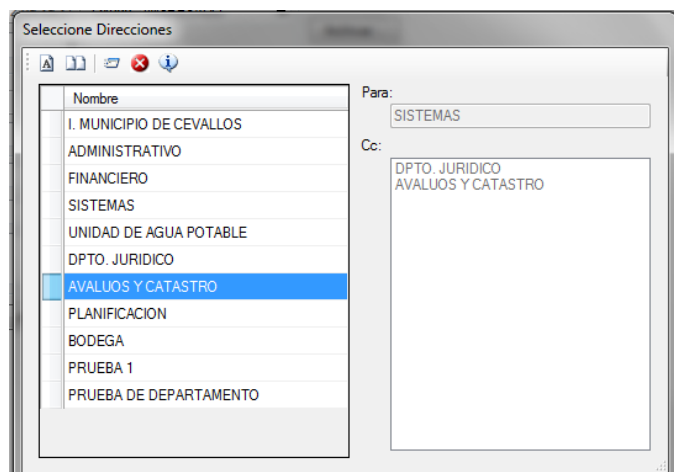


Figura A5.15.6 Selección de Direcciones

Opción Finalizar

Para determinar que el estado del trámite está finalizado, damos clic en la opción Finalizar, a continuación aparecerá una pantalla donde nos permite confirmar la acción.

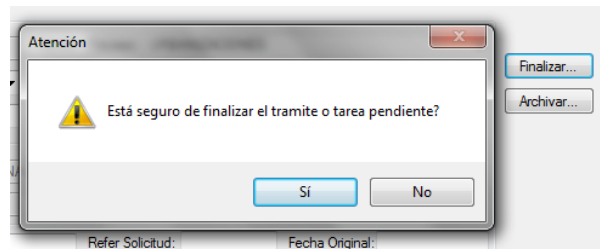


Figura A5.15.7 Opción Finalizar

Opción Archivar

Esta opción hace referencia a la ubicación física en donde vamos a guardar los documentos.

Al escoger esta opción aparece la siguiente ventana:

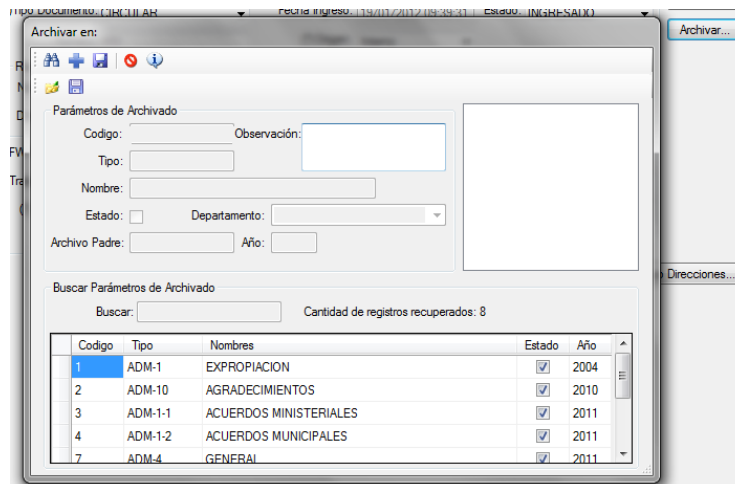
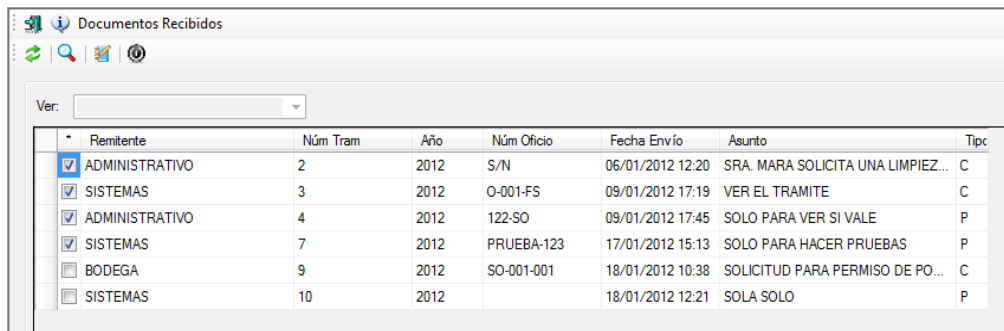


Figura A5.15.8 Archivar

Aquí podemos seleccionar el archivo físico en donde vamos a almacenar dicho trámite y poner una observación para posterior mente guardar, en esta ventana podemos también crear un nuevo archivo físico.

Documentos Recibidos

Luego aparece la siguiente pantalla con todos los documentos que nos han enviado a nuestro departamento o jefatura.



Ver:	Remitente	Núm Tram	Año	Núm Oficio	Fecha Envío	Asunto	Tipc
<input checked="" type="checkbox"/>	ADMINISTRATIVO	2	2012	S/N	06/01/2012 12:20	SRA. MARA SOLICITA UNA LIMPIEZ...	C
<input checked="" type="checkbox"/>	SISTEMAS	3	2012	O-001-FS	09/01/2012 17:19	VER EL TRAMITE	C
<input checked="" type="checkbox"/>	ADMINISTRATIVO	4	2012	122-SO	09/01/2012 17:45	SOLO PARA VER SI VALE	P
<input checked="" type="checkbox"/>	SISTEMAS	7	2012	PRUEBA-123	17/01/2012 15:13	SOLO PARA HACER PRUEBAS	P
<input type="checkbox"/>	BODEGA	9	2012	SO-001-001	18/01/2012 10:38	SOLICITUD PARA PERMISO DE PO...	C
<input type="checkbox"/>	SISTEMAS	10	2012		18/01/2012 12:21	SOLA SOLO	P

Figura A5.16 Documentos Recibidos

Si deseamos obtener el listado de trámites por departamento damos clic en la flecha de la parte superior izquierda, para lo cual primero debemos desbloquear y aparecerá un listado de los departamentos que existen en la municipalidad.

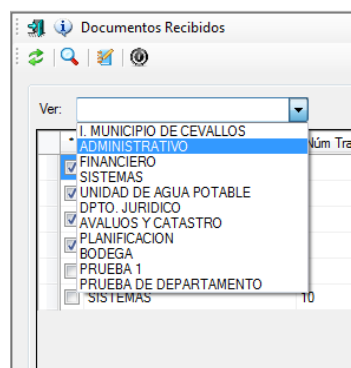


Figura A5.16.1 Documentos Recibidos 1

Cuando los registros recibidos aparecen acompañados con un visto, esto significa que ya lo hemos recibido.

Para receiptar un trámite que nos ha llegado damos doble clic alado del registro a receiptar.

<input checked="" type="checkbox"/>	SISTEMAS	7	2012	PRUEBA-123	17/01/2012 15:13	SOLO PARA HACER PRUEBAS	P
<input checked="" type="checkbox"/>	BODEGA	9	2012	SO-001-001	18/01/2012 10:38	SOLICITUD PARA PERMISO DE PO...	C
<input type="checkbox"/>	SISTEMAS	10	2012		18/01/2012 12:21	SOLA SOLO	P

Figura A5.16.2 Documentos Recibidos 2

A continuación aparecerá la siguiente pantalla con los datos del documento enviado, para que procedamos a continuar con el trámite respectivo.

Figura A5.16.3 Documentos Recibidos 3

En la pestaña derecha Tareas el sistema nos permite delegar tareas a funcionarios dentro del departamento o jefatura, para esto debemos seleccionar el tipo de documentos, ingresar el número de oficio, la fecha no es necesario el sistema automáticamente lo pone, En el campo de motivo podemos ingresar una breve descripción de la tarea a cumplir, para determinar el responsable de la tarea, damos un clic en el este campo y aparecerá una ventanita con todos los funcionarios municipales, en donde nosotros simplemente tenemos que escoger para luego proceder a guardar los cambio realizados, Una vez concluida la tarea procedemos a llenar los campos adicionales y por último a finalizar dicha tarea.

Opción Remitir

Esta opción nos permite remitir o delegar la atención del trámite a otro departamento o jefatura.

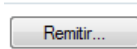


Figura A5.16.4 Botón Remitir

Luego aparecerá la siguiente pantalla con información del oficio original, aquí, si es el caso debemos ingresar datos adicionales, caso contrario guardamos y procedemos como si fuera un envío de documentación.

Envío de Documentos
Se encuentra: REMITIENDO

Ingreso de Documentos

Departamento / Sección:
AÑO: 2012 FW: Trámite - Proceso: AVALUOS

(*)Tipo Documento: Fecha Ingreso: 19/01/2012 11:21:44 Estado:
Número: (*) Origen: Interno

Remite Externo
Nombres: Dirección: Inst:

FW Origen: 7 Oficio Refer: PRUEBA-123 Refer Solicitud: Fecha Original: 12/01/2012 14:41

Tratamiento:
(*) Asunto:

Documentos Habil... Anexos... Tareas Internas Envío Direcciones...

Figura A5.16.5 Remitir

Documentos Finalizados

En esta opción podemos revisar todos los documentos que fueron enviados a nuestro departamento y ya fueron finalizados.

También los podemos revisar filtrados por departamento.

Ver:	Remitente	Núm Tram	Año	Núm Oficio	Fecha Envío	Asunto	Tipc
<input checked="" type="checkbox"/>	SISTEMAS	10	2012		18/01/2012 12:21	SOLA SOLO	P

Figura A5.17 Documentos Finalizados

Heredar Trámites

En esta ventana podemos heredar un trámite que depende de uno ya ingresado con anterioridad y fue creado como nuevo.

Documentos Anexos Archivado en

Código: 13 FW: 8 FW Original: 11
Año: 2012 Año: 2012

Tipo Documento: FACTURAS Origen: Interno

Asunto: SOLO PARA DAR PRUEBA DE REMITIDO

Remite: SISTEMAS

Responsable: BOMBON RAMOS VICTOR ALFONSO

F. Ingreso: 17/01/2012 15:14:10 Oficio: 11RR

Estado: FINALIZADO

Tratamiento: SEGUIR EL TRATAMIENTO

Observación:

Heredar Trámites

Año: 2012 FW: Código:
ID Padre: Estado:
Motivo:
Fecha Ingreso:

Figura A5.18 Heredar Trámite

1. Primero debemos buscar el trámite que deseamos heredar haciendo clic en el botón Buscar.
 - a. Esto visualizara la siguiente ventana.

Consulta por Trámite(s)

Busqueda

ID: 8 Año: 2012


Tipo de Documento: FACTURAS Fecha Ingreso: 17/01/2012 15:14:10

Número Oficio: 11RR Origen: Interno

Remitente ext:

Motivo: SOLO PARA DAR PRUEBA DE REMITIDO

Figura A15.18.1 Consulta de Trámites

- b. En la caja de texto ID digitamos el numero de tramite a heredar y hacemos clic en Buscar.
 - c. Si este es el trámite correcto a heredar hacemos clic en Aceptar.
2. En la parte de Heredar Trámite procedemos a buscar el trámite al que queremos heredar, la búsqueda se realiza seleccionando el año y digitando el FW de trámite y enter.
 3. Si es el trámite correcto hacemos clic en el botón  y se heredara el trámite.

Archivo Documentos

En esta pantalla podemos realizar el archivo de trámites cualquiera.

Archivo Documento

Archivo de Documentos

Departamento / Sección:

AÑO: 2012 FW: 8 Trámite - Proceso: AVALUOS Finalizar...

Tipo Documento: FACTURAS Fecha Ingreso: 17/01/2012 15:14:10 Estado: FINALIZADO Archivar

Número: 11RR Origen: Interno

Remitente Externo

Nombres:

Dirección: Inst:

FW Origen: 11 Oficio Refer: PRUEBA-123 Refer Solicitud: Fecha Original: 12/01/2012

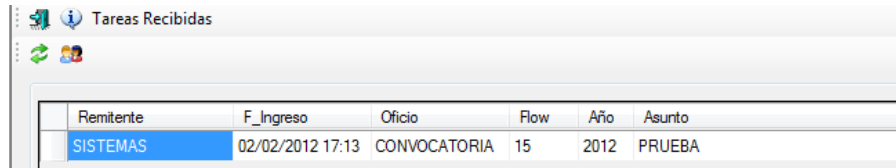
Tratamiento: SEGUIR EL TRATAMIENTO

Asunto: SOLO PARA DAR PRUEBA DE REMITIDO

Figura A5.19 Archivo Documentos

Tareas Recibidas

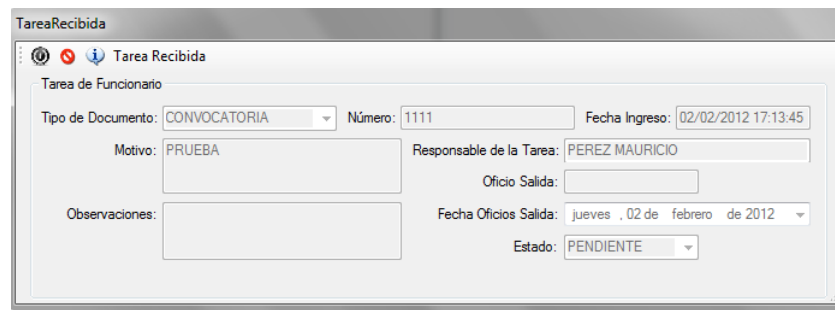
En esta opción podemos revisar los trámites asignados a este usuario específicamente como tarea interna.



Remitente	F_Ingreso	Oficio	Flow	Año	Asunto
SISTEMAS	02/02/2012 17:13	CONVOCATORIA	15	2012	PRUEBA

Figura A5.20 Tareas Recibidas

1. Al hacer doble clic en la tarea se visualiza la siguiente pantalla.



TareaRecibida

Tarea Recibida

Tarea de Funcionario

Tipo de Documento: CONVOCATORIA Número: 1111 Fecha Ingreso: 02/02/2012 17:13:45

Motivo: PRUEBA Responsable de la Tarea: PEREZ MAURICIO

Oficio Salida:

Observaciones:

Fecha Oficios Salida: jueves , 02 de febrero de 2012

Estado: PENDIENTE

Figura A5.20.1 Tareas Recibidas 1

2. Aquí podemos revisar lo que nos piden y proceder a finalizarlo mediante el botón de finalizar de la barra de herramientas.

Tareas Finalizadas

En esta opción podemos revisar las tareas que nos han ingresado y que ya se encuentran finalizadas.

Remitente	F_Ingreso	Oficio	Flow	Año	Asunto
SISTEMAS	02/02/2012 17:13	CONVOCATORIA	15	2012	PRUEBA

Figura A5.21 Tareas Finalizadas

CONSULTAS Y REPORTES

Grupo de opciones que nos permiten realizar consultas rápidas y veraces de un determinado trámite y su ubicación dentro de alguna dependencia municipal.

Hoja de Guía

Para consultar o imprimir una hoja de guía de la documentación enviada a los diversos departamentos ingresamos por esta opción.

Luego aparece la siguiente pantalla, en donde podemos parametrizar en tipo de guía y las fechas que se desea imprimir, luego damos clic en el botón de Ejecutar y si queremos podemos generar un PDF o Excel para imprimir.

Título del Reporte
Reporte de guía

Tipo de Guía: Tarea Departamento

Período: Desde: 2012-01-01 Hasta: 2012-01-23

Obtener guía solamente de documentos enviados a:
SISTEMAS

Vista:

Trámite	Oficio	Motivo	Destino	Fecha Ingreso	Remitente Externo
7	PRUEBA-123	SOLO PARA HACER PRUEBAS	SISTEMAS	12/01/2012 14:41	ANDAGANA ABEL

Figura A5.22 Hojas de Guía

A continuación desplegará en la pantalla la hoja de guía con la información, necesaria para despachar a los distintos departamentos si es el caso.

Documentos Enviados

Esta opción sirve para que podamos realizar una consulta de un determinado trámite y podamos observar a donde hemos enviado.

La búsqueda de un trámite lo podemos realizar por cualquier campo del encabezado del registro.

Tipo	Destino	Fecha Envío	Recibido	Fecha Recibido	Usuario	Estado
PARA	SISTEMAS	17/01/2012 15:13	RECIBITO	19/01/2012 10:21	vico	PENDIENTE
COPIA	I. MUNICIPIO DE CEVAL...	17/01/2012 15:13	PENDIENTE			PENDIENTE

Figura A5.23 Documentos Enviados

Envío General

Al ingresarnos a esta opción, aparece la siguiente pantalla en donde podemos parametrizar los campos de nuestra consulta, y podemos consultar todos los trámites enviados a los diferentes departamentos de la municipalidad.

Título	Unidad Destino	Remitente Externo	Fecha Inicial	Fecha Final
Hoja de Guía de la documentación			2012-01-23	2012-01-23

Figura A5.24 Envío General

Para buscar al remitente externo que queremos consultar, hacemos clic en la casilla remitente externo y se despliega la siguiente ventana en la cual ubicamos al remitente según los métodos de búsqueda.

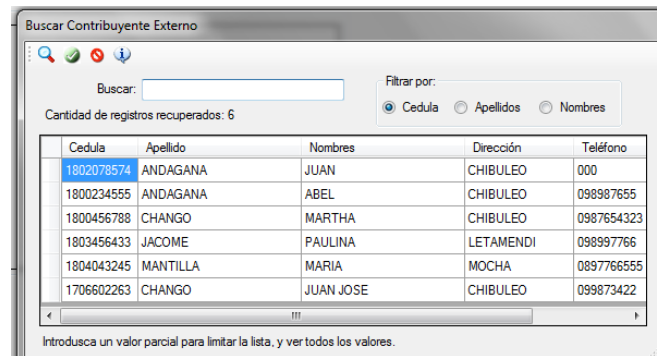


Figura A5.25 Búsqueda Contribuyente Externo

Si deseamos obtener la consulta de un determinado departamento damos un clic en el cuadro ubicado a la derecha de la Unidad Destino, entonces se despliega un listado de todos los departamentos existentes, lo que nos va a permitir escoger la dependencia deseada.

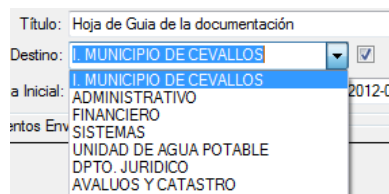


Figura A5.26 Selección Departamentos

Una vez que hemos ingresado los parámetros de nuestra consulta damos clic en la opción ejecutar, y nos desplegará un listado con la información solicitada.

Aquí podemos seleccionar el documento que deseamos ver el seguimiento y dar doble clic para que se despliegue la siguiente ventana en donde podemos ver los distintos documentos que se generaron e imprimir la hoja de guía.

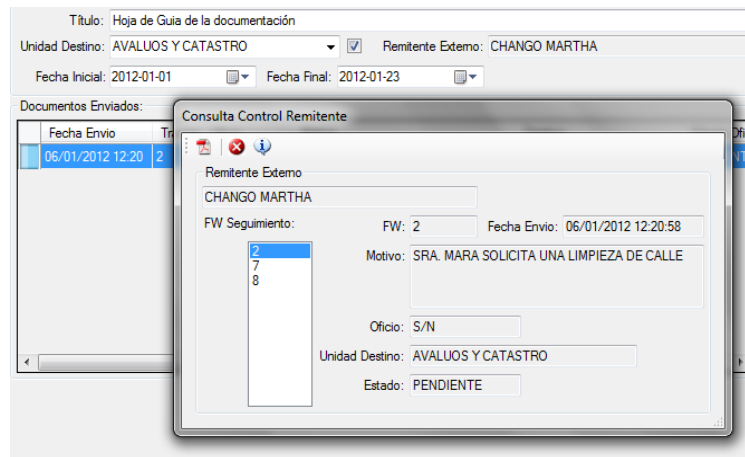


Figura A5.27 Consulta Control Remitente

GOBIERNO AUTONOMO DESCENTRALIZADO MUNICIPAL DEL CANTON CEVALLOS

lunes, enero 23 2012

CONTROL DEPARTAMENTAL DE DOCUMENTOS POR REMITENTE

Hoja de Guía solicitada

REMITENTE EXTERNO: CHANGO MARTHA

FW: 2
 Fecha de envío: 06/01/2012 12:20:58
 Numero de Tramite: S/N
 Motivo de Tramite: SRA. MARA SOLICITA UNA LIMPIEZA DE CALLE
 Departamento Destino: AVALUOS Y CATASTRO
 Estado de Tramite: PENDIENTE

FW: 7
 Fecha de envío: 17/01/2012 15:13:02
 Numero de Tramite: PRUEBA-123
 Motivo de Tramite: SOLO PARA HACER PRUEBAS
 Departamento Destino: SISTEMAS
 Estado de Tramite: PENDIENTE

FW: 8
 Fecha de envío: 17/01/2012 15:20:28
 Numero de Tramite: 11RR
 Motivo de Tramite: SOLO PARA DAR PRUEBA DE REMITIDO
 Departamento Destino: BODEGA
 Estado de Tramite: FINALIZADO

Figura A5.28 Hoja de Guía Solicitada

Consulta General

Al ingresar en esta opción aparece la siguiente pantalla:

Figura A5.29 Consulta General

Aquí podemos ubicar un trámite sin importar en que dependencia se encuentre.

Para activar la consulta presionamos Buscar, luego ingresamos la información que queremos buscar en el campo específico y posteriormente presionamos en la lupa, para que se ejecute la búsqueda.

Figura A5.30 Consulta de Trámites

Para observar el detalle de cada una de las dependencia por las cuales paso el trámite damos clic en la opción Aceptar y aparecerá la siguiente pantalla.

Consulta General

Trámites

Documento | Anexos | Archivo en | U.C.A. | Interesado

Seguimiento

Año: 2012 | FW: 8 | FW Original: 11 | Año: 2012

Días: 0

Tipo Documento: FACTURAS | Origen: Interno

Asunto: SOLO PARA DAR PRUEBA DE REMITIDO

Remite: SISTEMAS

Responsable: BOMBON RAMOS VICTOR ALFONSO

F. Ingreso: 17/01/2012 15:14:10 | Oficio: 11RR

Estado: FINALIZADO | Usuario env: wico

Tratamiento: SEGUIR EL TRATAMIENTO

Referencia

Ofic: PRUEBA-123

Solic:

Recibido

Recibido por: IPANA

Fecha recibido: 18/01/2012 12:02:41

FW Seguimiento:

8
7
2

Figura A5.31 Consulta General

En la parte izquierda presenta en forma general en número de FW del trámite, y en la parte derecha aparece el detalle con toda la información que fue generada en cada una de las dependencias por las cuales paso el trámite, hasta llegar a establecer su ubicación actual

Manual de Usuarios

Esta documentación tiene como finalidad brindar al usuario una guía práctica, facilitando su comprensión, manejo y agilizando sus procesos.

Es recomendable que el digitador o usuario deba tener conocimientos básicos de computación, además de tener las precauciones necesarias para el uso del sistema, por lo cual se recomienda que antes de empezar a operar el sistema leer este manual.

INGRESO AL SISTEMA



Figura A6.1 Sistema Usuario

Para ingresarnos al sistema Doc-Tram, en el escritorio de nuestro computador tenemos ícono indicado.

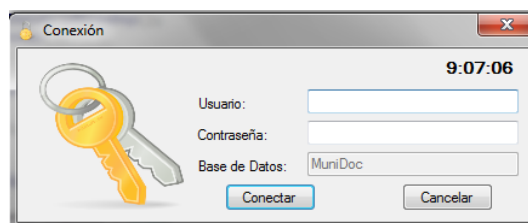


Figura A5.2 Conexión

Luego de dar doble clic, ingresamos a una pantalla de identificación del usuario, donde debemos ingresar nuestro usuario, la clave de acceso y la cadena de la base de datos que esta proporcionada por defecto.

Luego de ingresar estos datos damos clic en Conectar y nos aparecerá la pantalla de menú principal del Sistema.

MENU PRINCIPAL

El menú principal está compuesto de tres grupos de opciones:

Parámetros, Movimientos y Consultas y Reportes, además de los básicos como Archivo, Seguridad y Ayuda.

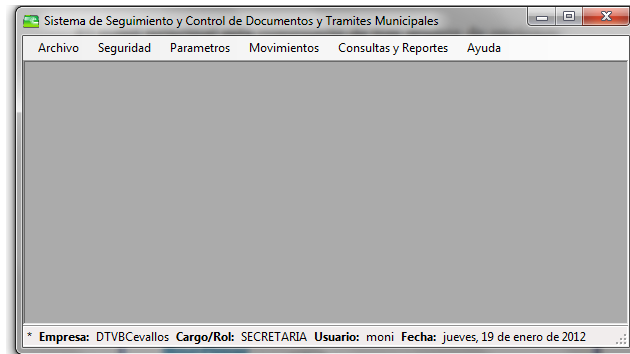














Figura A5.3 Ventana Principal

ICONOS DE BARRA DE HERRAMIENTAS

Para una mejor y rápida aplicación de las opciones, debemos utilizar los iconos establecidos en la Barra de Herramientas ubicados en la parte superior de cada una de las pantallas que conforman el sistema.

Cada Icono realiza una función diferente dependiendo de la pantalla en la que nos encontremos, los más destacados son:

-  Para salir de una ventana.
-  Nuevo registro
-  Cancelar
-  Procesar
-  Imprimir Ventana
-  Información de Ventana
-  Grabar
-  Refrescar
-  Buscar

-  Cerrar
 -  Eliminar
 -  Editar
- Entre otros que al situarse en ellos podrá leer su función.

PARAMETROS

Opción del sistema que permite parametrizar campos o valores que posteriormente serán utilizados conforme se va alimentando de datos las aplicaciones.

Archivo Físico

Nos permite registrar la ubicación física, en donde se van a almacenar los documentos, para que su localización sea más ágil en caso de necesitarla.



Tipo	Nombre	Estado	Año
ADM-1	EXPROPIACION	<input checked="" type="checkbox"/>	2004
ADM-10	AGRADECIMIENTOS	<input checked="" type="checkbox"/>	2010
ADM-1-1	ACUERDOS MINISTERIALES	<input checked="" type="checkbox"/>	2011
ADM-1-2	ACUERDOS MUNICIPALES	<input checked="" type="checkbox"/>	2011
ADM-4	GENERAL	<input checked="" type="checkbox"/>	2011
ADM-6	PRUEBAS	<input checked="" type="checkbox"/>	2011
ADM-1	EXPROPIACION	<input checked="" type="checkbox"/>	2004
ADM-2	GENERAL M	<input checked="" type="checkbox"/>	2004
		<input type="checkbox"/>	

Figura A6.4 Archivo Físico

Jefes / Directores

Nos permite conocer a los diferentes jefes departamentales.

Departamento / Sección	Jefe / Director	Activo
I. MUNICIPIO DE CEVALLOS	CONSTANTE ESPINOZA BAYARDO	<input checked="" type="checkbox"/>
ADMINISTRATIVO	MANTILLA GUERRERO MONICA ALEXANDRA	<input checked="" type="checkbox"/>
FINANCIERO	BARRIONUEVO CAIZA FRANKLIN RICARDO	<input checked="" type="checkbox"/>
SISTEMAS	FREIRE VINICIO	<input checked="" type="checkbox"/>
UNIDAD DE AGUA POTABLE	PALATE VICENTE	<input checked="" type="checkbox"/>
DPTO. JURIDICO	FREIRE SORIA ARMANDO	<input checked="" type="checkbox"/>
AVALUOS Y CATASTRO	GUERRERO TRAJANO	<input checked="" type="checkbox"/>
PLANIFICACION	RAMOS MOYA JOSE	<input checked="" type="checkbox"/>
BODEGA	PANA IVAN	<input checked="" type="checkbox"/>
PRUEBA 1	RAMOS JULIAN	<input checked="" type="checkbox"/>

Figura A6.5 Jefes / Directores

MOVIMIENTOS

Opciones que nos permiten dar el mantenimiento a la base de datos, es decir alimentar nuestro sistema con información relacionada a la institución.

Envío de Documentos

Lo cual nos permitirá ingresarnos a la pantalla siguiente:

Ingreso de Documentos

Departamento / Sección:

AÑO: 2012 FW: Trámite - Proceso:

(*) Tipo Documento: Fecha Ingreso: 19/01/2012 09:34:14 Estado:

Número: (*) Origen: Interno

Remitente Externo

Nombres:

Dirección: Inst:

FW Origen: Oficio Refer: Refer Solicitud: Fecha Original:

Tratamiento:

(*) Asunto:

Documentos Habil... Anexos... Tareas Internas Envío Direcciones...

Figura A6.6 Envío de Documentos

En esta pantalla debemos ingresar los datos referentes a nuestro oficio, tomando en cuenta las siguientes consideraciones:

- (*) son campos obligatorios, es decir que deben ser llenados, caso contrario no podremos guardar el registro,
- **FW**: número de identificación del trámite, el sistema pone automáticamente.
- **Fecha Ingreso**, de igual manera no es ingresado por el usuario, coge la fecha y hora del sistema,
- **Estado**, el sistema lo interpreta dependiendo del tipo de movimiento que realiza el usuario,
- **Trámite-Proceso**, está directamente relacionado con la opción **Documentos Habilitantes** de la parte inferior de la pantalla, al escogemos un tipo de proceso, se habilita esta opción, y podemos revisar los documentos parametrizados para este tipo de proceso de la siguiente manera:

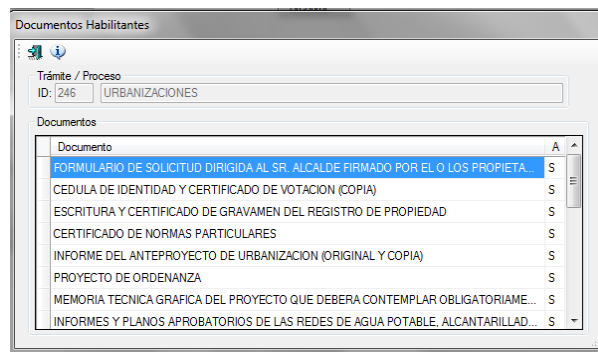


Figura A6.7 Documentos Habilitantes

- **Remitente Externo**, cuando se trata de una solicitud externa que ingresa a una dependencia municipal, es necesario que ingresemos en nombre del remitente, con la finalidad de establecer quién es el responsable de la solicitud, para eso debemos dar un clic en el botón con una lupa para buscar el nombre del remitente, luego de lo cual nos aparece otra pantalla en donde podemos buscar al remitente ya sea por cedula, apellido o nombre y posteriormente dar clic en el botón buscar.

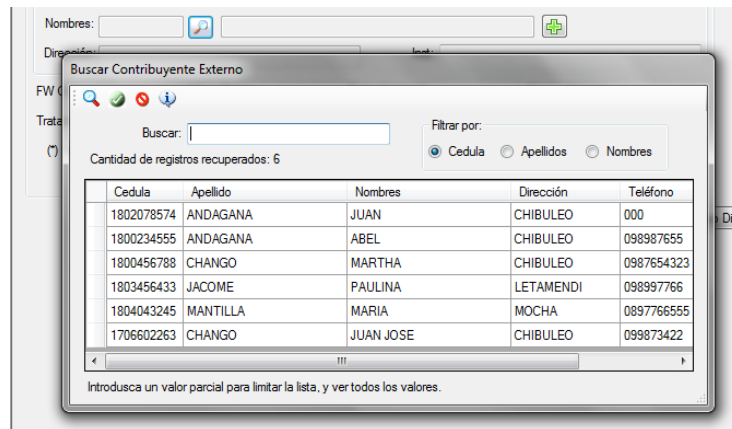


Figura A6.8 Buscar Contribuyente Externo

Aquí debemos ubicar el remitente y presionar aceptar, para continuar con el ingreso de nuestro documento.

- **Campos de Referencia**, permiten hacer una referencia a un trámite o solicitud anterior.
- **Tratamiento**, es utilizado solamente para casos puntuales tales como: Favorable, Negado, Pendiente, etc. Aunque puede ser llenado según lo crea conveniente la persona que ingresa el documento.
- **Asunto**, en este campo debemos ingresar un breve resumen del contenido del oficio, tratando de ser lo más concreto posible.

Opción de Anexos

Al dar clic en la opción de anexos, aparece otra pantalla en donde debemos especificar los documentos que van anexados al trámite, en caso de poseerlos.

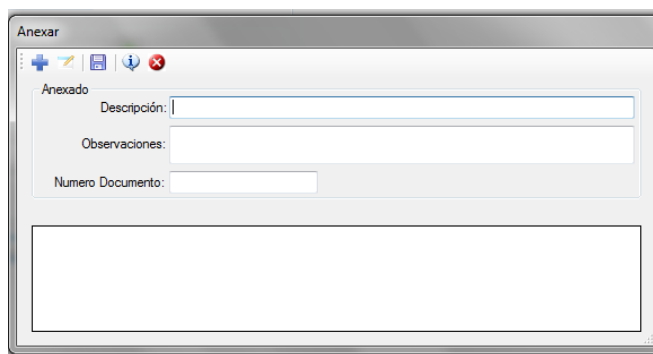


Figura A6.9 Anexar

Opción de Tareas Internas

Al dar clic en esta opción aparece la siguiente pantalla, en donde debemos especificar las tareas que están asignadas a cada uno de los funcionarios dentro del departamento o de la jefatura.

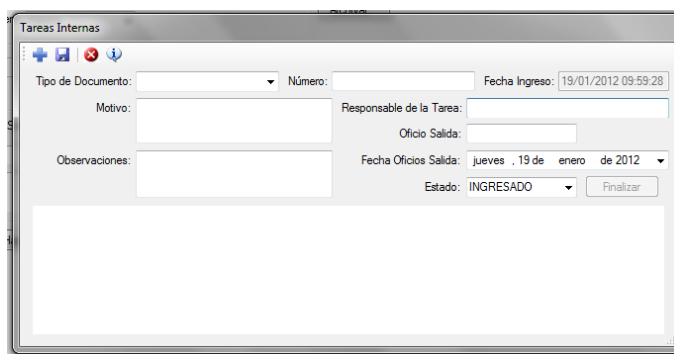


Figura A6.10 Tareas Internas

Opción Envío Direcciones

Para el envío de documentos a otras dependencias municipales damos clic en la opción.

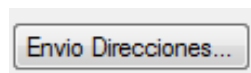


Figura A6.11 Botón Envío

A continuación aparecerá la siguiente pantalla, en donde podemos seleccionar el departamento a donde vamos a enviar el documento, podemos indicar si es original o copia conjunta si es el caso, luego presionamos la opción enviar ubicado en la parte inferior de la pantalla.

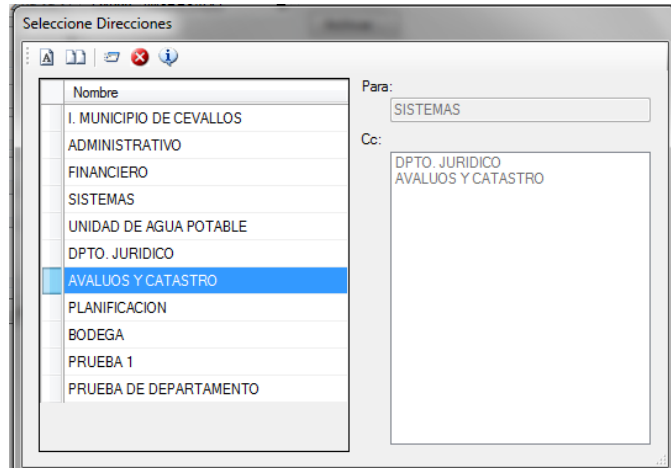


Figura A6.12 Selección Direcciones

Opción Finalizar

Para determinar que el estado del trámite está finalizado, damos clic en la opción Finalizar, a continuación aparecerá una pantalla donde nos permite confirmar la acción.

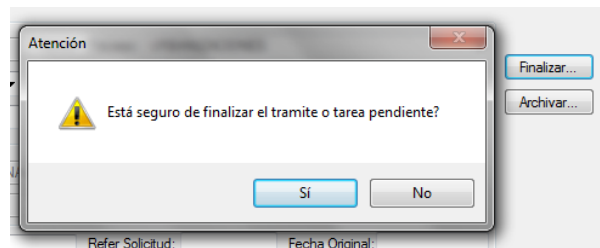


Figura A6.13 Opción Finalizar

Opción Archivar

Esta opción hace referencia a la ubicación física en donde vamos a guardar los documentos.

Al escoger esta opción aparece la siguiente ventana:

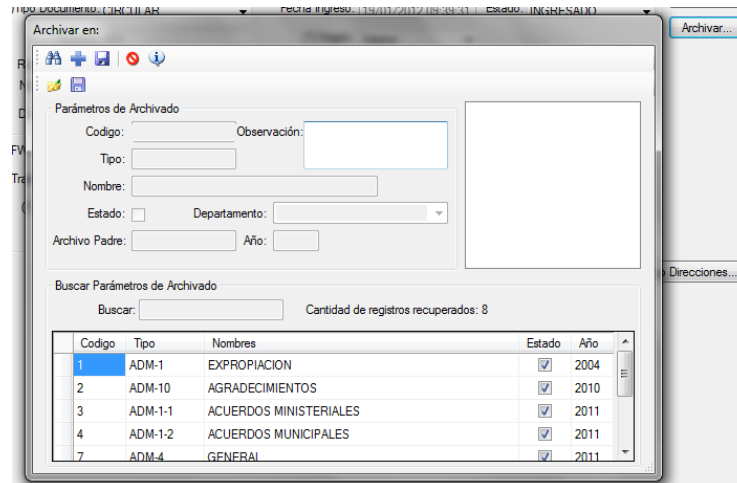


Figura A6.14 Archivar

Aquí podemos seleccionar el archivo físico en donde vamos a almacenar dicho trámite y poner una observación para posterior mente guardar, en esta ventana podemos también crear un nuevo archivo físico.

Documentos Recibidos

Luego aparece la siguiente pantalla con todos los documentos que nos han enviado a nuestro departamento o jefatura.

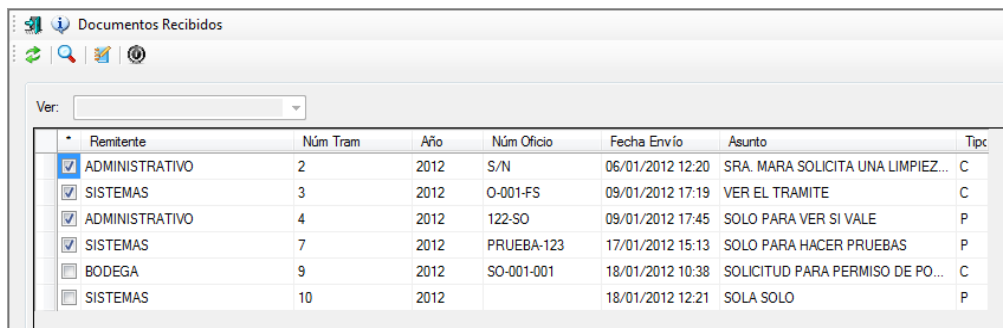


Figura A6.15 Documentos Recibidos

Si deseamos obtener el listado de trámites por departamento damos clic en la flecha de la parte superior izquierda, para lo cual primero debemos desbloquear y aparecerá un listado de los departamentos que existen en la municipalidad.

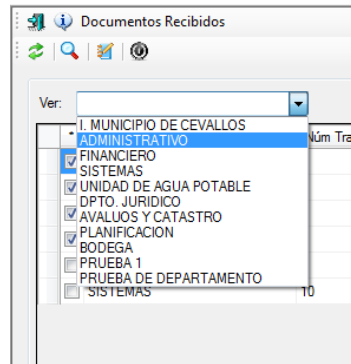


Figura A6.16 Documentos Recibidos 1

Cuando los registros recibidos aparecen acompañados con un visto, esto significa que ya lo hemos recibido.

Para receptor un trámite que nos ha llegado damos doble clic alado del registro a receptor.

<input checked="" type="checkbox"/>	SISTEMAS	7	2012	PRUEBA-123	17/01/2012 15:13	SOLO PARA HACER PRUEBAS	P
<input checked="" type="checkbox"/>	BODEGA	9	2012	SO-001-001	18/01/2012 10:38	SOLICITUD PARA PERMISO DE PO...	C
<input type="checkbox"/>	SISTEMAS	10	2012		18/01/2012 12:21	SOLA SOLO	P

Figura A6.17 Documentos Recibidos 2

A continuación aparecerá la siguiente pantalla con los datos del documento enviado, para que procedamos a continuar con el trámite respectivo.

Figura A6.18 Documento Recibido

En la pestaña derecha Tareas el sistema nos permite delegar tareas a funcionarios dentro del departamento o jefatura, para esto debemos seleccionar el tipo de documentos, ingresar el número de oficio, la fecha no es necesario el sistema automáticamente lo pone, En el campo de motivo podemos ingresar una breve descripción de la tarea a cumplir, para determinar el responsable de la tarea, damos un clic en el este campo y aparecerá una ventanita con todos los funcionarios municipales, en donde nosotros simplemente tenemos que escoger para luego proceder a guardar los cambio realizados, Una vez concluida la tarea procedemos a llenar los campos adicionales y por último a finalizar dicha tarea.

Opción Remitir

Esta opción nos permite remitir o delegar la atención del trámite a otro departamento o jefatura.

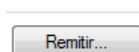


Figura A6.19 Botón Remitir

Luego aparecerá la siguiente pantalla con información del oficio original, aquí, si es el caso debemos ingresar datos adicionales, caso contrario guardamos y procedemos como si fuera un envío de documentación.

Figura A6.20 Remitir

CONSULTAS Y REPORTE

Grupo de opciones que nos permiten realizar consultas rápidas y veraces de un determinado trámite y su ubicación dentro de alguna dependencia municipal.

Hoja de Guía

Para consultar o imprimir una hoja de guía de la documentación enviada a los diversos departamentos ingresamos por esta opción.

Luego aparece la siguiente pantalla, en donde podemos parametrizar en tipo de guía y las fechas que se desea imprimir, luego damos clic en el botón de Ejecutar y si queremos podemos generar un PDF o Excel para imprimir.

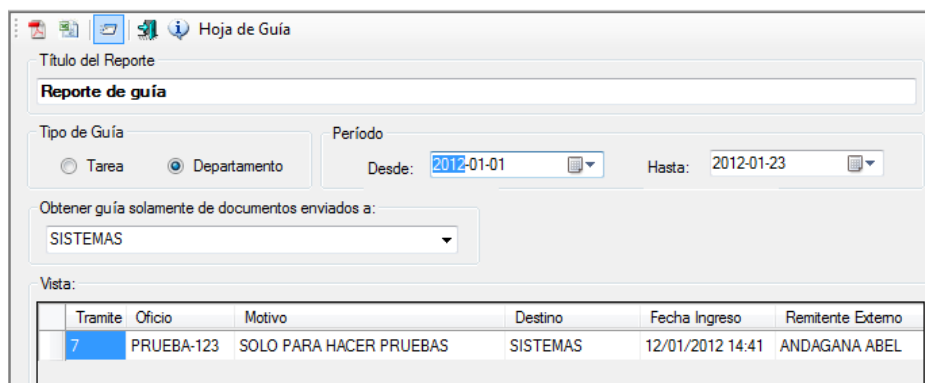


Figura A6.21 Hoja de Guía

A continuación desplegara en la pantalla la hoja de guía con la información, necesaria para despachar a los distintos departamentos si es el caso.

Documentos Enviados

Esta opción sirve para que podamos realizar una consulta de un determinado trámite y podamos observar a donde hemos enviado.

La búsqueda de un trámite lo podemos realizar por cualquier campo del encabezado del registro.

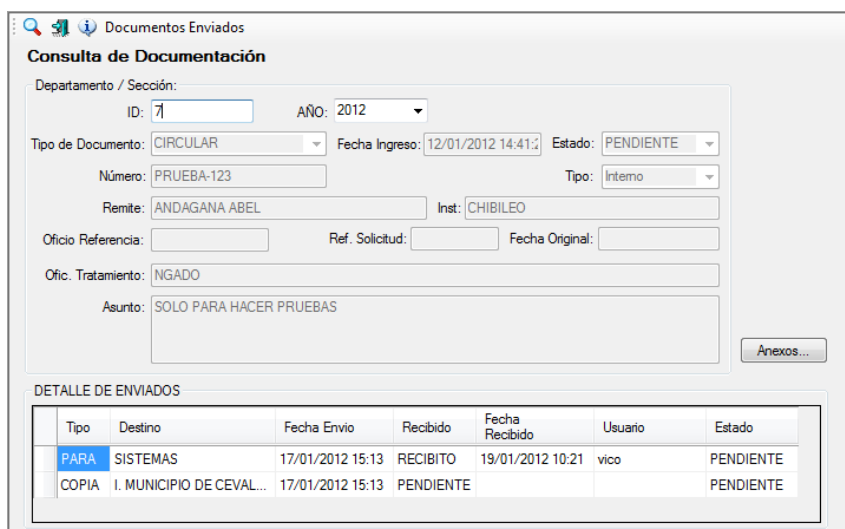
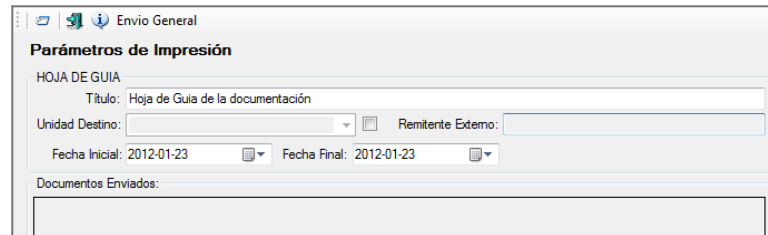


Figura A6.22 Documentos Enviados

Envío General

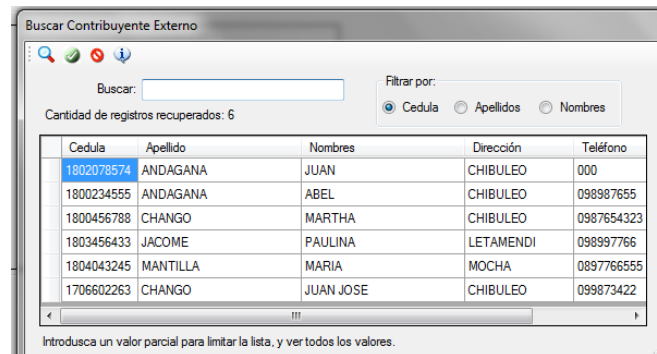
Al ingresarnos a esta opción, aparece la siguiente pantalla en donde podemos parametrizar los campos de nuestra consulta, y podemos consultar todos los trámites enviados a los diferentes departamentos de la municipalidad.



The screenshot shows a web application window titled 'Envío General'. Below the title bar is a section titled 'Parámetros de Impresión'. It contains several input fields: 'HOJA DE GUIA' with a title 'Hoja de Guia de la documentación', 'Unidad Destino:' with a dropdown menu, 'Remitente Externo:' with a text input field, 'Fecha Inicial:' set to '2012-01-23', and 'Fecha Final:' set to '2012-01-23'. Below these fields is a section labeled 'Documentos Enviados:' which is currently empty.

Figura A6.23 Envío General

Para buscar al remitente externo que queremos consultar, hacemos clic en la casilla remitente externo y se despliega la siguiente ventana en la cual ubicamos al remitente según los métodos de búsqueda.



The screenshot shows a search window titled 'Buscar Contribuyente Externo'. It features a search input field, a 'Filtrar por:' section with radio buttons for 'Cedula', 'Apellidos', and 'Nombres', and a table of search results. The table has columns for 'Cedula', 'Apellido', 'Nombres', 'Dirección', and 'Teléfono'. Below the table is a pagination bar and a note: 'Introduzca un valor parcial para limitar la lista, y ver todos los valores.'

Cedula	Apellido	Nombres	Dirección	Teléfono
1802078574	ANDAGANA	JUAN	CHIBULEO	000
1800234555	ANDAGANA	ABEL	CHIBULEO	098987655
1800456789	CHANGO	MARTHA	CHIBULEO	0987654323
1803456433	JACOME	PAULINA	LETAMENDI	098997766
1804043245	MANTILLA	MARIA	MOCHA	0897766555
1706602263	CHANGO	JUAN JOSE	CHIBULEO	099873422

Figura A6.24 Buscar Contribuyente Externo

Si deseamos obtener la consulta de un determinado departamento damos un clic en el cuadrado ubicado a la derecha de la Unidad Destino, entonces se despliega un listado de todos los departamentos existentes, lo que nos va a permitir escoger la dependencia deseada.

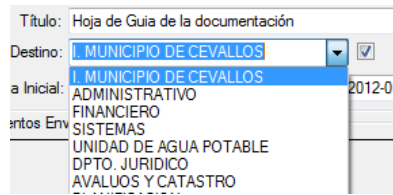


Figura A6.25 Búsqueda Destino

Una vez que hemos ingresado los parámetros de nuestra consulta damos clic en la opción ejecutar, y nos desplegará un listado con la información solicitada.

Aquí podemos seleccionar el documento que deseamos ver el seguimiento y dar doble clic para que se despliegue la siguiente ventana en donde podemos ver los distintos documentos que se generaron e imprimir la hoja de guía.

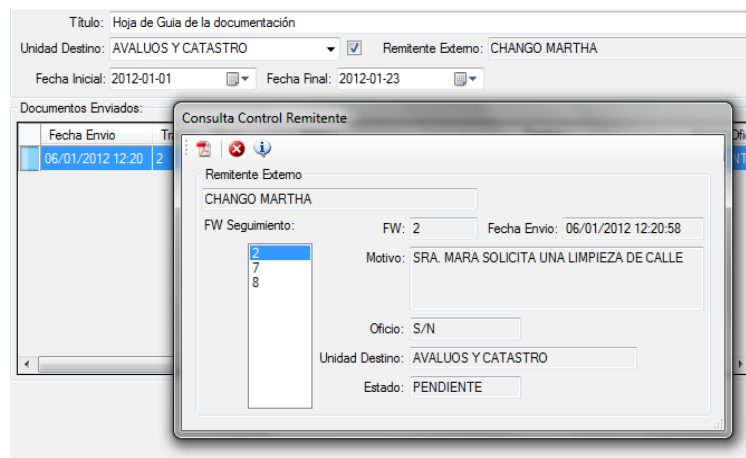


Figura A6.26 Consulta Control Remitente

Hoja de Guía solicitada

REMITENTE EXTERNO: CHANGO MARTHA

FW: 2
 Fecha de envío: 06/01/2012 12:20:58
 Numero de Tramite: S/N
 Motivo de Tramite: SRA. MARA SOLICITA UNA LIMPIEZA DE CALLE
 Departamento Destino: AVALUOS Y CATASTRO
 Estado de Tramite: PENDIENTE

FW: 7
 Fecha de envío: 17/01/2012 15:13:02
 Numero de Tramite: PRUEBA-123
 Motivo de Tramite: SOLO PARA HACER PRUEBAS
 Departamento Destino: SISTEMAS
 Estado de Tramite: PENDIENTE

FW: 8
 Fecha de envío: 17/01/2012 15:20:28
 Numero de Tramite: 11RR
 Motivo de Tramite: SOLO PARA DAR PRUEBA DE REMITIDO
 Departamento Destino: BODEGA
 Estado de Tramite: FINALIZADO

Figura A6.27 Hoja de Guía Solicitada

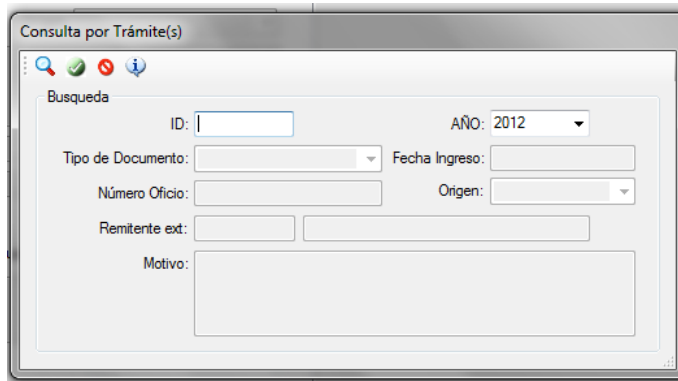
Consulta General

Al ingresar en esta opción aparece la siguiente pantalla:

Figura A6.28 Consulta General

Aquí podemos ubicar un trámite sin importar en que dependencia se encuentre.

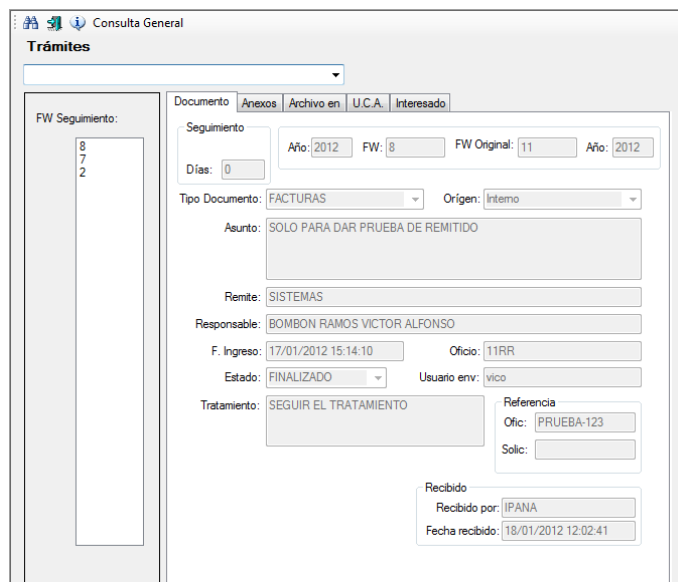
Para activar la consulta presionamos Buscar, luego ingresamos la información que queremos buscar en el campo específico y posteriormente presionamos en la lupa, para que se ejecute la búsqueda.



The screenshot shows a search window titled "Consulta por Trámite(s)". It contains a search form with the following fields: "ID:" (text input), "AÑO:" (dropdown menu set to "2012"), "Tipo de Documento:" (dropdown menu), "Fecha Ingreso:" (text input), "Número Oficio:" (text input), "Origen:" (dropdown menu), "Remitente ext:" (text input), and "Motivo:" (text area). There are also search icons (magnifying glass, refresh, stop, and help) at the top left of the form area.

Figura A6.29 Consulta por Trámites

Para observar el detalle de cada una de las dependencia por las cuales paso el trámite damos clic en la opción Aceptar y aparecerá la siguiente pantalla.



The screenshot shows a detailed view window titled "Consulta General". It features a "Trámites" dropdown menu at the top. Below it, there are several sections: "Seguimiento" with fields for "Año: 2012", "FW: 8", "FW Original: 11", and "Año: 2012"; "Días: 0"; "Tipo Documento: FACTURAS" and "Origen: Interno"; "Asunto: SOLO PARA DAR PRUEBA DE REMITIDO"; "Remite: SISTEMAS"; "Responsable: BOMBON RAMOS VICTOR ALFONSO"; "F. Ingreso: 17/01/2012 15:14:10" and "Oficio: 11RRR"; "Estado: FINALIZADO" and "Usuario env: vico"; "Tratamiento: SEGUIR EL TRATAMIENTO"; "Referencia" with "Ofic: PRUEBA-123" and "Solic:"; and "Recibido" with "Recibido por: IPANA" and "Fecha recibido: 18/01/2012 12:02:41". There are also tabs for "Documento", "Anexos", "Archivo en", "U.C.A.", and "Interesado".

Figura A6.30 Consulta General

En la parte izquierda presenta en forma general en número de FW del trámite, y en la parte derecha aparece el detalle con toda la información que fue generada en cada una de las dependencias por las cuales paso el trámite, hasta llegar a establecer su ubicación actual.