



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL**

**CARRERA DE SISTEMAS COMPUTACIONALES
E INFORMÁTICOS**

TEMA:

**“SISTEMA DE GESTIÓN UTILIZANDO SOFTWARE LIBRE PARA COBROS
Y REGISTROS DE USUARIOS DE LA JUNTA DE AGUAS CHACÓN
SEVILLA”**

Trabajo de Graduación. Modalidad: TEMI. Trabajo Estructurado de Manera Independiente, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos.

AUTOR: Luis Israel Mesías Haro.

TUTOR: Ing. Franklin Mayorga.

AMBATO – ECUADOR

Julio -2012

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de investigación sobre el tema: “**SISTEMA DE GESTIÓN UTILIZANDO SOFTWARE LIBRE PARA COBROS Y REGISTROS DE USUARIOS DE LA JUNTA DE AGUAS CHACÓN SEVILLA**”, del señor Luis Israel Mesías Haro, 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 el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el Art. 16 del Capítulo II , del Reglamento de Graduación para Obtener el Título Terminal del Tercer Nivel de la Universidad Técnica de Ambato.

Ambato, Julio 2012

.....
Ing. Franklin Mayorga

TUTOR

AUTORÍA

El presente trabajo de investigación titulado **SISTEMA DE GESTIÓN UTILIZANDO SOFTWARE LIBRE PARA COBROS Y REGISTROS DE USUARIOS DE LA JUNTA DE AGUAS CHACÓN SEVILLA**. Es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, Julio 2012

.....

Luis Israel Mesías Haro
C.I. 1600457483

APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. M.Sc. Oswaldo Paredes, Ing. Darío Robayo, Ing. Pilar Urrutia, revisó y aprobó el Informe Final del trabajo de graduación titulado: “Sistema de Gestión utilizando software libre para cobros y registros de usuarios de la Junta de Aguas Chacón Sevilla” , presentado por el señor Luis Israel Mesías Haro 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, M.Sc.
PRESIDENTE DEL TRIBUNAL

Ing. Darío Robayo
DOCENTE CALIFICADOR

Ing. Pilar Urrutia
DOCENTE CALIFICADOR

DEDICATORIA

Este proyecto está dedicado a dios por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, a mis padres, por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor y paciencia.

Luis Israel Mesías Haro

AGRADECIMIENTOS

Primeramente doy infinitamente gracias a Dios, por haberme dado fuerza y valor para terminar estos estudios.

Agradezco también la confianza y el apoyo de mis padres y hermanos, porque han contribuido positivamente para llevar a cabo esta difícil jornada.

*A la **Universidad Técnica de Ambato** y en especial a la **Facultad de Ingeniería en Sistemas** que me dieron la oportunidad de formar parte de ellas.*

Luis Israel Mesías Haro

ÍNDICE

CAPITULO I

EL PROBLEMA DE INVESTIGACIÓN

1.1 Tema de Investigación.....	1
1.2 Planteamiento del Problema.....	1
1.2.1 Contextualización.....	2
1.2.2 Análisis Crítico.....	3
1.2.3 Prognosis.....	3
1.2.4 Formulación del Problema.....	4
1.2.5 Preguntas Directrices.....	4
1.2.6 Delimitación del Problema.....	4
1.3 Justificación.....	5
1.4 Objetivos de la Investigación.....	6
1.4.1 Objetivo General.....	6
1.4.2 Objetivo Específico.....	6

CAPITULO II

MARCO TEÓRICO

2.1 Antecedentes Investigativos.....	7
2.2 Fundamentación Legal.....	7-8
2.3 Categorías Fundamentales.....	9-10
2.3.1 Sistema de Gestión.....	11-13
2.3.2 Software Libre.....	13-14
2.3.3 Sistema de Información.....	15-18
2.3.4 Cobros, Registro Usuarios.....	19-20
2.3.5 Base de Datos.....	20-23

2.3.6 Programación.....	23-26
2.4 Hipótesis.....	27
2.5 Señalamiento de variable de Hipótesis.....	27
2.5.1 Variable Independiente.....	27
2.5.2 Variable Dependiente.....	27

CAPITULO III

METODOLOGÍA

3.1 Enfoque.....	28
3.2 Modalidad básica de la Investigación.....	28
3.3 Niveles o tipos de Investigación.....	29
3.4 Población y Muestra.....	29
3.5 Operacionalización de variables.....	30-31
3.6 Recolección de Información.....	32
3.7 Procesamiento y análisis.....	32

CAPITULO IV

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1 Análisis de la necesidad.....	33
4.2 Análisis e Interpretación de los resultados.....	33
4.2.1 Análisis de los resultados de las encuestas.....	34-42

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.2 Conclusiones.....	44
5.3 Recomendaciones.....	45

CAPÍTULO VI

PROPUESTA

6.1 Tema.....	46
6.2 Datos Informativos.....	46
6.3 Antecedentes.....	47
6.4 Justificación.....	47
6.5 Objetivos.....	48
6.6 Análisis de Factibilidad.....	48
6.7 Fundamentación.....	50
6.8 Metodología.....	52
6.9 Modelo Operativo.....	52
6.9.1 Análisis del Sistema.....	53
6.9.1.1 Análisis y requerimientos del Sistema.....	53
6.9.1.1.1 Diagrama de Caso de Uso.....	54
6.9.1.1.2 Prototipo de Interfaz de Usuario.....	63
6.9.1.1.3 Diagrama de Clases.....	71
6.9.1.1.4 Diagrama de Secuencias.....	72
6.9.1.1.5 Diagrama de Estados.....	79
6.9.1.1.6 Diagrama de Actividades.....	81
6.9.2 Diseño del Sistema.....	91
6.9.2.1 Diseño de la Base de Datos.....	91
6.9.2.2 Diccionario de Datos.....	92
6.9.3 Pruebas.....	100
6.9.3.1 Pruebas de caja negra.....	100
6.9.3.2 Pruebas de caja blanca.....	101
6.9.3.3 Pruebas de verificación y validación.....	102
6.9.3.4 Implantación.....	103
6.10 Conclusiones y Recomendaciones.....	104
6.10.1 Conclusiones.....	104
6.10.2 Recomendaciones.....	105
6.11 Bibliografía.....	106
6.11.1 Información bibliográfica.....	106

6.11.2 Referencias bibliográficas Web.....	106
Glosario de términos.....	109

ANEXOS

ANEXO 1 Cronograma.....	112
ANEXO 2: Estructura del Cuestionario.....	103
ANEXO 3: Manual de Configuración.....	115
ANEXO 4: Manual de Instalación.....	150
ANEXO 5: Manual de Usuario.....	154

ÍNDICE DE FIGURAS

Figura 1.0 Constelación de Ideas(Causa -Efecto).....	2
Figura 1.1 Inclusión interrelacionadas de la variable independiente.....	9
Figura 1.2 Inclusión interrelacionadas de la variable dependiente.....	9
Figura 1.3 Constelación de Ideas variable independiente.....	10
Figura 1.4 Constelación de Ideas variable dependiente.....	10
Figura 1.5 Sistema de Información.....	15
Figura 1.6 Gráfico porcentual pregunta 1.....	34
Figura 1.7 Gráfico porcentual pregunta 2.....	35
Figura 1.8 Gráfico porcentual pregunta 3.....	36
Figura 1.9 Gráfico porcentual pregunta 4.....	37
Figura 2.0 Gráfico porcentual pregunta 5.....	38
Figura 2.1 Gráfico porcentual pregunta 6.....	39
Figura 2.2 Gráfico porcentual pregunta 7.....	40
Figura 2.3 Gráfico porcentual pregunta 8.....	41
Figura 2.4 Gráfico porcentual pregunta 9.....	42
Figura 2.5 Gráfico porcentual pregunta 10.....	43

Figura 2.6 Diagrama de casos de uso Sistema de Cobros.....	54
Figura 2.7 Caso de uso individual Ingresar al Sistema.....	55
Figura 2.8 Caso de uso individual administrar junta.....	56
Figura 2.9 Caso de uso individual administrar óvalos.....	57
Figura 3.0 Caso de uso individual administrar usuarios.....	58
Figura 3.1 Caso de uso individual administrar cuenta.....	59
Figura 3.2 Caso de uso individual administrar contribuciones.....	60
Figura 3.3 Caso de uso individual administrar Actividades.....	61
Figura 3.4 Caso de uso individual Facturar.....	62
Figura 3.5 Ventana Inicio de Sesión.....	63
Figura 3.6 Pantalla menú principal.....	63
Figura 3.7 Pantalla opción principal.....	64
Figura 3.8 Pantalla opción datos.....	65
Figura 3.9 Pantalla opción cobros.....	66
Figura 4.0 Pantalla opción informes.....	67
Figura 4.1 Pantalla opción finanzas.....	68
Figura 4.2 Pantalla opción Configuraciones.....	69
Figura 4.3 Pantalla opción ayuda.....	70
Figura 4.4 Diagrama de clases.....	71
Figura 4.5 Diagrama de secuencia de Junta.....	72
Figura 4.6 Diagrama de secuencia de Sector.....	72
Figura 4.7 Diagrama de secuencia de Óvalos.....	73
Figura 4.8 Diagrama de secuencia de Usuarios.....	73
Figura 4.9 Diagrama de secuencia de Cuentas.....	74
Figura 5.0 Diagrama de secuencia de Moras.....	74
Figura 5.1 Diagrama de secuencia de Contribuciones.....	75
Figura 5.2 Diagrama de secuencia de Actividades.....	75
Figura 5.3 Diagrama de secuencia de Registrar Actividades.....	76
Figura 5.4 Diagrama de secuencia de Aportes Extras.....	76
Figura 5.5 Diagrama de secuencia de Registrar Aportes Extras.....	77
Figura 5.6 Diagrama de secuencia de Registrar Egresos.....	77
Figura 5.7 Diagrama de secuencia de Pagos.....	78
Figura 5.8 Diagrama de secuencia de Anular Factura.....	78
Figura 5.9 Diagrama de estado Usuarios.....	79

Figura 6.0 Diagrama de estado Cuenta.....	79
Figura 6.1 Diagrama de estado Factura.....	80
Figura 6.2 Diagrama de Actividades Sistema de Cobros.....	81
Figura 6.3 Diagrama de actividades ingresar al sistema.....	82
Figura 6.4 Diagrama de actividades administrar junta.....	83
Figura 6.5 Diagrama de Actividades administrar sectores.....	84
Figura 6.6 Diagrama de Actividades administrar óvalos.....	85
Figura 6.7 Diagrama de Actividades administrar usuarios.....	86
Figura 6.8 Diagrama de Actividades administrar cuenta.....	87
Figura 6.9 Diagrama de Actividades administrar actividades.....	88
Figura 6.10 Diagrama de Actividades administrar contribuciones.....	89
Figura 6.11 Diagrama de Actividades Factura.....	90
Figura 6.12 Diseño de Base de Datos.....	91
Figura 6.13 Validación Ruc.....	103
Figura 6.14 Validación del campo Ruc incompleto.....	103
Figura 6.15 Validación cédula.....	103
Figura A1 Cronograma.....	112
Figura 6.16 Pantalla Setup.....	150
Figura 6.17 Pantalla asistente para instalación.....	151
Figura 6.18 Pantalla seleccionar carpeta de instalación.....	151
Figura 6.19 Pantalla confirmar instalación.....	152
Figura 6.20 Pantalla proceso instalación.....	153
Figura 6.21 Figura Pantalla instalación completada.....	154
Figura 6.22 Ventana de inicio de sesión.....	156
Figura 6.23 Pantalla Menú Principal.....	158
Figura 6.24 Pantalla Administrador.....	158
Figura 6.25 Pantalla administrar junta.....	159
Figura 6.26 Pantalla modificar junta.....	159
Figura 6.27 Pantalla administrar óvalos.....	160
Figura 6.29 Pantalla de ingreso óvalo.....	160
Figura 6.30 Pantalla modificación óvalo.....	161
Figura 6.31 Pantalla de eliminación de óvalo.....	161
Figura 6.32 Pantalla administrar usuarios.....	162
Figura 6.33 Pantalla ingreso de usuarios.....	162

Figura 6.34 Pantalla ingreso de usuarios.....	163
Figura 6.35 Pantalla eliminación de usuarios.....	163
Figura 6.36 Pantalla administrar cuenta.....	164
Figura 6.37 Pantalla ingreso cuenta.....	164
Figura 6.38 Pantalla modificación cuenta.....	165
Figura 6.39 Pantalla eliminación cuenta.....	165
Figura 6.40 Pantalla administrar actividad.....	166
Figura 6.41 Pantalla ingreso de actividad.....	166
Figura 6.42 Pantalla modificación de actividad.....	167
Figura 6.43 Pantalla eliminación de actividad.....	167
Figura 6.44 Pantalla administrar asistencias.....	168
Figura 6.45 Pantalla ingreso asistencia.....	168
Figura 6.46 Pantalla administración contribuciones.....	169
Figura 6.47 Pantalla ingreso contribución.....	169
Figura 6.48 Pantalla opción cobros.....	170
Figura 6.49 Pantalla factura.....	170
Figura 6.50 Pantalla Factura.....	171
Figura 6.51 Pantalla anular factura.....	172
Figura 6.52 Pantalla anular.....	172
Figura 6.53 Pantalla opción informes.....	173
Figura 6.54 Pantalla Informes Facturas Emitidas.....	173
Figura 6.55 Pantalla visualizar informe factura.....	174
Figura 6.56 Pantalla Informes Económicos.....	174
Figura 6.57 Pantalla visualizar informes económicos.....	175
Figura 6.58 Pantalla configurar IVA.....	175

ÍNDICE DE TABLAS

Tabla 1.0 Operacionalización de variable independiente.....	30
Tabla 1.1 Operacionalización de variable dependiente.....	31
Tabla 1.2 Cuadro porcentual pregunta 1.....	34
Tabla 1.3 Cuadro porcentual pregunta 2.....	35
Tabla 1.4 Cuadro porcentual pregunta 3.....	36
Tabla 1.5 Cuadro porcentual pregunta 4.....	37
Tabla 1.6 Cuadro porcentual pregunta 5.....	38
Tabla 1.7 Cuadro porcentual pregunta 6.....	39
Tabla 1.8 Cuadro porcentual pregunta 7.....	40
Tabla 1.9 Cuadro porcentual pregunta 8.....	41
Tabla 2.0 Cuadro porcentual pregunta 9.....	42
Tabla 2.1 Cuadro porcentual pregunta 10.....	43
Tabla 2.2 Especificación ingresar al sistema.....	55
Tabla 2.3 Especificación Administrar Junta.....	56
Tabla 2.4 Especificación Administrar Ovalo.....	57
Tabla 2.5 Especificación Administrar Usuarios.....	58
Tabla 2.6 Especificación Administrar Cuenta.....	59
Tabla 2.7 Especificación Administrar Contribuciones.....	60
Tabla 2.8 Especificación Administrar Actividades.....	61
Tabla 2.9 Especificación Facturar.....	62
Tabla 3.0 Tabla Junta.....	92
Tabla 3.1 Tabla Sectores.....	92
Tabla 3.2 Tabla Óvalos.....	93
Tabla 3.3 Tabla Usuarios.....	93
Tabla 3.4 Tabla Cuenta.....	94
Tabla 3.5 Tabla Moras.....	94
Tabla 3.6 Tabla Aportes Extras.....	95
Tabla 3.7 Tabla Actividades.....	95
Tabla 3.8 Tabla Registrar Actividades.....	96
Tabla 3.9 Tabla Contribuciones.....	96
Tabla 4.1 Tabla Pagos.....	97

Tabla 4.2 Tabla Detalle Aportes Extras.....	97
Tabla 4.3 Tabla Detalle Aportes Extras.....	98
Tabla 4.4 Tabla Detalle Contribuciones.....	98
Tabla 4.5 Tabla Egresos.....	99
Tabla 4.6 Tabla IVA.....	99
Tabla 4.7 Casos de Prueba.....	102

CAPÍTULO I.

EL PROBLEMA.

1.1 TEMA:

Sistema de gestión utilizando software libre para cobros y registros de usuarios de la Junta de Aguas Chacón Sevilla.

1.2 PLANTEAMIENTO DEL PROBLEMA.

1.2.1 CONTEXTUALIZACIÓN.

Debido a la globalización existe una gran competitividad en nuestro mundo actual, es por ello que todas las instituciones de nuestro País deben estar siempre a la vanguardia de los avances tecnológicos que se presentan cada día, para así obtener un excelente rendimiento y eficiencia en todos los aspectos que en ella se realicen especialmente en los servicios que se prestan.

Los procesos rutinarios en forma manual procesan información lenta, llevarla a un sistema de gestión garantizar un mejor trato de información.

La mayoría de Juntas de agua de riego del País, han venido experimentando un moderado pero sostenido crecimiento en el servicio de riego, lo que ha significado un

cierto aumento del volumen de usuarios. La problemática tiene su génesis en que varias de estas juntas carecen de un sistema de gestión que permita agilizar el servicio de cobros y mejorar el almacenamiento de información de los usuarios.

En la provincia de Tungurahua, la carencia de herramientas tecnológicas en las juntas de agua de riego provoca conflictos con el almacenamiento de la información haciéndole insegura y desconfiable para los usuarios, puesto que muchas de estas siguen llevando sus informes de forma manual.

La Junta de aguas Chacón Sevilla de la provincia de Tungurahua cantón Ambato, actualmente cuenta con 1000 usuarios, pero carece de tecnología, funcionando de forma manual y originando un alto grado de desorganización, lentitud en los cobros, proceso que a su vez produce agotamiento del personal. La carencia de un Sistema de Gestión, ocasiona retrasos en los pagos, y un mal almacenamiento de la información de los usuarios.



Figura 1.0 Constelación de Ideas (Causa - Efecto)

1.2.2 ANÁLISIS CRÍTICO.

En la Junta de Aguas Chacón Sevilla existe una carencia de las herramientas tecnológicas, como un sistema de gestión con las funciones que la Junta requiera, esto se debe a la falta de recursos económicos que se limita poder adquirirlos, conllevando a que la junta no posea un registro adecuado de los usuarios, ocasionando la pérdida de la información.

La limitada decisión de los Directivos de la Junta de Aguas Chacón Sevilla en la adquisición de un software que aporte en el mejoramiento del servicio de cobros ha provocado, que la junta siga llevando todos sus informes manualmente, por lo que resulta lento y problemático dando lugar a una insatisfacción de los usuarios.

El no contar con personas capacitadas para poder realizar un software específico para el beneficio de la Junta de Aguas Chacón Sevilla, provoca que la información no pueda ser llevada de forma ágil y segura, lo que no facilita el normal funcionamiento de la junta ocasionando pérdidas económicas, pérdida de tiempo e información no confiable.

1.2.3 PROGNOSIS.

El no contar con una tecnología adecuada para el servicio de los usuarios de la Junta, esta seguirá llevando todos sus informes de forma manual, y la información que brinda no será ágil ni segura, trayendo consigo un retraso en los cobros, además de pérdidas de información y clientes insatisfechos.

Por lo dicho anteriormente la Empresa se desprestigiará dentro de la comunidad al prestar un mal servicio, cuyos miembros se verían obligados a buscar alternativas que les proporcione un servicio eficiente.

Por lo que se hace necesario desarrollar e Implantar un Sistema de Gestión utilizando software libre para cobros y registros de usuarios.

1.2.4 FORMULACIÓN DEL PROBLEMA.

¿Qué incidencia tiene un sistema de gestión utilizando software libre en los cobros y registros de usuarios en la Junta de Aguas Chacón Sevilla?

1.2.5 PREGUNTAS DIRECTRICES.

¿Qué características deberá tener el sistema de gestión?

¿Qué características deben reunir las herramientas de software libre para implementar un sistema de gestión?

¿Cuáles serían las ventajas y desventajas de un sistema de gestión que lleve el control de cobros y registros de usuarios.

¿Cuáles son los requerimientos necesarios para implementar un sistema de gestión?

1.2.6 DELIMITACIÓN DEL PROBLEMA.

El presente proyecto se desarrollará en la Junta de aguas Chacón Sevilla ubicado en la parroquia de Izamba, se trabajará con herramienta de software libre el cual requerirá de conocimientos básicos, el tiempo estimado para el desarrollo del proyecto es de seis meses a partir de su aprobación, se trabajará con cinco miembros de la directiva de la junta.

1.3 JUSTIFICACIÓN.

Los procesos automatizados, son factores de vital importancia en la organización, en esta era se ha observado una gran innovación y desarrollo de tecnologías de la información que ha permitido la evolución de computadores que son capaces de producir sistemas de gestión a la sociedad con aportes a los distintos sectores económicos, sociales, políticos, educativos, entre otros.

El estudio para el diseño del sistema de gestión se lo realizará enfocado en mejorar los procesos actuales de la Junta de aguas Chacón Sevilla, se aplicará los conocimientos teóricos -prácticos adquiridos durante la carrera, así como también la información que brindarán los miembros de la junta, ya que conocen del funcionamiento del los procesos.

La conveniencia de desarrollar un sistema de gestión, procesada y almacenada de una forma más efectiva para agilizar el proceso de cobros y lograr un control integral de los usuarios, proporcionará una mejor efectividad en el manejo del flujo y procesamiento de los datos.

El sistema tendrá un impacto positivo en la comunidad, la cual proporcionará a la misma, una información confiable, agilizando y facilitando el trabajo y el control de los cobros y los usuarios.

Los usuarios, miembros de la Junta y la misma, se verán beneficiados al implementar el sistema de gestión, ya que se sentirán seguros y confiados de que la información cuenta con un control eficiente, y el mismo puede servir como ejemplo a seguir en otros sectores que tengan la necesidad de lograr con éxito los propósitos e iniciar una nueva era de avances en pro de la comunidad.

La información será manejada con eficiencia y eficacia al momento de atender las exigencias de control que se requiere.

1.4 OBJETIVOS.

1.4.1 GENERAL:

- Desarrollar un sistema de gestión utilizando software libre para cobros y registros de usuarios de la Junta de Aguas Chacón Sevilla.

1.4.2 ESPECÍFICO:

- Analizar los procesos para el funcionamiento del sistema de gestión para cobros y registros de usuarios de la Junta de Aguas Chacón Sevilla.
- Determinar las herramientas adecuadas con las que se va a trabajar en el desarrollo del sistema de gestión.
- Implementar el sistema de gestión que permita mejorar el proceso de cobros y registros de usuarios en la junta de aguas Chacón Sevilla.

CAPITULO II.

MARCO TEÓRICO.

2.1 ANTECEDENTES INVESTIGATIVOS.

Revisados los archivos de la biblioteca de la facultad de Ingeniería en Sistemas Electrónica e Industrial se encontró que existe varios trabajos similares con los temas: “Desarrollo e implantación de un sistema para el control y facturación del agua potable en el municipio de Tisaleo” que fue implantado en el año 2008 por el Ing. Hugo Freire. “Sistema de facturación para el control automatizado de las tarifas recaudadas en las juntas administradoras de agua potable adscritas al parlamento agua del gobierno provincial de Tungurahua” que fue implantado en el año 2011 por la Ing. Erika Magdalena Llerena Ortíz.

2.2 FUNDAMENTACIÓN LEGAL.

ESTATUTOS DEL DIRECTORIO DE AGUAS DE LA ACEQUIA CHACON SEVILLA.

ORGANIZMO, FINES Y ADMINISTRACION.

CAPITULO I.

Art 1.- Los Usuarios de la acequia Chacón – Sevilla, acogiéndonos a la ley de aguas publicada con decreto No 369 del 18 de mayo de 1972 y el reglamento de la misma,

constituidos en la ciudad de Ambato, provincia de Tungurahua, se organizan en el directorio de aguas de la acequia Chacón Sevilla.

Art 2.- El objeto de esta organización, es la administración, conservación y mejor aprovechamiento de las aguas de este canal, según las concesiones otorgadas y las que se otorgaren por parte del Instituto Ecuatoriano de Recurso Hidráulicos.

CAPITULO III.

CONSEJO DE AGUAS.

Art 10.- Son deberes y atribuciones del Consejo de Aguas:

- A. Hacer respetar los derechos que corresponden a cada usuario.
- B. Elaborar y mantener actualizado el padrón o registro de usuarios, con todos los datos necesarios.
- C. Exigir que el caudal que corresponde a cada usuario sea respetado.
- D. Aplicar las sanciones legales a los usuarios incumplidos.
- E. Los Pagos y más gastos se realizarán con la autorización del Presidente.
- F. Sesionar, una vez al mes, por lo menos en forma ordinaria y extraordinariamente, cada vez que lo convoque el Presidente.
- G. Determinar el porcentaje de recargo por morosidad, de acuerdo con la ley de Aguas, a los usuarios incumplidos.

Art 24.- El usuario que incurra en sustracciones de aguas será sancionado con una multa equivalente al 10% del salario mínimo vital, y si reincide se sancionará con el doble.

Art 28.- Es obligatorio, para todos los usuarios, acudir a las limpiezas generales y quienes no lo hagan pagarán el doble del jornal fijado.

2.3 CATEGORÍAS FUNDAMENTALES.

VARIABLE INDEPENDIENTE

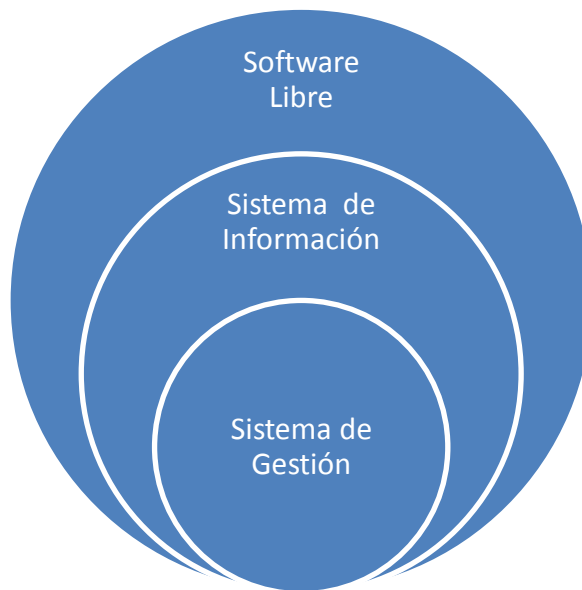


Figura 1.1 Inclusión interrelacionadas de la variable independiente

VARIABLE DEPENDIENTE

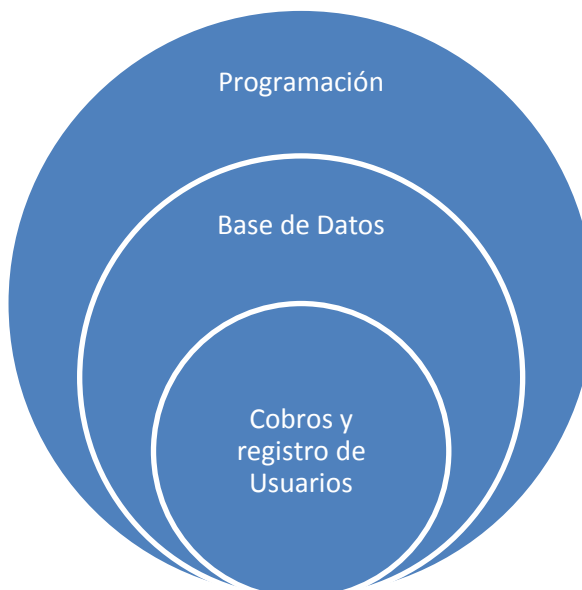


Figura 1.2 Inclusión interrelacionadas de la variable dependiente

CONSTELACION DE IDEAS VARIABLE INDEPENDIENTE

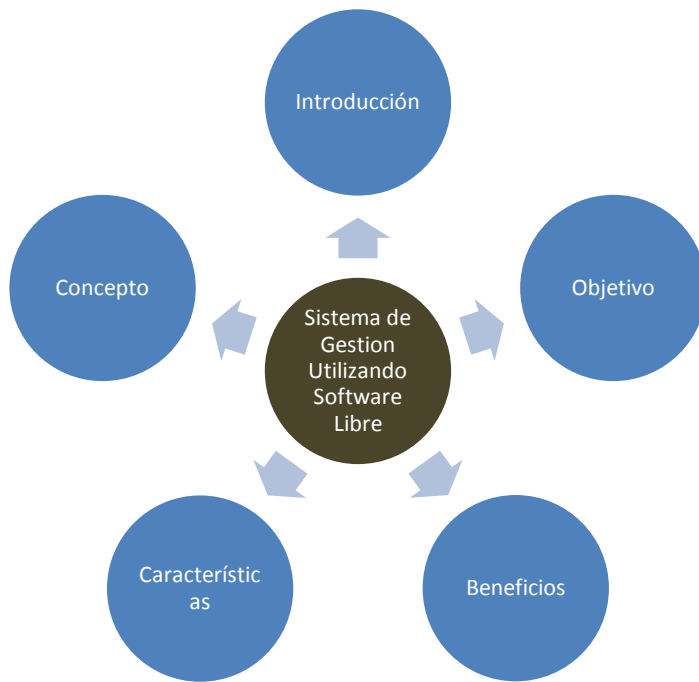


Figura 1.3 Constelación de Ideas variable independiente

CONSTELACION DE IDEAS VARIABLE DEPENDIENTE

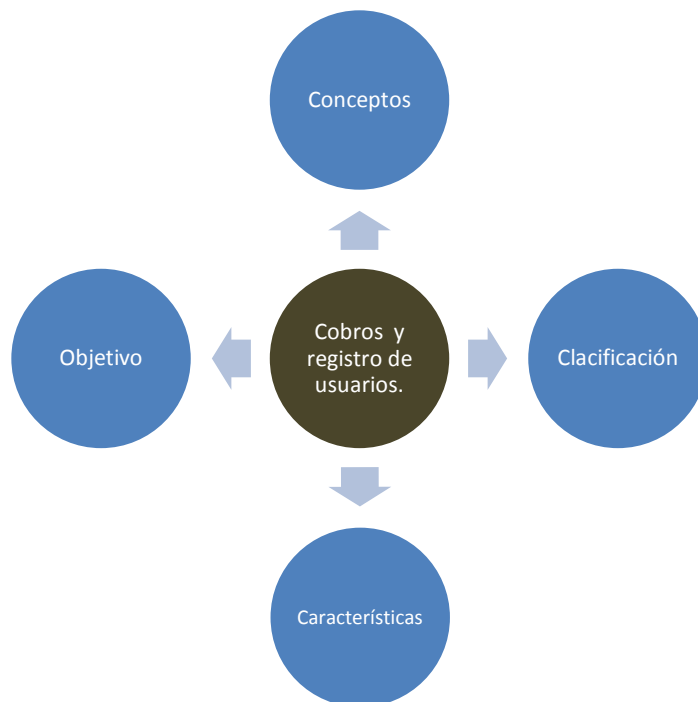


Figura 1.4 Constelación de Ideas variable dependiente

2.3.1 Categorías Fundamentales de la Variable Independiente:

SISTEMAS DE GESTIÓN.

Introducción.

Es muy importante que tengamos en cuenta que la mayoría de las estrategias que suelen desarrollar las empresas dependen de las herramientas e instrumentos tecnológicos correspondientes a los sistemas de gestión que se emplean en algún momento dado.

Siempre que se trate de la administración de una empresa los sistemas de gestión cumplen con un papel fundamental ya que aquellas empresas que cuentan con una información que se caracterice por ser rápida, fluida y enfocada a las tácticas y estrategias de la misma.

Concepto.

Se trata de sistemas que optimizan los procesos internos de las empresas mediante un software que permite compartir la información de las áreas principales:

- Finanzas.
- Fabricación.
- Logística.
- Ventas y marketing.
- Recursos humanos.

Los sistemas de gestión hacen referencia a todos aquellos sistemas informáticos con los que se apoya una empresa para poder desarrollar correctamente las actividades de la misma.

Objetivo.

Los sistemas de gestión tienen por objetivo particular proporcionar la información que se requiera y se considere necesaria para poder desempeñar correctamente, todas las responsabilidades que las diferentes funciones del negocio implican; las mismas son, en cada caso, las necesarias para que la organización sea capaz de cumplir con todos sus objetivos, y en muchos otros casos, también para seguir una estrategia de negocio determinada.

Beneficios.

Independientemente de qué se trate de la industria, las empresas que operen con los sistemas de gestión de la más alta calidad, permanecerán y se convertirán en aquellas organizaciones más fuertes en cuanto a la competencia que se plantea en el ámbito del mercado empresarial, mientras que aquellas empresas que no logren una mejora en los sistemas de gestión correspondientes a su información, quedarán a la zaga de aquellas que lo han logrado. Además, debemos considerar el hecho de contar con información de calidad proporcionada por los sistemas de gestión de más alto rendimiento, nos ayudará y resultará muy útil para la planificación, dirección y el control de todos los aspectos que comprende la gestión de una empresa.

Características.

- Los sistemas de gestión de la información que se maneja en una empresa para poder lograr una correcta administración de los recursos, se encuentra completamente al servicio de la empresa y del logro de sus objetivos, y por tal motivo, no es independiente de los mismos.
- Los sistemas de gestión no representan los objetivos, sino que simplemente, constituyen las armas competitivas mediante las cuales las empresas pueden acudir para alcanzar las metas de los gerentes, de productos y servicios excelentes y especialmente de una mayor productividad.

- La información de la empresa que se gestiona mediante estos sistemas de gestión, debe **transmitir una ilustración que resulte muy clara al receptor**, lo cual, si consideramos esto, la misma requerirá una presentación en forma gráfica.
- Los sistemas de gestión de información de una empresa no resultan independientes de su estrategia, debido a que los mismos deben contribuir a poner dicha estrategia en práctica, y siempre cumplimiento con la meta de ser planificada y diseñada a medida de la empresa.

SOFTWARE LIBRE.

Concepto.

Es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y por tanto, una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente.

Según la Free Software Fundación, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado.

Libertades.

De acuerdo a la definición, el software es libre si se garantizan las 4 libertades siguientes:

- **Libertad 0:** La libertad de usar el programa, con cualquier propósito (privado, educativo, público, comercial, militar, etc.).
- **Libertad 1:** La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (para lo cual es necesario poder acceder al código fuente).
- **Libertad 2:** La libertad de distribuir copias del programa, con lo cual puedes ayudar a tu prójimo.

- **Libertad 3:** La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.

Ventajas.

- Escrutinio Público: Al ser muchos las personas que tienen acceso al código fuente, eso lleva a un proceso de corrección de errores muy dinámico, no hace falta esperar que el proveedor del software saque una nueva versión.
- Independencia del proveedor.
- Existen aplicaciones para todas las plataformas (Linux, Windows, Mac Os).
- La existencia de virus es casi nula.
- Mejor distribución de la riqueza y conocimiento.
- El mantenimiento de la computadora es menor.
- Transparencia en el manejo de nuestros datos Sector público y privado.

Desventaja.

- Mayores costos de implantación e interoperabilidad: dado que el software constituye "algo nuevo", ello supone afrontar un costo de aprendizaje, de instalación, de migración, de interoperabilidad, etc.

SISTEMAS DE INFORMACIÓN.

Concepto.

Es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su posterior uso, generados para cubrir una necesidad.

Elementos.

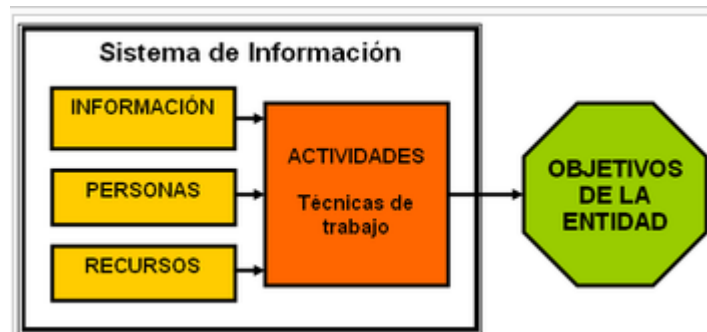


Figura 1.5 Sistema de Información

Dichos elementos formarán parte de alguna de estas categorías:

- Personas.
- Datos.
- Actividades o técnicas de trabajo.
- Recursos materiales en general (típicamente recursos informáticos y de comunicación, aunque no tienen por qué ser de este tipo obligatoriamente).

Todos estos elementos interactúan entre sí para procesar los datos (incluyendo procesos manuales y automáticos) dando lugar a información más elaborada y distribuyéndola de la manera más adecuada posible en una determinada organización en función de sus objetivos.

Un sistema de información representa todos los elementos que forman parte de la administración, el procesamiento, el transporte y la distribución de la información dentro de la compañía. En términos prácticos, el alcance del término "sistema de información" puede variar notablemente entre una organización y otra y, según el caso, puede abarcar todos o algunos de los siguientes elementos:

- Bases de datos de la compañía.
- Software de gestión integral de empresas.
- Herramienta para la Gestión de relaciones con los clientes.
- Herramienta para la Gestión de la cadena de suministro.

- Solicitudes de empleo.
- Infraestructura de red.
- Servidores de datos y sistemas de almacenamiento.
- Servidor de aplicaciones.
- Dispositivos de seguridad.

Objetivos.

Los objetivos de los Sistemas de Información van a depender del nivel de la organización en el que se apliquen para resolver determinados problemas. Los sistemas de información cumplirán **tres objetivos** básicos dentro de las organizaciones:

1. Automatizar los procesos operativos.
2. Proporcionar información que sirva de apoyo a la toma de decisiones.
3. Lograr ventajas competitivas a través de su implantación y uso.

Ciclo de vida.

Existen pautas básicas para el desarrollo de un Sistema de Información para una organización:

- **Conocimiento de la Organización:** analizar y conocer todos los sistemas que forman parte de la organización, así como los futuros usuarios del Sistema de Información.
- **Identificación de problemas y oportunidades:** el segundo paso es relevar las situaciones que tiene la organización y de las cuales se puede sacar una ventaja competitiva (Por ejemplo: una empresa con un personal capacitado en manejo informático reduce el costo de capacitación de los usuarios).
- **Determinar las necesidades:** este proceso también se denomina elicitación de requerimientos. En el mismo, se procede identificar a través de algún método de recolección de información (el que más se ajuste a cada caso) la información relevante para el Sistema de Información que se propondrá.

- **Diagnóstico:** En este paso se elabora un informe resaltando los aspectos positivos y negativos de la organización. Este informe formará parte de la propuesta del Sistema de Información y, también, será tomado en cuenta a la hora del diseño.
- **Propuesta:** Contando ya con toda la información necesaria acerca de la organización es posible elaborar una propuesta formal dirigida hacia la organización donde se detalle el presupuesto, relación costo-beneficio, presentación del proyecto de desarrollo del Sistema de Información.
- **Diseño del sistema:** Una vez aprobado el proyecto, se comienza con la elaboración del diseño lógico; la misma incluye el diseño del flujo de la información dentro del sistema, los procesos que se realizarán dentro del sistema, etc. En este paso es importante seleccionar la plataforma donde se apoyará el SI y el lenguaje de programación a utilizar.
- **Codificación:** Con el algoritmo ya diseñado, se procede a su reescritura en un lenguaje de programación establecido (programación), es decir, en códigos que la máquina pueda interpretar y ejecutar.
- **Implementación:** Este paso consta de todas las actividades requeridas para la instalación de los equipos informáticos, redes y la instalación del programa generado en el paso anterior.
- **Mantenimiento:** proceso de retroalimentación, a través del cual se puede solicitar la corrección, el mejoramiento o la adaptación del Sistema de Información ya creado a otro entorno. Este paso incluye el soporte técnico acordado anteriormente.

Tipos.

Se puede clasificar a los sistemas de información en:

- **Sistemas de procesamiento de transacciones, TPS:** estos sistemas se encargan de registrar y procesar datos resultantes de transacciones comerciales y de la automatización de tareas y procesos operativos. Los resultados se utilizan para actualizar las bases de datos. Estas bases de datos posteriormente

proporcionan los datos que pueden ser utilizados por los sistemas de apoyo a la toma de decisiones.

- **Sistemas de información administrativa, MIS:** Proporcionan una variedad de informes y presentaciones a la gerencia. Estos sistemas recuperan de las bases de datos información sobre operaciones internas que han sido actualizadas mediante sistemas de procesamiento de transacciones.
- **Sistemas de soporte a la toma de decisiones, DSS:** estos sistemas combinan datos y modelos analíticos avanzados para apoyar los procesos de toma de decisiones. Ayudan a la toma de decisiones relacionadas con problemas específicos.
- **Sistemas para la toma de decisión en grupo, GDSS:** estos sistemas tienen como tarea apoyar la toma de decisiones en grupo. Permiten a los miembros de un grupo tener acceso simultáneo a archivos comunes y a bases de datos, de tal manera que los miembros del grupo puedan trabajar en la misma tarea, en el mismo lugar y al mismo tiempo.
- **Sistemas de apoyo a ejecutivos, EIS:** estos sistemas ayudan a los funcionarios de alto nivel a dirigir una organización. Su meta es proporcionar un acceso inmediato y fácil a información selectiva sobre factores clave que son fundamentalmente para el logro de los objetivos estratégicos de una empresa.
- **Sistemas expertos de soportes a la toma de decisiones, EDSS:** son sistemas de información basados en el conocimiento, es decir, que utilizan su conocimiento sobre un área específica para actuar como un consultor experto para los usuarios.

2.3.2 Categorías Fundamentales de la Variable Dependiente:

COBROS Y REGISTROS DE USUARIOS.

Cuentas por Cobrar.

Son derechos exigibles originados por ventas o prestación de servicios. Las cuentas por cobrar son, al igual que cualquier activo, recursos económicos propiedad de una empresa que la generarán un beneficio en el futuro Forman parte del activo circulante Entre las principales cuentas que están incluidas en este rubro se encuentran: Clientes.

Objetivo.

El objetivo de las cuentas por cobrar es estimular las ventas y ganar clientes. Se tiende a considerarlas como un medio para vender productos y superar la competencia mediante el ofrecimiento de facilidades de crédito como partes de los servicios de la empresa para favorecer a sus clientes.

Característica.

Las cuentas por cobrar representan el crédito que concede la empresa a sus clientes, sin más garantías que la promesa de pago en un plazo determinado.

Clasificación.

Las cuentas por cobrar se clasifican atendiendo a su origen, a cargo de clientes, de compañías afiliadas, empleados y otros deudores, separando los que provengan de ventas y servicios de los que tengan otro origen. También se clasifican atendiendo a su disponibilidad, inmediata o corto plazo y a largo plazo.

Registro.

Es un tipo o conjunto de datos almacenados en un sistema. Para la informática, existen distintos tipos de registros, pero en todos los casos hay una referencia al concepto de almacenar datos o información sobre el estado, procesos o uso de la computadora.

Usuario.

Para la informática es un usuario aquella persona que utiliza un dispositivo o un ordenador y realiza múltiples operaciones con distintos propósitos. A menudo es un usuario aquel que adquiere una computadora o dispositivo electrónico y que lo emplea para comunicarse con otros usuarios, generar contenido y documentos, utilizar software de diverso tipo y muchas otras acciones posibles.

BASE DE DATOS.**Concepto.**

Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Una base de datos (cuya abreviatura es BD) es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible. Diferentes programas y diferentes usuarios deben poder utilizar estos datos.

Existen programas denominados sistemas gestores de bases de datos, abreviados SGBD, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Tipos.

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se esté manejando, la utilidad de las mismas o las necesidades que satisfagan:

Según la variabilidad de los datos almacenados

- **Bases de datos estáticas:** Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.
- **Bases de datos dinámicas:** Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de un supermercado, una farmacia, un videoclub o una empresa.

Según el contenido

- **Bases de datos bibliográficas:** Solo contienen un representante de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque si no, estaríamos en presencia de una base de datos a texto completo. Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio.
- **Bases de datos de texto completo:** Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

Modelos.

Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos:

- **Bases de datos relacionales:** Éste es el modelo utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.
- **Bases de datos jerárquicas:** Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica
- **Base de datos de red:** Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres.
- **Bases de datos transaccionales:** Éste es el modelo utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

Ventajas.

- Uso de reglas lógicas para expresar las consultas.
- Permite responder consultas recursivas.
- Cuenta con negaciones estratificadas
- Capacidad de obtener nueva información a través de la ya almacenada en la base de datos mediante inferencia.
- Uso de algoritmos de optimización de consultas.
- Soporta objetos y conjuntos complejos.

Desventajas.

- Crear procedimientos eficaces de deducción para evitar caer en bucles infinitos.
- Encontrar criterios que decidan la utilización de una ley como regla de deducción.
- Replantear las convenciones habituales de la base de datos.

PROGRAMACIÓN.

Concepto:

La programación es el proceso de diseñar, escribir, probar, depurar y mantener el código fuente de programas computacionales. El código fuente es escrito en un lenguaje de programación.

El propósito de la programación es crear programas que exhiban un comportamiento deseado.

Programas y algoritmos

Un algoritmo es una secuencia no ambigua, finita y ordenada de instrucciones que han de seguirse para resolver un problema. Un programa normalmente implementa (traduce a un lenguaje de programación concreto) uno o más algoritmos. Un algoritmo puede expresarse de distintas maneras: en forma gráfica, como un diagrama de flujo, en forma de código como en pseudocódigo o un lenguaje de programación, en forma explicativa, etc.

Según Niklaus Wirth, un programa está formado por los algoritmos y la estructura de datos.

Se han propuesto diversas técnicas de programación cuyo objetivo es mejorar tanto el proceso de creación de software como su mantenimiento. Entre ellas, se pueden mencionar las siguientes:

- Programación Estructurada.
- Programación Modular.
- Programación Orientada a Objetos (POO).
- Programación Declarativa.

Compilación.

El programa escrito en un lenguaje de programación fácilmente comprensible por el programador es llamado programa fuente y no se puede ejecutar directamente en una computadora. La opción más común es compilar el programa obteniendo un módulo objeto, aunque también puede ejecutarse en forma más directa a través de un intérprete informático.

El código fuente del programa se debe someter a un proceso de traducción para convertirlo en lenguaje máquina, código éste directamente ejecutable por el procesador. A este proceso se le llama compilación.

Normalmente la creación de un programa ejecutable .exe para Microsoft Windows o DOS conlleva dos pasos. El primer paso se llama compilación propiamente dicho y traduce el código fuente escrito en un lenguaje de programación almacenado en un archivo a código en bajo nivel normalmente en código objeto, no directamente a lenguaje máquina. El segundo paso se llama enlazado en el cual se enlaza el código de bajo nivel generado de todos los ficheros y subprogramas que se han mandado compilar y se añade el código de las funciones que hay en las bibliotecas del compilador para que el ejecutable pueda comunicarse directamente con el sistema operativo, traduciendo así finalmente el código objeto a código máquina, y generando un módulo ejecutable.

Programación e ingeniería del software.

Existe una tendencia a identificar el proceso de creación de un programa informático con la programación, que es cierta cuando se trata de programas pequeños para uso personal, y que dista de la realidad cuando se trata de grandes proyectos.

El proceso de creación de software, desde el punto de vista de la ingeniería, incluye los siguientes pasos:

1. Reconocer la necesidad de un programa para solucionar un problema o identificar la posibilidad de automatización de una tarea.
2. Recoger los requisitos del programa. Debe quedar claro qué es lo que debe hacer el programa y para qué se necesita.
3. Realizar el análisis de los requisitos del programa. Debe quedar claro cómo debe realizar el programa las cosas que debe hacer. Las pruebas que comprueben la validez del programa se pueden especificar en esta fase.
4. Diseñar la arquitectura del programa. Se debe descomponer el programa en partes de complejidad abordable.
5. Implementar el programa. Consiste en realizar un diseño detallado, especificando completamente todo el funcionamiento del programa, tras lo cual la codificación debería resultar inmediata.
6. Implantar (instalar) el programa. Consiste en poner el programa en funcionamiento junto con los componentes que pueda necesitar (bases de datos, redes de comunicaciones, etc.).

Objetivos de la programación.

La programación debe perseguir la obtención de programas de calidad. Para ello se establece una serie de factores que determinan la calidad de un programa. Algunos de los factores de calidad más importantes son los siguientes:

- **Corrección.** Un programa es correcto si hace lo que debe hacer tal y como se estableció en las fases previas a su desarrollo.

- Claridad. Es muy importante que el programa sea lo más claro y legible posible, para facilitar así su desarrollo y posterior mantenimiento.
- Eficiencia. Se trata de que el programa, además de realizar aquello para lo que fue creado (es decir, que sea correcto), lo haga gestionando de la mejor forma posible los recursos que utiliza.
- Portabilidad. Un programa es portable cuando tiene la capacidad de poder ejecutarse en una plataforma, ya sea hardware o software, diferente a aquélla en la que se elaboró.

2.4 HIPÓTESIS.

El Desarrollo e Implantación de un Sistema de Gestión utilizando software libre mejorará los cobros y registros de usuarios de la Junta de Aguas Chacón Sevilla.

2.5 VARIABLES.

2.5.1 VARIABLE INDEPENDIENTE.

Sistema de gestión utilizando software libre.

2.5.2 VARIABLE DEPENDIENTE.

Cobros y registros de usuarios de la Junta Chacón Sevilla.

CAPÍTULO III.

3.1 ENFOQUE.

La presente investigación tendrá un enfoque cuali-cuantitativo porque se realizará una investigación de todas las causas y factores referentes al tema del proyecto, y la información proporcionada servirá como referencia para interpretarla como un sustento científico y profesional así como el tratamiento estadísticos de los datos, con lo que se pretende solucionar el problema.

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.

Se realizará una investigación bibliográfica - documental para poder obtener información más profunda con respecto a problemas similares, de esta manera se recopilará información valiosa que servirá como sustento científico del proyecto.

3.2.3 PROYECTO FACTIBLE.

Se realizará una propuesta en base al desarrollo de un Sistema de Gestión utilizando software libre para cobros y registro de usuarios, que es un modelo práctico que permitirá solucionar los problemas detectados en la Junta de aguas Chacón Sevilla.

3.3 NIVELES O TIPOS DE INVESTIGACIÓN

La investigación a realizarse utilizará un nivel exploratorio para detectar las características del problema, determinar si es factible o no solucionarse.

3.4 POBLACIÓN Y MUESTRA.

3.4.1 POBLACIÓN.

La población con la que se va a trabajar en la recopilación de información es un total de 5 personas miembros de la Junta de Aguas Chacón Sevilla.

3.4.2 MUESTRA.

Se trabajará con la totalidad del universo.

3.5 OPERACIONALIZACIÓN DE VARIABLES.

Variable Independiente: Sistema de Gestión utilizando software libre

CONCEPTUALIZACIÓN	DIMENSIONES	INDICADORES	ÍTEMS	TÉCNICAS E INSTRUMENTOS
Son sistemas que optimizan los procesos internos de las empresas mediante un software que permite compartir la información de las áreas principales.	<p>Procesos Internos</p> <p>Software</p> <p>Información de Áreas</p>	<p>Entrada de datos</p> <p>Salida de datos</p> <p>Herramientas</p> <p>Administración</p> <p>Registros</p> <p>Usuarios</p> <p>Cobros</p>	<p>¿Cree usted que el servicio de cobros es rápido?</p> <p>¿La Junta utiliza herramientas informáticas para almacenar los datos?</p> <p>¿La información de los recursos económicos de la Junta se encuentra automatizada?</p> <p>¿Existe problemas con la información al momento de pagar por el servicio?</p>	<ul style="list-style-type: none"> • Encuestas

Tabla 1.0 Operacionalización de variable independiente

Variable Dependiente: Cobros y registro de usuarios

CONCEPTUALIZACIÓN	DIMENSIONES	INDICADORES	ÍTEMS	TÉCNICAS E INSTRUMENTOS
<p>Son derechos exigibles originados por ventas o prestación de servicios.</p> <p>Conjunto de datos almacenados en un sistema.</p> <p>Persona que utiliza un ordenador para realizar múltiples operaciones con distintos propósitos</p>	<p>Servicios</p> <p>Datos almacenados</p> <p>Ordenador</p> <p>Operaciones</p>	<p>Riego</p> <p>Cobros</p> <p>Estructura</p> <p>Hardware</p> <p>Software</p> <p>Facturas</p>	<p>¿El servicio de cobros que presta actualmente es eficiente?</p> <p>¿La junta almacena la información correctamente?</p> <p>¿La junta utiliza los suficientes equipos de cómputos para el servicio que presta?</p> <p>¿La junta lleva un control automatizado de las facturas emitidas?</p>	<ul style="list-style-type: none"> • Encuestas

Tabla 1.1 Operacionalización de variable dependiente

3.6 RECOLECCIÓN DE INFORMACIÓN.

3.6.1 PLAN DE RECOLECCIÓN DE INFORMACIÓN.

La técnica de investigación que se aplicó a los miembros de la Junta de aguas Chacón Sevilla para la recolección de la información es la encuesta.

3.7 PROCESAMIENTO Y ANÁLISIS.

3.7.1 PLAN DE PROCESAMIENTO DE INFORMACIÓN.

Lo primero que se realizará al recopilar la información, será seleccionar los datos que se requiere para el desarrollo del proyecto los mismos que serán analizados en relación con el problema para poder establecer las conclusiones respectivas asegurando que los datos sean lo más reales posibles.

3.7.2 PLAN DE ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS.

Los resultados obtenidos a través de las encuestas realizadas a los miembros que integran la Junta de aguas Chacón Sevilla 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.

La Junta de aguas Chacón Sevilla, actualmente no cuenta con un sistema acorde a sus necesidades, llevando a cabo un deficiente almacenamiento de información ocasionando problemas al momento de las recaudaciones del servicio de regadío, motivo por el que se ve en la necesidad de implementar un sistema de gestión que ayude a llevar un control automatizado de los cobros y el registro de usuarios.

En vista de la necesidad la Junta requiere del desarrollo e implantación del sistema de gestión para cobros y registro de usuarios.

4.2 ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS.

Para la realización del análisis e interpretación de resultados se aplicó 5 encuestas a miembros administrativos de la Junta de Aguas Chacón Sevilla.

4.2.1 ANÁLISIS DE LOS RESULTADOS DE LAS ENCUESTAS.

1. De qué manera se almacena la información financiera de las tarifas recaudadas y de los usuarios?

Objetivo:

Determinar cómo se encuentra almacenada la información financiera de las tarifas recaudadas para tener una idea clara de la necesidad primordial de la Junta.

CUADRO PORCENTUAL PREGUNTA 1

RESPUESTA	CANTIDAD	PORCENTAJE
Manualmente	3	60%
Archivos Digitales	0	0%
De ambas formas	2	40%
Ninguna	0	0%
Total	5	100%

Tabla 1.2 Cuadro porcentual pregunta 1

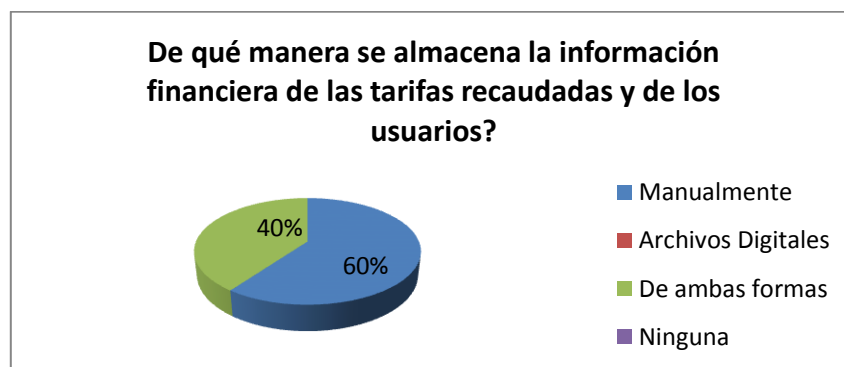


Figura 1.6 Gráfico porcentual pregunta 1

Interpretación:

La gráfica refleja claramente que el 60% de los miembros de la Junta encuestados respondieron que la información se almacena manualmente, mientras que 40% respondieron que De ambas formas.

Análisis:

La mayoría de los miembros encuestados manifestaron que en la Junta de Aguas Chacón Sevilla se almacena la información en forma manual.

2. ¿La Junta Chacón Sevilla cuenta con un Sistema Informático de cobros para las recaudaciones?

Objetivo:

Determinar la necesidad de desarrollo e implementación de un Sistema Gestión para el control de los cobros y el registro de usuarios.

CUADRO PORCENTUAL PREGUNTA 2

RESPUESTA	CANTIDAD	PORCENTAJE
Si	0	0%
No	5	100%
Total	5	100%

Tabla 1.3 Cuadro porcentual pregunta 2

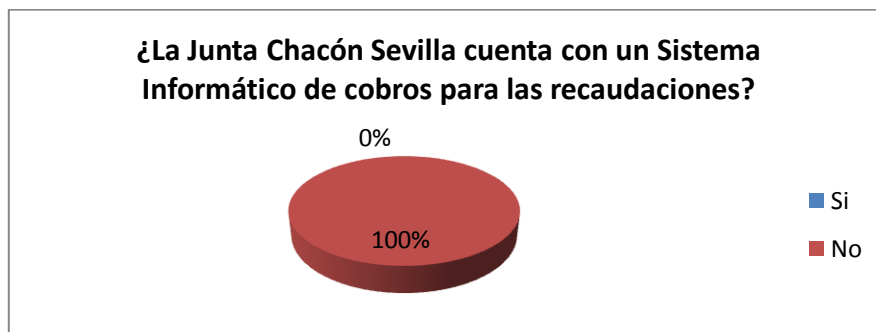


Figura 1 .7 Gráfico porcentual pregunta 2

Interpretación:

La gráfica refleja que el 100% de los miembros de la Junta encuestados respondieron que no cuentan con un sistema Informático para cobros.

Análisis:

Se demuestra que en la Junta de Aguas Chacón Sevilla no cuenta con un Sistema informático que ayude a mejorar los cobros, lo cual confirma la necesidad de automatizar este proceso.

3. Qué programas informáticos utilizan para almacenar la información?

Objetivo:

Verificar si la Junta posee algún tipo programa para almacenar la información.

CUADRO PORCENTUAL PREGUNTA 3

RESPUESTA	CANTIDAD	PORCENTAJE
Excel	4	80%
Sistema de Facturación	0	0%
Otros	0	0%
Ninguno	1	20%
Total	5	100%

Tabla 1.4 Cuadro porcentual pregunta 3

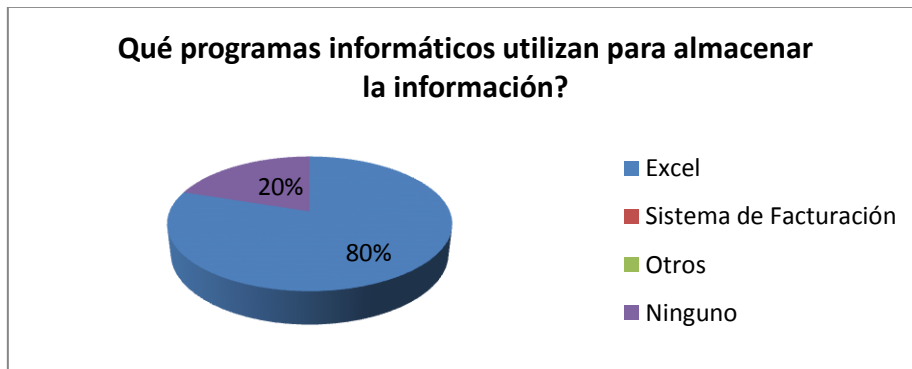


Figura 1.8 Gráfico porcentual pregunta 3

Interpretación:

La gráfica indica que el 80% de los miembros de la Junta encuestados respondieron que utilizan el Excel para almacenar la información, mientras que el 20% respondieron que no utilizan ninguno.

Análisis:

Se demuestra que en la Junta de Aguas Chacón Sevilla utiliza el Excel como herramienta para almacenar la Información.

4. ¿La forma de administrar los recursos económicos de la Junta es?

Objetivo:

Determinar cómo administran los recursos económicos en la Junta Chacón Sevilla.

CUADRO PORCENTUAL PREGUNTA 4

RESPUESTA	CANTIDAD	PORCENTAJE
Muy Buena	0	0%
Buena	1	20%
Regular	4	80%
Total	5	100%

Tabla 1.5 Cuadro porcentual pregunta 4

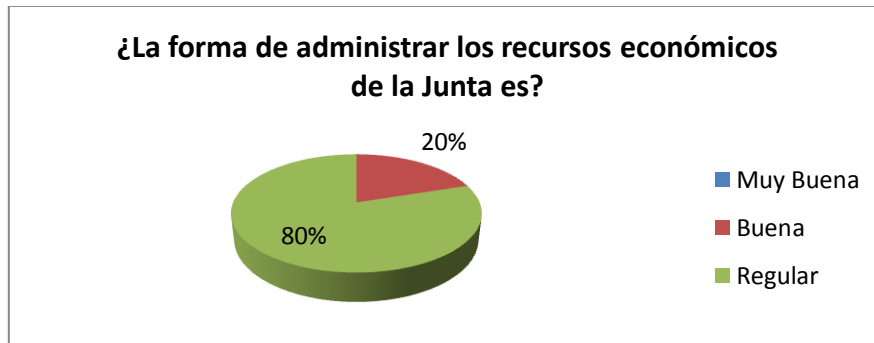


Figura 1.9 Gráfico porcentual pregunta 4

Interpretación:

La gráfica refleja que el 80% de los miembros de la Junta encuestados respondieron que la forma de administrar los recursos económicos es Regular, mientras que el 20% manifiesta que la administración es Buena.

Análisis:

La mayoría de miembros de la Junta encuestados están consientes de que la forma de de administrar la información es Regular.

5. ¿Con que frecuencia tienen problemas al momento de recaudar el dinero por consumo de agua de riego?

Objetivo:

Conocer si existen problemas en la recaudación por el servicio de consumo del agua de regadío.

CUADRO PORCENTUAL PREGUNTA 5

RESPUESTA	CANTIDAD	PORCENTAJE
Siempre	3	60%
Ocasionalmente	2	40%
Nunca	0	0%
Total	5	100%

Tabla 1.6 Cuadro porcentual pregunta 5

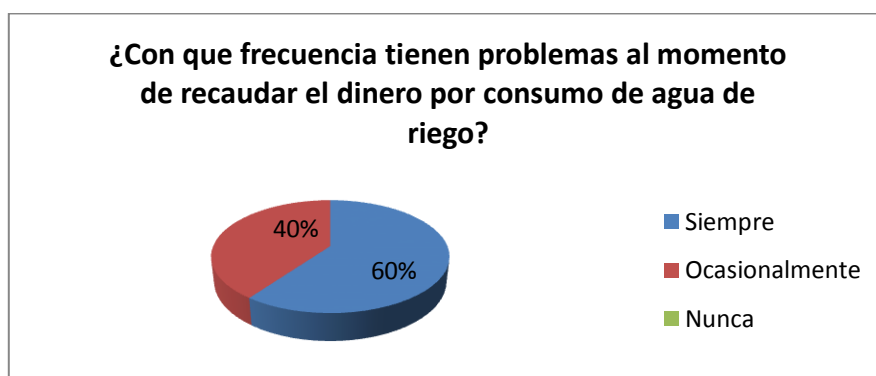


Figura 2.0 Gráfico porcentual pregunta 5

Interpretación:

La gráfica refleja que el 60% de los miembros de la Junta encuestados manifiestan que siempre se presentan problemas al momento de realizar la recaudación por concepto de consumo del agua de regadío; mientras que el 40% indica que ocasionalmente tiene inconvenientes.

Análisis:

Los encuestados mayoritariamente manifiestan que se presentan problemas en la recaudación de los valores por consumo de agua de regadío.

6. El sistema o mecanismo utilizado actualmente para los cobros de tarifas dificulta el proceso de recaudación?

Objetivo:

Determinar si el mecanismo actual de la Junta para el proceso de los cobros dificulta las recaudaciones.

CUADRO PORCENTUAL PREGUNTA 6

RESPUESTA	CANTIDAD	PORCENTAJE
Si	5	100%
No	0	0%
Total	5	100%

Tabla 1.7 Cuadro porcentual pregunta 6

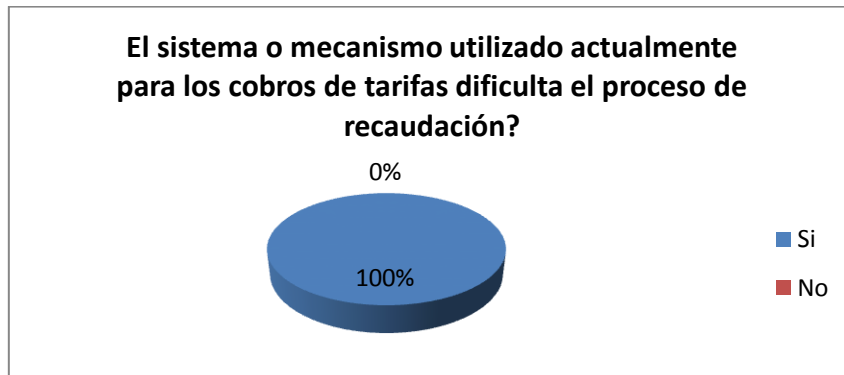


Figura 2.1 Gráfico porcentual pregunta 6

Interpretación:

La gráfica refleja que el 100% de los miembros encuestados manifestaron que el mecanismo utilizado actualmente para los cobros de tarifas dificulta el proceso de recaudación.

Análisis:

La mayoría de los miembros de la Junta encuestados están consientes el mecanismo actual para cobros dificultan las recaudaciones.

7. Considera usted que la información actual de los usuarios es confiable:

Objetivo:

Determinar si la información de los usuarios es segura.

CUADRO PORCENTUAL PREGUNTA 7

RESPUESTA	CANTIDAD	PORCENTAJE
Si	0	0%
No	5	100%
Total	5	100%

Tabla 1.8 Cuadro porcentual pregunta 7



Figura 2.2 Gráfico porcentual pregunta 7

Interpretación:

La gráfica refleja que el 100% de los miembros encuestados manifiestan que la información actual de los usuarios no es confiable.

Análisis:

Los miembros de la junta encuestados están consientes que la información actual de los usuarios no es confiable ya que el almacenamiento es manual y están expuestos a pérdidas de la información.

8. ¿La información de los usuarios de la Junta se encuentra almacenada en alguna Base de Datos?

Objetivo:

Conocer si la información se almacena en una Base de Datos.

CUADRO PORCENTUAL PREGUNTA 8

RESPUESTA	CANTIDAD	PORCENTAJE
Si	0	0%
No	4	80%
Parcialmente	1	20%
Total	5	100%

Tabla 1.9 Cuadro porcentual pregunta 8

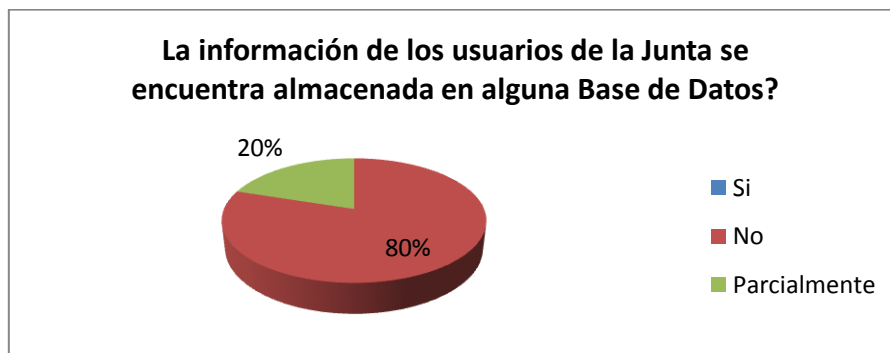


Figura 2.3 Gráfico porcentual pregunta 8

Interpretación:

La gráfica refleja que el 80% de los miembros encuestados manifiestan que la información de los usuarios no se encuentra en una base de datos, mientras que 20% responden que Parcialmente.

Análisis:

La mayoría de los miembros de la Junta manifestaron que la información de los usuarios no se almacena en una base de datos.

9.- ¿La Junta posee los suficientes Equipos de Cómputo para el Servicio que presta?

Objetivo:

Determinar si existen los suficientes recursos informáticos para el servicio prestado.

CUADRO PORCENTUAL PREGUNTA 9

RESPUESTA	CANTIDAD	PORCENTAJE
Si	0	0%
No	5	100%
Total	5	100%

Tabla 2.0 Cuadro porcentual pregunta 9

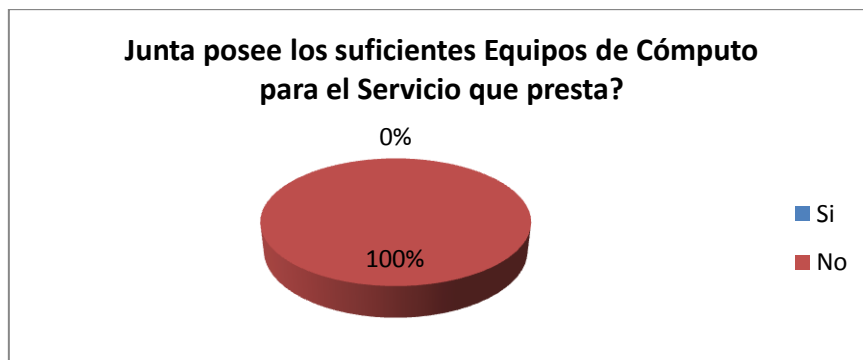


Figura 2.4 Gráfico porcentual pregunta 9

Interpretación:

La gráfica refleja que el 100% de los miembros encuestados manifiestan que no poseen los suficientes equipos de cómputo para el servicio prestado.

Análisis:

Los miembros de la junta están consientes que no cuentan con los suficientes equipos de cómputo para el servicio prestado ya que en la actualidad posee un solo computador.

10.- ¿Considera Ud. que implementado un Sistema de Gestión para cobros y registro de usuarios mejorará la productividad de la Junta y evitará la Información inexacta?

Objetivo:

Conocer las necesidades de la Junta para mejorar la productividad de la misma.

CUADRO PORCENTUAL PREGUNTA 10

RESPUESTA	CANTIDAD	PORCENTAJE
Si	5	100%
No	0	0%
Total	5	100%

Tabla 2.1 Cuadro porcentual pregunta 10

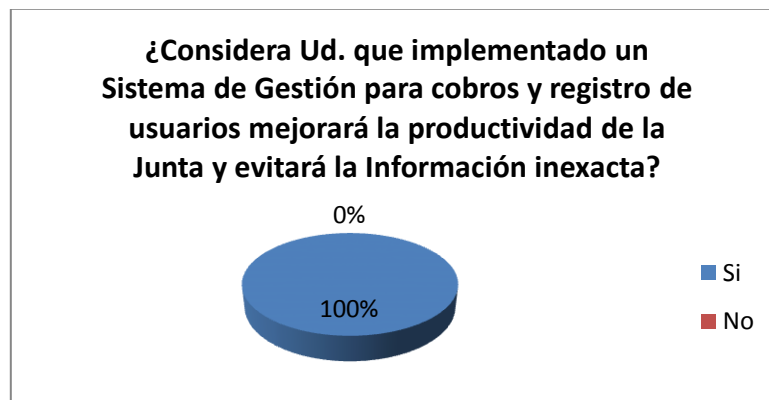


Figura 2.5 Gráfico porcentual pregunta 10

Interpretación:

La gráfica refleja que el 100% de los miembros encuestados consideran que un Sistema de Gestión de cobros y registro de usuarios mejorará la productividad de la Junta y evitara problemas.

Análisis:

Los miembros de la Junta manifiestan su interés por contar con un Sistema de Gestión de cobros y registro de usuarios que les permita tener información real y confiable para la toma de decisiones rápidas.

CAPÍTULO V.

CONCLUSIONES Y RECOMENDACIONES.

5.1 CONCLUSIONES.

- En la actualidad no se realiza un control adecuado sobre los recursos económicos de la Junta Chacón Sevilla.
- Debido a que el sistema se implementará en una entidad pública, requiere ser desarrollado mediante herramientas libres, razón por la que se escogió a SharpDevelop y SQL Server Express, herramientas de código abierto que brindan seguridad, confiabilidad en la información, resultando fácil de aprender su utilización.
- SharpDevelop es un entorno de desarrollo integrado libre, que permite hacer grades aplicaciones el mismo que ayudará con el desarrollo de todos los requerimientos que necesitan la Junta.
- SQL Server Express es una herramienta gráfica de administración gratuita y fácil de usar que permite almacenar grandes cantidades de datos.
- Trabajar con las personas responsables del manejo de recursos económicos de la Junta, facilitó la recopilación de información el cual servirá de gran ayuda para solucionar los problemas de la Institución.
- Al estar la información almacenada en una base de datos resultará fácil su recuperación, aumentará los niveles de seguridad y confiabilidad de los datos.

5.2 RECOMENDACIONES.

- Se recomienda analizar la herramienta seleccionada para el desarrollo del sistema revisando las características como: licencias, soporte de sistemas operativos, soporte de gestores de base de datos, entre otros.
- Todas las personas encargadas de la administración de los datos deben estar capacitadas sobre el funcionamiento del “Sistema de Gestión para cobros y registro de usuarios”, para evitar errores y optimizar el tiempo.
- Se recomienda presentar avances a los interesados después de cada fase del desarrollo del sistema para evitar errores del mismo.
- En caso de requerir un nuevo módulo para el sistema, se recomienda la investigación de herramientas para el desarrollo de software que sean bajo licencias libres.

CAPÍTULO VI.

PROPUESTA.

6.1 TEMA.

SISTEMA DE GESTIÓN UTILIZANDO SOFTWARE LIBRE PARA COBROS Y REGISTROS DE USUARIOS DE LA JUNTA DE AGUAS CHACÓN SEVILLA.

6.2 DATOS INFORMATIVOS.

Institución Ejecutoriada:	Junta de Aguas Chacón Sevilla.
Beneficiarios:	Usuarios y miembros administrativos de la Junta.
Ciudad:	Ambato.
Dirección:	Parroquia de Izamba Modesto Chacón y Rafael García, Barrio San Francisco de Asís.
Investigador:	Luis Israel Mesías Haro.
Tiempo:	22 de Septiembre 2011 hasta 22 de Marzo 2012
Tutor:	Ing. Franklin Mayorga.

6.3 ANTECEDENTES.

Desde sus inicios la Junta de aguas Chacón Sevilla, ha venido realizando el proceso de recaudaciones del servicio de regadío y el registros de los usuarios de manera manual, ocasionados inconsistencias y pérdidas de información, esto ha generado un malestar en los usuarios, ya que son los principales afectados de esta problemática que tiene su génesis en la carencia del uso de tecnologías que ayuden a mejorar estos procesos.

Partiendo de estos inconvenientes surge la necesidad de desarrollar un sistema de gestión con la finalidad de garantizar un control eficiente del servicio, y permitir que los procesos que llevan a cabo, se los ejecute de forma rápida y segura, proporcionando una mejor atención al usuario.

Por ser la Junta Chacón Sevilla una institución pública y basándose en la política de estado, se ha optado por realizar el proyecto con tecnologías de bajo costo como es la utilización de software libre para el desarrollo e implantación del sistema.

Teniendo en cuenta la exigencia en cuanto a seguridad, fiabilidad, veracidad de la información se decidió establecer como SGBD a SQL Server Express y como lenguaje de programación a SharpDevelop siendo ambas herramientas libres.

6.4 JUSTIFICACIÓN.

Las necesidades de implementación tecnológica en todos los procesos empresariales optimizan el tiempo como todos los recursos, siendo estas una importante herramienta de apoyo a la gestión administrativa.

La Implantación de un Sistema de Gestión para cobros y registro de usuarios tiene como objetivo organizar, manipular, almacenar de manera ordenada la información y optimizar el tiempo en las recaudaciones de las tarifas, para poder brindar servicio de calidad.

6.5 OBJETIVOS

6.5.1 OBJETIVO GENERAL

- Desarrollar un Sistema de Gestión mediante el uso de software libre para cobros y registro de usuarios de la Junta de Aguas Chacón Sevilla.

6.5.2 OBJETIVOS ESPECÍFICOS

- Analizar las necesidades administrativas y financieras de la Junta de Aguas Chacón Sevilla, para definir las características operacionales del sistema a desarrollarse.
- Diseñar la estructura de la base de datos y probar la veracidad de la información.
- Implantar el Sistema de Gestión que permita el control de los cobros y el registro de los usuarios.

6.6 ANÁLISIS DE FACTIBILIDAD

6.6.1 FACTIBILIDAD OPERATIVA

Debido a que el Sistema fue desarrollado de acuerdo a las necesidades de los usuarios, este interactúa directamente con el usuario, por lo tanto no se requiere de una capacitación extensa para poder utilizarlo; además de esto cuenta con interfaces sencillas, amigables y fáciles donde podrán realizar el proceso de cobros, dando a conocer los valores que deberán pagar los usuarios por el servicio de regadío, de esta forma a los administradores de la Junta se les hace más fácil obtener la información financiera para poder tomar decisiones rápidas y oportunas.

El sistema cuenta con dos tipos de usuarios (Administrador, Secretario(a))

Administrador.- Es la persona encargada de ingresar, eliminar, modificar y anular los datos de las tablas.

Secretario(a)- Es la persona encargada de la facturación, consultas, e informes.

6.6.2 FACTIBILIDAD ECONÓMICA

Es factible el proyecto, ya que se lo desarrollo con herramientas de Software Libre, puesto que no tienen ningún costo para la adquisición del lenguaje.

Con el desarrollo de este Sistema de Gestión la Junta de Aguas Chacón Sevilla, controlará de mejor manera sus recursos económicos, optimizando el tiempo, y teniendo así una mejora en la atención de los usuarios.

6.6.3 FACTIBILIDAD TÉCNICA

Para la creación del sistema de gestión para cobros y registro de usuarios se manejó SharpDevelop herramienta que nos permite configurar un entorno de desarrollo de aplicaciones Windows bajo licencia GPL y SQL Server Express como motor de base de datos.

SOFTWARE

- Lenguaje de programación SharpDevelop.
- Como motor de base de datos SQL Server Express.
- Crystal Report Redistribuable para la creación de reportes.

HARDWARE

La Junta Chacón Sevilla cuenta con una computadora de escritorio en el cual se realizan los cobros del servicio de agua de riego, y aunque este Equipo no sea de última tecnología el sistema correrá perfectamente ya que este no cuenta con procesos pesados que sobrecarguen el uso de memoria.

6.7 FUNDAMENTACIÓN.

6.7.1 PROGRAMACIÓN ORIENTADA A OBJETOS.

La **Programación Orientada a Objetos (POO u OOP** según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo y encapsulamiento. Su uso se popularizó a principios de la década de 1990. Actualmente son muchos los lenguajes de programación que soportan la orientación a objetos.

Los objetos son entidades que combinan estado, comportamiento e identidad:

- El estado está compuesto de datos, será uno o varios atributos a los que se habrán asignado unos valores concretos (datos).
- El comportamiento está definido por los procedimientos o métodos con que puede operar dicho objeto, es decir, qué operaciones se pueden realizar con él.
- La identidad es una propiedad de un objeto que lo diferencia del resto, dicho con otras palabras, es su identificador (concepto análogo al de identificador de una variable o una constante).

La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener, reutilizar y volver a utilizar.

Los métodos (comportamiento) y atributos (estado) están estrechamente relacionados por la propiedad de conjunto. Esta propiedad destaca que una clase requiere de métodos para poder tratar los atributos con los que cuenta. El programador debe pensar indistintamente en ambos conceptos, sin separar ni darle mayor importancia a ninguno de ellos. Hacerlo podría producir el hábito erróneo de crear clases contenedoras de información por un lado y clases con métodos que manejen a las primeras por el otro. De esta manera se estaría realizando una programación estructurada camuflada en un lenguaje de programación orientado a objetos.

Esto difiere de la programación estructurada tradicional, en la que los datos y los procedimientos están separados y sin relación, ya que lo único que se busca es el procesamiento de unos datos de entrada para obtener otros de salida. La programación estructurada anima al programador a pensar sobre todo en términos de procedimientos o funciones, y en segundo lugar en las estructuras de datos que esos procedimientos manejan. En la programación estructurada sólo se escriben funciones que procesan datos. Los programadores que emplean éste nuevo paradigma, en cambio, primero definen objetos para luego enviarles mensajes solicitándoles que realicen sus métodos por sí mismos.

6.7.2 SHARPDEVELOP

"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#.

6.7.3 CRYSTAL REPORTS

Crystal Reports es una aplicación de inteligencia empresarial utilizada para diseñar y generar informes desde una amplia gama de fuentes de datos.

Varias aplicaciones, como Microsoft Visual Studio, incluyen una versión OEM de Crystal Reports como una herramienta de propósito general del informes/reportes. Crystal

Reports se convirtió en el escritor de informes estándar cuando Microsoft lo liberó con Visual Basic.

Incorpora la posibilidad de crear contenido interactivo con calidad de presentación al entorno de Windows. Con Crystal Reports se puede crear informes complejos y profesionales en un programa basado en GUI. Después puede conectar el informe a casi todos los orígenes de base de datos, así como a datos proxy, como un conjunto de resultados por ejemplo, un ADO.NET DataSet. Los asistentes del diseñador de GUI le permiten establecer fácilmente los criterios de formato, agrupamiento y gráficos, etc.

Puede almacenar el informe en una aplicación Web o para Windows, con uno de los controles de visores de Crystal Reports para Visual Studio 2005. La presentación de informes, tanto en clientes Windows como en HTML 3.2 ó 4.0, es muy interactiva y proporciona funciones como la profundización en gráficos, la exploración de informes y la búsqueda de texto.

6.8 METODOLOGÍA

Para el presente sistema se ha procedido a utilizar la metodología de desarrollo de software denominado Modelo de Cascada o Ciclo de Vida Clásico, dicho paradigma consiste en descomponer la actividad global en fases que se van desarrollando en forma lineal, es decir una detrás de la otra de tal forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior; esta metodología permite tener una mejor administración y llegar al objetivo propuesto.

Para el análisis del sistema se utilizó UML (Lenguaje Unificado de Modelado) el cual es un conjunto de notaciones y diagramas para modelar sistemas orientados a objetos; dando diferentes perspectivas a un sistema.

6.9 MODELO OPERATIVO

6.9.1 ANÁLISIS DEL SISTEMA

6.9.1.1 ANÁLISIS Y REQUERIMIENTOS DEL SISTEMA

De los resultados obtenidos en la encuesta que se aplicaron a los miembros de la Junta de Aguas Chacón Sevilla se determinaron las necesidades de estos organismos.

Se realizó reuniones con los representantes de la Junta para establecer los requerimientos del sistema y evaluar las alternativas de solución planteadas.

Entre los requerimientos solicitados por los beneficiarios está la seguridad a la información, estableciendo restricciones de usuarios, delegando permisos a cada uno de ellos y se determino de acuerdo a las funciones que realiza cada miembro.

Luego de haber analizado los requerimientos que solicita el departamento se ha determinado que el sistema deberá contar con:

- Un módulo para ingreso, modificación, eliminación y control de: tarifas recaudadas.
- Un módulo de seguridad que permita el acceso a la información solo a personas autorizadas.
- Un módulo egresos para registrar los gastos de la Junta de aguas.
- Generación de reportes de: facturas emitidas, contribuciones, actividades, aportes extras.
- Además, el sistema deberá contar con interfaces sencillas y amigables para su manipulación.

6.9.1.1.1 DIAGRAMA DE CASOS DE USO

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

CASO DE USO SISTEMA COBROS



Figura 2.6 Diagrama de casos de uso Sistema de Cobros

Caso de uso: Ingresar al Sistema

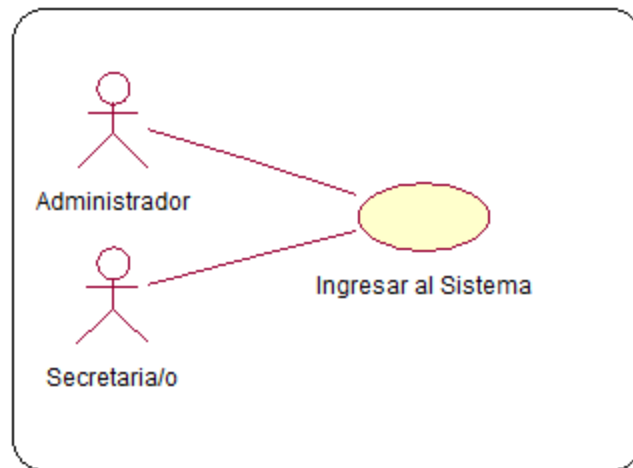


Figura 2.7 Caso de uso individual Ingresar al Sistema

Especificación caso de uso: Ingresar al Sistema

	Caso de uso: Ingresar al Sistema
Actores:	Usuario: Administrador. Usuario: Secretaria(o).
Descripción:	Validar a un usuario ya registrado para el uso del sistema.
Precondiciones:	El usuario debe estar registrado en el sistema.
Pos condiciones:	Validación realizada con éxito.
Flujo normal de eventos	Validar <ol style="list-style-type: none"> 1. El usuario ingresa el nombre de usuario y contraseña. 2. El sistema valida los datos ingresados por el usuario. 3. Una vez validado el usuario, el sistema muestra el menú opciones.
Flujos alternos	Usuario no registrado <ol style="list-style-type: none"> 1. En el paso 2 del flujo normal, si el usuario no existe en el sistema se muestra un mensaje donde se indica que los datos de usuario o contraseña son erróneos.

Tabla 2.2 Especificación ingresar al sistema

Caso de uso: Administrar Junta

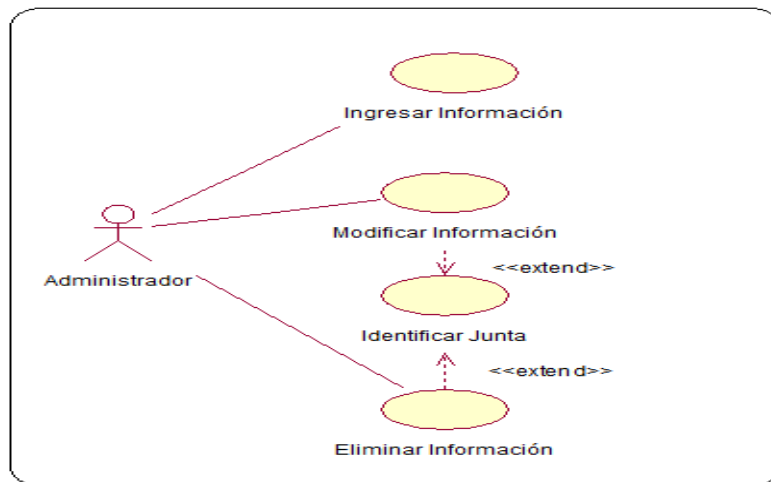


Figura 2.8 Caso de uso individual administrar junta

Especificación caso de uso: Administrar Junta

	Caso de uso: Administrar Junta
Actores:	Usuario: Administrador.
Descripción:	Agregar, modificar, eliminar Junta del sistema.
Precondiciones:	El usuario debe estar registrado como administrador. Datos de permisos ingresados con anterioridad.
Pos condiciones:	Resultados almacenados.
Flujo normal de eventos	<p>Agregar</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción administrar Junta. 2. El sistema muestra las opciones: agregar, modificar, eliminar. 3. El usuario escoge la opción agregar. 4. El sistema muestra los campos para ingresar la información de la junta. 5. El usuario ingresa la información. 6. El sistema verifica el ingreso correcto de la información.
Flujos alternos	<p>Modificar</p> <ol style="list-style-type: none"> 7. En el paso 3 del Flujo Normal, el usuario selecciona la opción modificar. 8. El usuario muestra los campos con la información de la junta a modificar. 9. El usuario modifica la información. <p>Eliminar</p> <ol style="list-style-type: none"> 10. En el paso 3 Flujo Normal, el usuario selecciona la opción eliminar. 11. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación.

Tabla 2.3 Especificación Administrar Junta

Caso de uso: Administrar Óvalos

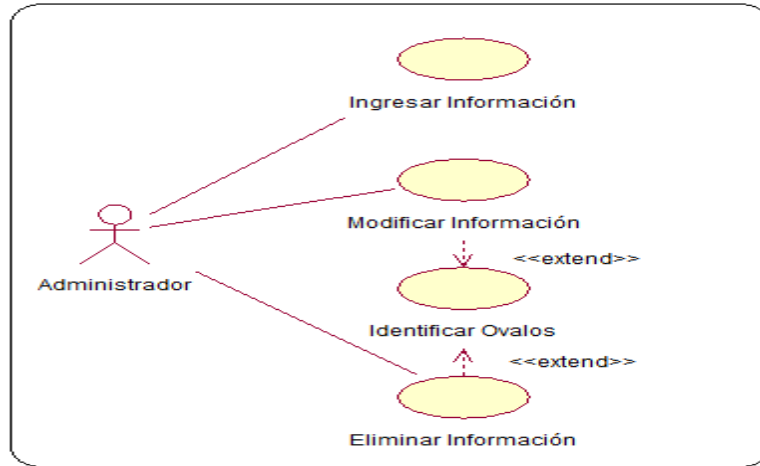


Figura 2.9 Caso de uso individual administrar óvalos

Especificación caso de uso: Administrar Óvalos

Caso de uso: Administrar Óvalos	
Actores:	Usuario: Administrador.
Descripción:	Agregar, modificar, eliminar Óvalos del sistema.
Precondiciones:	El usuario debe estar registrado como administrador. Datos de permisos ingresados con anterioridad.
Pos condiciones:	Resultados almacenados.
Flujo normal de eventos	<p>Agregar</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción administrar Óvalos. 2. El sistema muestra las opciones: agregar, modificar, eliminar. 3. El usuario escoge la opción agregar. 4. El sistema muestra los campos para ingresar la información del ovalo. 5. El usuario ingresa la información. 6. El sistema verifica el ingreso correcto de la información.
Flujos alternos	<p>Modificar</p> <ol style="list-style-type: none"> 7. En el paso 3 del Flujo Normal, el usuario selecciona la opción modificar. 8. El usuario muestra los campos con la información del óvalo a modificar. 9. El usuario modifica la información. <p>Eliminar</p> <ol style="list-style-type: none"> 10. En el paso 3 Flujo Normal, el usuario selecciona la opción eliminar. 11. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación.

Tabla 2.4 Especificación Administrar Ovalo

Caso de uso: Administrar Usuarios

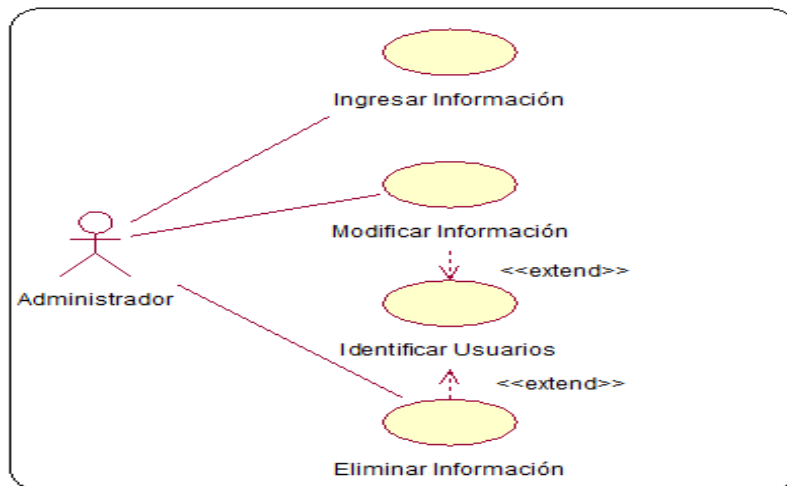


Figura 3.0 Caso de uso individual administrar usuarios.

Especificación caso de uso: Administrar Usuarios

Caso de uso: Administrar Usuarios	
Actores:	Usuario: Administrador.
Descripción:	Agregar, modificar, eliminar Usuarios del sistema.
Precondiciones:	El usuario debe estar registrado como administrador. Datos de permisos ingresados con anterioridad.
Pos condiciones:	Resultados almacenados.
Flujo normal de eventos	<p>Agregar</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción administrar Usuarios. 2. El sistema muestra las opciones: agregar, modificar, eliminar. 3. El usuario escoge la opción agregar. 4. El sistema muestra los campos para ingresar la información de los usuarios. 5. El usuario ingresa la información. 6. El sistema verifica el ingreso correcto de la información.
Flujos alternos	<p>Modificar</p> <ol style="list-style-type: none"> 7. En el paso 3 del Flujo Normal, el usuario selecciona la opción modificar. 8. El usuario muestra los campos con la información de los usuarios a modificar. 9. El usuario modifica la información. <p>Eliminar</p> <ol style="list-style-type: none"> 10. En el paso 3 del Flujo Normal, el usuarios selecciona la opción eliminar. 11. El sistema muestra un mensaje donde indica si el usuario está seguro de la eliminación.

Tabla 2.5 Especificación Administrar Usuarios.

Caso de uso: Administrar Cuenta

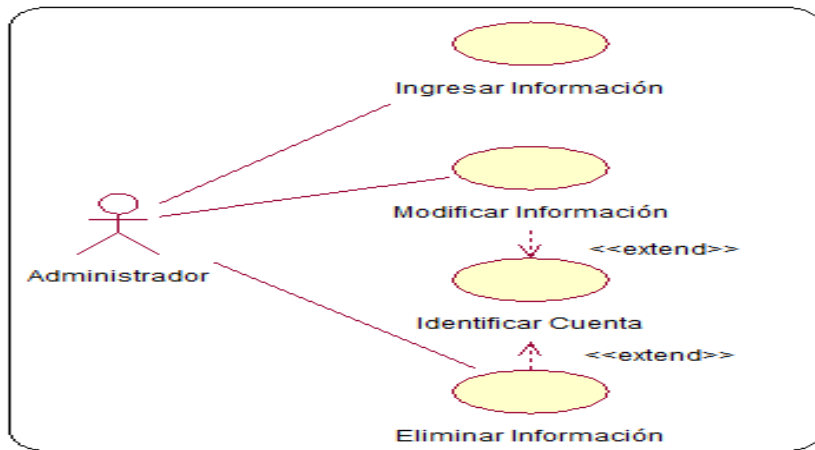


Figura 3.1 Caso de uso individual administrar cuenta.

Especificación caso de uso: Administrar Cuenta

	Caso de uso: Administrar Cuenta
Actores:	Usuario: Administrador.
Descripción:	Agregar, modificar, eliminar Cuenta del sistema.
Precondiciones:	El usuario debe estar registrado como administrador. Datos de permisos ingresados con anterioridad.
Pos condiciones:	Resultados almacenados.
Flujo normal de eventos	<p>Agregar</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción administrar Cuenta. 2. El sistema muestra las opciones: agregar, modificar, eliminar. 3. El usuario escoge la opción agregar. 4. El sistema muestra los campos para ingresar la información de la cuenta. 5. El usuario ingresa la información. 6. El sistema verifica el ingreso correcto de la información.
Flujos alternos	<p>Modificar</p> <ol style="list-style-type: none"> 7. En el paso 3 del Flujo Normal, el usuario selecciona la opción modificar. 8. El usuario muestra los campos con la información de la cuenta a modificar. 9. El usuario modifica la información. <p>Eliminar</p> <ol style="list-style-type: none"> 10. En el paso 3 del Flujo Normal, el usuario selecciona la opción eliminar. 11. El sistema muestra un mensaje donde indica si el usuario está seguro de la eliminación.

Tabla 2.6 Especificación Administrar Cuenta.

Caso de uso: Administrar Contribuciones

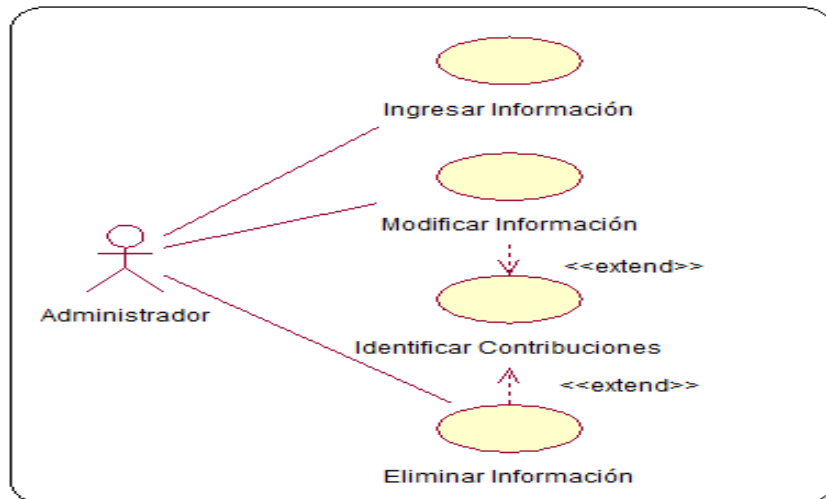


Figura 3.2 Caso de uso individual administrar contribuciones

Especificación caso de uso: Administrar Contribuciones

Caso de uso: Administrar Contribuciones	
Actores:	Usuario: Administrador.
Descripción:	Agregar, modificar, eliminar Contribuciones del sistema.
Precondiciones:	El usuario debe estar registrado como administrador. Datos de permisos ingresados con anterioridad.
Pos condiciones:	Resultados almacenados.
Flujo normal de eventos	<p>Agregar</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción administrar Contribuciones. 2. El sistema muestra las opciones: agregar, modificar, eliminar. 3. El usuario escoge la opción agregar. 4. El sistema muestra los campos para ingresar la información de las contribuciones. 5. El usuario ingresa la información. 6. El sistema verifica el ingreso correcto de la información.
Flujos alternos	<p>Modificar</p> <ol style="list-style-type: none"> 7. En el paso 3 del Flujo Normal, el usuario selecciona la opción modificar. 8. El usuario muestra los campos con la información de las contribuciones a modificar. 9. El usuario modifica la información. <p>Eliminar</p> <ol style="list-style-type: none"> 10. En el paso 3 del Flujo Normal, el usuario selecciona la opción eliminar. 11. El sistema muestra un mensaje donde indica si el usuario está seguro de la eliminación.

Tabla 2.7 Especificación Administrar Contribuciones.

Caso de uso: Administrar Actividades

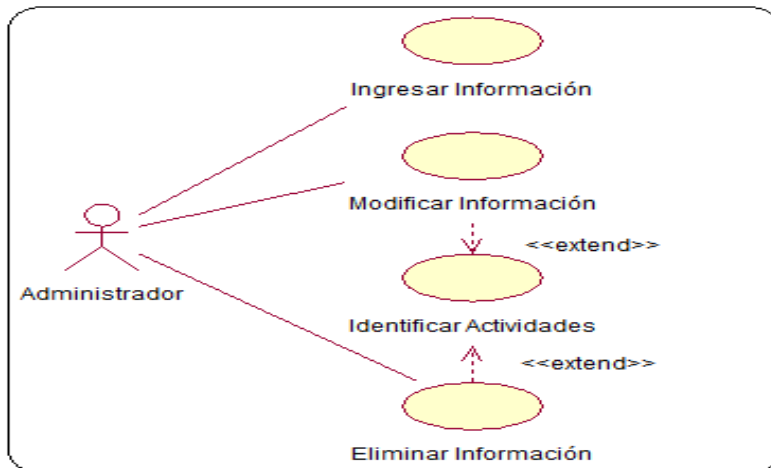


Figura 3.3 Caso de uso individual administrar Actividades

Especificación caso de uso: Administrar Actividades

Caso de uso: Administrar Actividades	
Actores:	Usuario: Administrador.
Descripción:	Agregar, modificar, eliminar Actividades del sistema.
Precondiciones:	El usuario debe estar registrado como administrador. Datos de permisos ingresados con anterioridad.
Pos condiciones:	Resultados almacenados.
Flujo normal de eventos	<p>Agregar</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción administrar Actividades. 2. El sistema muestra las opciones: agregar, modificar, eliminar. 3. El usuario escoge la opción agregar. 4. El sistema muestra los campos para ingresar la información de las actividades. 5. El usuario ingresa la información. 6. El sistema verifica el ingreso correcto de la información.
Flujos alternos	<p>Modificar</p> <ol style="list-style-type: none"> 7. En el paso 3 del Flujo Normal, el usuario selecciona la opción modificar. 8. El usuario muestra los campos con la información de las actividades a modificar. 9. El usuario modifica la información. <p>Eliminar</p> <ol style="list-style-type: none"> 10. En el paso 3 del Flujo Normal, el usuario selecciona la opción modificar. 11. El sistema muestra un mensaje donde indica si el usuario está seguro de la eliminación

Tabla 2.8 Especificación Administrar Actividades.

Caso de uso: Facturar

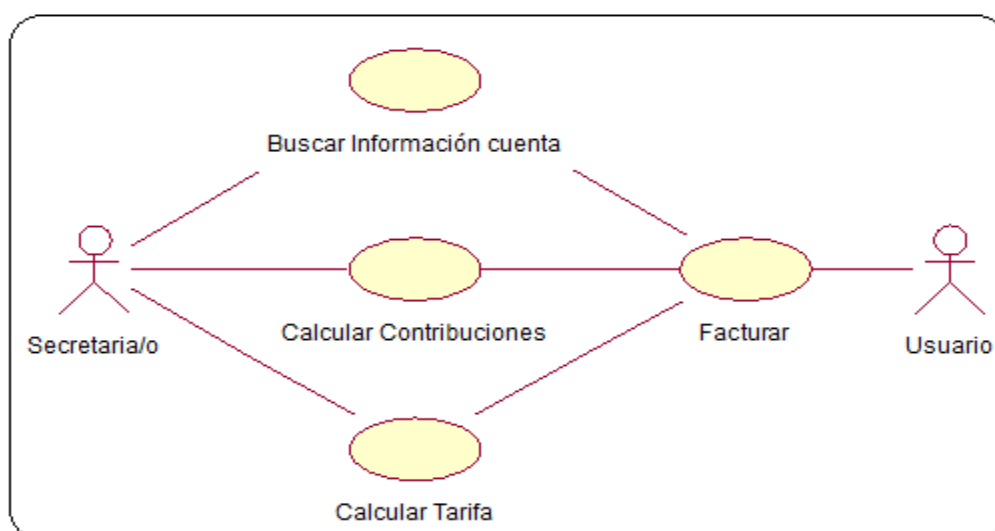


Figura 3.4 Caso de uso individual Facturar

Especificación caso de uso: Facturar

	Caso de uso: Facturar
Actores:	Usuario: Secretaria/o.
Descripción:	Buscar cuenta, calcular contribuciones, calcular tarifa, Guardar.
Precondiciones:	El usuario debe estar registrado como Secretaria/o. Datos de permisos ingresados con anterioridad.
Pos condiciones:	Resultados almacenados.
Flujo normal de eventos	<p>Buscar Cuenta</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción buscar cuenta. 2. El sistema muestra la información del las cuentas de los usuarios. <p>Calcular Contribuciones</p> <ol style="list-style-type: none"> 1. El usuario selecciona la opción calcular contribuciones. 2. El sistema muestra las contribuciones a cancelar.
Flujos alternos	<p>Calcular Tarifas</p> <ol style="list-style-type: none"> 3. En el paso 3 del Flujo Normal, el usuario selecciona la opción calcular tarifa. 4. El sistema muestra las tarifas por tiempo de regadío. <p>Guardar</p> <ol style="list-style-type: none"> 5. En el paso 3 del Flujo Normal, el usuario selecciona la opción guardar. 6. El sistema muestra un mensaje donde indica si desea guardar e imprimir la factura.

Tabla 2.9 Especificación Facturar.

6.9.1.1.2 PROTOTIPO DE INTERFAZ DE USUARIO

De acuerdo a las necesidades determinadas en Junta Chacón Sevilla, se procedió a realizar una interfaz sencilla y amigable, para que el usuario pueda manejar el sistema de la mejor manera.

Ventana Inicio de Sesión

Inicio de Sesión

Usuario:

Contraseña:

Aceptar Cancelar

Figura 3.5 Ventana Inicio de Sesión

En esta ventana el usuario que desea entrar al sistema debe iniciar sesión e ingresar el nombre de usuario y su contraseña datos que son validados y permiten su acceso; además se asignan los permisos de manipulación de datos concedidos al usuario.

Pantalla Menú Principal

Principal Datos Cobros Informes Finanzas Configuraciones Avuda

1

Sistema de cobros

Figura 3.6 Pantalla menú principal

1. En el Menú se desplegarán las opciones de: **Principal, Datos, Cobros Informes, Finanzas, Configuraciones, Ayuda.**

Descripción Opción Principal

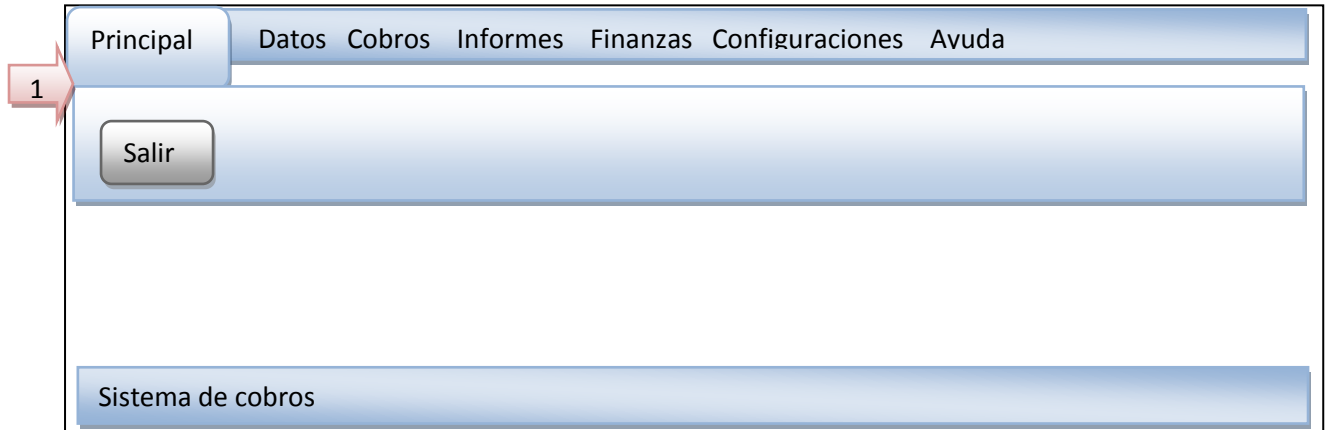


Figura 3.7 Pantalla opción principal

- La Opción **Principal** tiene un el Submenú Salir el cual permite salir del sistema.

Descripción Opción Datos

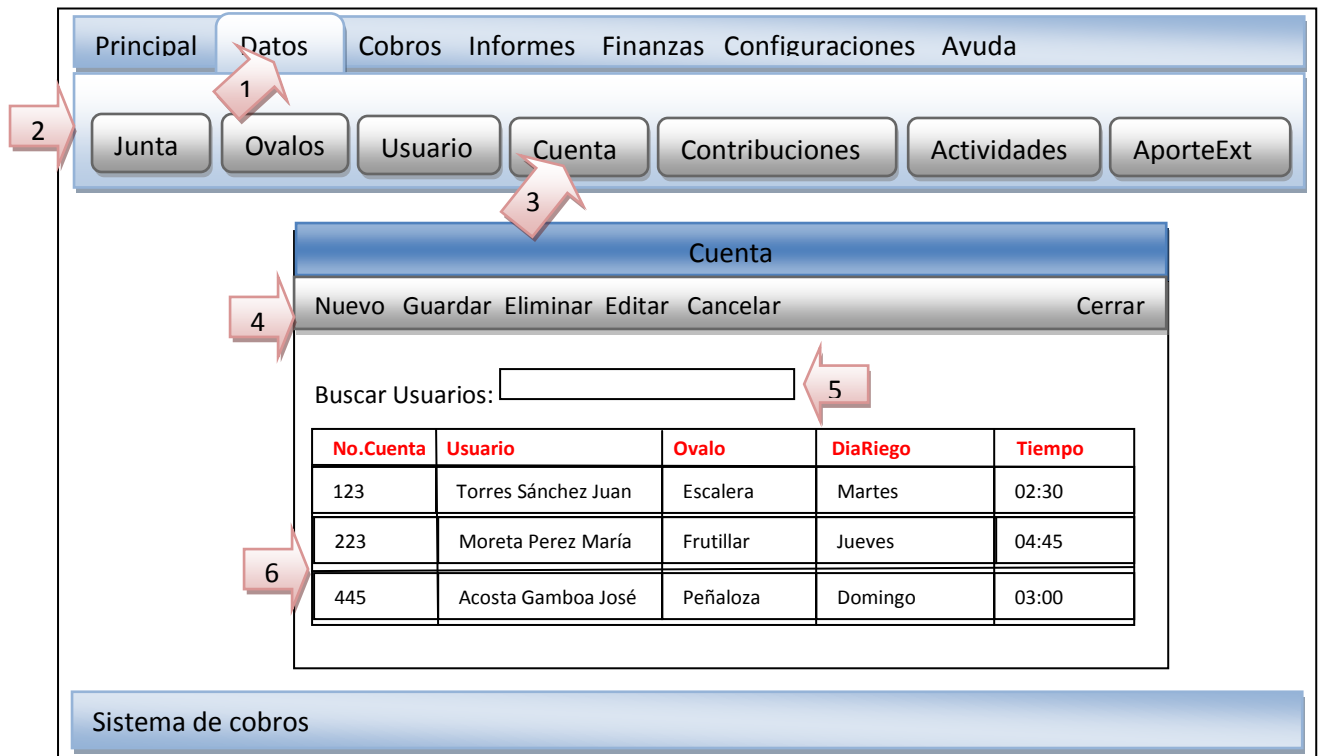


Figura 3.8 Pantalla opción datos

1. La Opción **Datos** contiene los Submenús **Junta, Óvalos, Usuarios, Contribuciones, Actividades, Aportes Extras**.
2. Muestra los Botones donde podremos administrar la Información siguiente:
 - Junta.-** Donde se registrará la información de la Junta.
 - Óvalos.-** Donde se registrará la información de los Óvalos.
 - Usuarios.-** Donde se registrará la información de los Usuarios.
 - Cuenta.-** Donde se registrará la información de las Cuentas.
 - Contribuciones.-** Donde se registrará la información de las Contribuciones.
 - Actividades.-** Donde se registrará la información de las actividades de los Usuarios.
 - Aportes Extras.-** Donde se registrará la información de los aportes extras de los Usuarios.
3. Muestra la información de las cuentas de todos los usuarios registrados.
4. Barra de Mantenimiento como **Nuevo, Guardar, Eliminar, Editar, Cancelar, Cerrar**, el cual permite administrar los datos del formulario Cuentas.
5. Caja de Texto que permite el buscar a los usuarios registrados.
6. Campos recuperados o que se van a recuperar.

Descripción Opción Cobros

Principal Datos **Cobros** Informes Finanzas Configuraciones Avuda

Factura Anular Factura

Factura

Cerrar

No. Factura:

Fecha Pago:

No. Cuenta: Usuario: CI/Ruc:

Ovalo: Día Riego: Hora Riego: **Tiempo Riego**

Total Contribuciones:

Buscar

Mingas/ Sesiones

Codigo	Concepto	Falta	Fecha	Costo

Total Actividades:

Aportes Extras

Codigo	Concepto	Año	Fecha	Costo

Total Aportes Extras:

Subtotal

IVA %

Total

Guardar Cancelar

Sistema de cobros

Figura 3.9 Pantalla opción cobros

1. La Opción **Cobros** contiene los Submenús **Factura**, **Anular Factura**.
2. **Factura**.- Donde se podrá facturar las contribuciones, actividades, aportes extras de las cuentas.
3. Barra de Mantenimiento **Cerrar**, el cual permite cerrar la Ventana.
4. Caja de Texto donde se debe ingresar el No de Factura.
5. Caja de Texto donde se debe ingresar el No de Cuenta de cada usuario.

6. Botón Buscar Permite Buscar las deudas de las contribuciones, aportes extras, actividades anteriores.
7. Caja de Texto donde se visualizará el total de las Contribuciones.
8. Caja de Texto donde se visualizará el total de las Actividades.
9. Caja de Texto donde se visualizará el total de las Aportes Extras.
10. Caja de Texto donde se visualizará el valor total de la Factura.
11. Botón que permite Guardar e Imprimir la Factura.
12. Botón que permite Cancelar la Factura.

Descripción Opción Informes



Figura 4.0 Pantalla opción informes

1. **Facturas Emitidas.**- Botón que permitirá visualizar todas las Facturas Emitidas de las Cuentas.
2. Barra de Mantenimiento **Cerrar**, el cual permite cerrar la Ventana.
3. Selección Fecha pagos Desde.
4. Selección Fecha pagos Hasta.

5. Muestra la información de las Facturas Emitidas
6. Botón que permite visualizar el informe de las facturas emitidas para la impresión.
7. Botón que permite Imprimir el Informe.

Descripción Opción Finanzas

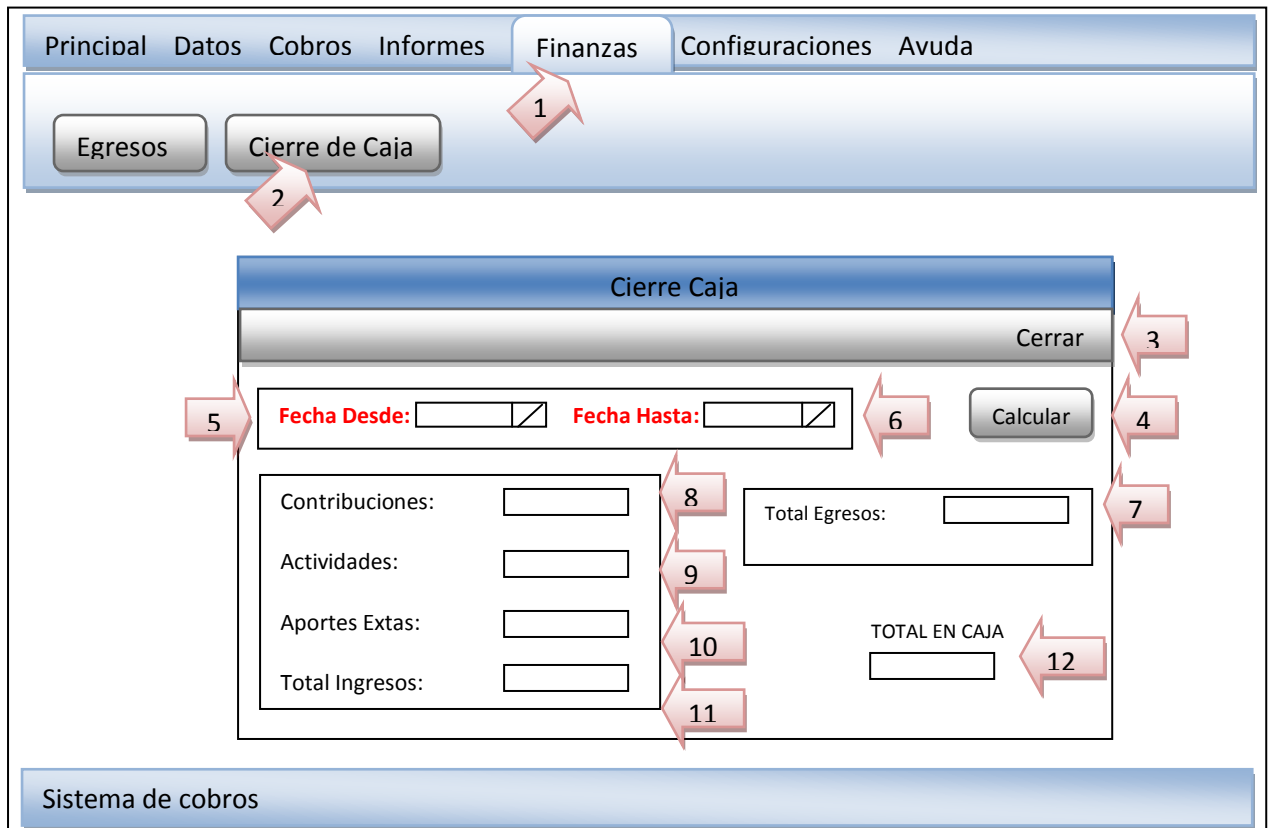


Figura 4.1 Pantalla opción finanzas

1. La Opción **Finanzas** contiene los Submenús **Egresos, Cierre de Caja**.
2. **Cierre de Caja**.- Botón que permitirá hacer un cierre de caja, es decir total en ingresos y gastos.
3. Barra de Mantenimiento **Cerrar**, el cual permite cerrar la Ventana.
4. Botón que permitirá calcular los totales.
5. Selección Fecha Desde.
6. Selección Fecha Hasta.
7. Caja de Texto donde se visualizará el valor total de los Egresos.
8. Caja de Texto donde se visualizará el valor de las Contribuciones.
9. Caja de Texto donde se visualizará el valor de las Actividades.
10. Caja de Texto donde se visualizará el valor de los Aportes Extras.

11. Caja de Texto donde se visualizará el valor total de Ingresos.

12. Caja de Texto donde se visualizará el Total en Caja.

Descripción Opción Configuraciones

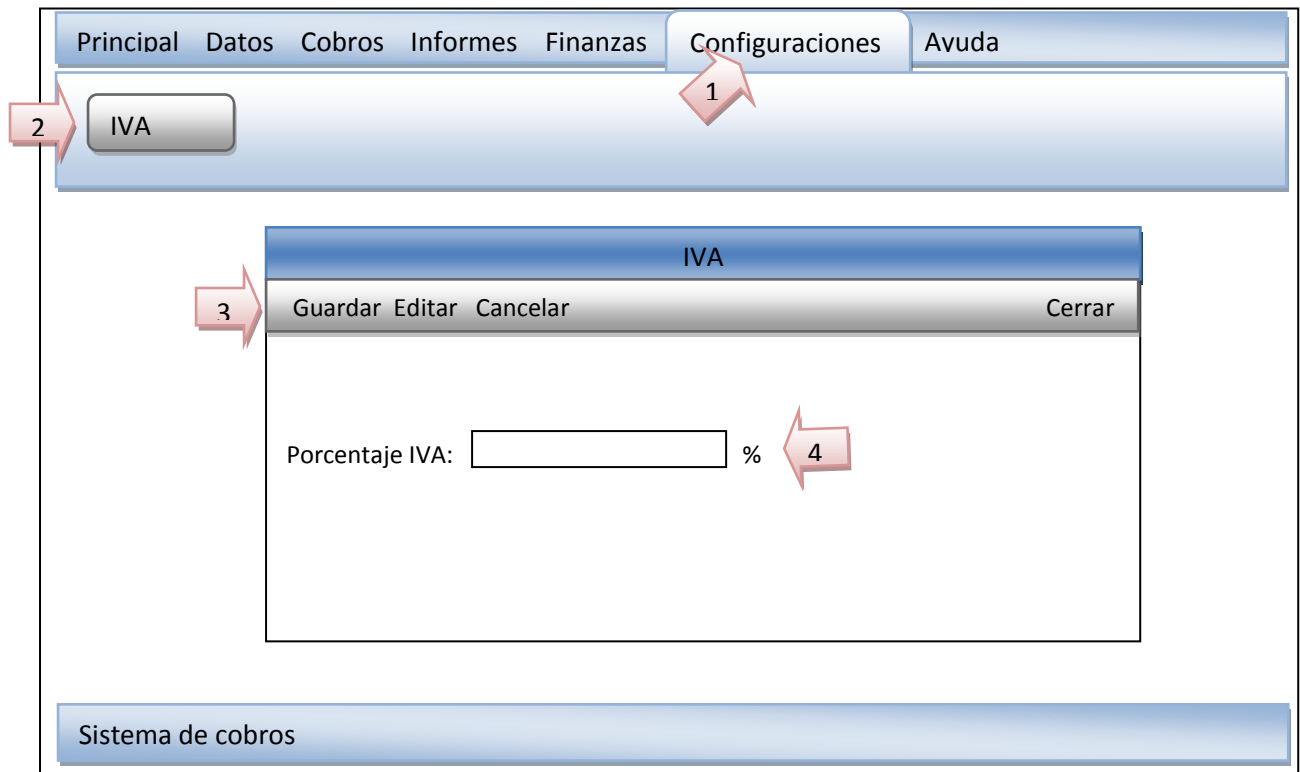


Figura 4.2 Pantalla opción Configuraciones

1. La Opción **Configuraciones** contiene los Submenús **IVA**.
2. **IVA**.- Botón que permitirá configurar el porcentaje IVA.
3. Barra de Mantenimiento como **Guardar, Editar, Cancelar, Cerrar**, el cual permite administrar los datos del formulario IVA.
4. Caja de Texto que permite el ingreso de porcentaje IVA.

Descripción Opción Ayuda



Figura 4.3 Pantalla opción ayuda

1. La Opción **Ayuda** contiene los Submenús **Ayuda, Acerca de**.
2. **Ayuda.-** Botón que permitirá visualizar la ayuda del sistema.
3. Permitirá visualizar los contenidos de la ayuda.
4. Botones que permitirán el desplazamiento de una página a otra.
5. Muestra la Información de la Ayuda.

6.9.1.1.3 DIAGRAMA DE CLASES

Son diagramas estáticos que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Se utilizan para modelar la visión estática de un sistema; esta visión soporta los requisitos funcionales del sistema, en concreto, los servicios que el sistema debería proporcionar a sus usuarios finales.

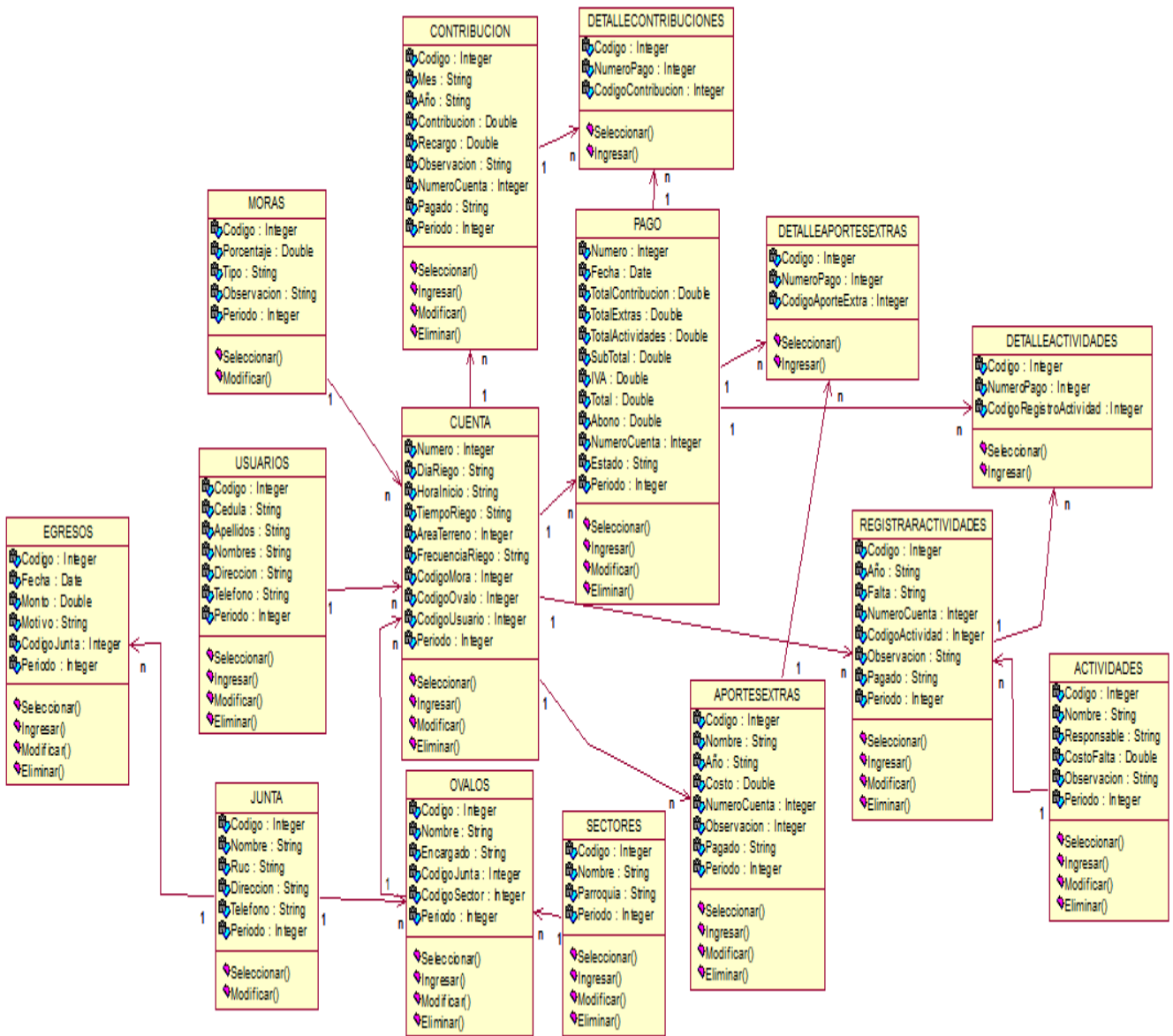


Figura 4.4 Diagrama de clases

6.9.1.1.4 DIAGRAMA DE SECUENCIAS

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso.

Junta

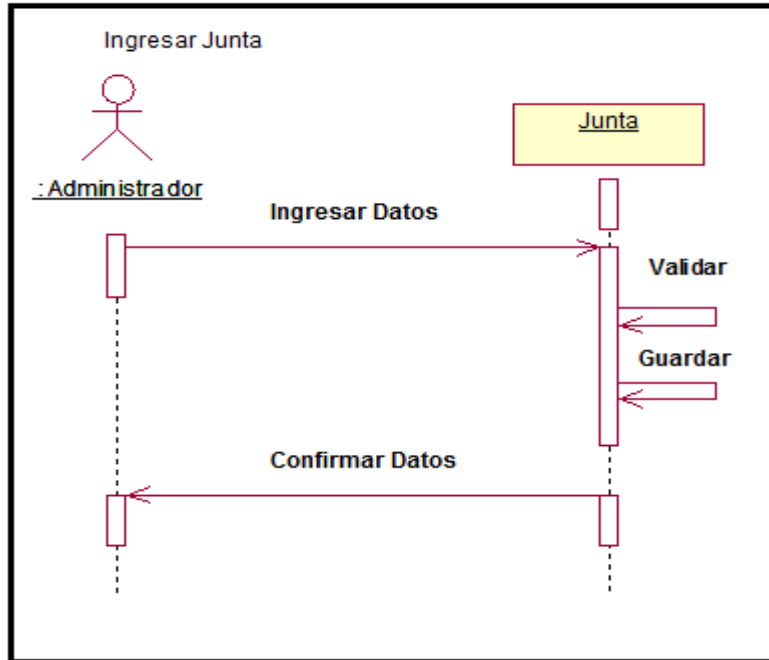


Figura 4.5 Diagrama de secuencia de Junta

Sector

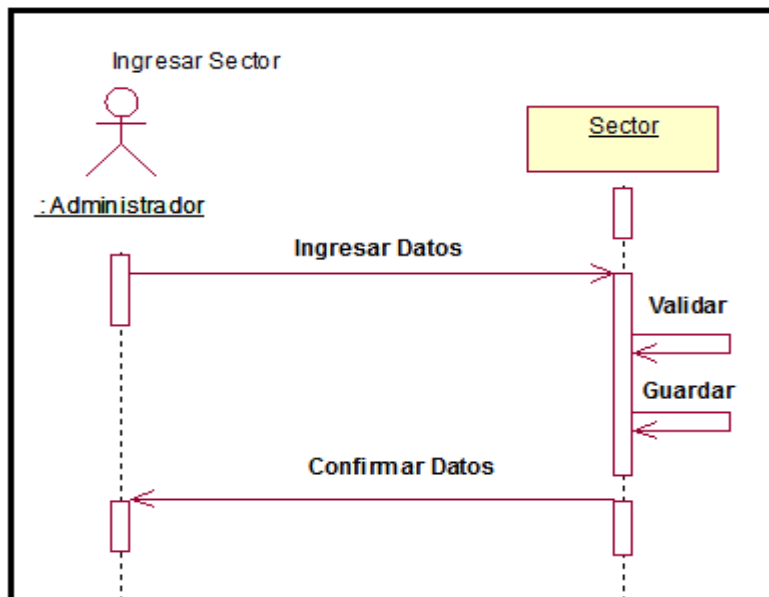


Figura 4.6 Diagrama de secuencia de Sector

Óvalos

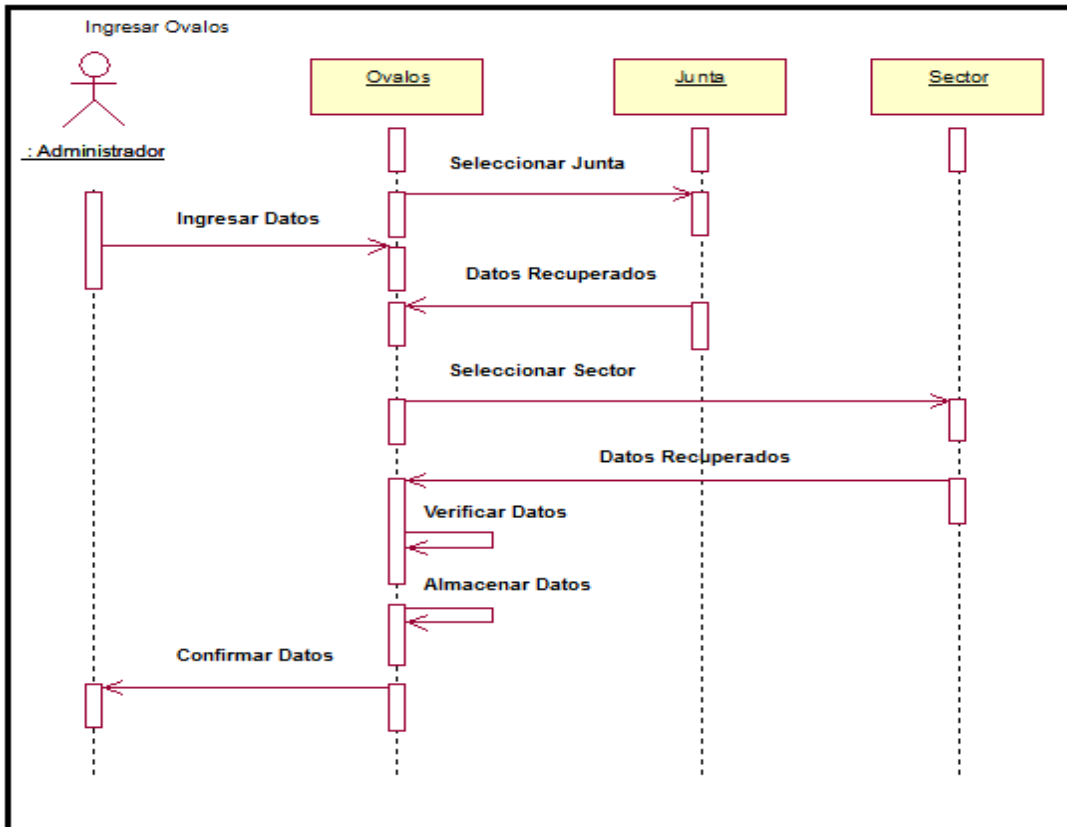


Figura 4.7 Diagrama de secuencia de Óvalos.

Usuarios

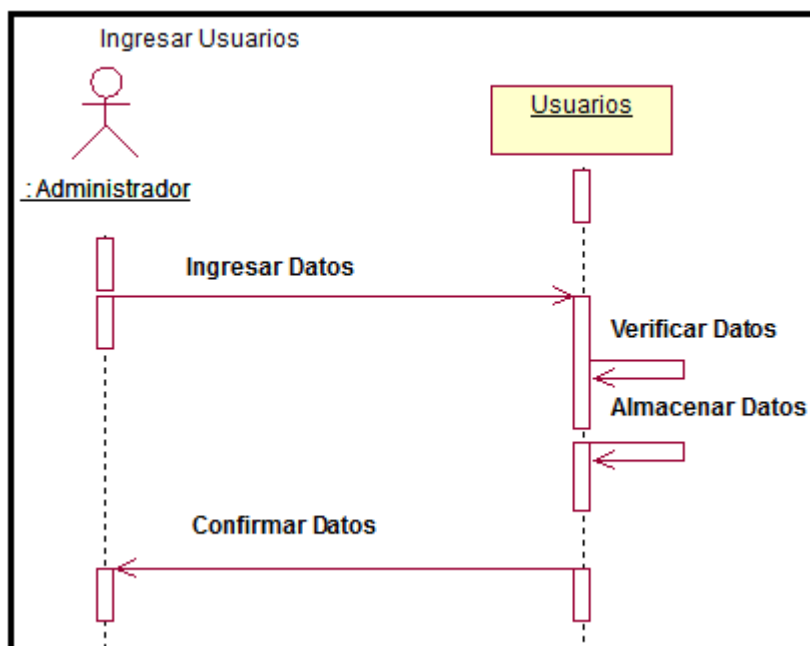


Figura 4.8 Diagrama de secuencia de Usuarios.

Cuentas

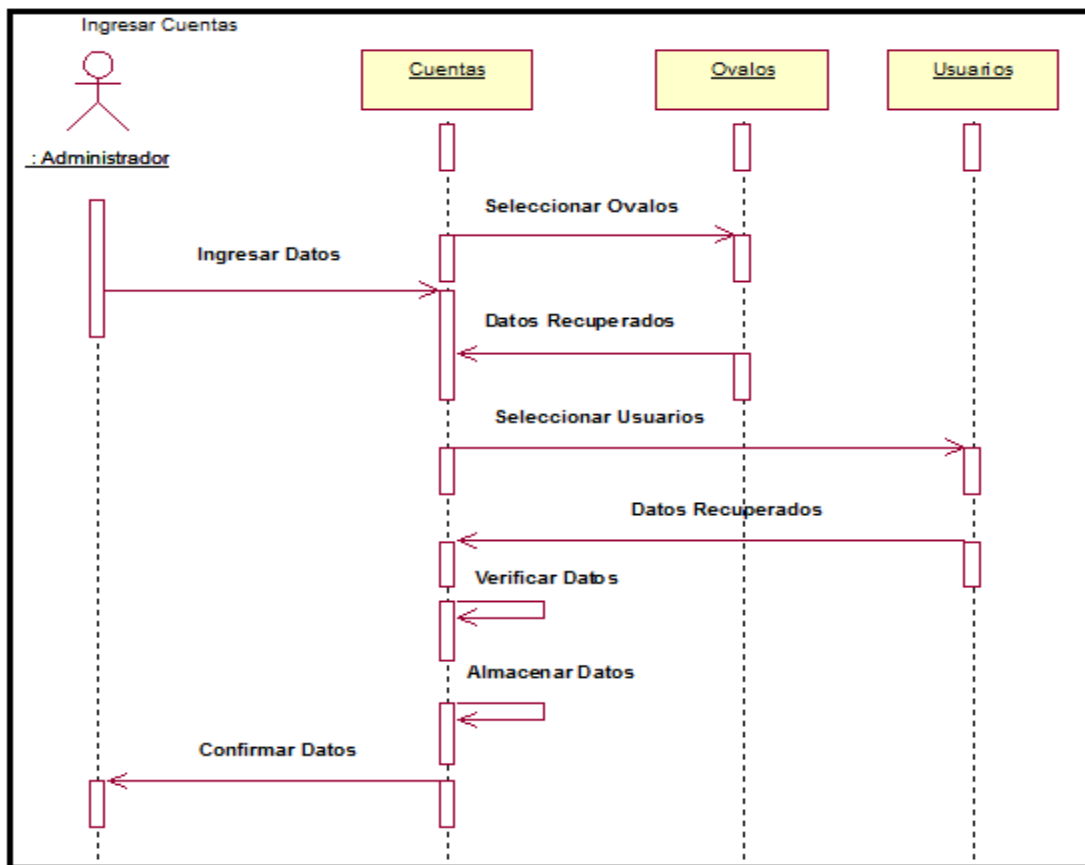


Figura 4.9 Diagrama de secuencia de Cuentas

Moras

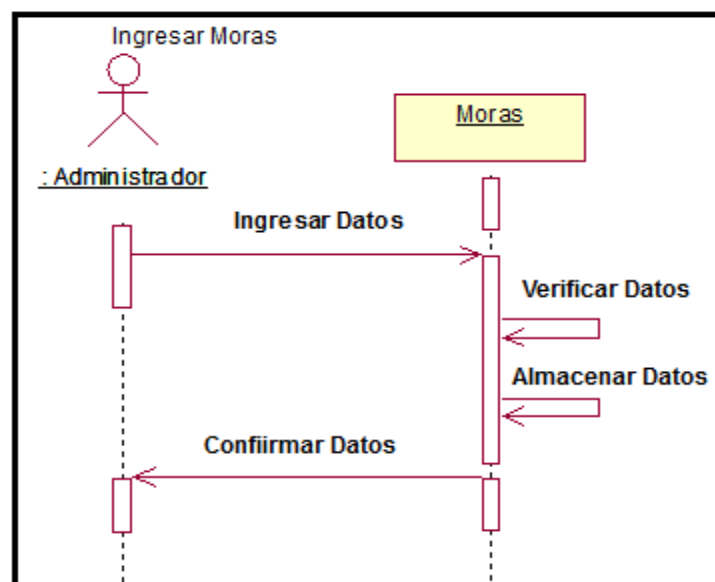


Figura 5.0 Diagrama de secuencia de Moras

Contribuciones

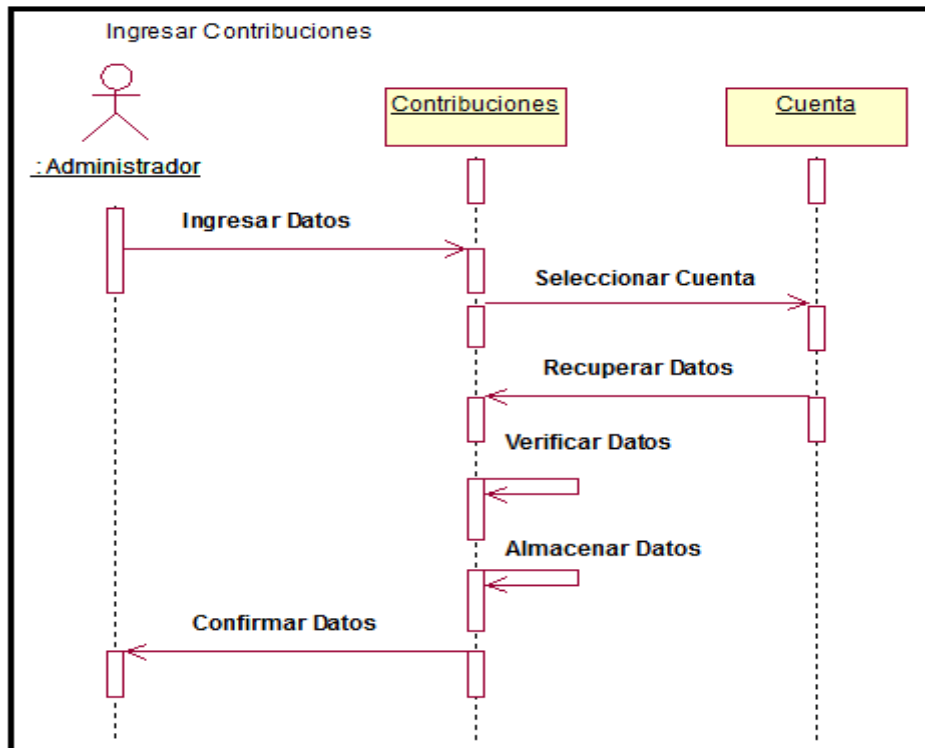


Figura 5.1 Diagrama de secuencia de Contribuciones

Actividades

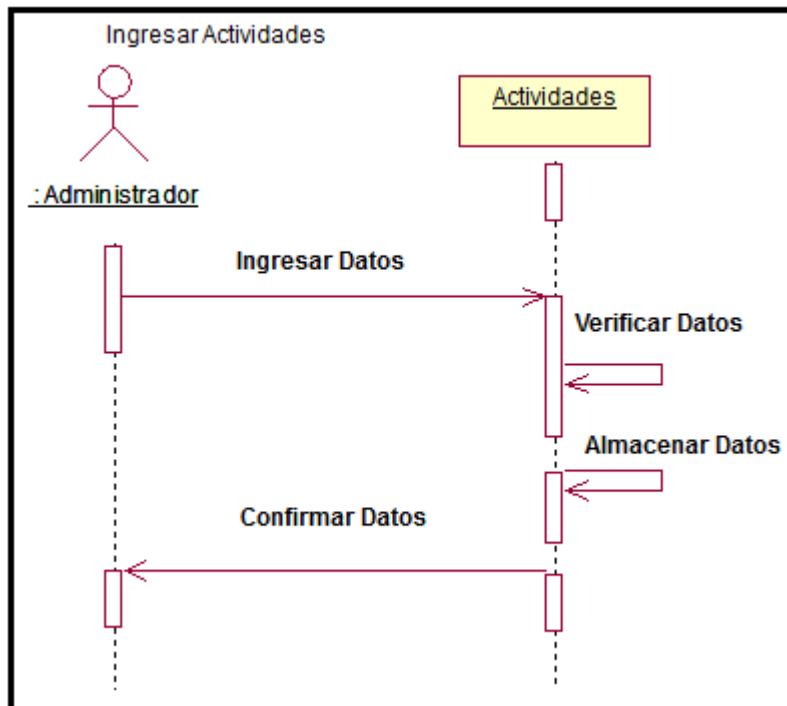


Figura 5.2 Diagrama de secuencia de Actividades.

Registrar Actividades

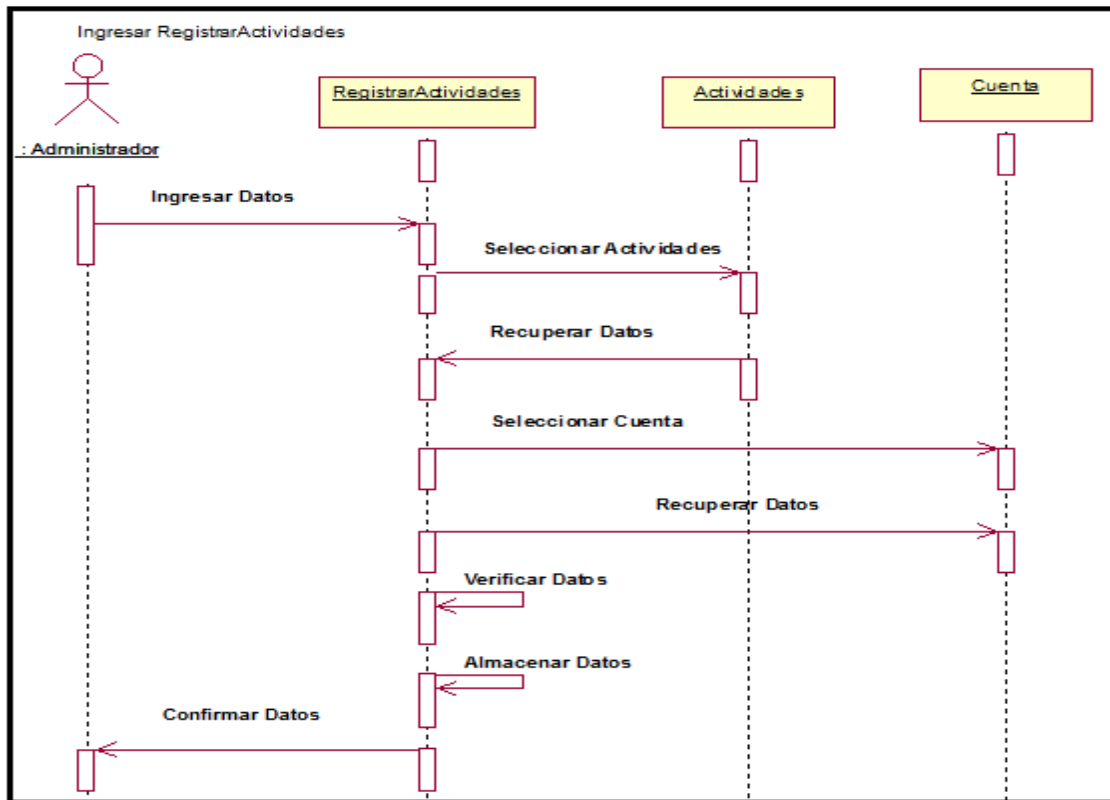


Figura 5.3 Diagrama de secuencia de RegistrarActividades

Aportes extras

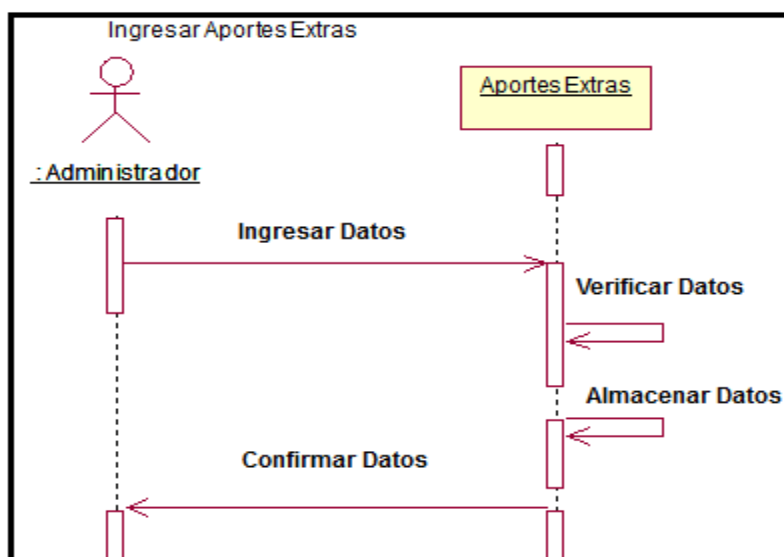


Figura 5.4 Diagrama de secuencia de Aportes Extras

Registrar Aportes Extras

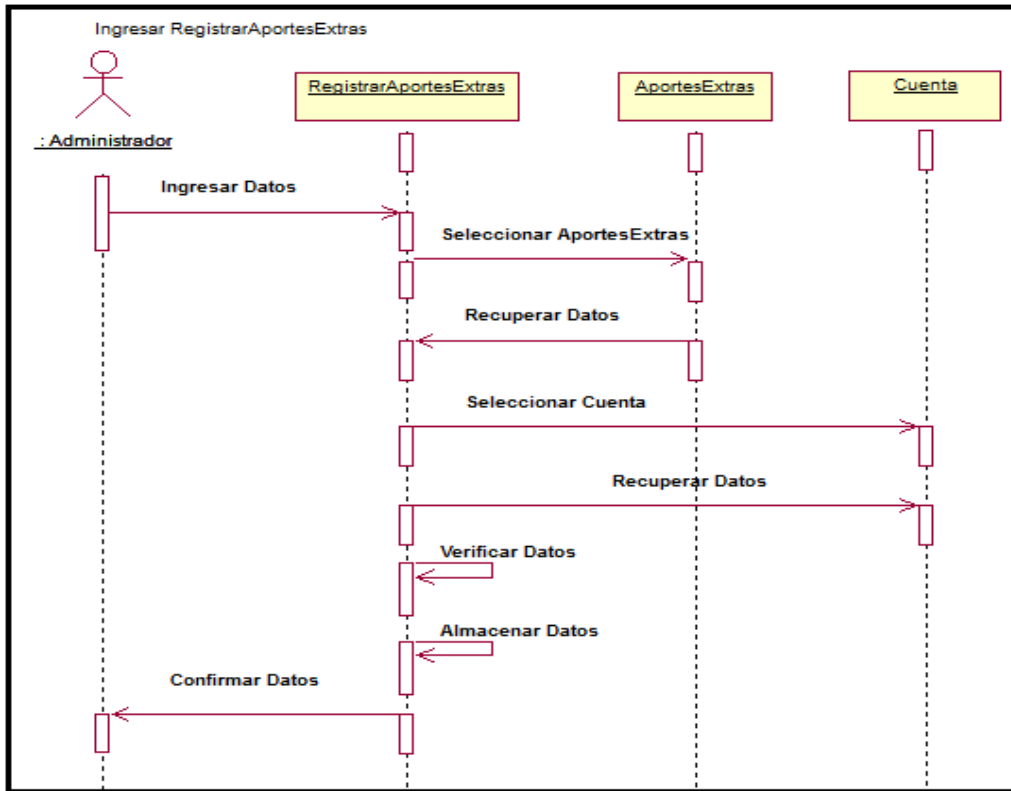


Figura 5.5 Diagrama de secuencia de Registrar Aportes Extras

Egresos

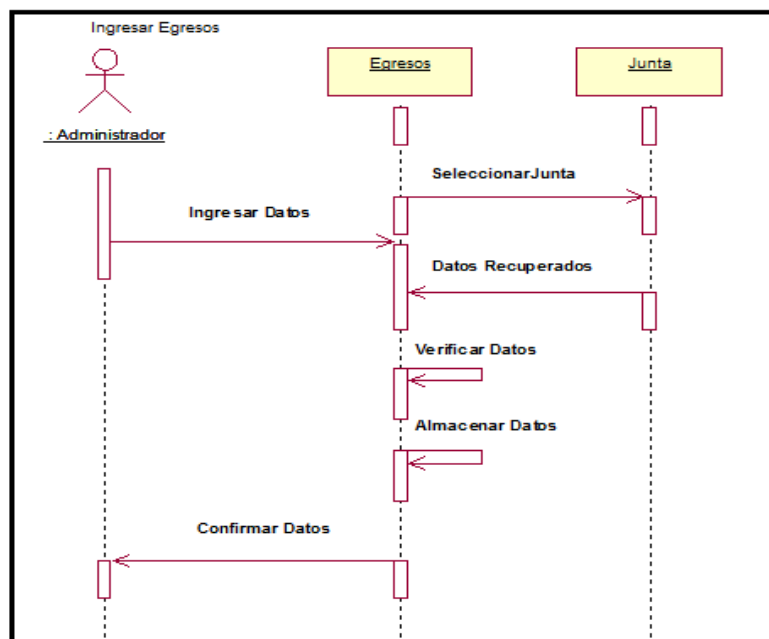


Figura 5.6 Diagrama de secuencia de Registrar Egresos

Pagos

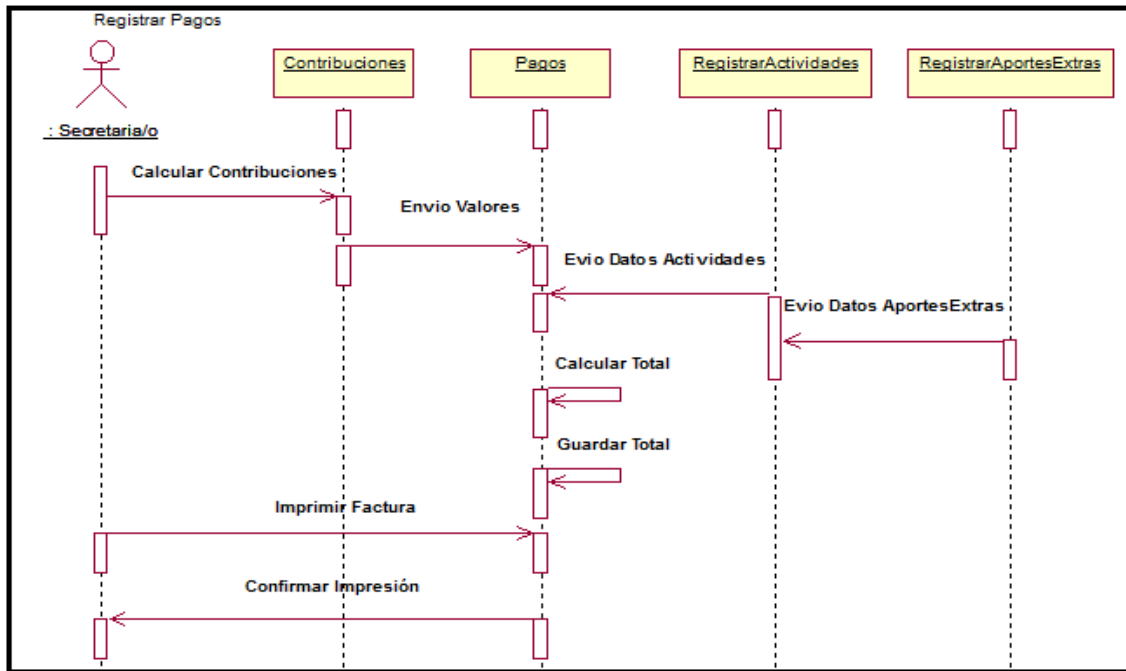


Figura 5.7 Diagrama de secuencia de Pagos

Factura

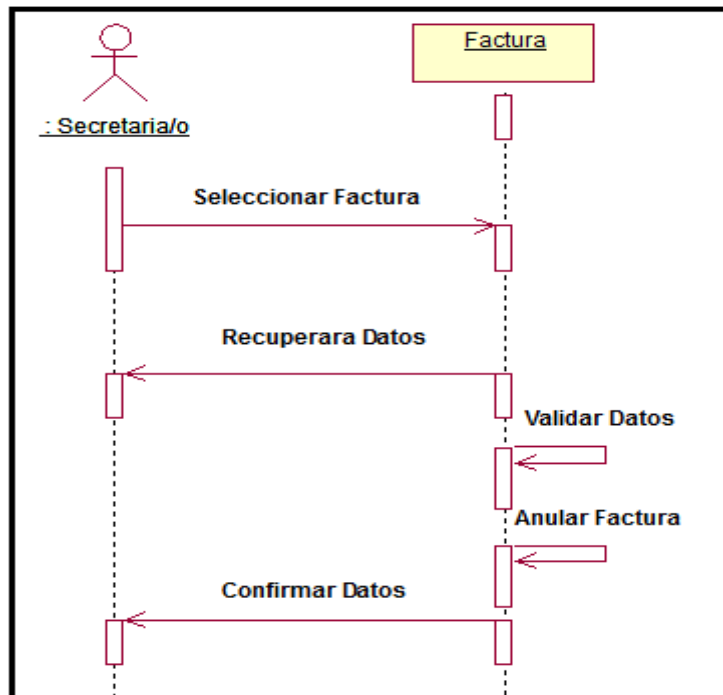


Figura 5.8 Diagrama de secuencia de Anular Factura

6.9.1.1.5 DIAGRAMA DE ESTADOS

Los diagramas de estado son una técnica importante en la descripción del comportamiento de un sistema. Detallan todos los estados posibles por los que puede pasar un objeto a lo largo de su vida y los estímulos que lo provocan.

Usuario

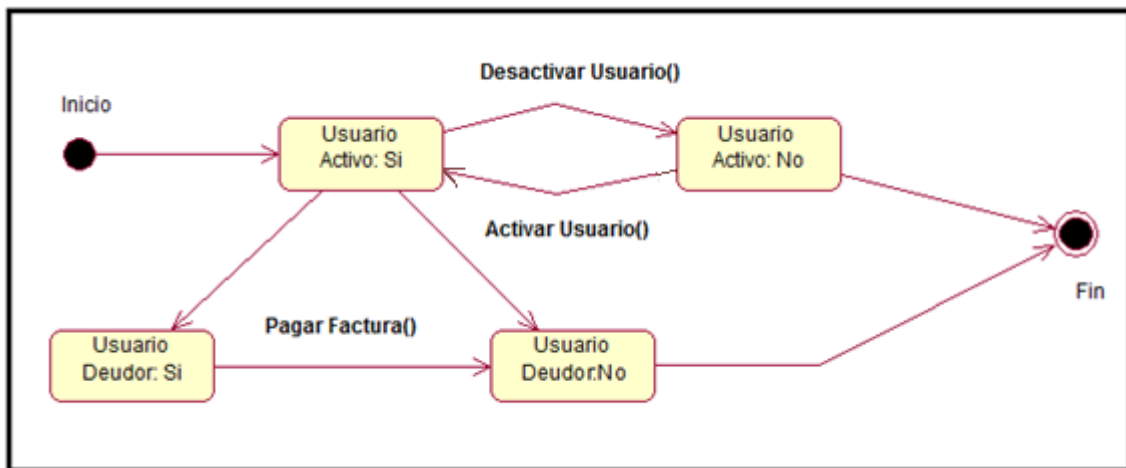


Figura 5.9 Diagrama de estado Usuarios

Cuenta

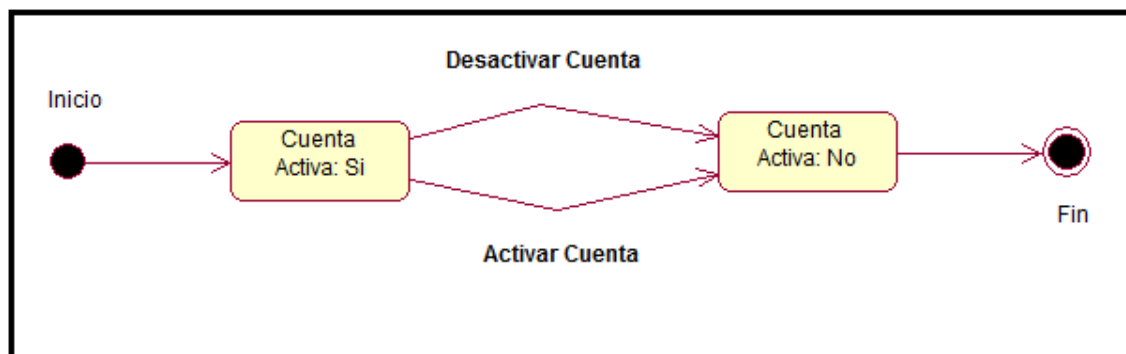


Figura 6.0 Diagrama de estado Cuenta

Factura

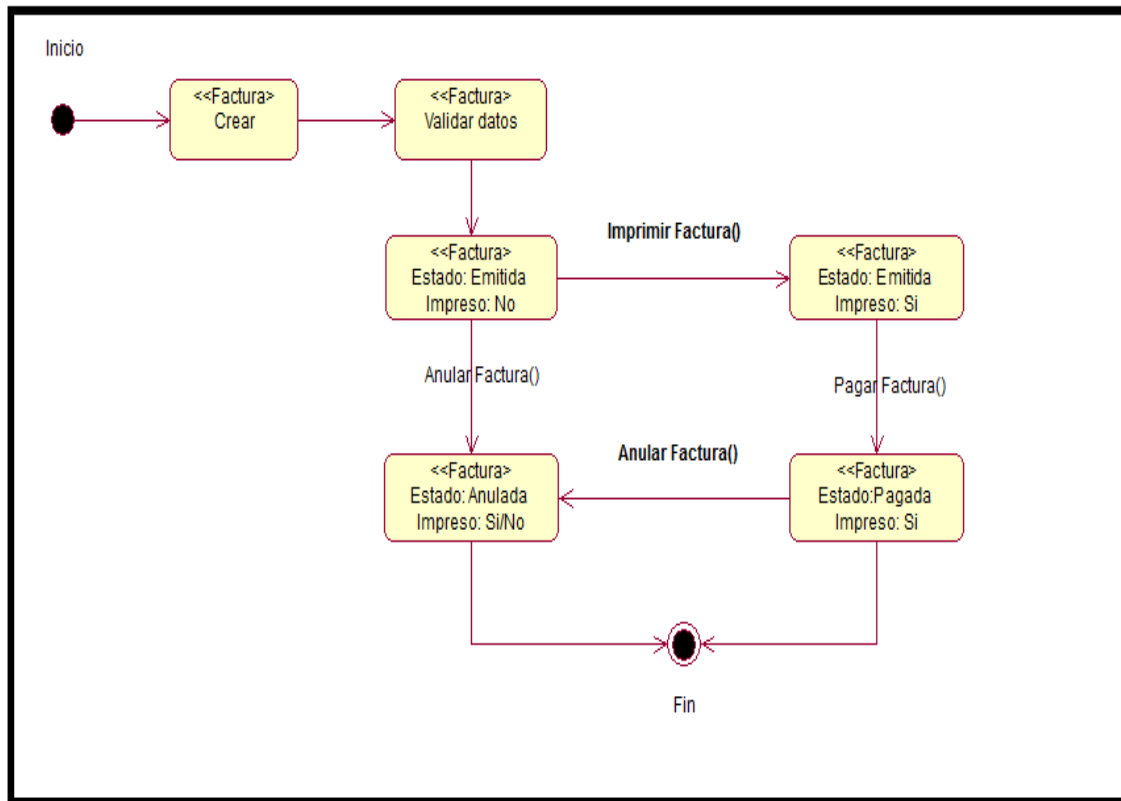


Figura 6.1 Diagrama de estado Factura

6.9.1.1.6 DIAGRAMA DE ACTIVIDADES

Representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema. Un Diagrama de Actividades muestra el flujo de control general. Este tipo de diagrama es útil para representar las operaciones de un objeto y los procesos de negocio.

Diagrama actividades Sistema Cobros

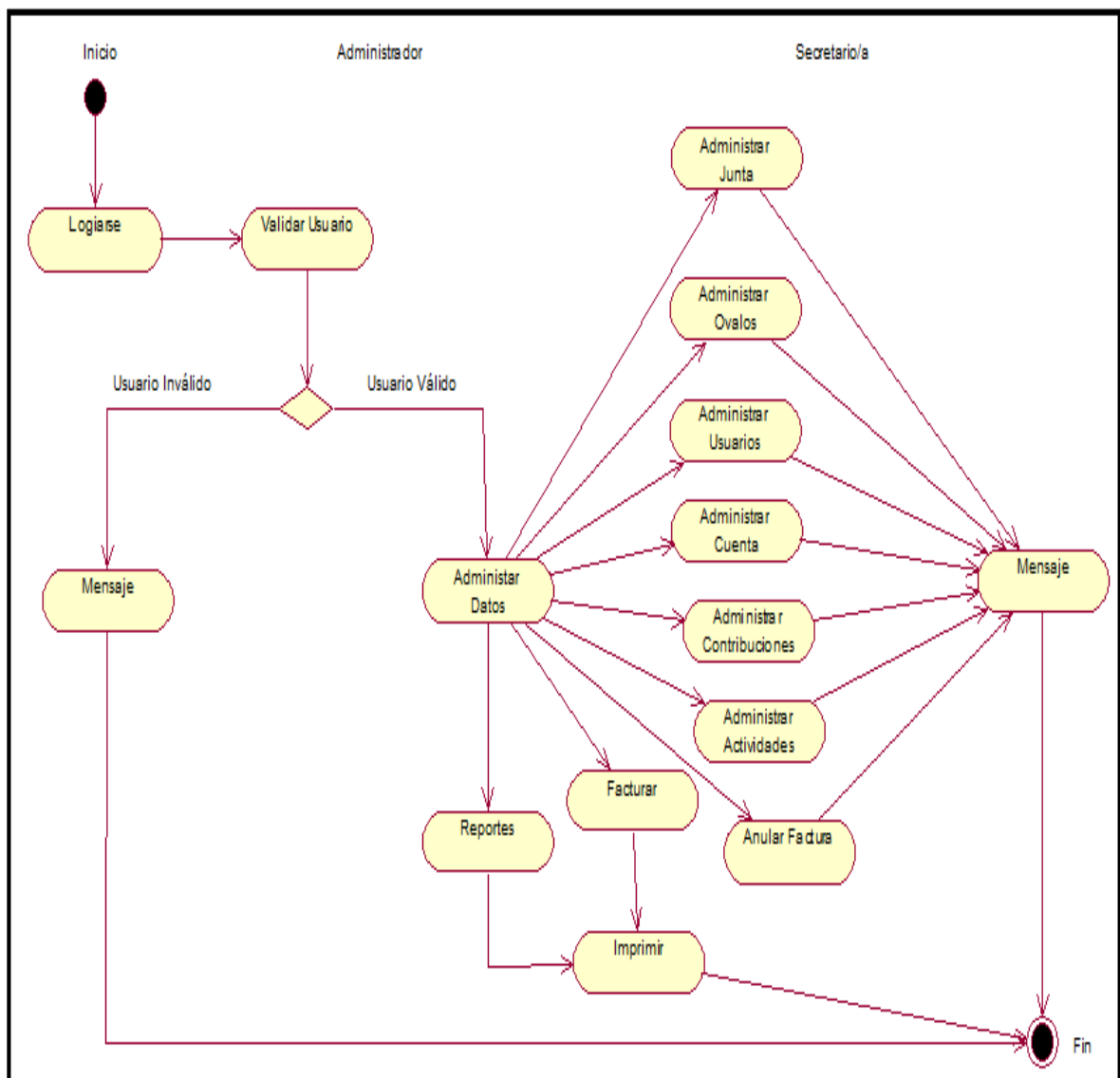


Figura 6.2 Diagrama de Actividades Sistema de Cobros

Ingresar al Sistema

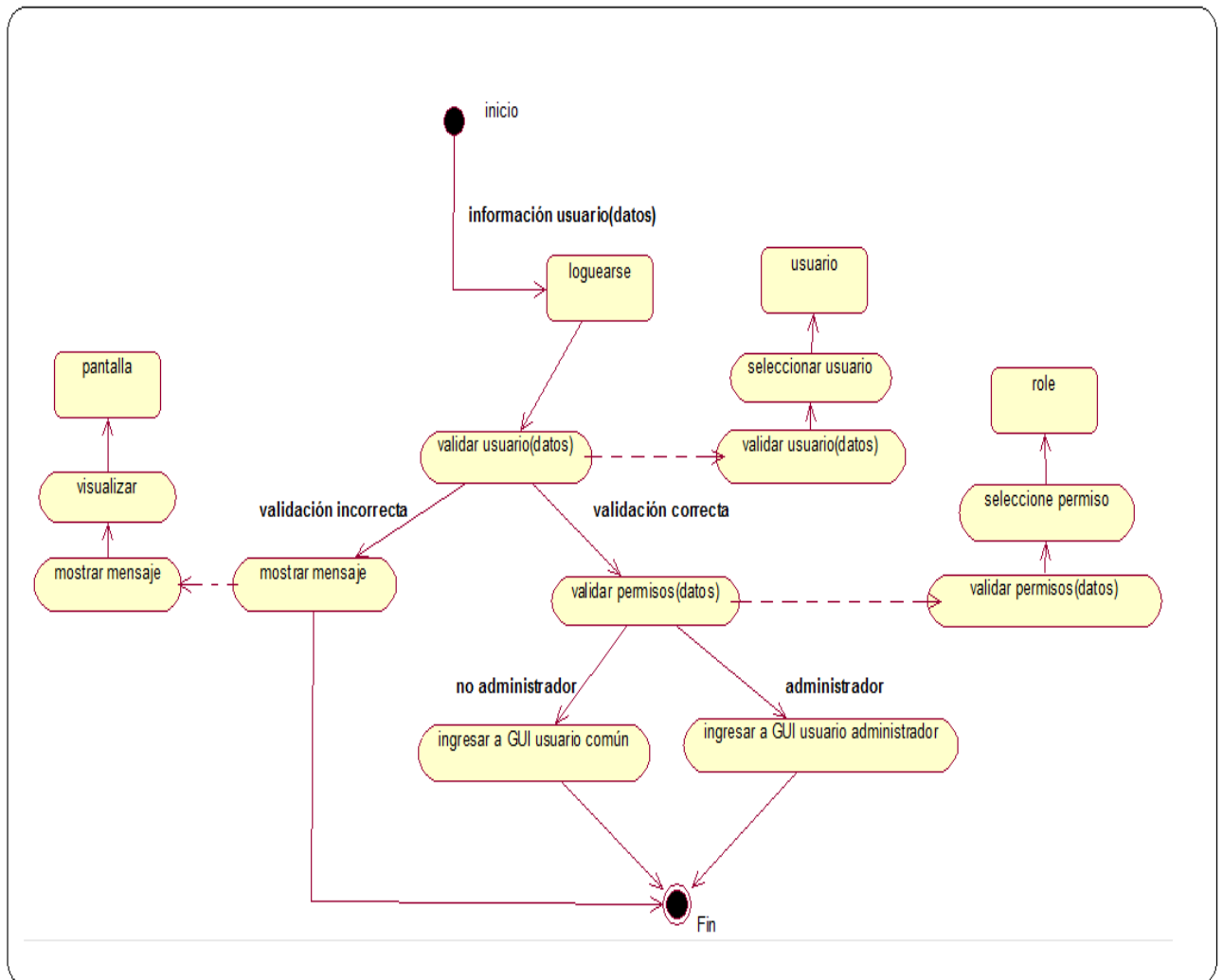


Figura 6.3 Diagrama de actividades ingresar al sistema

Junta

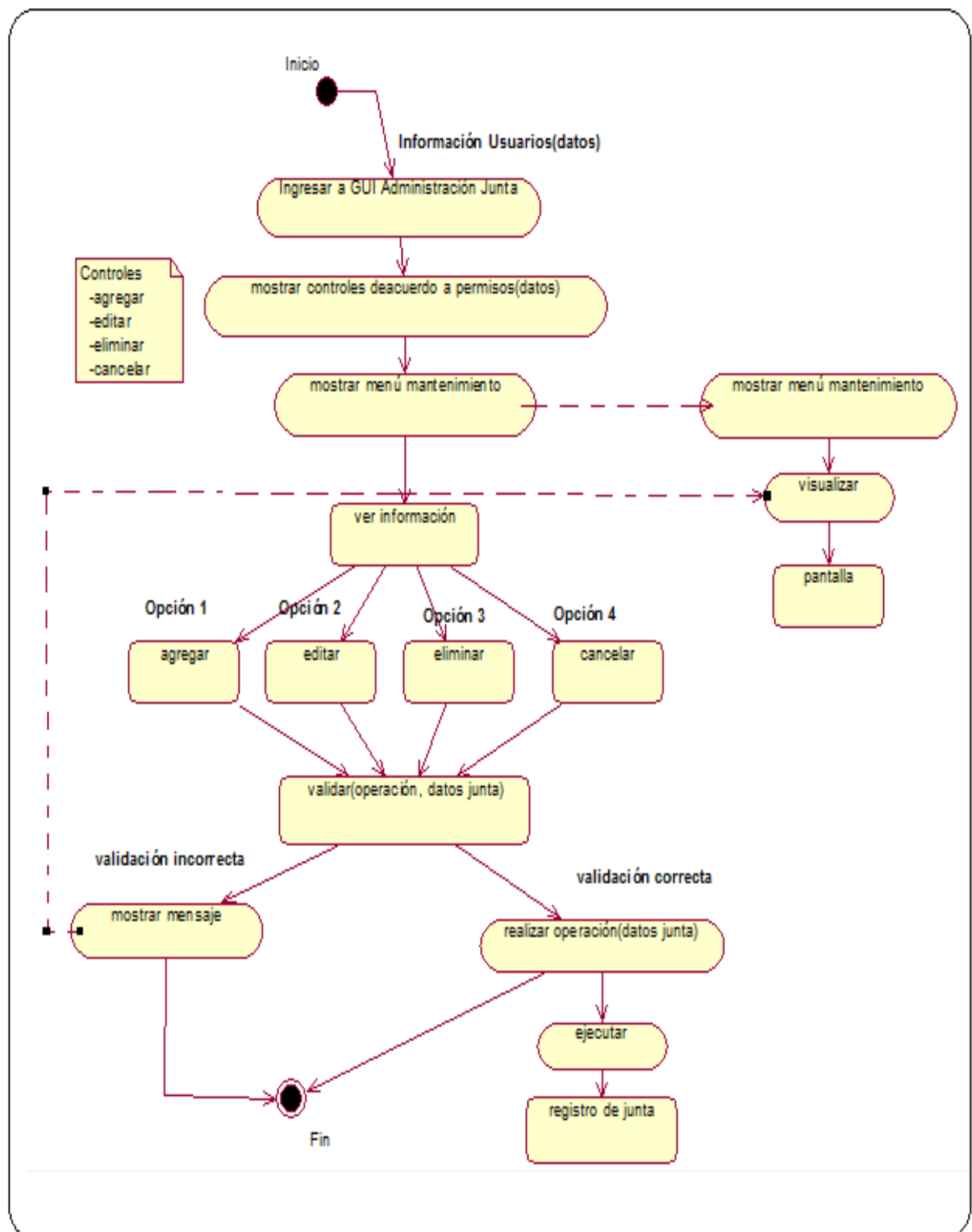


Figura 6.4 Diagrama de actividades administrar junta

Sectores

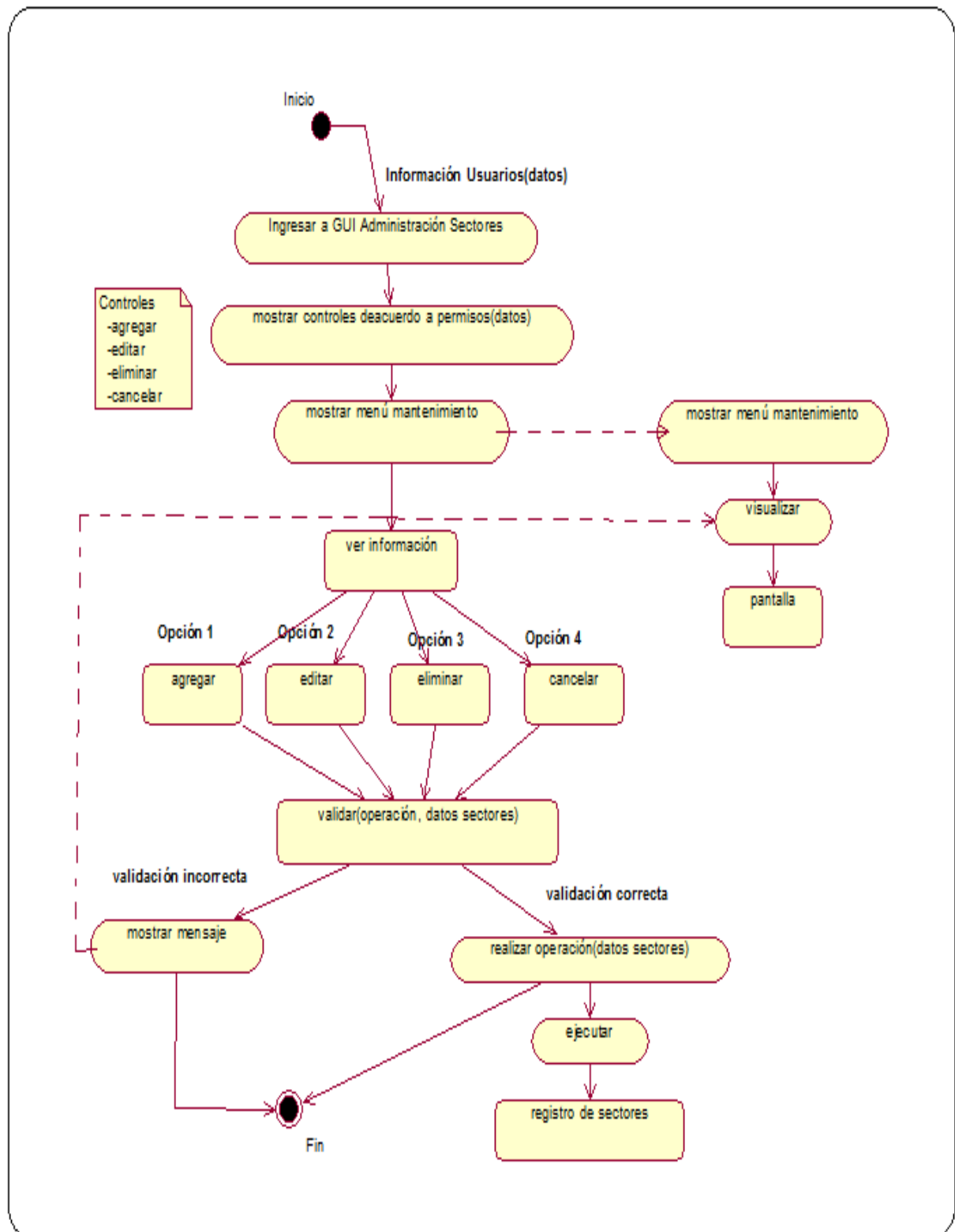


Figura 6.5 Diagrama de Actividades administrar sectores

Óvalos

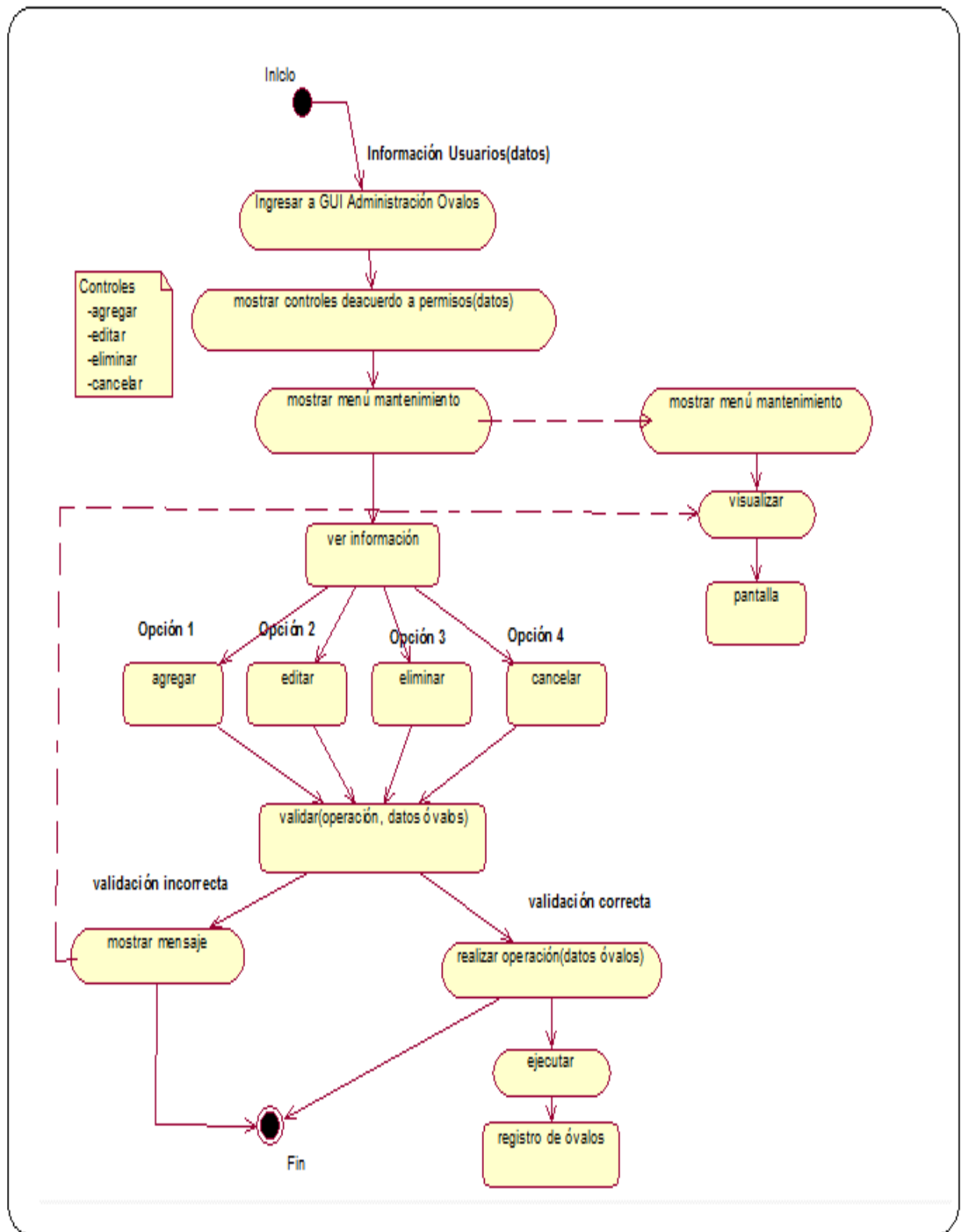


Figura 6.6 Diagrama de Actividades administrar óvalos

Usuarios

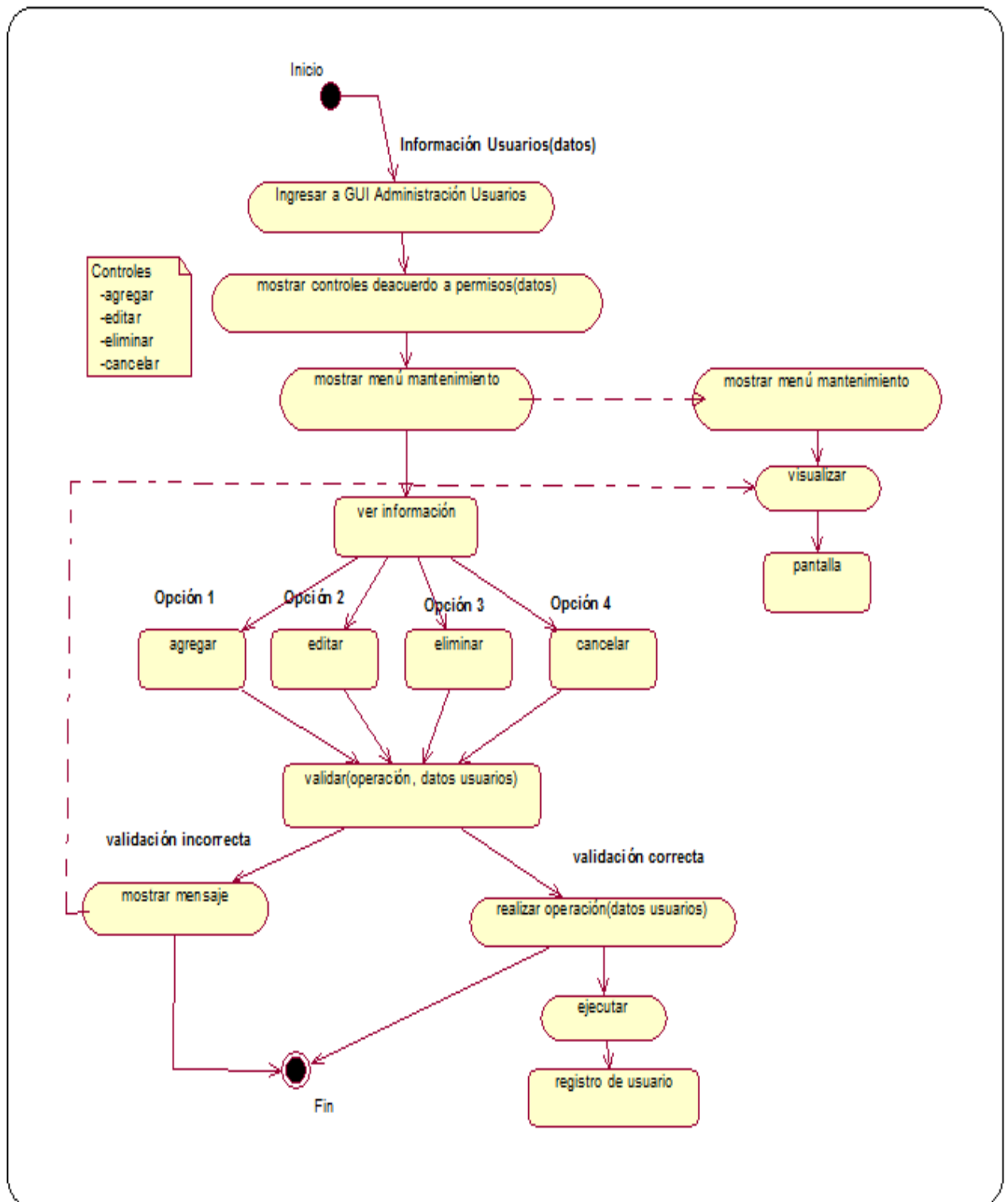


Figura 6.7 Diagrama de Actividades administrar usuarios

Cuenta

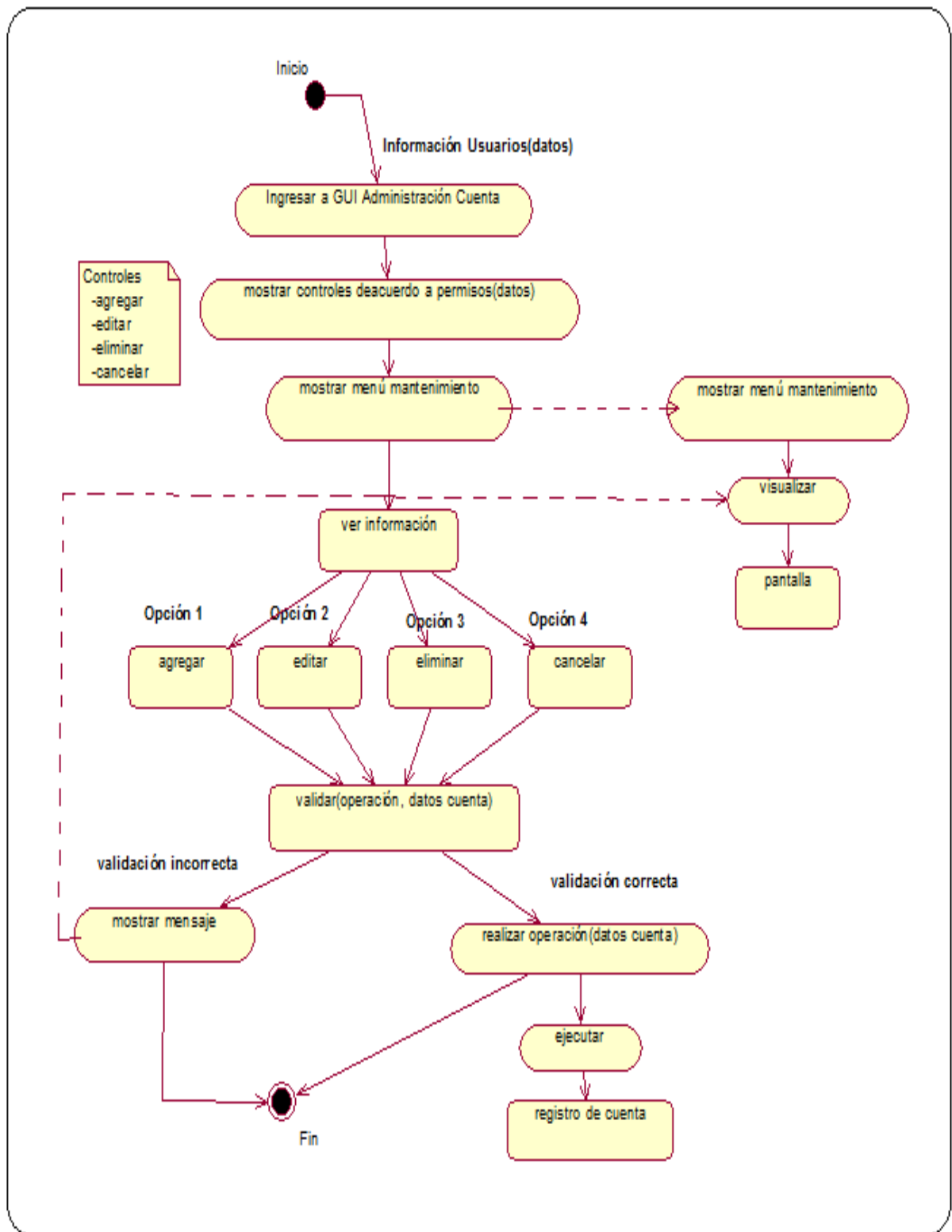


Figura 6.8 Diagrama de Actividades administrar cuenta

Actividades

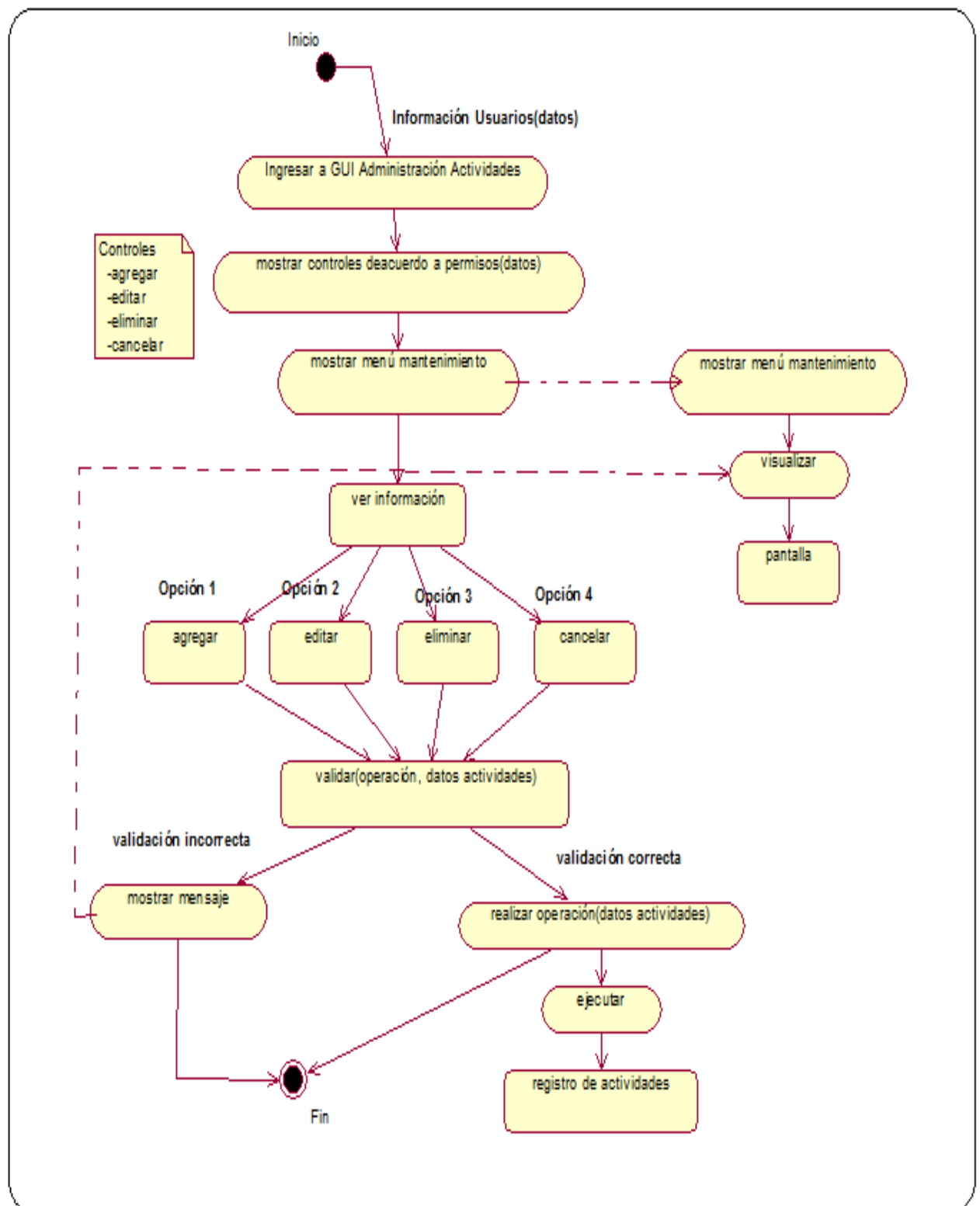


Figura 6.9 Diagrama de Actividades administrar actividades

Contribuciones

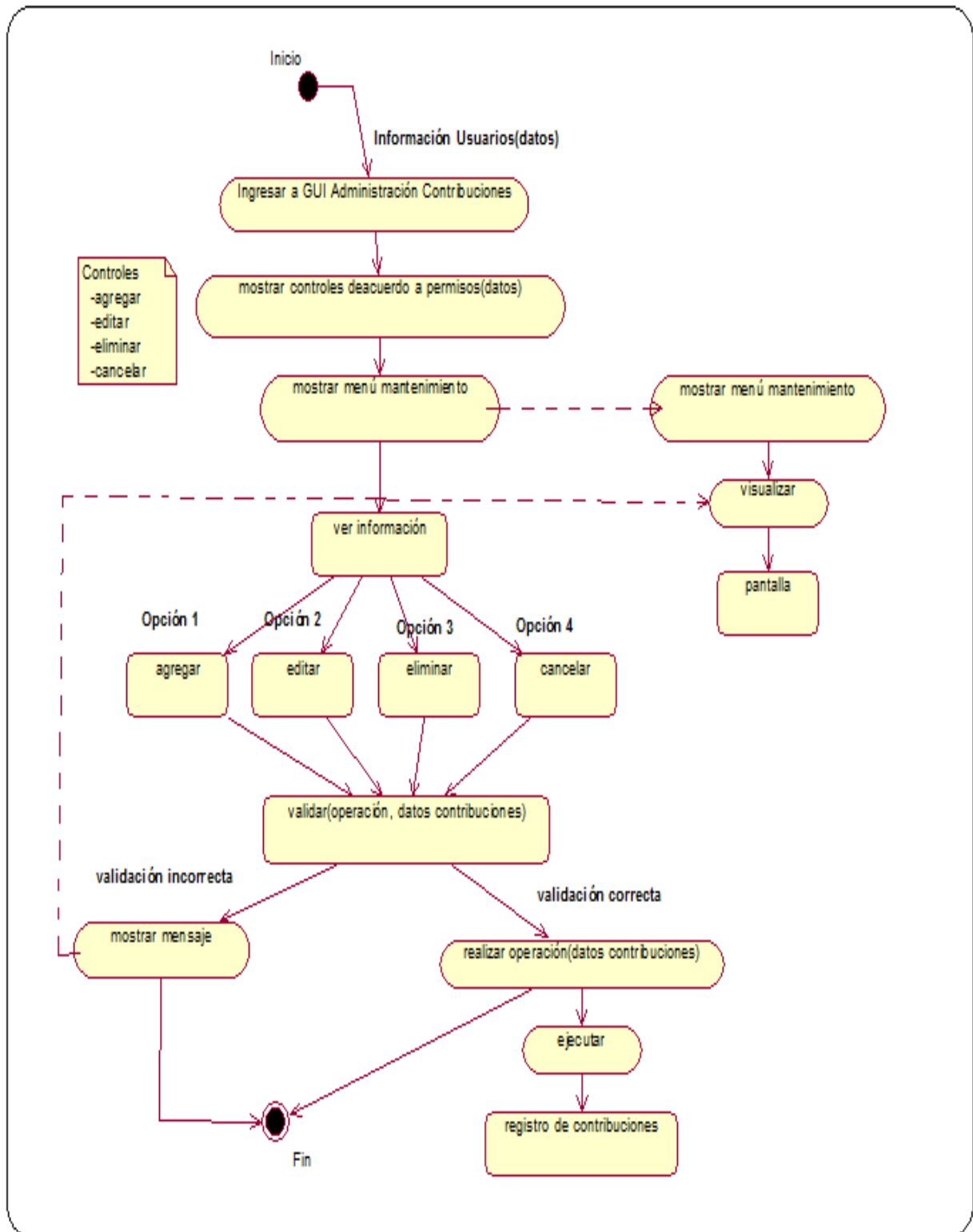


Figura 6.10 Diagrama de Actividades administrar contribuciones

Factura

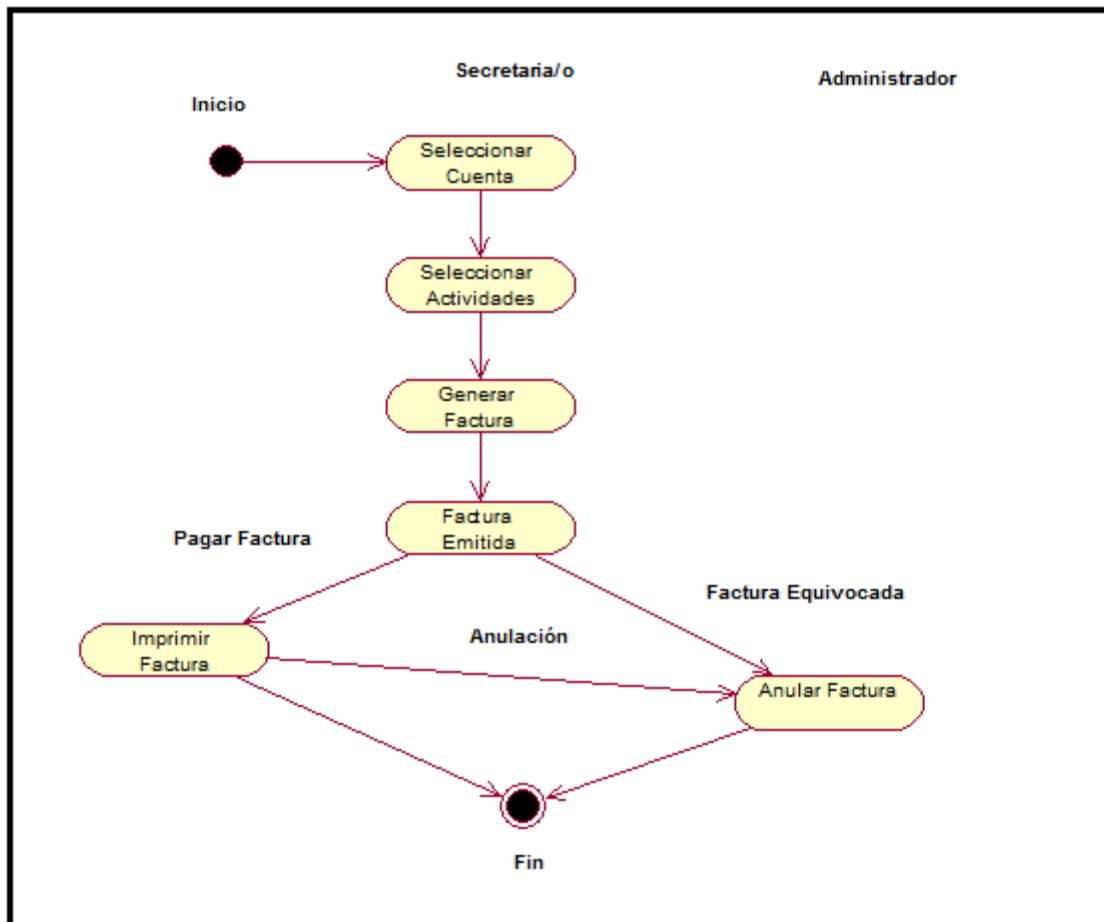


Figura 6.11 Diagrama de Actividades Factura

6.9.2 DISEÑO DEL SISTEMA

6.9.2.1 DISEÑO DE LA BASE DE DATOS

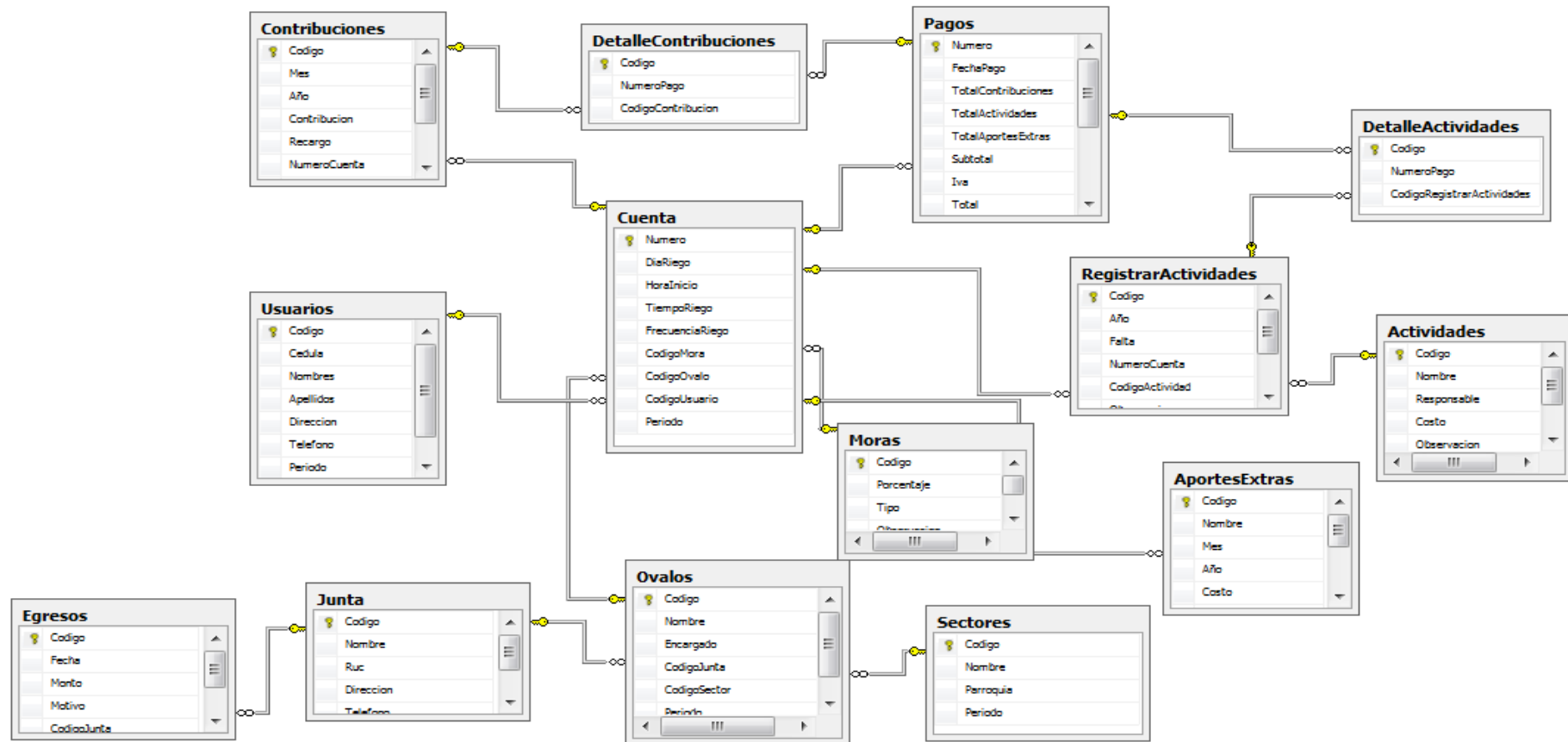


Figura 6.12 Diseño de la Base de Datos

6.9.2.2 DICCIONARIO DE DATOS

Es un conjunto de metadatos que contienen las características lógicas y puntuales de los datos que se van a utilizar en el sistema.

JUNTA

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Codigo	int	Código de la Junta	Primary Key
Nombre	nvarchar(100)	Nombre de la Junta	Not Null
Ruc	nvarchar(13)	Ruc de la Junta	Not Null
Dirección	nvarchar(100)	Dirección de la Junta	Null
Teléfono	nvarchar(9)	Teléfono de la Junta	Null
Periodo	int	Periodo actual de la Junta	Not Null

Tabla 3.0 Tabla Junta

SECTORES

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Codigo	int	Código del Sector	Primary Key
Nombre	nvarchar(100)	Nombre del Sector	Not Null
Parroquia	nvarchar(100)	Parroquia del Sector	Null
Periodo	int	Periodo actual del Sector	Not Null

Tabla 3.1 Tabla Sectores

ÓVALOS

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Codigo	int	Código del Ovalo	Primary Key
Nombre	nvarchar(100)	Nombre del Ovalo	Not Null
Encargado	nvarchar(50)	Encargado del Ovalo	Null
CodigoJunta	int	Código de la Junta	Not Null
CodigoSector	int	Código del Sector	Not Null
Periodo	int	Periodo actual de los Ovalos	Not Null

Tabla 3.2 Tabla Óvalos

USUARIOS

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Codigo	int	Código del Usuario	Primary Key
Cedula/Ruc	nvarchar(13)	Cédula / Ruc del Usuario	Not Null
Apellidos	nvarchar(50)	Apellidos del Usuario	Not Null
Nombres	nvarchar(50)	Nombres del Usuario	Not Null
Dirección	nvarchar(100)	Dirección del Usuario	Null
Telefono	nvarchar(12)	Dirección del Telefono	Null
Periodo	int	Periodo actual de Usuarios	Not Null

Tabla 3.3 Tabla Usuarios

CUENTA

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Numero	int	Número de la Cuenta	Primary Key
DiaRiego	nvarchar(10)	Día Riego de la Cuenta	Not Null
Horainicio	nvarchar(10)	Hora Inicio del Riego	Not Null
TiempoRiego	nvarchar(10)	Tiempo del Riego	Not Null
FrecuenciaRiego	nvarchar(10)	Frecuencia del Riego	Not Null
CodigoOvalo	int	Código del Ovalo	Not Null
CodigoUsuario	int	Código del Usuario	Not Null
CodigoMora	int	Código de la Mora	Not Null
Periodo	int	Periodo actual de la Cuenta	Not Null

Tabla 3.4 Tabla Cuenta

MORAS

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Codigo	int	Código de la Mora	Primary Key
Porcentaje	Decimal (7,2)	Porcentaje de la Mora	Not Null
Tipo	nvarchar(20)	Tipo de Mora	Not Null
Observación	nvarchar(100)	Observación de las Moras	Null
Periodo	int	Periodo actual de la Moras	Not Null

Tabla 3. 5 Tabla Moras

APORTESEXTRAS

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Codigo	int	Código del Aporte Extra	Primary Key
Nombre	nvarchar(100)	Nombre del Aporte Extra	Not Null
Año	int	Año del Aporte Extra	Not Null
Costo	decimal(7,2)	Costo del Aporte Extra	Not Null
NumeroCuenta	int	Número de Cuenta	Not Null
Pagado	nvarchar (2)	Estado del Aporte Extra	Not Null
Observacion	nvarchar(100)	Observación Aporte Extra	Null
Periodo	int	Periodo actual Aporte Extra	Not Null

Tabla 3. 6 Tabla Aportes Extras

ACTIVIDADES

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Codigo	int	Código de la Actividad	Primary Key
Nombre	nvarchar(100)	Nombre de la Actividad	Not Null
Responsable	nvarchar(50)	Responsable de la Actividad	Not Null
Costo	decimal(7,2)	Costo x Falta de la Actividad	Not Null
Fecha	datetime	Fecha de la Actividad	Not Null
Observacion	nvarchar(100)	Observación de la Actividad	Null
Periodo	int	Periodo actual de la Actividad	Not Null

Tabla 3. 7 Tabla Actividades

REGISTRAR ACTIVIDADES

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Codigo	int	Código de Registrar Actividad	Primary Key
Año	nvarchar(4)	Año de Registrar Actividad	Not Null
Falta	nvarchar(2)	Asistencia de Registrar Actividad	Not Null
NumeroCuenta	int	Número de Cuenta	Not Null
CodigoActividad	int	Código de la Actividad	Not Null
Pagado	nvarchar(2)	Estado de Registrar Actividad	Not Null
Observacion	nvarchar(100)	Observación Registrar Actividad	Null
Periodo	int	Periodo act. Registrar Actividad	Not Null

Tabla 3.8 Tabla Registrar Actividades

CONTRIBUCIONES

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Codigo	int	Código de la Contribución	Primary Key
Mes	nvarchar(12)	Mes de la Contribución	Not Null
Año	int	Año de la Contribución	Not Null
Contribucion	decimal(7,2)	Valor de la Contribución	Not Null
Recargo	decimal(7,2)	Recargo de Contribución	Not Null
PorcentajeMora	decimal(7,2)	Porcentaje Mora	Not Null
NumeroCuenta	int	Número de Cuenta	Not Null
Pagado	nvarchar (2)	Estado de la Contribución	Not Null
Observacion	nvarchar(100)	Observación Contribución	Null
Periodo	int	Periodo actual Contribución	Not Null

Tabla 3.9 Tabla Contribuciones

PAGOS

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Numero	int	Número de Pago	Primary Key
FechaPago	datetime	Fecha de Pago	Not Null
TotalContribuciones	decimal(7,2)	Total Contribuciones	Not Null
TotalActividades	decimal(7,2)	Total Actividades	Not Null
TotalAportesExtras	decimal(7,2)	Total Aportes Extras	Not Null
Subtotal	decimal(7,2)	Subtotal Factura	Not Null
IVA	decimal(7,2)	IVA Factura	Not Null
Total	decimal(7,2)	Total Factura	Not Null
Abono	decimal(7,2)	Abono Factura	Not Null
Recargo	decimal(7,2)	Recargo Contribuciones	Not Null
NumeroCuenta	int	Número de Cuenta	Not Null
Estado	nvarchar (9)	Estado Factura	Not Null
Periodo	int	Periodo actual Factura	Not Null

Tabla 4.1 Tabla Pagos

DETALLE APORTES EXTRAS

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Codigo	Int	Código Detalle Aportes Extra	Primary Key
NumeroPago	Int	Número de Pago	Not Null
CodigoAporteExtra	Int	Código Aporte Extra	Null
Periodo	Int	Periodo actual detalle	Not Null

Tabla 4.2 Tabla Detalle Aportes Extras

DETALLE ACTIVIDADES

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Codigo	int	Código Detalle Actividades	Primary Key
NumeroPago	int	Numero de Pago	Not Null
CodigoRegistroAct	int	Código Registrar Actividades	Null
Periodo	int	Periodo actual detalle	Not Null

Tabla 4.3 Tabla Detalle Aportes Extras

DETALLE CONTRIBUCIONES

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Codigo	int	Código Detalle Contribución	Primary Key
NumeroPago	int	Numero de Pago	Not Null
CodigoContribu	int	Código Contribuciones	Null
Periodo	int	Periodo actual detalle	Not Null

Tabla 4.4 Tabla Detalle Contribuciones

EGRESOS

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Codigo	int	Código de los Egresos	Primary Key
Fecha	datetime	Fecha de los Egresos	Not Null
Monto	decimal(7,2)	Monto de los Egresos	Not Null
Motivo	nvarchar(100)	Motivo de los Egresos	Null
CodigoJunta	int	Código de la Junta	Not Null
Periodo	int	Periodo actual de la Moras	Not Null

Tabla 4.5 Tabla Egresos

IVA

CAMPO	TIPO DE DATO	DESCRIPCIÓN	RESTRICCIÓN
Codigo	int	Código IVA	Primary Key
Porcentaje	Decimal (7,2)	Porcentaje IVA	Not Null
Periodo	int	Periodo actual IVA	Not Null

Tabla 4.6 Tabla IVA

6.9.3 PRUEBAS

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

La fase de pruebas añade un valor al producto que se maneja. Todos los programas tienen errores y en esta fase se los descubre. El objetivo principal es encontrar errores.

Una vez concluido con el desarrollo del proyecto se procedió a realizar las pruebas con la intención de garantizar la seguridad, confiabilidad, integridad de la información que se va a manejar en el sistema de gestión.

6.9.3.1 PRUEBAS DE CAJA NEGRA

En este tipo de pruebas no se considera la codificación dentro de sus parámetros a evaluar, es decir, no está basado en el conocimiento interno del sistema. Estas pruebas se enfocan en la funcionalidad y requerimientos especificados del software.

Una vez aplicado este tipo de pruebas en el sistema, como conclusión se pudo determinar qué:

- Las funciones que realiza el sistema son claras y precisas.
- El ingreso al sistema es fácil y sencillo.
- La seguridad de acceso y permisos de usuario se cumplen a cabalidad.
- Las interfaces del sistema no permiten confusión en su manipulación.
- Se mantiene la integridad del sistema a lo largo de su utilización.

6.9.3.2 PRUEBAS DE CAJA BLANCA

Las pruebas de caja blanca se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente.

Mediante estas pruebas se garantizó qué:

- Se ejecuten al menos una vez todos los caminos independientes de cada módulo.
- Se utilicen las decisiones en su parte verdadera y falsa.
- Se ejecuten todos los bucles en sus límites.
- Se utilicen todas las estructuras de datos internas.

```

public boo ValidarCedula()
{
    bool bandera = true;
    string cadena;
    int suma =0;
    int digitoverificador;

    int valor=0;

    cadena = txtCedula.Text.Trim();

    if(cadena.Length==10)
    {
        for(int i=0;i<=8;i++)
        {
            if(i%2==0)
            {
                valor = Convert.ToInt32(cadena.Substring(i,1))*2;
                if(valor>9)
                    valor-=9;
                suma+=valor;
            }
            else
            {
                suma+= Convert.ToInt32(cadena.Substring(i,1));
            }
        }
        digitoverificador = 10 -(suma%10);
        if(digitoverificador != Convert.ToInt32(cadena.Substring(9,1)))
        {
            bandera = false;
        }
    }
    else
    {
        bandera = false;
    }
}

```

```

    return bandera;
}

```

#	CASOS DE PRUEBA	OBJETIVO
1	La cédula del usuario es correcta (la condición es verdadera)	Verdadero
2	La cédula del usuario no es correcta (la condición es falsa)	Falso

Tabla 4.7 Casos de Prueba.

Como resultado de la ejecución de estas pruebas en el sistema se verificó y se aseguró que su estructura interna esté de acuerdo con las especificaciones.

6.9.3.3 PRUEBAS DE VERIFICACIÓN Y VALIDACIÓN

La verificación es un aspecto muy importante dentro de las pruebas ya que nos permite conocer si el sistema cumple con las especificaciones planteadas y si ejecuta la tarea para la cual fue creado, en cuanto a la validación es el proceso de comprobar que lo que se ha especificado es lo que el usuario realmente quería.

Verificación.- Este proceso determinó que el sistema satisface las condiciones impuestas al comienzo de este proyecto, este concuerda y cumple con las especificaciones planteadas.

RESTRICCIONES

Restricciones de usuario

Según lo acordado con los directivos de la junta se crearon dos tipos de usuarios uno que tiene acceso total llamado administrador y uno con restricciones de eliminación y de anulación de factura llamado secretario.

Solo el usuario administrador tiene la potestad para crear nuevos usuarios, anular facturas y ver los reportes de auditoría de las tablas usuarios y cuenta.

Eliminación

Solo el usuario administrador puede eliminar registros.

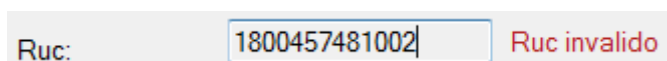
Validación.- Esta prueba verificó si el sistema cumple con las expectativas del usuario.

VALIDACIÓN DE DATOS

El sistema valida el ingreso de datos como:

Validación del RUC

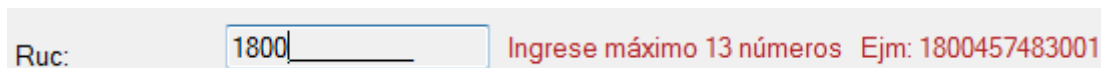
Al ingresar el RUC incorrecto aparecerá el siguiente mensaje:



A screenshot of a web form showing a validation error. On the left, the label 'Ruc:' is followed by a text input field containing the value '1800457481002'. To the right of the input field, the text 'Ruc invalido' is displayed in red, indicating the error.

Figura 6.13 Validación Ruc

Al ingresar un RUC incompleto aparecerá el siguiente mensaje:

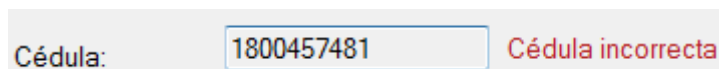


A screenshot of a web form showing a validation error for an incomplete RUC. The label 'Ruc:' is followed by a text input field containing '1800' and a cursor. To the right of the input field, the text 'Ingrese máximo 13 números Ejm: 1800457483001' is displayed in red, indicating the error.

Figura 6.14 Validación del campo Ruc incompleto

Validación del Cédula

Al ingresar la cedula incorrecta aparecerá el siguiente mensaje:



A screenshot of a web form showing a validation error for an incorrect ID number. The label 'Cédula:' is followed by a text input field containing the value '1800457481'. To the right of the input field, the text 'Cédula incorrecta' is displayed in red, indicating the error.

Figura 6.15 Validación cédula

6.9.3.4 IMPLANTACIÓN

Luego de que el Sistema de gestión para cobros y registro de usuarios de la Junta de Aguas Chacón Sevilla concluyó satisfactoriamente la fase de pruebas, se procedió a su implantación. Se realizó una implantación directa, verificando la información del manual en consiguiente el sistema entró en funcionamiento inmediatamente.

6.10 CONCLUSIONES Y RECOMENDACIONES

6.10.1 CONCLUSIONES

- Se desarrolló e implantó un sistema de gestión para cobros y registro de usuarios que permitió dar solución al manejo inadecuado de la información en la Junta de Aguas Chacón Sevilla.
- Con la implementación del sistema, se pudo conseguir un control eficiente de las tarifas recaudadas.
- Con la aplicación de pruebas en el sistema se pudo encontrar y corregir errores que permitieron asegurar el correcto funcionamiento en su implantación.
- El sistema se ajusta fácilmente a las diversas formas de administrar los recursos financieros de la Junta.
- La interfaz del sistema es de fácil navegación y tiene búsquedas inteligentes, para que el usuario pueda interactuar de mejor manera con el sistema y recupere datos fácilmente.

6.10.2 RECOMENDACIONES

- Se recomienda designar a una persona capacitada para el mantenimiento y control del sistema de gestión, para asegurar un mejor desempeño y el correcto funcionamiento del mismo.
- Efectuar respaldos de la Base de Datos consecutivamente y almacenarlos en dispositivos externos seguros, con la fecha y descripción que los identifique para esto se debe utilizar la opción de reportes del sistema.
- Crear usuarios en el sistema tomando en consideración los módulos que van a manipular.
- Se recomienda revisar los manuales de usuario en caso dudas o detección de errores.

6.11 BIBLIOGRAFÍA

6.11.1 INFORMACIÓN BIBLIOGRÁFICA

- DATE, CJ. (2001). "Introducción a los Sistemas de base de datos". Séptima Edición. Pearson Educación
- LA WRENCE, Shari (2002) Ingeniería de Software Teoría y práctica, Editorial Pentice Hall
- BOOCH, J. RUMBAUGH, I. JACOBSON. El Lenguaje Unificado de Modelado (1999), Addison Wesley Iberoamericana. – Conceptos UML, características, diagramas y ejemplos

6.11.2 REFERENCIAS BIBLIOGRÁFICAS WEB

Software Libre (2011). Extraído el 20 de Octubre del 2011 desde http://es.wikipedia.org/wiki/Software_libre

Base de Datos (2011). Extraído el 20 de Octubre del 2011 desde http://es.wikipedia.org/wiki/Base_de_datos

Sistema de Información (2011). Extraído el 20 de Octubre del 2011 desde http://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n

Sistema de Gestión (2011). Extraído el 20 de Octubre del 2011 desde http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos

Sistema de Gestión (2011). Extraído el 2 de Noviembre del 2011 desde <http://es.wikipedia.org/wiki/Programaci%C3%B3n>

Registro (2011). Extraído el 2 de Noviembre del 2011 desde <http://www.definicionabc.com/tecnologia/registro-2.php>

Usuario (2011). Extraído el 2 de Noviembre del 2011 desde
<http://www.definicionabc.com/tecnologia/usuario.php>

Egresos (2011). Extraído el de 23 Noviembre del 2011 desde
<http://definicion.de/egresos>

Software (2011). Extraído el de 23 Noviembre del 2011 desde
<http://es.wikipedia.org/wiki/Software>

SharpDevelop Software (2011). Extraído el de 5 Diciembre del 2011 desde
<http://es.wikipedia.org/wiki/SharpDevelop>

Framework (2011). Extraído el de 5 Diciembre del 2011 desde
<http://es.wikipedia.org/wiki/Framework>

SQL Server Express (2011). Extraído el de 5 Diciembre del 2011 desde
<http://msdn.microsoft.com/es-es/express/aa718378>

SQL Server Express (2011). Extraído el de 5 Diciembre del 2011 desde
<http://msdn.microsoft.com/es-es/express/aa718378>

SQL Server Express (2012). Extraído el de 3 Enero del 2012 desde
http://es.wikipedia.org/wiki/Crystal_Reports

Diagrama Caso de Usos (2012). Extraído el de 7 Febrero del 2012 desde
http://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso

Diagrama de Secuencias (2012). Extraído el de 7 Febrero del 2012 desde
http://es.wikipedia.org/wiki/Diagrama_de_secuencia

UML (2012). Extraído el de 7 Febrero del 2012 desde
http://en.wikipedia.org/wiki/Unified_Modeling_Language

Diagrama Actividades (2012). Extraído el de 12 Marzo del 2012 desde
http://es.wikipedia.org/wiki/Diagrama_de_actividades

Diagrama Estados (2012). Extraído el de 12 Marzo del 2012 desde
http://es.wikipedia.org/wiki/Diagrama_de_estados

Técnicas de Pruebas (2012). Extraído el de 12 Marzo del 2012 desde
<http://indalog.ual.es/mtorres/LP/Prueba.pdf>

Pruebas de Caja Negra (2012). Extraído el de 24 Marzo del 2012 desde
http://es.wikipedia.org/wiki/Caja_negra_%28sistemas%29

Pruebas de Caja Blanca (2012). Extraído el de 24 Marzo del 2012 desde
http://es.wikipedia.org/wiki/Pruebas_de_caja_blanca

Pruebas de Validación (2012). Extraído el de 24 Marzo del 2012 desde
http://es.wikipedia.org/wiki/Pruebas_de_validaci%C3%B3n

GLOSARIO DE TÉRMINOS.

Diagrama de actividades.- Representa las operaciones de un objeto y los procesos de negocio.

Diagramas de caso de uso.- Es una técnica que resume la funcionalidad completa de un sistema, presentando las respectivas interacciones con los agentes.

Diagramas de clase.- Presenta las clases del sistema con sus relaciones estructurales y de herencia.

Diagrama de estado: Presenta los estados en los que puede encontrarse un objeto junto con las transacciones entre los estados.

Diagramas de secuencia.- Muestra la secuencia de los mensajes entre objetos durante un escenario concreto.

Framework.- es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado.

Lenguaje de programación.- Un lenguaje de programación es aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.

Sistema de Gestión.- Se trata de sistemas que optimizan los procesos internos de las empresas mediante un software que permite compartir la información.

Software.- Es el equipamiento lógico o soporte lógico de un sistema informático, comprende el conjunto de los componentes **lógicos** necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.

SQL Server Express.- Es un producto de gestión de datos eficaz y confiable que ofrece funciones completas, protección de datos y rendimiento para clientes de aplicaciones incrustadas, aplicaciones ligeras y almacenes de datos locales.

SGBS: (Sistema de Gestor de Base de Datos).- Es una agrupación de programas que sirven para definir, construir y manipular una base de datos.

Software.- Consiste en un código en un lenguaje máquina específico para un procesador individual. El código es una secuencia de instrucciones ordenadas que cambian el estado del hardware de una computadora.

UML.- 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: Cronograma

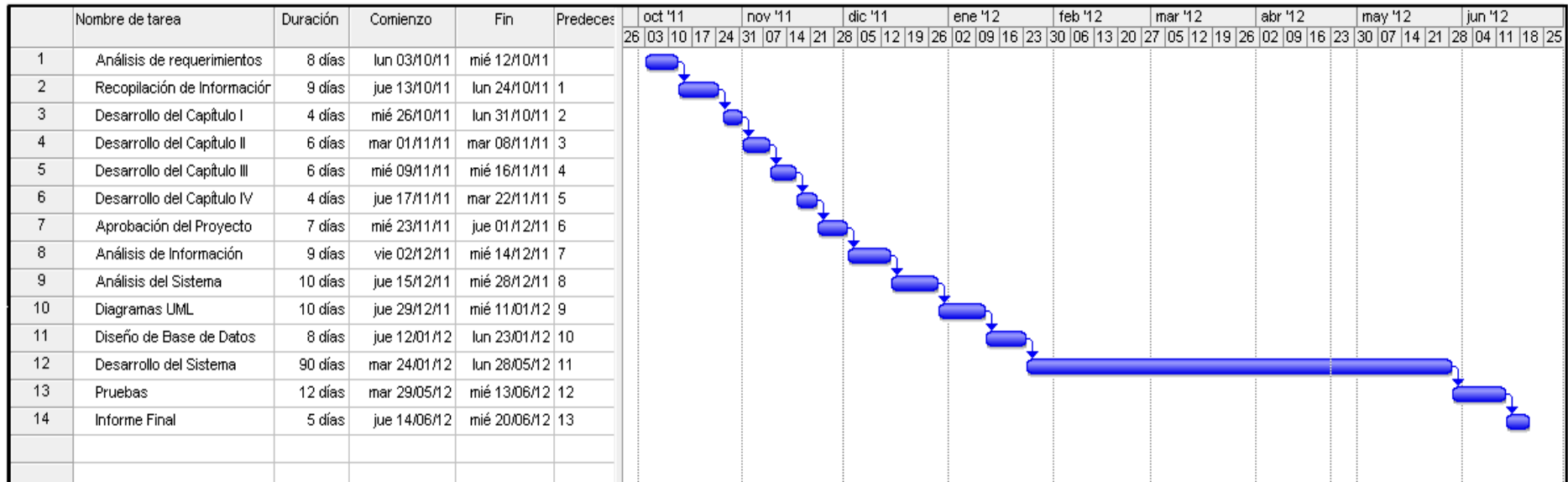


Figura A1. Cronograma

ANEXO 2: Estructura del Cuestionario

FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL

CARRERA DE INGENIERÍA EN SISTEMAS

OBJETIVO DEL CUESTIONARIO:

El presente cuestionario se ha planificado con el objeto de recabar información referente al Sistema de Facturación, la misma que será manejada en forma responsable y exclusivamente para solucionar un problema de la Institución, la información es anónima y tendrá la reserva del caso, le solicito comedidamente conteste con la verdad en el siguiente cuestionario.

INSTRUCCIONES:

Marque con una X o escriba la respuesta que UD. considere conveniente.

1. La información financiera que se recopila por la Junta se encuentra almacenada de forma:

- Manual
- Archivos digitales
- De ambas formas
- Ninguna

2. ¿Qué programas informáticos utilizan para almacenar información?

- Sistema de Cobros
- Excel
- Otros (especifique)
- Ninguno

3. ¿La junta cuenta con un sistema informático para las recaudaciones de los valores de regadío?

- Si
- No

4. ¿El mecanismo utilizado actualmente para los cobros de tarifas dificulta el proceso de recaudación?

- Si

No

5. Marque con una (X) lo que corresponda: La junta de agua bajo su administración cuenta con:

- Computadora
- Impresora
- Internet
- Teléfono celular
- Teléfono convencional
- Fax

6. Considera usted que el proceso de facturación utilizado actualmente es:

- Muy bueno
- Bueno
- Regular
- Malo

7. El tiempo que se utiliza para la recaudación es óptimo:

- Si
- No

8. ¿Considera usted necesario implementar un sistema informático de control financiero y administrativo que le permita tener información de calidad para la toma de decisiones y optimización del tiempo?

- Si
- No

9. ¿El personal encargado de los procesos de recaudación y facturación de Junta de su competencia tiene capacitación en informática?

- Si
- No

10. Con que frecuencia tienen problemas al momento de recaudar el dinero por consumo de riego?

- Siempre
- Ocasionalmente
- Nunca

ANEXO 3: Manual de Configuración

INSTALACIÓN DEL SISTEMA

PRIMER PASO: BASE DE DATOS

Creación de la Base de Datos.

```
CREATE TABLE Junta
```

```
(  
Codigo int IDENTITY (1, 1) PRIMARY KEY,  
Nombre nvarchar (100) NOT NULL,  
Ruc nvarchar (14) NOT NULL,  
Direccion nvarchar (50) NULL,  
Telefono nvarchar (12) NULL,  
Periodo int NOT NULL  
);
```

```
CREATE TABLE Egresos
```

```
(  
Codigo int IDENTITY (1, 1) PRIMARY KEY,  
Fecha datetime NOT NULL,  
Monto decimal (7, 2) NOT NULL,  
Motivo nvarchar (100) NULL,  
CodigoJunta int NOT NULL,  
Periodo int NOT NULL  
CONSTRAINT FK_Egresos_Junta FOREIGN KEY (CodigoJunta)  
REFERENCES Junta (Codigo)  
)
```

```

CREATE TABLE Sectores
(
Codigo int IDENTITY (1, 1) PRIMARY KEY,
Nombre nvarchar (100) NOT NULL,
Parroquia nvarchar (50) NULL,
Periodo int NOT NULL
);

```

```

CREATE TABLE Ovalos
(
Codigo int IDENTITY (1, 1) PRIMARY KEY,
Nombre nvarchar (100) NOT NULL,
Encargado nvarchar (30) NOT NULL,
CodigoJunta int NOT NULL,
CodigoSector int NOT NULL,
Periodo int NOT NULL,
CONSTRAINT FK_Ovalos_Junta FOREIGN KEY (CodigoJunta)
REFERENCES Junta (Codigo),
CONSTRAINT FK_Ovalos_Sectores FOREIGN KEY (CodigoSector)
REFERENCES Sectores (Codigo) ON DELETE CASCADE ON UPDATE
CASCADE
);

```

```

CREATE TABLE Usuarios
(
Codigo int IDENTITY (1, 1) PRIMARY KEY,
Cedula nvarchar (13) NOT NULL,
Nombres nvarchar (20) NOT NULL,
Apellidos nvarchar (20) NOT NULL,
Direccion nvarchar (100) NULL,
Telefono nvarchar (9) NULL,
Periodo int NOT NULL
);

```

);

CREATE TABLE Moras

(

Codigo int IDENTITY (1, 1) PRIMARY KEY,

Porcentaje decimal (7,2) NOT NULL,

Tipo nvarchar (10) NOT NULL,

Observacion nvarchar (100) NULL,

Periodo int NOT NULL

);

CREATE TABLE Cuenta

(

Numero int IDENTITY (1, 1) PRIMARY KEY,

DiaRiego nvarchar (10) NOT NULL,

Horainicio nvarchar (10) NOT NULL,

TiempoRiego nvarchar (10) NOT NULL,

FrecuenciaRiego nvarchar (10) NOT NULL,

CodigoMora int NOT NULL,

CodigoOvalo int NOT NULL,

CodigoUsuario int NOT NULL,

Periodo int NOT NULL,

CONSTRAINT FK_Cuenta_Ovalos FOREIGN KEY (CodigoOvalo)

REFERENCES Ovalos (Codigo) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FK_Cuenta_Moras FOREIGN KEY (CodigoMora)

REFERENCES Moras (Codigo) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FK_Cuenta_Usuarios FOREIGN KEY (CodigoUsuario)

REFERENCES Usuarios (Codigo) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE AportesExtras

(

```
Codigo int IDENTITY (1, 1) PRIMARY KEY,  
Nombre nvarchar (100) NOT NULL,  
Mes nvarchar (12) NOT NULL,  
Año int NOT NULL,  
Costo decimal (7,2) NOT NULL,  
NumeroCuenta int NOT NULL,  
Observacion nvarchar (100) NULL,  
Pagado nvarchar (2) NOT NULL,  
Periodo int NOT NULL,  
CONSTRAINT FK_AportesExtras_Cuenta FOREIGN KEY (NumeroCuenta)  
REFERENCES Cuenta (Numero) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE Actividades  
(  
Codigo int IDENTITY (1, 1) PRIMARY KEY,  
Nombre nvarchar (100) NOT NULL,  
Responsable nvarchar (100) NULL,  
Costo decimal (7,2) NOT NULL,  
Observacion nvarchar (100) NOT NULL,  
Periodo int NOT NULL  
);
```

```
CREATE TABLE RegistrarActividades  
(  
Codigo int IDENTITY (1, 1) PRIMARY KEY,  
Año nvarchar (4) NOT NULL,  
Falta nvarchar (2) NOT NULL,  
NumeroCuenta int NOT NULL,  
CodigoActividad int NOT NULL,  
Observacion nvarchar (100) NULL,  
Pagado nvarchar (2) NOT NULL,
```

```
Periodo int NOT NULL,  
CONSTRAINT FK_RegistrarActividades_Cuenta FOREIGN KEY (NumeroCuenta)  
REFERENCES Cuenta (Numero) ON DELETE CASCADE ON UPDATE CASCADE,  
CONSTRAINT FK_RegistrarActividades_Actividades FOREIGN KEY (CodigoActividad)  
REFERENCES Actividades (Codigo) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE Contribuciones  
(  
Codigo int IDENTITY (1, 1) PRIMARY KEY,  
Mes nvarchar (12) NOT NULL,  
Año int NOT NULL,  
Contribucion decimal (7, 2) NOT NULL,  
Recargo decimal (7, 2) NOT NULL,  
NumeroCuenta int NOT NULL,  
Observacion nvarchar (100) NULL,  
Pagado nvarchar (2) NOT NULL,  
Periodo int NOT NULL,  
CONSTRAINT FK_Contribuciones_Cuenta FOREIGN KEY (NumeroCuenta)  
REFERENCES Cuenta (Numero) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE Pagos  
(  
Numero int PRIMARY KEY,  
FechaPago datetime NOT NULL,  
TotalContribuciones decimal (7, 2) NOT NULL,  
TotalActividades decimal (7, 2) NOT NULL,  
TotalAportesExtras decimal (7, 2) NOT NULL,  
Subtotal decimal (7, 2) NOT NULL,  
Iva decimal (7, 2) NOT NULL,
```

```

Total decimal (7, 2) NOT NULL,
Abono decimal (7, 2) NOT NULL,
Recargo decimal (7, 2) NOT NULL,
NumeroCuenta int NOT NULL,
Estado nvarchar (9) NOT NULL,
Periodo int NOT NULL,
CONSTRAINT FK_Pagos_Cuenta FOREIGN KEY (NumeroCuenta)
REFERENCES Cuenta (Numero) ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE DetalleAportesExtras
(
Codigo int IDENTITY (1, 1) PRIMARY KEY,
NumeroPago int NOT NULL,
CodigoAportesExtras int NOT NULL,
CONSTRAINT FK_AportesExtras_DetalleAportesExtras FOREIGN KEY
(CodigoAportesExtras) REFERENCES AportesExtras (Codigo) ON DELETE CASCADE ON
UPDATE CASCADE,
CONSTRAINT FK_Pagos_DetalleAportesExtras FOREIGN KEY (NumeroPago)
REFERENCES Pagos (Numero)
);

```

```

CREATE TABLE DetalleActividades
(
Codigo int IDENTITY (1, 1) PRIMARY KEY,
NumeroPago int NOT NULL,
CodigoRegistrarActividades int NOT NULL,
CONSTRAINT FK_RegistrarActividades_DetalleActividades FOREIGN KEY
(CodigoRegistrarActividades) REFERENCES RegistrarActividades (Codigo) ON DELETE
CASCADE ON UPDATE CASCADE,
CONSTRAINT FK_Pagos_DetalleActividades FOREIGN KEY (NumeroPago) REFERENCES
Pagos (Numero)
);

```

);

CREATE TABLE DetalleContribuciones

(

Codigo int IDENTITY (1, 1) PRIMARY KEY,

NumeroPago int NOT NULL,

CodigoContribucion int NOT NULL,

CONSTRAINT FK_Contribuciones_DetalleContribuciones FOREIGN KEY

(CodigoContribucion) REFERENCES Contribuciones (Codigo) ON DELETE CASCADE ON
UPDATE CASCADE,

CONSTRAINT FK_Pagos_DetalleContribuciones FOREIGN KEY (NumeroPago)

REFERENCES Pagos (Numero)

);

CREATE TABLE Iva

(

Codigo int IDENTITY (1, 1) PRIMARY KEY,

Porcentaje decimal (7, 2) NOT NULL,

Periodo int NOT NULL

);

CREATE TABLE Tarifas

(

Codigo int IDENTITY (1, 1) PRIMARY KEY,

ValorHora decimal (7,2) NOT NULL,

ValorMinuto decimal (7,2) NOT NULL

);

CREATE TABLE AditoriaUsuarios

(

Operacion nvarchar (20),

Fecha datetime,

Usuario nvarchar (30),
CodigoUsuario int,
Cedula nvarchar (13),
Apellidos nvarchar (20),
Nombres nvarchar (20),
Direccion nvarchar (100),
Telefono nvarchar (9)
);

SEGUNDO PASO: CONFIGURACIONES

Este paso es fundamental para el funcionamiento del sistema, para que este se conecte con la base de datos creada, se necesita configurar la cadena de conexión.

La cadena de conexión cuenta con 4 parámetros para lo cual se debe colocar en el orden correspondiente, el primero es el servidor, el segundo es la base datos, el tercero es el usuario que se conecta al Gestor de Base de Datos y el cuarto es la clave del usuario configurado anteriormente, en cuanto a la función.

```
ConexionBD.CadenaDeConexion = @"Data Source=" + txtServidor.Text + ";Initial  
Catalog=" + txtBaseDatos.Text + ";User ID=" + txtUsuario.Text + ";Password=" +  
txtContraseña.Text + ";
```

Esta configuración se lo debe hacer en el formulario inicio de sesión para conectar el frontal con la base de datos

TERCER PASO: CÓDIGO FUENTE

Debido a que el código fuente utilizado para la elaboración de este proyecto es demasiado extenso, se procederá a indicar como muestra la administración de la información de las cuentas de usuarios y el código más relevante del sistema.

VALIDACIÓN DE USUARIOS PARA EL INGRESO AL SISTEMA

CLASE CONEXIÓN

```
using System;

namespace JuntasDLL
{
    public static class ConexionBD
    {
        //Creamos los parámetros de conexión para la Conexión.
        private static string _cadenaDeConexion;
        private static string _usuario;
        private static string _clave;
        private static string _baseDeDatos;
        private static string _servidor;

        //Creamos las Propiedades para la manipulación de los Campos.
        public static string CadenaDeConexion
        {
            get
            {
                return _cadenaDeConexion;
            }
            set
            {
                _cadenaDeConexion = value;
            }
        }

        public static string BaseDatos
        {
```

```

    get
    {
        return _baseDeDatos;
    }
    set
    {
        _baseDeDatos = value;
    }
}
public static string Clave
{
    get
    {
        return _clave;
    }
    set
    {
        _clave = value;
    }
}

```

```

public static string Servidor
{
    get
    {
        return _servidor;
    }
    set

```

```

        {
            _servidor = value;
        }
    }

    public static string Usuario
    {
        get
        {
            return _usuario;
        }
        set
        {
            _usuario = value;
        }
    }
}

```

MÉTODO PARA INICIAR SESION.

```

public void ConectarBaseDeDatos()
{
    SqlConnection cnConexion = new SqlConnection ();

    try
    {
        usuario = txtUsuario.Text;

        clave = txtContraseña.Text;

        baseDeDatos =txtBaseDeDatos.Text;

        ConexionBD.CadenaDeConexion = @"Data Source=" + txtServidor.Text +
        ";Initial Catalog=" + txtBaseDatos.Text + ";User ID=" + txtUsuario.Text +
        ";Password=" + txtContraseña.Text + ";

        ConexionBD.Usuario = txtUsuario.Text;
    }
}

```

```

        cnConexion.ConnectionString = ConexionBD.CadenaDeConexion;

        cnConexion.Open();
    }

    catch(SqlException error)
    {
        ControDelErrores.TratarErroresWindows(error);
    }

    finally
    {
        cnConexion.Close();
        cnConexion.Dispose();
    }
}

```

CLASE CUENTAS

```

using System;
using System.Text;
using System.Data;
using System.Data.SqlClient;
namespace JuntasDLL
{
    public class Cuentas
    {
        //Creamos los parámetros para la conexión de tabla Cuentas.
        private int _numerocuenta;
        private string _diariego;
        private string _horainicio;
        private string _tiemporiego;
        private string _frecuenciariego;
    }
}

```

```

private int _areaterreno;

private int _codigoovalo;

private int _codigousuario;

private int _periodo;

//Creamos el constructor para inicializar las Variables.

public Cuentas (int numerocuenta,string diariego, string horainicio, string tiemporiego,
string frecuenciariego,int areaterreno,int codigoovalo, int codigousuario, int periodo )

{

this._numerocuenta=numerocuenta;

this._diariego = diariego;

this._horainicio = horainicio;

this._tiemporiego = tiemporiego;

this._frecuenciariego = frecuenciariego;

this._areaterreno = areaterreno;

this._codigoovalo = codigoovalo;

this._codigousuario = codigousuario;

this._periodo = periodo;

}

//Creamos las propiedades para la manipulación de los Campos.

public int NumeroCuenta

{

    get

    {

        return this._numerocuenta;

    }

    set

    {

        this._numerocuenta = value;

    }

}

```

```

}
public string DiaRiego
{
    get
    {
        return this._diariego;
    }
    set
    {
        this._diariego = value;
    }
}
public string Horalnicio
{
    get
    {
        return this._horainicio;
    }
    set
    {
        this._horainicio = value;
    }
}
public string TiempoRiego
{
    get
    {

```

```

        return this._tiemporiego;
    }
    set
    {
        this._tiemporiego = value;
    }
}
public string FrecuenciaRiego
{
    get
    {
        return this._frecuenciariego;
    }
    set
    {
        this._frecuenciariego = value;
    }
}
public int AreaTerreno
{
    get
    {
        return this._areaterreno;
    }
    set
    {
        this._areaterreno = value;
    }
}

```



```
}  
public int CodigoOvalo  
{  
    get  
    {  
        return this._codigoovalo;  
    }  
    set  
    {  
        this._codigoovalo = value;  
    }  
}  
public int CodigoUsuario  
{  
    get  
    {  
        return this._codigousuario;  
    }  
    set  
    {  
        this._codigousuario = value;  
    }  
}  
public int Periodo  
{  
    get  
    {  
        return this._periodo;  
    }  
}
```

```

    }
    set
    {
        this._periodo = value;
    }
}

```

//Creamos el primer método de la clase Cuentas el cual servirá para insertar nuevas Cuentas.

```

public bool Insertar(Cuentas objetoCuentas, string nombreProcemientoAlmacenado)
{
    SqlConnection cnConexionCuentas = new SqlConnection();
    try
    {
        cnConexionCuentas.ConnectionString= ConexionBD.CadenaDeConexion;
        //Creamos un objeto de tipo Command.
        SqlCommand cmdInsert = new SqlCommand();
        cmdInsert.CommandText = nombreProcemientoAlmacenado;
        cmdInsert.CommandType = CommandType.StoredProcedure;
        cmdInsert.Connection = cnConexionCuentas;
        //Creamos los parámetros del procedimiento almacenado.
        cmdInsert.Parameters.Add("@DiaRiego", SqlDbType.NVarChar, 10);
        cmdInsert.Parameters.Add("@HoraInicio", SqlDbType.NVarChar, 10);
        cmdInsert.Parameters.Add("@TiempoRiego", SqlDbType.NVarChar, 10);
        cmdInsert.Parameters.Add("@FrecuenciaRiego", SqlDbType.NVarChar, 20);
        cmdInsert.Parameters.Add("@AreaTerreno", SqlDbType.Int);
        cmdInsert.Parameters.Add("@CodigoOvalo", SqlDbType.Int);
        cmdInsert.Parameters.Add("@CodigoUsuario", SqlDbType.Int);
    }
}

```

```

cmdInsert.Parameters.Add("@Periodo", SqlDbType.Int)

//Enviamos los valores para los parámetros.

cmdInsert.Parameters["@DiaRiego"].Value= objetoCuentas. DiaRiego;
cmdInsert.Parameters["@HoralInicio"].Value= objetoCuentas.HoralInicio;
cmdInsert.Parameters["@TiempoRiego"].Value=objetoCuentas.TiempoRiego;

cmdInsert.Parameters["@FrecuenciaRiego"].Value=objetoCuentas.FrecuenciaRiego;

cmdInsert.Parameters["@AreaTerreno"].Value=objetoCuentas.Arreaterreno;

cmdInsert.Parameters["@CodigoOvalo"].Value=objetoCuentas.CodigoOvalo;

cmdInsert.Parameters["@CodigoUsuario"].Value=objetoCuentas.CodigoUsuario;

cmdInsert.Parameters["@Periodo"].Value=objetoCuentas.Periodo;

//Abrimos la conexión con la BD.

cnConexionCuentas.Open();

int registrosAfectados = cmdInsert.ExecuteNonQuery();

bool selInserto;

if (registrosAfectados != 0)
    selInserto = true;

else
    selInserto = false;

return selInserto;
}

catch (SqlException error)
{
throw error;
}

catch (Exception error)

```

```

    {
        throw error;
    }
    finally
    {
        //Cerramos la Conexion con la BD Juntas.
        cnConexionCuentas.Close();
        //Liberamos memoria
        cnConexionCuentas.Dispose();
    }
}

```

//Método para actualizar Cuentas.

```

public bool Actualizar(Cuentas objetoCuentas, string nombreProcemientoAlmacenado)
{
    SqlConnection cnConexionCuentas = new SqlConnection();

    try
    {
        cnConexionCuentas.ConnectionString= ConexionBD.CadenaDeConexion;
        //Creamos un objeto de tipo Command.

        SqlCommand cmdUpdate = new SqlCommand();

        cmdUpdate.CommandText = nombreProcemientoAlmacenado;
        cmdUpdate.CommandType = CommandType.StoredProcedure;
        cmdUpdate.Connection = cnConexionCuentas;

        //Creamos los parámetros del procedimiento almacenado.
        cmdUpdate.Parameters.Add("@Codigo", SqlDbType.Int);
        cmdUpdate.Parameters.Add("@DiaRiego", SqlDbType.NVarChar, 10);
        cmdUpdate.Parameters.Add("@HoraInicio", SqlDbType.NVarChar, 10);
    }
}

```

```

cmdUpdate.Parameters.Add("@TiempoRiego", SqlDbType.NVarChar, 10);
cmdUpdate.Parameters.Add("@FrecuenciaRiego", SqlDbType.NVarChar,
20);
cmdUpdate.Parameters.Add("@AreaTerreno", SqlDbType.Int);
cmdUpdate.Parameters.Add("@CodigoOvalo", SqlDbType.Int);
cmdUpdate.Parameters.Add("@CodigoUsuario", SqlDbType.Int);
cmdUpdate.Parameters.Add("@Periodo", SqlDbType.Int);
//Enviamos los valores para los parámetros.
cmdUpdate.Parameters["@Codigo"].Value= objetoCuentas.Codigo;
cmdUpdate.Parameters["@DiaRiego"].Value= objetoCuentas. DiaRiego;
cmdUpdate.Parameters["@HoraInicio"].Value=objetoCuentas.HoraInicio;
cmdUpdate.Parameters["@TiempoRiego"].Value=objetoCuentas.TiempoRiego;
cmdUpdate.Parameters["@FrecuenciaRiego"].Value=objetoCuentas.FrecuenciaRiego;
cmdUpdate.Parameters["@AreaTerreno"].Value=objetoCuentas.Arreaterreno;
cmdUpdate.Parameters["@CodigoOvalo"].Value=objetoCuentas.CodigoOvalo;
cmdUpdate.Parameters["@CodigoUsuario"].Value=objetoCuentas.CodigoUsuario;
cmdUpdate.Parameters["@Periodo"].Value=objetoCuentas.Periodo;

//Abrimos la conexión con la BD.
cnConexionCuentas.Open();
int registrosAfectados = cmdInsert.ExecuteNonQuery();
bool seActualizo;
if (registrosAfectados != 0)
    seActualizo = true;
else

```

```

        seActualizo = false;
    return seActualizo;
}

catch (SQLException error)
{
    throw error;
}

catch (Exception error)
{
    throw error;
}

finally
{
    //Cerramos la Conexion con la BD Juntas.
    cnConexionCuentas.Close();
    //Liberamos memoria
    cnConexionCuentas.Dispose();
}
}

```

//Método para eliminar Cuentas.

```

public bool Eliminar(Cuentas objetoCuentas, string nombreProcemientoAlmacenado)
{
    SqlConnection cnConexionCuentas = new SqlConnection();
    try
    {
        cnConexionCuentas.ConnectionString= ConexionBD.CadenaDeConexion;
    }
}

```

```

//Creamos un objeto de tipo Command.
SqlCommand cmdDelete= new SqlCommand();
cmdDelete.CommandText = nombreProcedimientoAlmacenado;
cmdDelete.CommandType = CommandType.StoredProcedure;
cmdDelete.Connection = cnConexionCuentas;
//Creamos los parámetros del procedimiento almacenado.
cmdDelete.Parameters.Add("@Codigo", SqlDbType.Int);
//Enviamos los valores para los parámetros.
cmdDelete.Parameters["@Codigo"].Value=objetoCuentas.Codigo;

//Abrimos la conexión con la BD.
cnConexionCuentas.Open();
int registrosAfectados = cmdDelete.ExecuteNonQuery();
bool seElimino;
if (registrosAfectados != 0)
    seElimino = true;
else
    seElimino = false;
return seElimino;
}

catch (SqlException error)
{
throw error;
}

catch (Exception error)
{
throw error;
}
}

```

```

        finally
        {
            //Cerramos la Conexion con la BD Juntas.
            cnConexionCuentas.Close();
            //Liberamos memoria
            cnConexionCuentas.Dispose();
        }
    }

//Método para Seleccionar los datos de la tabla Cuentas
public DataSet Seleccionar( string nombreProcemientoAlmacenado)
{
    SqlConnection cnConexionCuentas = new SqlConnection();
    try
    {
        cnConexionCuentas.ConnectionString= ConexionBD.CadenaDeConexion;
        //Crearemos un objeto de tipo Command.
        SqlCommand cmdSelect= new SqlCommand();
        cmdSelect.CommandText = nombreProcemientoAlmacenado;
        cmdSelect.CommandType = CommandType.StoredProcedure;
        cmdSelect.Connection = cnConexionCuentas;

        SqlDataAdapter daCuentas = new SqlDataAdapter();
        daCuentas.SelectCommand = cmdSelect;
        DataSet dsDatosCuentas = new DataSet();
        daCuentas.Fill(dsDatosCuentas, "Cuenta");

        return dsDatosCuentas;
    }

    catch (SqlException error)

```



```

        {
            throw error;
        }
        catch (Exception error)
        {
            throw error;
        }
        finally
        {
            //Cerramos la Conexion con la BD Juntas.
            cnConexionCuentas.Close();
            //Liberamos memoria
            cnConexionCuentas.Dispose();
        }
    }
}

```

Codigo para mostrar cuentas

```

private void MostrarCuentas()
{
    registroActual = this.BindingContext[dsDatosCuentas, "Cuentas"].Position;
    totalRegistros = this.BindingContext[dsDatosCuentas, "Cuentas"].Count;
    txtNumeroCuenta.Text=
    dsDatosCuentas.Tables[0].Rows[registroActual]["NumeroCuenta"].ToString();
    cboDiaRiego.Text=
    dsDatosCuentas.Tables[0].Rows[registroActual]["DiaRiego"].ToString();
}

```

```

txtHoralInicio.Text=
dsDatosCuentas.Tables[0].Rows[registroActual]["HoralInicio"].ToString();
txtTiempoRiego.Text=
dsDatosCuentas.Tables[0].Rows[registroActual]["TiempoRiego"].ToString();
txtAreaTerreno.Text=
dsDatosCuentas.Tables[0].Rows[registroActual][" AreaTerreno"].ToString();
cboFrecuenciaRiego.Text=
dsDatosCuentas.Tables[0].Rows[registroActual][" FrecuenciaRiego"].ToString();
cboOvalos.Text=
dsDatosCuentas.Tables[0].Rows[registroActual]["CodigoOvalo"].ToString();
cboUsuarios.Text=
dsDatosCuentas.Tables[0].Rows[registroActual]["CodigoUsuario"].ToString();
txtPeriodo.Text=
dsDatosCuentas.Tables[0].Rows[registroActual]["Periodo"].ToString();
//Mostramos en el navegador el registro actual y el total de registros.
lblTotalRegistros.Text = " " + (registroActual + 1) + " de " + totalRegistros;
}

```

Codigo para vincular los ovalos al Combobox

```

private void VincularOvalos()
{
Ovalos ovalos = new Ovalos();
cboOvalos.DataSource=ovalos.Seleccionar("SeleccionarOvalos").Tables["Ovalos"];
cboOvalos.DisplayMember="Nombre";
cboOvalos.ValueMember="Codigo";
}

```

Codigo para vincular los usuarios al Combobox

```

private void VincularUsuarios()

```

```

{
    Usuarios usuarios = new Usuarios();

    cboUsuarios.DataSource=usuarios.Seleccionar("SeleccionarUsuarios").Tables["Usuarios"];

    cboUsuarios.DisplayMember="NombresCompleto";

    cboUsuarios.ValueMember="Codigo";
}

```

Codigo para insertar nuevos registros de las Cuentas

```

private void InsertarRegistro()
{
    if (!EstanDatosRequeridosIngresados())
    {
        return;
    }

    Cuentas cuentas = new Cuentas();

    DataTable dtDatosCuenta;

    dtDatosCuenta=cuentas.MetodoVerificarExistencia("VerificarExistencia")

    if(dtDatosCuenta.Rows.Count ==0)
    {
        cuentas.DiaRiego = cboDiaRiego.Text.Trim();
        cuentas.HoraInicio = txtHoraInicio.Text.Trim();
        cuentas.TiempoRiego = txtTiempoRiego.Text.Trim();

        cuentas.FrecuenciaRiego = cboFrecuencia.Text.Trim();
        cuentas.AreaTerreno = Convert.ToInt32( txtAreaTerreno.Text.Trim());
        cuentas.CodigoOvalo = Convert.ToInt32(cboOvalos.SelectedValue);
        cuentas.CodigoUsuario = Convert.ToInt32(cboUsuario.SelectedValue);
        cuentas.Periodo = Convert.ToInt32(txtPeriodo.Text.Trim());

        bool seInserto = cuentas.Insertar(cuentas, "InsertarCuentas");
    }
}

```

```

DeshabilitarControles();
}
else
{
    MessageBox.Show("La Cuenta ya existe, Por favor ingrese una nueva cuenta" , "
Error ", MessageBoxButtons.OK, MessageBoxIcon.Error);
    LimpiarControles();
}
}

```

Codigo para actualizar registros de las Cuentas

```

private void ActualizarRegistro()
{
    if (!EstanDatosRequeridosIngresados())
    {
        return;
    }
    Cuentas cuentas = new Cuentas();
    Cuentas.NumeroCuenta = Convert.ToInt32( txtNumeroCuenta.Text.Trim());
    cuentas.DiaRiego = cboDiaRiego.Text.Trim();
    cuentas.HoralInicio = txtHoralInicio.Text.Trim();
    cuentas.TiempoRiego = txtTiempoRiego.Text.Trim();
    cuentas.FrecuenciaRiego = cboFrecuencia.Text.Trim();
    cuentas.AreaTerreno = Convert.ToInt32( txtAreaTerreno.Text.Trim());
    cuentas.CodigoOvalo = Convert.ToInt32(cboOvalos.SelectedValue);
    cuentas.CodigoUsuario = Convert.ToInt32(cboUsuario.SelectedValue);
    cuentas.Periodo = Convert.ToInt32(txtPeriodo.Text.Trim());
    bool seActualizo = cuentas.Actualizar (cuentas, "ActualizarCuentas");
    DeshabilitarControles();
}

```

Código para eliminar registros de las Cuentas

```
private void EliminarRegistro()
{
    if (DialogResult.Yes == MessageBox.Show("¿Está seguro que desea borrar la
cuenta"+txtNumerCuenta.Text+"?", "Eliminar", MessageBoxButtons.YesNo, Mess
ageBoxIcon.Warning))
    {
        Cuentas cuentas = new Cuentas();
        cuentas.NumeroCuenta=Convert.ToInt32(txtNumeroCuenta.Text);
        bool seElimino = cuentas.Eliminar (cuentas, "EliminarCuentas");
        DeshabilitarControles();
    }
}
```

Código que permite filtrar los apellidos de los Usuarios.

```
private void BuscarPorApellidos()
{
    //Establecemos el filtro para el campo.
    string clausulaWhere = "Usuario LIKE '" + txtBuscarxApellidos.Text.Trim() + '%"
";
    Cuentas cuentas = new Cuentas();
    //Filtramos los datos.
    dvCuentas=cuentas.Where(dvCuentas,clausulaWhere);
    //Vinculamos la vista filtrada la DataGridView.
    dgvDatosCuentas.DataSource=dvCuentas;
}
```

Código que permite filtrar la cedula de los Usuarios.

```
private void BuscarPorCedula()
{
    //Establecemos el filtro para el campo.
    string clausulaWhere = "Cedula LIKE '" + txtBuscarxCedula.Text.Trim() + '%"
";
```

```

Cuentas cuentas = new Cuentas();

//Filtramos los datos.
dvCuentas=cuentas.Where(dvCuentas,clausulaWhere);

//Vinculamos la vista filtrada la DataGridView.
dgvDatosCuentas.DataSource=dvCuentas;
}

```

Código para calcular Contribuciones

```

private void CalcularContribuciones()
{
    //Tiempo Riego
    numerocuenta = Convert.ToInt32(txtNumeroCuenta.Text);
    Cuentas cuentas = new Cuentas();
    DataTable dtDatos;
    dtDatos = cuentas.SeleccionarPorClavePrimaria(numerocuenta);
    dgvDatosCuentas.DataSource= dtDatos;

    txtTiempoRiego.Text=
    dgvDatosCuentas.Rows[0].Cells["TiempoRiego"].Value.ToString();

    Tarifas tarifas = new Tarifas()
    dsDatosTarifas= tarifas.Seleccionar("SeleccionarTarifas");

    txtValorHora.Text=
    dsDatosTarifas.Tables[0].Rows[registroActual]["ValorHora"].ToString();

    txtValorMinuto.Text=
    dsDatosTarifas.Tables[0].Rows[registroActual]["ValorMinuto"].ToString(
    );

    double totalCostoHoras=0;

    double totalCostoFraccion=0;

    double totalcontribucion=0;

    txtTiempoRiego.Select(0,2);
    if (txtTiempoRiego.Text != "")
    {

```

```

totalCostoHoras=
Convert.ToDouble(txtTiempoRiego.SelectedText)*Convert.ToDouble(txt
ValorHora.Text);
txtTiempoRiego.Select(3,5);
if(Convert.ToInt32(txtTiempoRiego.SelectedText)!=0)
{
    if(Convert.ToInt32(txtTiempoRiego.SelectedText)<=30)
    {
        totalCostoFraccion=Convert.ToDouble(txtValorMinuto.Text);
    }
    else
    {
        totalCostoFraccion=Convert.ToDouble(txtValorHora.Text);
    }
}
totalcontribucion=totalCostoHoras+totalCostoFraccion;
txtContribucion.Text= totalcontribucion.ToString();
}
}

```

Código para calcular actividades a Pagar.

```

private void ActividadesAPagar()
{
    string cadena = txtNumeroCuenta.Text;
    numerocuenta= Convert.ToInt32(cadena);
    RegistrarActividades registraractividades = new RegistrarActividades();
    DataTable dtDatos;

```

```

dtDatos=
registraractividades.SeleccionarActividadesAPagar(numerocuenta,"Seleccionar
ActividadesAPagar");

dgvActividades.DataSource= dtDatos;

double totalactividades = 0;

foreach (DataGridViewRow row in dgvActividades.Rows)
{
    if(Convert.ToString(row.Cells["Faltó"].Value)!="")
    {
        if(Convert.ToString(row.Cells["Faltó"].Value)=="NO")
        {
            row.Cells["CostoFalta"].Value="0";
        }
        if(Convert.ToString(row.Cells["Faltó"].Value)=="SI")
        {
            totalactividades+=Convert.ToDouble(row.Cells["CostoFalt
a"].Value);
        }
    }
}

txtTotalActividades.Text=Convert.ToString(totalactividades);
}

```

Código para calcular Mora

```

private void CalcularMoraContribuciones()
{
    Contribuciones contribuciones = new Contribuciones ();

    DataTable dtDatos;

    string pagado="NO";

    dtDatos=
contribuciones.SeleccionarContribuciones(numerocuenta,pagadoconsumo);

```



```

dgvContribuciones.DataSource= dtDatos;

double totalcontribuciones = 0;

double totalrecargos = 0;

foreach (DataGridViewRow row in dgvContribuciones.Rows)
{
totalcontribuciones+=Convert.ToDouble(row.Cells["Contribucion"].Value);
totalrecargos +=Convert.ToDouble(row.Cells["Recargo"].Value);
periodoanteriorcontribucion = Convert.ToInt32(row.Cells["Año"].Value);
valorcontribucion = Convert.ToDouble(row.Cells["Contribucion"].Value);
porcentajemora= Convert.ToDouble(row.Cells["PorcentajeMora"].Value);
if(periodoanteriorcontribucion < PeriodoActual && pagado == "NO")
{
recargocontribucion= (valorcontribucion *porcentajemora)/100;
row.Cells["Recargo"].Value=recargocontribucion;

bool seActualizoContribuciones=
contribuciones.ActualizarValorRecargos(Contribuciones,"ActualizarValor
Recargos");
}
}
}
}

```

Codigo para Ingresar Pagos

```

private void InsertarPagos()
{
if (!EstanDatosRequeridosIngresados())
{
return;
}

if(DialogResult.Yes == MessageBox.Show("Desea guardar e imprimir la
factura?", "Mensaje", MessageBoxButtons.YesNo, MessageBoxIcon.Question))

```

```

{

//Insertar Datos Pagos.

Pagos pagos = new Pagos();

pagos.NumeroPago =Convert.ToInt32( txtNumeroPago.Text.Trim());

pagos.FechaPago= txtFechaPago.Text.Trim();

pagos.TotalContribucion=
Convert.ToDecimal(txtTotalContribucion.Text.Trim());

pagos.TotalActividades=
Convert.ToDecimal(txtTotalActividades.Text.Trim());

pagos.TotalAportesExtras=
Convert.ToDecimal(txtTotalAportesExtras.Text.Trim());

pagos.Subtotal=Convert.ToDecimal(txtSubTotal.Text.Trim());

pagos.Iva = Convert.ToDecimal(txtIva.Text.Trim());

pagos.Total= Convert.ToDecimal(txtTotalAPagar.Text.Trim());

pagos.Abono= Convert.ToDecimal(txtAbono.Text.Trim());

pagos.Recargo = Convert.ToDecimal(txtRecargo.Text.Trim());

pagos.NumeroCuenta= Convert.ToInt32(txtNumeroCuenta.Text.Trim());

pagos.Periodo = Convert.ToInt32(txtPeriodo.Text.Trim());

bool seInserto = pagos.Insertar(pagos, "InsertarPagos")

//Insertar DetalleContribuciones

DetalleContribuciones detallecontribucion = new
DetalleContribuciones();

codigocontribucion=Convert.ToInt32(dgvContribuciones.Rows[0].Cells["
Codigo"].Value);

detallecontribucion.NumeroPago = txtNumeroPago.Text;

detalleconsumo.CodigoConsumo = codigocontribucion;

bool seInsertodetallecontribuciones=
detallecontribucion.Insertar(detallecontribucion,"InsertarDetalleContrib
ucion");

//Insertar DetalleActividades

```

```

DetalleActividades detalleactividades = new DetalleActividades();

codigoactividad=Convert.ToInt32(dgvActividades.Rows[0].Cells["Codigo"
].Value);

detalleactividades.NumeroPago = txtNumeroPago.Text;

detalleactividades.CodigoConsumo = codigoactividad;

bool seInsertodetalleactividades=
detalleactividad.Insertar(detalleactividad, "InsertarDetalleActividades");

//Actualizar Estado de las Contribuciones

Contribuciones contribuciones = new Contribuciones();

foreach (DataGridViewRow row in dgvContribuciones.Rows)
{

codigocontribucion = Convert.ToInt32(row.Cells["Codigo"].Value);

contribuciones.NumeroCuenta=numerocuenta;

contribuciones.Codigo=codigocontribucion;

contribuciones.Pagado="SI";

bool seActualizoContribuciones=
consumo.ActualizarEstadoContribuciones(contribuciones,"ActualizarEst
adoContribuciones");

}

//Actualizar Estado de las Actividades

RegistrarActividades registraractividades = new RegistrarActividades();

foreach (DataGridViewRow row in dgvActividades.Rows)
{

codigoactividad = Convert.ToInt32(row.Cells["Codigo"].Value);

registraractividades.NumeroCuenta=numerocuenta;

registraractividades.Codigo=codigoactividad;

registraractividades.Pagado="SI";

bool seActualizoActividades=
registraractividades.ActualizarEstadoActividades(registraractividades,"A
ctualizarEstadoActividades");

```

```

}
//Imprimir Pagos
prtVentana.Document = prtDocumento;
DialogResult dr = prtVentana.ShowDialog();
if (dr == DialogResult.OK)
{
    int numeroCopias = prtDocumento.PrinterSettings.Copies;
    int paginalInicial = prtDocumento.PrinterSettings.FromPage;
    int paginaFinal = prtDocumento.PrinterSettings.ToPage;
    string nombreImpresora =
prtDocumento.PrinterSettings.PrinterName;
    numeropago = txtNumeroPago.Text;
    //Crear el reporte.
    DsRepImprimirFactura datos =
DALInformelImprimirFactura.GetInformes(numerocuenta,
numeropago);
    ReportelImprimirFactura reporte = new
ReportelImprimirFactura();
    //Asociar el conjunto de datos con el reporte.
    reporte.SetDataSource(datos);
    reporte.PrintOptions.PrinterName = nombreImpresora;
    reporte.PrintToPrinter(numeroCopias, false, paginalInicial,
paginaFinal);
    LimpiarControles();
}
}

```

ANEXO 4: Manual de Instalación

INTRODUCCIÓN:

El presente documento lo guiará paso a paso durante el proceso de instalación y manejo del sistema denominado: SISTEMA DE GESTIÓN UTILIZANDO SOFTWARE LIBRE PARA COBROS Y REGISTROS DE USUARIOS DE LA JUNTA DE AGUAS CHACÓN SEVILLA.

REQUISITOS MÍNIMOS PARA EL FUNCIONAMIENTO DEL SISTEMA:

Hardware

- Computador con procesador Pentium 4 o superior, memoria RAM mínima 1G, espacio libre en disco duro 10 GB.

Software

- Sistema Operativo Windows XP SP2 o superior
- SQL Server Express Edition
- Framework 3.5 SP1

Instalación

Para instalar el programa haga doble clic en el archivo setup.



Figura 6.16 Pantalla Setup

A continuación se ejecutará el asistente que le guiará durante el proceso de instalación.

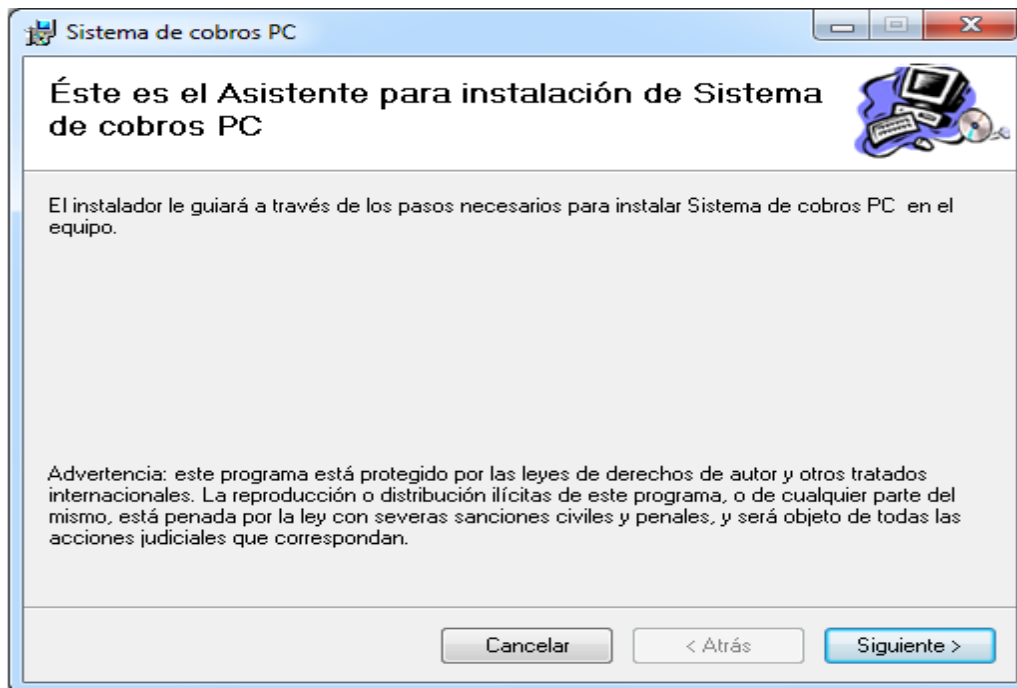


Figura 6.17 Pantalla asistente para instalación

En la siguiente ventana se deberá indicar el directorio donde desea que la aplicación se instale.

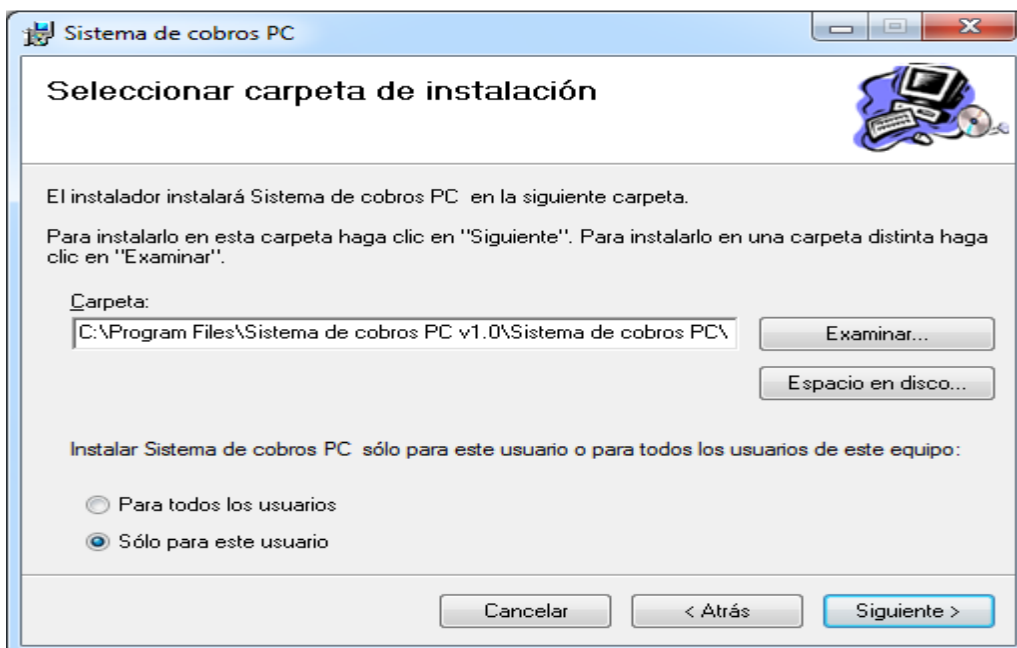


Figura 6.18 Pantalla seleccionar carpeta de instalación

Confirmamos la Instalación del Sistema, damos clic en siguiente:

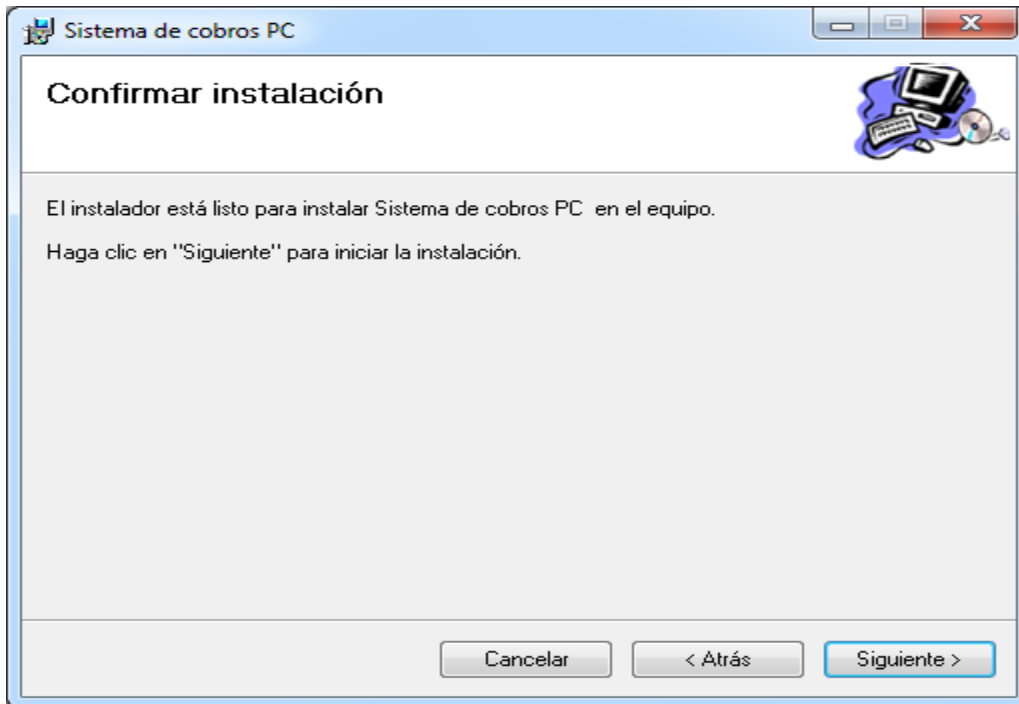


Figura 6.19 Pantalla confirmar instalación

La siguiente ventana muestra el proceso de instalación de los componentes del sistema.

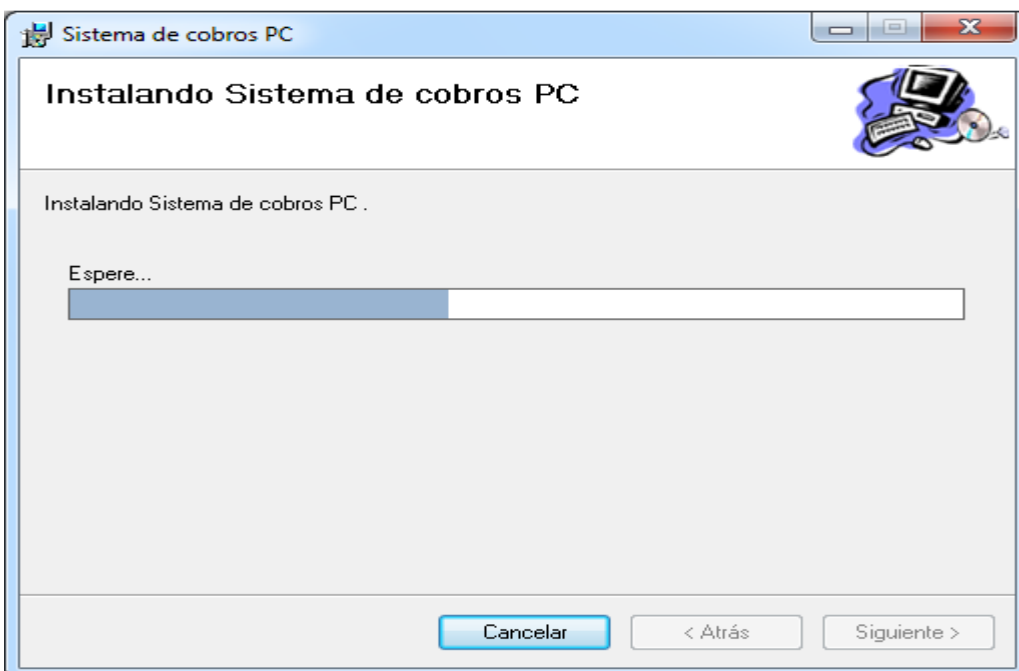
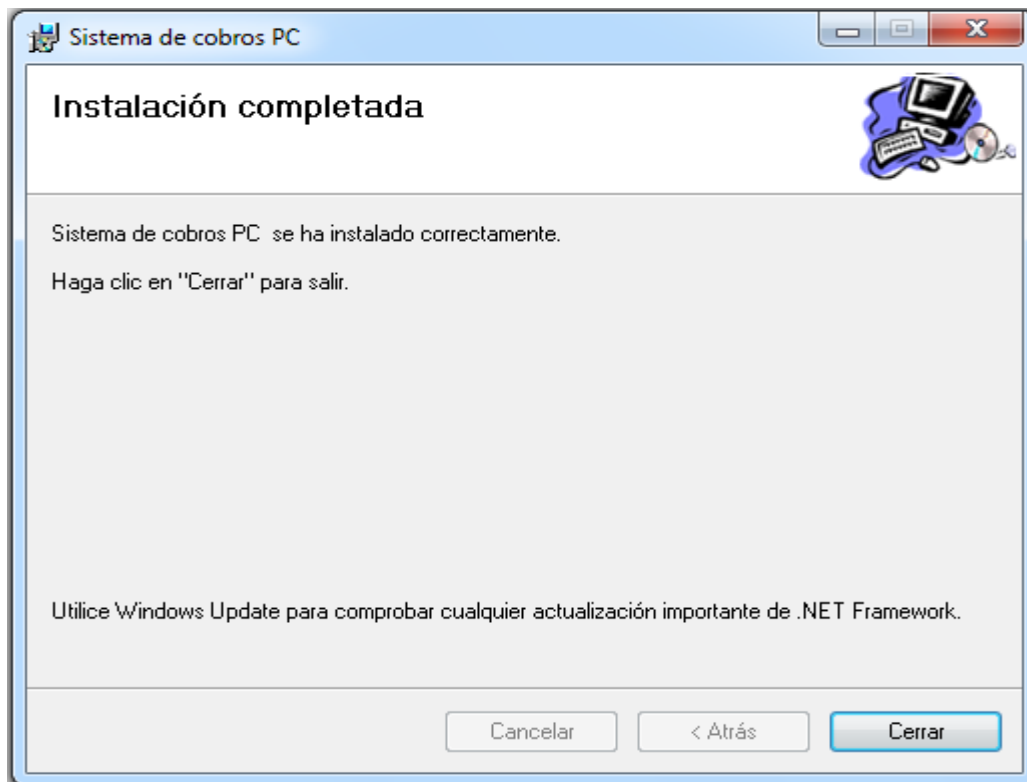


Figura 6.20 Pantalla proceso instalación

Finalmente se muestra la siguiente ventana donde indica que la Instalación se ha completado correctamente, damos clic en cerrar.



6.21 Figura Pantalla instalación completada

Se ha completado la instalación del Sistema de Cobros PC v1.0.

Manual Usuario del Sistema Cobros y registro de usuarios



Manual de usuario del Sistema de Cobros para administrar y controlar las tarifas recaudadas.

ÍNDICE

SISTEMA DE COBROS VERSIÓN 1.0.....	159
Inicio de Sesión.....	160
Perfiles de Usuario.....	160
Menú Principal.....	161
Manual Administrador.....	161
Manual Secretario(a).....	168

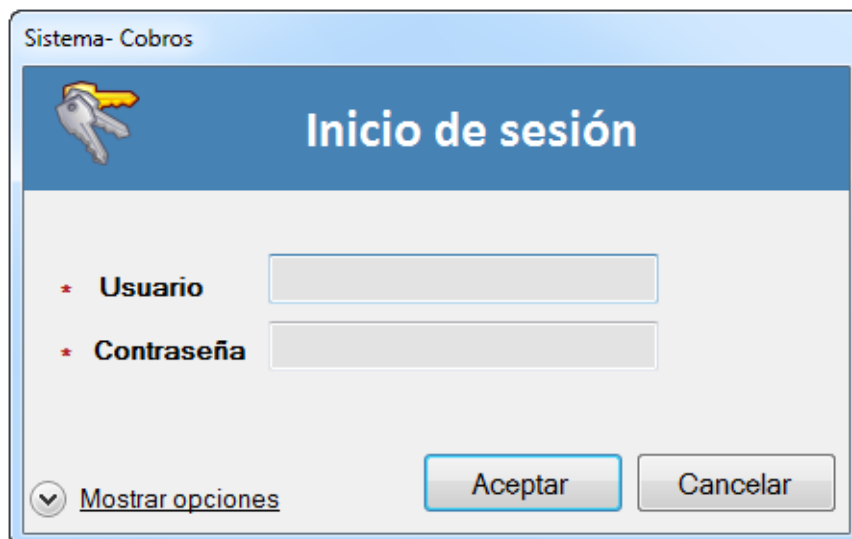
Sistema de Cobros Versión 1.0

Agradecemos la confianza depositada en nuestro software y le recomendamos leer este manual detenidamente, sobre todo la sección **“PRIMEROS PASOS”** para poder empezar a trabajar con la aplicación de forma inmediata.

Importante: Use **F1** para acceder al manual desde cualquier parte del sistema. Para acceder rápidamente a los diferentes módulos del programa, use el ratón desde el índice de este manual.

Inicio de Sesión

Cuando se accede al sistema la primera pantalla a la que es redirigido es la siguiente:



The image shows a screenshot of a software window titled "Sistema- Cobros". The window has a blue header bar with a key icon and the text "Inicio de sesión". Below the header, there are two input fields: "Usuario" and "Contraseña", each preceded by a red asterisk. At the bottom left, there is a dropdown menu labeled "Mostrar opciones" with a downward arrow. At the bottom right, there are two buttons: "Aceptar" and "Cancelar".

Figura 6.22 Ventana de inicio de sesión

En esta ventana se debe escribir el nombre del usuario y la contraseña. Únicamente los usuarios que se encuentren registrados en el sistema cobros podrán acceder al mismo.

Nota: Los campos de usuario y clave son sensibles a mayúsculas y minúsculas, deben ser ingresados de la misma manera como fueron dados de alta por el Administrador del sistema.

Perfiles de Usuario:

El sistema de cobros maneja 2 tipos de perfiles de usuario, cada uno de ellos tiene funciones específicas en el manejo del sistema, el cual define los limitantes con las que cuenta cada usuario. Los tipos de perfiles son los siguientes:

❖ **Administrador:**

Tiene control total en el sistema de cobros, no tiene limitaciones.

❖ **Secretario(a):**

Perfil utilizado para el manejo exclusivo de cobros, anulaciones facturas, generar reportes, hacer consultas. Limitaciones, no tiene permitido administración de los datos.

Menú Principal:

Cuando el usuario se encuentre habilitado aparecerá un menú con las siguientes opciones:

Principal, Datos, Cobros Informes, Finanzas, Configuraciones, Ayuda.



Figura 6.23 Pantalla Menú Principal

Manual Administrador:

El administrador del sistema será el encargado de administrar todo los datos sin restricciones



Figura 6.24 Pantalla Administrador

Pantalla Junta

Esta pantalla contiene la información básica de la junta, existen botones como guardar, editar, cancelar.

Junta

JUNTA

El Formulario permite la Administración de la información de la tabla Juntas

Guardar Editar Cancelar Cerrar

Nombre Junta: CHACON SEVILLA Periodo actual 2012

Ruc: _____-____

Dirección: _____

Teléfono: (____)____-____

Figura 6.25 Pantalla administrar junta

Pantalla Modificar Junta

Para poder modificar la información de la Junta se debe presionar el botón **Editar** que se encuentra en la barra de mantenimiento.

Junta

JUNTA

El Formulario permite la Administración de la información de la tabla Juntas

Guardar Editar Cancelar Cerrar

Nombre Junta: CHACON SEVILLA Periodo actual 2012

Ruc: 1800789921-001

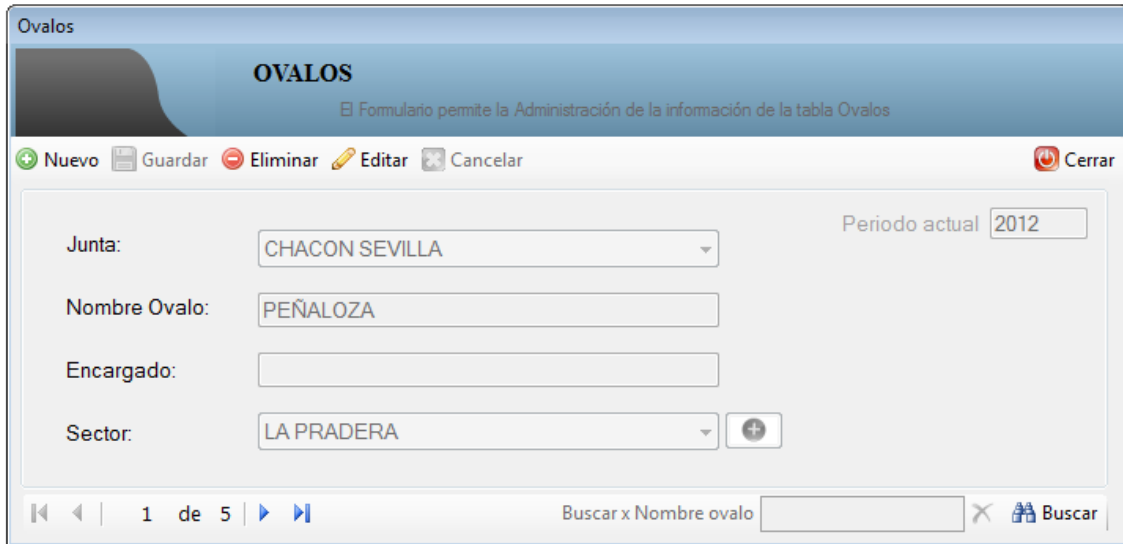
Dirección: AV.LAS PALMAS

Teléfono: (032)999-023

Figura 6.26 Pantalla modificar junta

Pantalla Óvalos

Esta pantalla contiene la información básica de los óvalos, existen botones como nuevo, guardar, editar, cancelar.

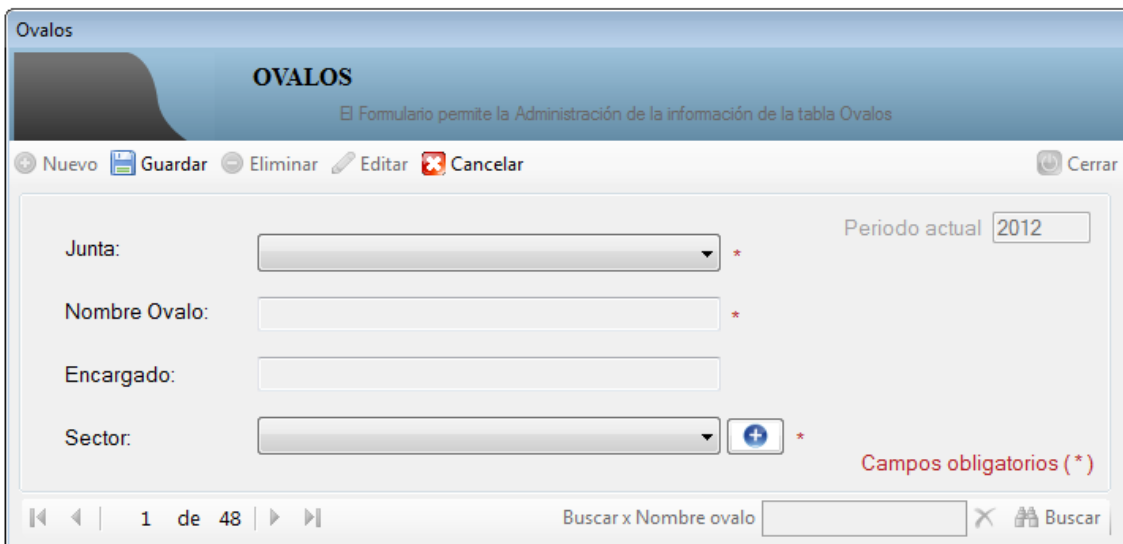


The screenshot shows a web application interface for managing 'Ovalos'. The title bar reads 'Ovalos' and 'OVALOS'. Below the title, a subtitle states 'El Formulario permite la Administración de la información de la tabla Ovalos'. A toolbar contains icons for 'Nuevo' (New), 'Guardar' (Save), 'Eliminar' (Delete), 'Editar' (Edit), 'Cancelar' (Cancel), and 'Cerrar' (Close). The main form area includes a dropdown menu for 'Junta' (set to 'CHACON SEVILLA'), a text input for 'Nombre Ovalo' (set to 'PEÑALOZA'), an empty text input for 'Encargado', and a dropdown menu for 'Sector' (set to 'LA PRADERA') with a '+' button. A 'Periodo actual' (Current Period) field is set to '2012'. At the bottom, there are navigation arrows, a page indicator '1 de 5', and a search bar labeled 'Buscar x Nombre ovalo' with a 'Buscar' button.

Figura 6.27 Pantalla administrar óvalos

Pantalla de ingreso del óvalo

Para poder ingresar la información de un nuevo óvalo se debe presionar el botón **Nuevo** que se encuentra en la barra de mantenimiento.

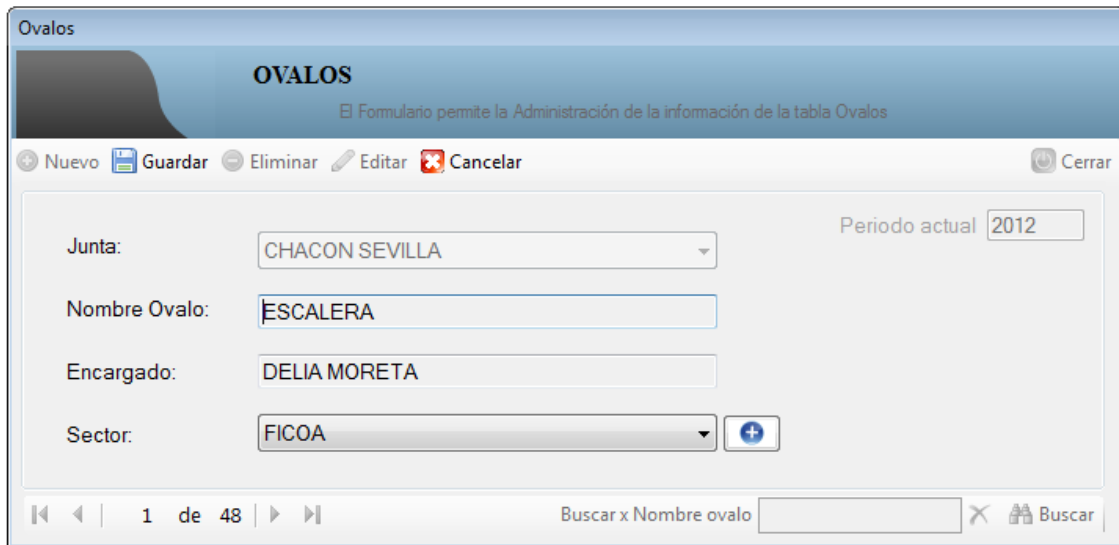


This screenshot shows the same 'Ovalos' interface, but with the 'Nuevo' button in the toolbar highlighted. The form fields are now empty, and red asterisks (*) are placed next to the 'Junta', 'Nombre Ovalo', and 'Sector' fields to indicate they are mandatory. A legend at the bottom right states 'Campos obligatorios (*)'. The page indicator now shows '1 de 48'.

Figura 6.29 Pantalla de ingreso óvalo

Pantalla modificación del óvalo

Para poder modificar la información de un óvalo se debe presionar el botón **Editar** que se encuentra en la barra de mantenimiento.



The screenshot shows a web application window titled 'Ovalos'. The main header contains the text 'OVALOS' and a subtitle 'El Formulario permite la Administración de la información de la tabla Ovalos'. Below the header is a toolbar with buttons for 'Nuevo', 'Guardar', 'Eliminar', 'Editar', 'Cancelar', and 'Cerrar'. The main form area contains several input fields: 'Junta:' with a dropdown menu showing 'CHACON SEVILLA', 'Nombre Ovalo:' with a text input field containing 'ESCALERA', 'Encargado:' with a text input field containing 'DELIA MORETA', and 'Sector:' with a dropdown menu showing 'FICOA' and a plus sign button. To the right of the 'Junta' field is a 'Periodo actual' field with the value '2012'. At the bottom of the window, there is a pagination control showing '1 de 48' and a search bar labeled 'Buscar x Nombre ovalo' with a search icon and the text 'Buscar'.

Figura 6.30 Pantalla modificación óvalo

Pantalla de eliminación óvalo

El usuario administrador puede eliminar el óvalo siempre y cuando no tenga asignado algún sector, seleccionar el óvalo y dar clic en eliminar se mostrará el siguiente mensaje de advertencia, si da clic en aceptar el óvalo se elimina caso contrario cancela la petición.

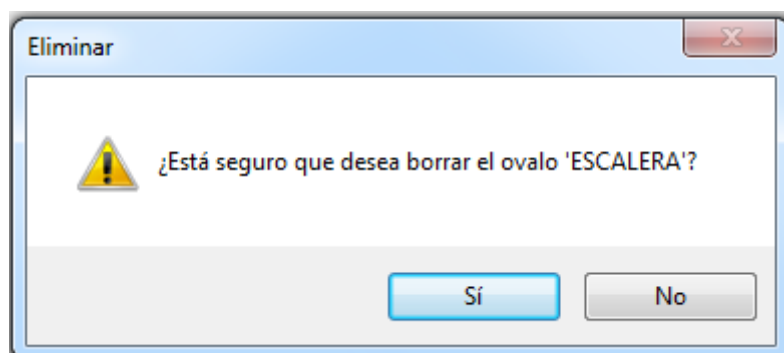



Figura 6.31 Pantalla de eliminación de óvalo

Pantalla Usuarios

Esta pantalla contiene la información básica de los usuarios, existen botones como nuevo, guardar, editar, cancelar.

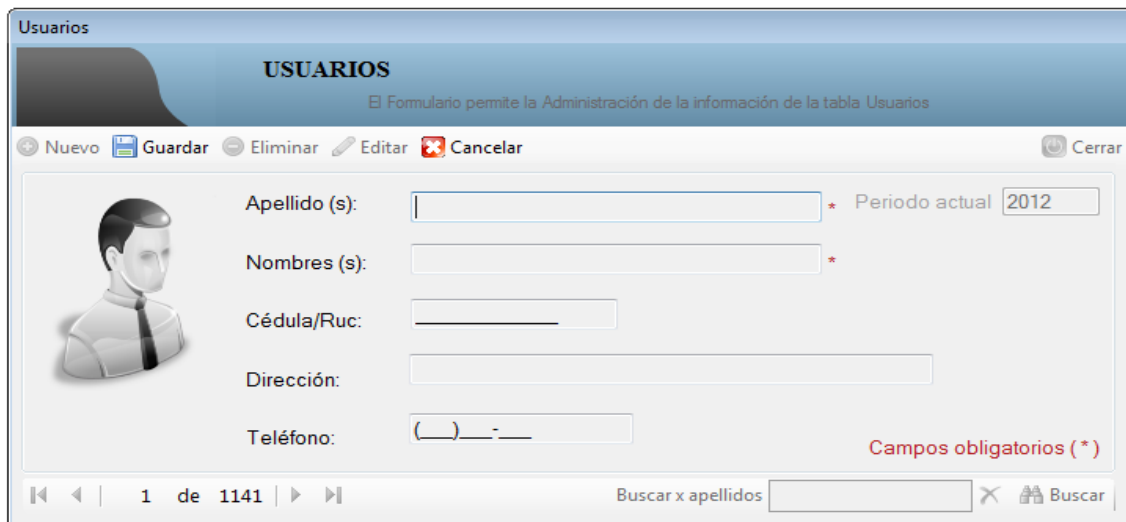


The screenshot shows a web application interface for user management. At the top, there's a header with the title 'USUARIOS' and a subtitle 'El Formulario permite la Administración de la información de la tabla Usuarios'. Below the header is a toolbar with buttons for 'Nuevo', 'Guardar', 'Eliminar', 'Editar', 'Cancelar', and 'Cerrar'. The main area contains a form for a user profile. On the left is a placeholder image of a man in a suit. The form fields are: 'Apellido (s):' with the value 'LASCANO PEREZ', 'Nombres (s):' with 'HUGO FERNANDO', 'Cédula:' with '1800441831', 'Dirección:' with 'AV. J DE SAN MARTIN - # 1', and 'Teléfono:' with '(032)856-754'. There is also a 'Periodo actual' field with the value '2012'. At the bottom, there are navigation controls showing '1 de 4' and a search bar with the text 'Buscar x' and a magnifying glass icon.

Figura 6.32 Pantalla administrar usuarios

Pantalla de ingreso usuarios

Para poder ingresar la información de un nuevo usuario se debe presionar el botón **Nuevo** que se encuentra en la barra de mantenimiento.

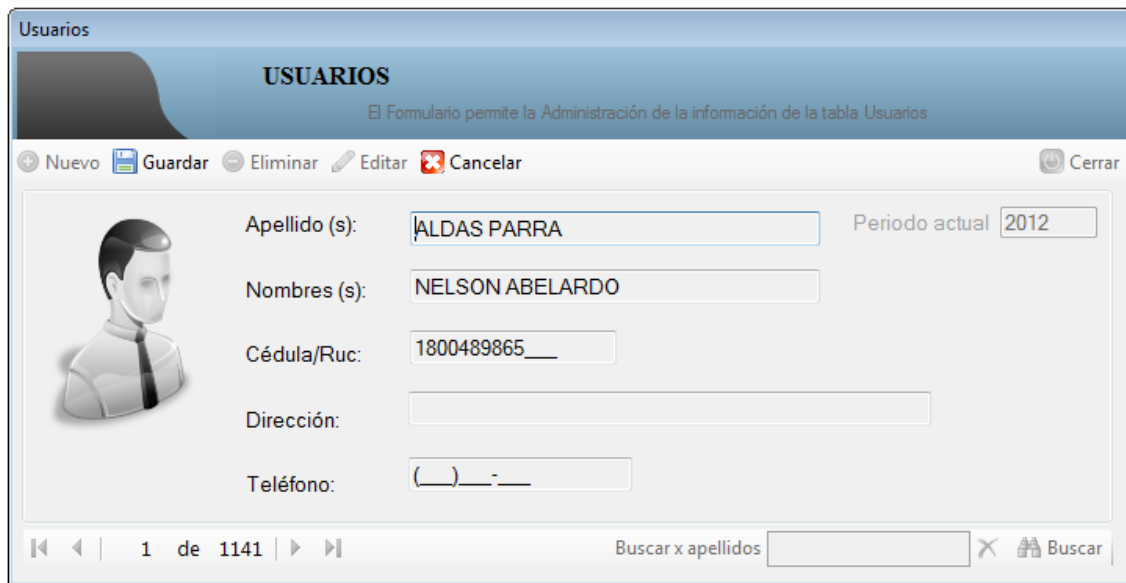


This screenshot shows the same 'USUARIOS' interface but in 'Nuevo' (New) mode. The toolbar now has a 'Nuevo' button highlighted. The form fields are empty, except for 'Apellido (s):' and 'Nombres (s):' which have red asterisks indicating they are required fields. The 'Cédula/Ruc:' field has a red asterisk. The 'Teléfono:' field has a red asterisk. At the bottom right, there is a red text label 'Campos obligatorios (*)'. The navigation controls at the bottom show '1 de 1141' and a search bar with the text 'Buscar x apellidos'.

Figura 6.33 Pantalla ingreso de usuarios

Pantalla modificación de usuario

Para poder modificar la información de un usuario se debe presionar el botón **Editar** que se encuentra en la barra de mantenimiento.



The screenshot shows a web application window titled 'Usuarios' with a sub-header 'USUARIOS'. Below the header, a message states: 'El Formulario permite la Administración de la información de la tabla Usuarios'. A toolbar contains buttons for 'Nuevo', 'Guardar', 'Eliminar', 'Editar', 'Cancelar', and 'Cerrar'. The main form area includes a user profile icon and several input fields: 'Apellido (s):' with the value 'ALDAS PARRA', 'Nombres (s):' with 'NELSON ABELARDO', 'Cédula/Ruc:' with '1800489865__', 'Dirección:', and 'Teléfono:'. A 'Periodo actual' field shows '2012'. At the bottom, there are navigation arrows, a page indicator '1 de 1141', and a search bar labeled 'Buscar x apellidos' with a magnifying glass icon and a 'Buscar' button.

Figura 6.34 Pantalla ingreso de usuarios

Pantalla de eliminación usuario

Para eliminar, seleccionar el usuario y dar clic en eliminar se mostrará el siguiente mensaje de advertencia, si da clic en aceptar el registro se perderá definitivamente caso contrario cancela la petición.

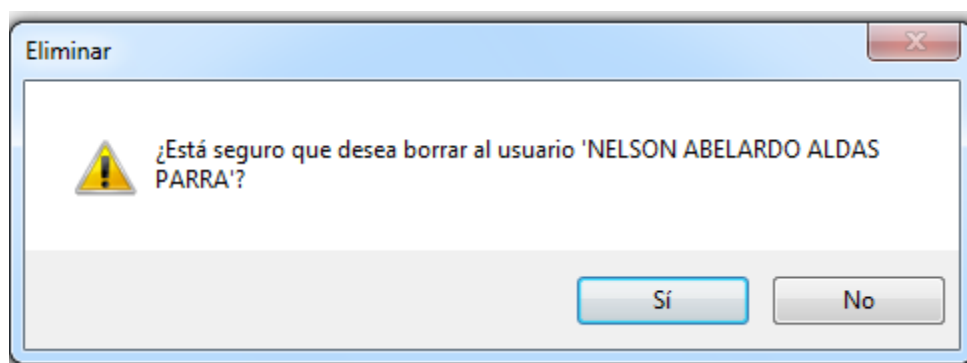


Figura 6.35 Pantalla eliminación de usuarios

Pantalla Cuenta

Esta pantalla contiene la información básica de las cuentas, existen botones como nuevo, eliminar, editar, cancelar.

	NumeroCuenta	Usuario	Ovalo	DiaRiego	TiempoRiego	Horainicio
▶	7	GARCES SANCHEZ ANGELA YADIRA	NUEVO OVALO	JUEVES	06:00	12:00
	8	GARCES SANCHEZ ANGELA YADIRA	PEÑALOZA	MIERCOLES	03:00	10:00
	2	LASCANO PEREZ HUGO FERNANDO	PEÑALOZA	MARTES	01:00	00:00
	3	MESIAS HARO LUIS ISRAEL	PEÑALOZA	VIERNES	02:00	00:00
	5	PAUCAR JAQUE DAVID JOSE	TORIBIO CAMINO	VIERNES	02:00	12:00

Figura 6.36 Pantalla administrar cuenta

Pantalla de ingreso cuenta

Para poder ingresar la información de una nueva cuenta se debe presionar el botón **Nuevo** que se encuentra en la barra de mantenimiento.

Periodo actual 2012

Datos Generales

Usuario: [dropdown] + *

Area del terreno: [input] m2 Ovalo: [dropdown] + *

Dia de riego: [dropdown] * Frecuencia: [dropdown] *

Tiempo de riego: [input] * Hora inicio: [input]

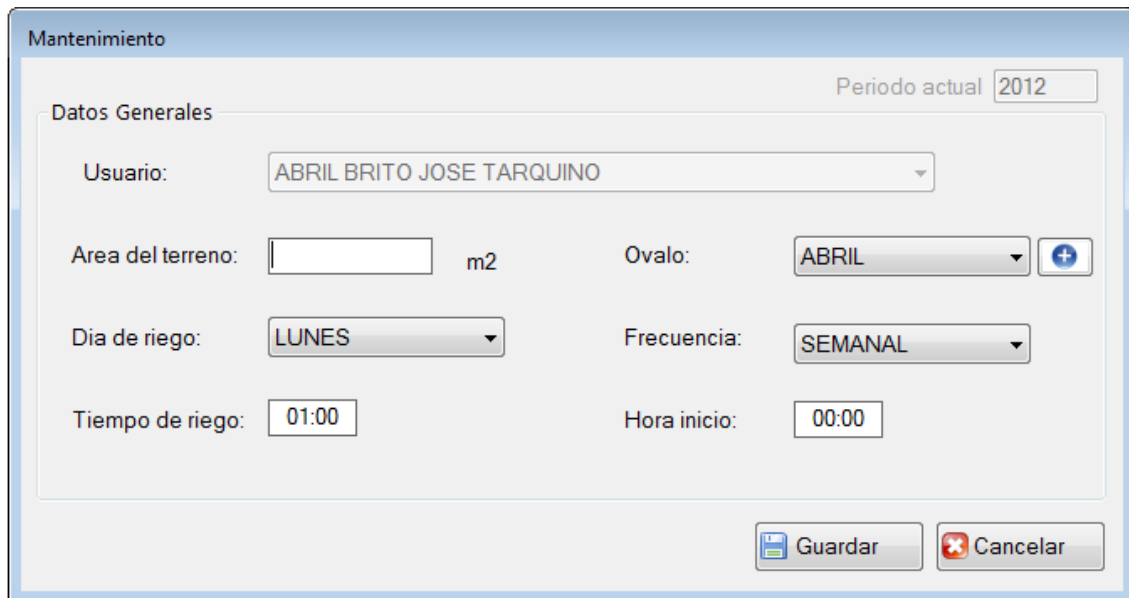
Campos obligatorios (*)

Guardar Cancelar

Figura 6.37 Pantalla ingreso cuenta

Pantalla modificación de cuenta

Para poder modificar la información de una cuenta se debe presionar el botón **Editar** que se encuentra en la barra de mantenimiento.

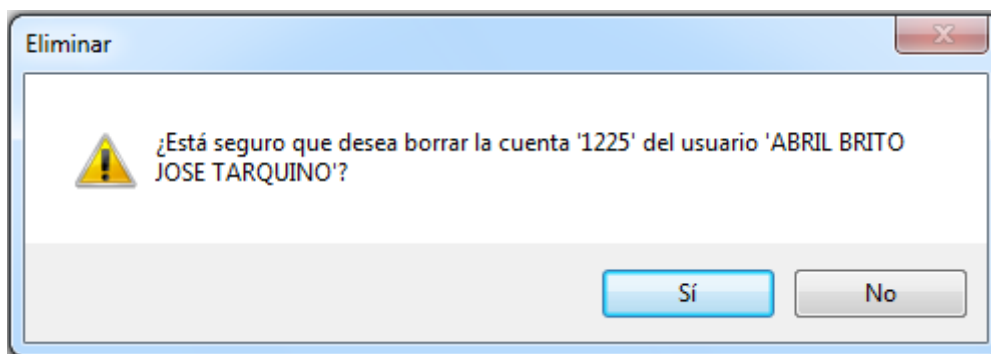


The screenshot shows a window titled "Mantenimiento" with a sub-section "Datos Generales". In the top right corner, there is a "Periodo actual" field set to "2012". The main area contains several fields: "Usuario" is a dropdown menu showing "ABRIL BRITO JOSE TARQUINO"; "Area del terreno" is a text input field followed by "m2"; "Ovalo" is a dropdown menu showing "ABRIL" with a "+" icon; "Dia de riego" is a dropdown menu showing "LUNES"; "Frecuencia" is a dropdown menu showing "SEMANAL"; "Tiempo de riego" is a text input field showing "01:00"; and "Hora inicio" is a text input field showing "00:00". At the bottom right, there are two buttons: "Guardar" (with a floppy disk icon) and "Cancelar" (with a red X icon).

Figura 6.38 Pantalla modificación cuenta

Pantalla de eliminación cuenta

Para eliminar, seleccionar la cuenta y dar clic en eliminar se mostrará el siguiente mensaje de advertencia, si da clic en aceptar el registro se perderá definitivamente caso contrario cancela la petición.



The screenshot shows a dialog box titled "Eliminar" with a yellow warning triangle icon. The text inside the dialog box reads: "¿Está seguro que desea borrar la cuenta '1225' del usuario 'ABRIL BRITO JOSE TARQUINO'?". At the bottom of the dialog box, there are two buttons: "Sí" and "No".

Figura 6.39 Pantalla eliminación cuenta

Pantalla Actividad

Esta pantalla contiene la información básica de las cuentas, existen botones como nuevo, eliminar, editar, cancelar.

The screenshot shows a web application window titled 'Actividad'. At the top, there is a header with the title 'ACTIVIDAD' and a subtitle 'El Formulario permite la Administración de la información de la tabla Actividad'. Below the header is a maintenance bar with icons for 'Nuevo', 'Guardar', 'Eliminar', 'Editar', 'Cancelar', and 'Cerrar'. The main form area contains the following fields: 'Actividad:' with the value '1 PEON', 'Responsable:' (empty), 'Costo (Falta):' with the value '10,00 \$', and 'Observación' (empty). On the right side, there is a 'Periodo actual' field with the value '2012'. At the bottom of the form, there is a pagination control showing '1 de 1' and a search bar labeled 'Buscar x Nombre actividad' with a 'Buscar' button.

Figura 6.40 Pantalla administrar actividad

Pantalla ingreso actividad

Para poder ingresar la información de una nueva actividad se debe presionar el botón **Nuevo** que se encuentra en la barra de mantenimiento.

This screenshot shows the same 'Actividad' screen as Figure 6.40, but with the 'Nuevo' button in the maintenance bar highlighted. The form fields are now empty. The 'Actividad:' field has a red asterisk next to it. The 'Costo (Falta):' field has a '\$' symbol and a red asterisk next to it. At the bottom right of the form, there is a red text label 'Campos obligatorios (*)'. The search bar and pagination control remain the same.

Figura 6.41 Pantalla ingreso de actividad

Pantalla modificación actividad

Para poder modificar la información de una actividad se debe presionar el botón **Editar** que se encuentra en la barra de mantenimiento.

Actividad

ACTIVIDAD

El Formulario permite la Administración de la información de la tabla Actividad

Nuevo Guardar Eliminar Editar Cancelar Cerrar

Actividad: Periodo actual

Responsable:

Costo (Falta):

Observación

2 de 2 | Buscar x Nombre actividad Buscar

Figura 6.42 Pantalla modificación de actividad

Pantalla de eliminación actividad

Para eliminar, seleccionar la cuenta y dar clic en eliminar se mostrará el siguiente mensaje de advertencia, si da clic en aceptar el registro se perderá definitivamente caso contrario cancela la petición.

Eliminar

¿Está seguro que desea borrar la Actividad 'SESION 2012'?

Sí No

Figura 6.43 Pantalla eliminación de actividad

Pantalla asistencias

Esta pantalla contiene la información básica de las cuentas, existen botones como nuevo, eliminar, editar, cancelar.

ASISTENCIA USUARIOS
El Formulario permite la Administración de la información de la tabla Asistencia Usuarios

Nuevo Eliminar Editar Cancelar Cerrar

Buscar Usuario

	NumeroCuenta	Usuario	Actividad	Fecha	Faltó	Periodo	Pagado
▶	2	LASCANO PEREZ HUGO FERNANDO	SESION 1	20/01/2012 18:04	SI	2012	SI
	2	LASCANO PEREZ HUGO FERNANDO	MINGAS	29/01/2012 10:00	NO	2012	SI
	3	MESIAS HARO LUIS ISRAEL	SESION 1	20/01/2012 18:04	SI	2012	NO
	5	PAUCAR JAQUE DAVID JOSE	SESION 1	20/01/2012 18:04	SI	2012	SI
	5	PAUCAR JAQUE DAVID JOSE	MINGAS	29/01/2012 10:00	NO	2012	SI
	5	PAUCAR JAQUE DAVID JOSE	OTRA	// :	NO	2012	SI

Figura 6.44 Pantalla administrar asistencias

Pantalla ingreso asistencia

Para poder ingresar la información de una nueva actividad se debe presionar el botón **Nuevo** que se encuentra en la barra de mantenimiento.

Mantenimiento Asistencia

Periodo actual 2012

Actividad: SESION 1 +

Usuario: LASCANO PEREZ HUGO FERNANDO No. Cuenta 2

Faltó: SI

Observación: EL USUARIO NO ASISTIO A LA SESION 1

Guardar Cancelar

Figura 6.45 Pantalla ingreso asistencia

Pantalla Contribuciones

Esta pantalla contiene la información básica de las cuentas, existen botones como nuevo, eliminar, editar, cancelar.

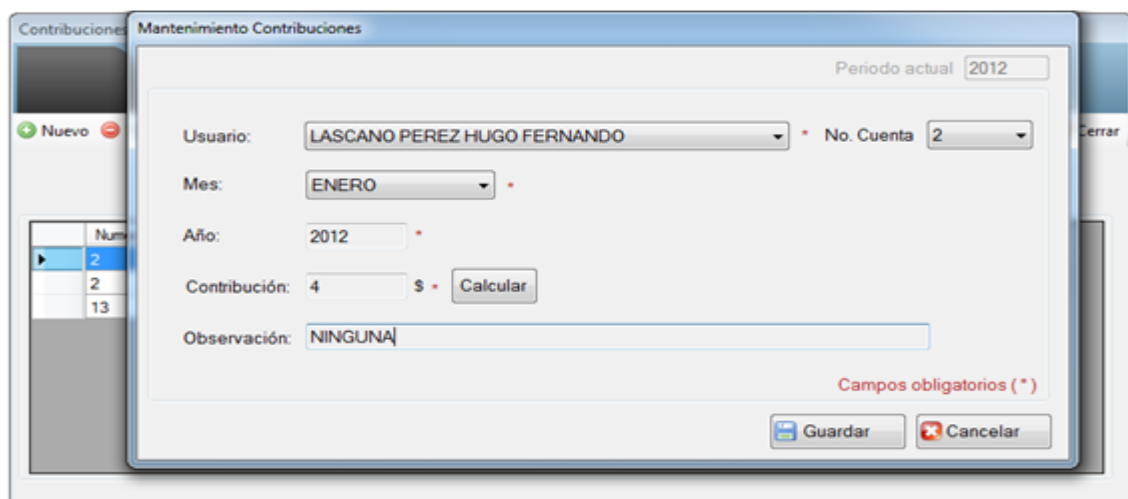


	NumeroCuenta	Usuario	Mes	Año	Contribución	Período	Pagado
▶	2	LASCANO PEREZ HUGO FERNANDO	ENERO	2011	4.00	2011	SI
	2	LASCANO PEREZ HUGO FERNANDO	MAYO	2012	4.00	2012	SI
	13	MATAMOROS MORALES CARLOS JOSE	ENERO	2012	16.00	2012	SI

Figura 6.46 Pantalla administración contribuciones

Pantalla ingreso contribuciones

Para poder ingresar la información de una nueva contribución se debe presionar el botón **Nuevo** que se encuentra en la barra de mantenimiento.



Periodo actual: 2012

Usuario: LASCANO PEREZ HUGO FERNANDO * No. Cuenta: 2

Mes: ENERO *

Año: 2012 *

Contribución: 4 \$ *

Observación: NINGUNA

Campos obligatorios (*)

Figura 6.47 Pantalla ingreso contribución

Manual Secretaria(o)

La Secretaria(o) del sistema será el encargado de facturar a los usuarios, tendrá restringido el ingreso de datos.

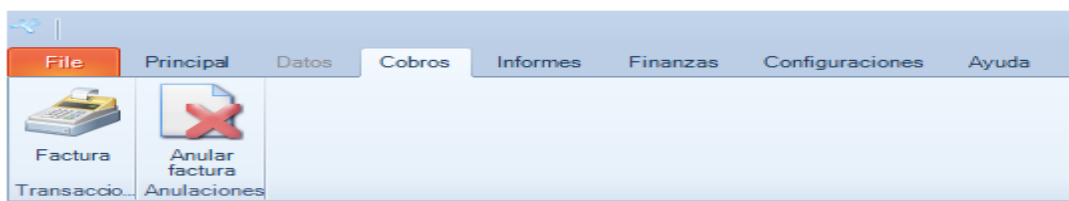
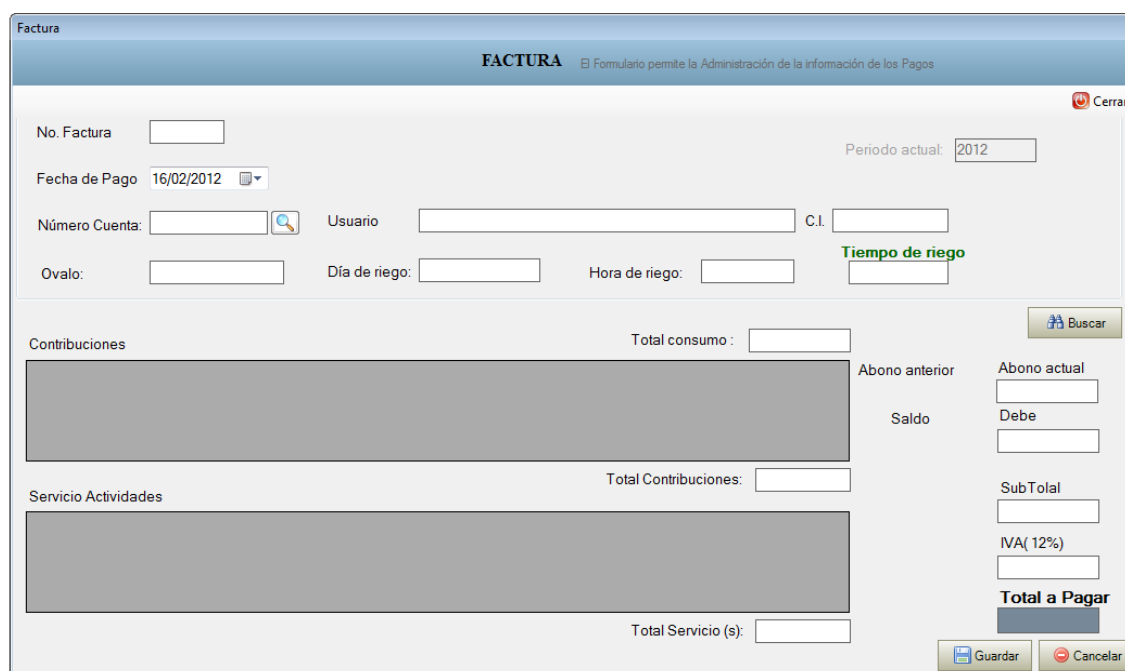


Figura 6.48 Pantalla opción cobros

Pantalla de Factura

Esta pantalla llamada factura contiene la información de los cobros, donde se detalla el número de factura, la fecha que se va a realizar el pago, la cuenta que se va a realizar el pago, el número de mes o meses que se va a cancelar, se deberá seleccionar los contribuciones que se van a pagar. Aparece además un detalle con todas las actividades o servicios que deben cancelar los que son facturables y los que no van hacer facturables.

The image shows a 'Factura' form with the following fields and sections:

- Header: 'FACTURA' and 'El Formulario permite la Administración de la información de los Pagos'. A 'Cerrar' button is in the top right.
- Form fields: 'No. Factura' (text box), 'Fecha de Pago' (calendar icon, value: 16/02/2012), 'Periodo actual' (value: 2012), 'Número Cuenta' (text box with search icon), 'Usuario' (text box), 'C.L.' (text box), 'Ovalo' (text box), 'Día de riego' (text box), 'Hora de riego' (text box), and 'Tiempo de riego' (text box).
- Buttons: 'Buscar' (magnifying glass icon).
- Summary section: 'Total consumo :', 'Abono anterior', 'Abono actual', 'Saldo', 'Debe', 'SubTotal', 'IVA(12%)', and 'Total a Pagar' (highlighted in a dark box).
- Footer: 'Total Servicio (s):' and 'Guardar' (floppy disk icon) and 'Cancelar' (red circle with X icon) buttons.

Figura 6.49 Pantalla factura

Pantalla ingreso nueva factura

Para poder registrar un nuevo cobro, se debe ingresar número factura, fecha de pago, número cuenta y presionar buscar para que busque las deudas de los usuarios. Para guardar e Imprimir se debe presionar en el botón guardar, la misma que se almacenarán siempre y cuando no existan errores en la validación de cada campo.

Factura

FACTURA El Formulario permite la Administración de la información de los Pagos

Cerrar

No. Factura

Fecha de Pago

Número Cuenta:

Ovalo:

Periodo actual:

Usuario C.I.

Día de riego: Hora de riego: Tiempo de riego

Contribuciones Total consumo:

▶	Codigo	Contribución	AñoContribución	Costo
▶	3	APORTE EXTRA	2010	10,00
*				

Total Contribuciones:

Abono anterior

Saldo 28

Abono actual

Debe

SubTotal

IVA(12%)

Total a Pagar

28

Servicio Actividades Total Servicio (s):

▶	Codigo	Actividad	Faltó	Fecha	CostoFalta
▶	17	SESION 1	SI	20/01/2012	10,00
	18	MINGAS	NO	29/01/2012	0
*					

Figura 6.50 Pantalla Factura

Pantalla de anulaciones

Esta pantalla permite anular la factura emitida.

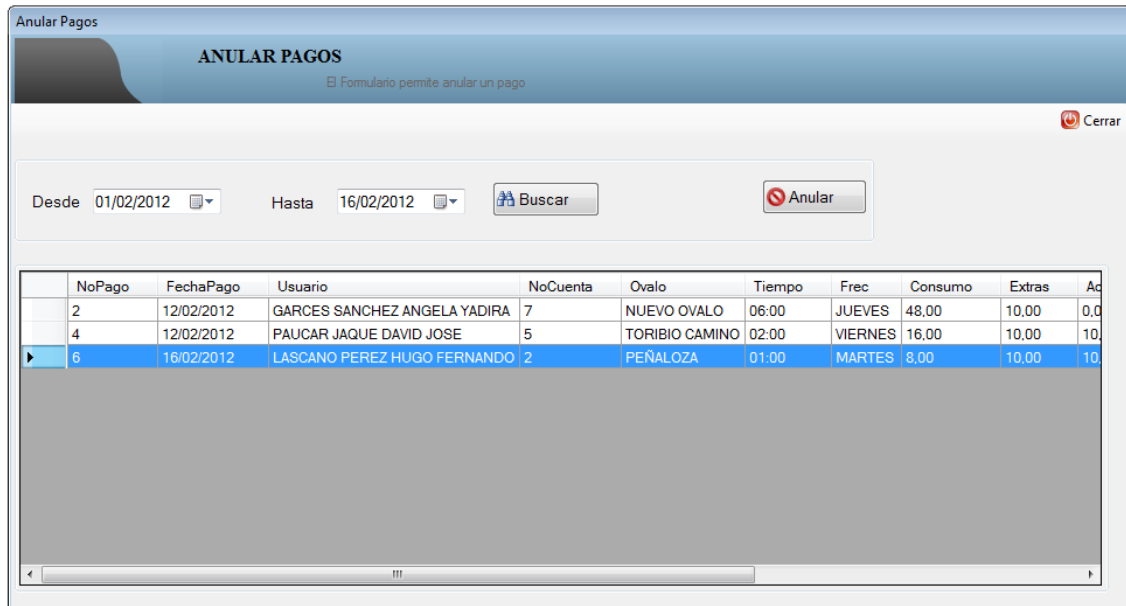


Figura 6.51 Pantalla anular factura

Pantalla de anular factura

Para poder anular una factura se debe seleccionar la factura presionar el botón **Anular**.

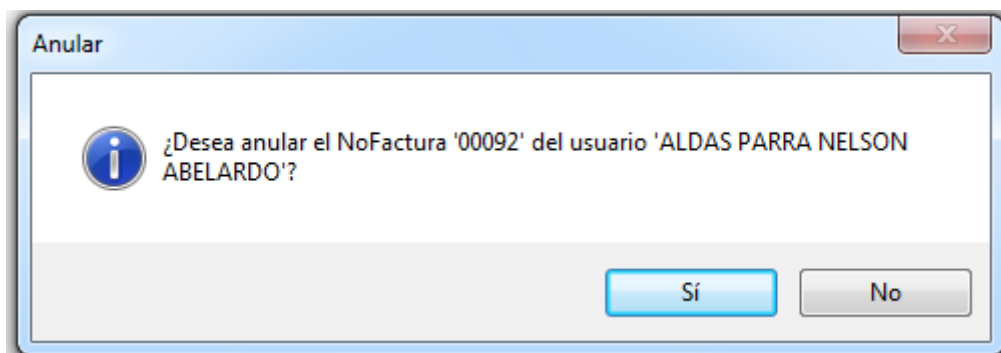


Figura 6.52 Pantalla anular

Pantalla Informes

La Secretario del sistema podrá visualizar los informes sin ninguna restricción



Figura 6.53 Pantalla opción informes

Pantalla Informes Facturas Emitidas

Esta pantalla permite visualizar las facturas emitidas.

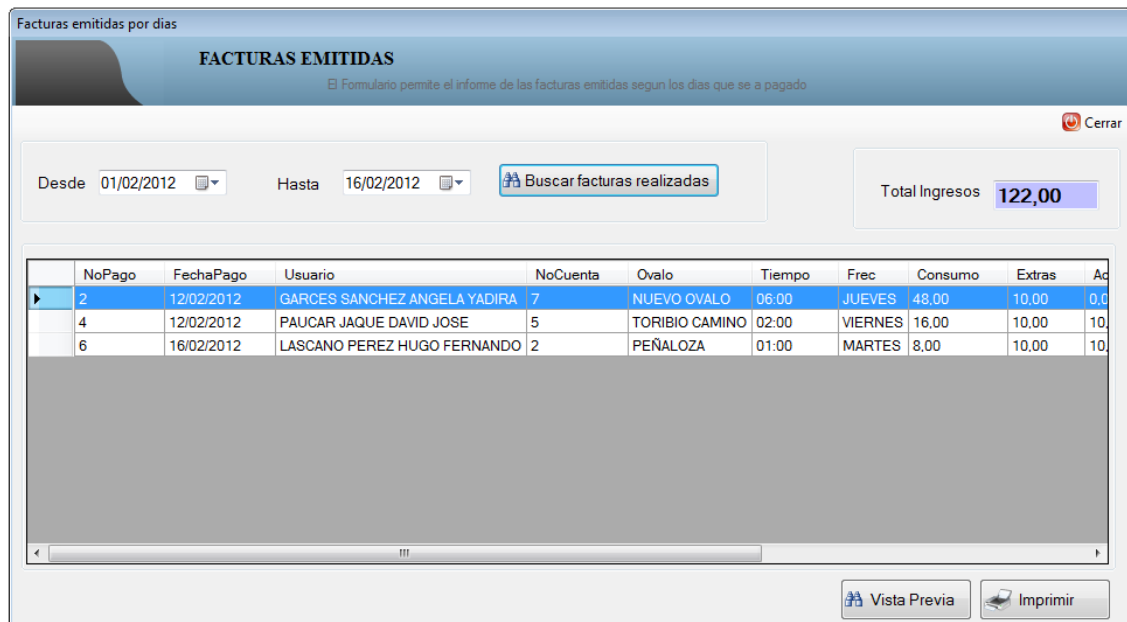


Figura 6.54 Pantalla Informes Facturas Emitidas

Pantalla visualizar informe facturas

Para visualizar e imprimir un informe se debe presionar el botón Vista Previa.

Reportes Pagos Realizados

Informe principal

Lista de cobros realizados segun fechas

Fecha de Impresión: 16/02/2012

NoPa	FechaPa	Usuario	NoCu	Ovalo	Tiempo	Cons	Act	Ext	Mor	SubTotal	IVA	Total	Abono	Estado
2	12/02/2012	GARCÉS SANCHEZ ANGELA YADIRA	7	NUEVO OVALO	06:00	48,00	0,00	10,00		58,00	0,00	58,00	58,00	PAGADO
4	12/02/2012	PAUCAR JAQUE DAVID JOSE	5	TORIBIO CAMINO	02:00	16,00	10,00	10,00		36,00	0,00	36,00	36,00	PAGADO
6	16/02/2012	LASCANO PEREZ HUGO FERNANDO	2	PEÑALOZA	01:00	8,00	10,00	10,00		28,00	0,00	28,00	28,00	PAGADO

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Vista Previa Imprimir

Figura 6.55 Pantalla visualizar informe factura

Pantalla de Informes Económicos

Esta pantalla contiene la información de informes económicos.

Informes Económicos

INFORMES ECONÓMICOS

El Formulario permite el informe económicos de la junta

Desde 01/02/2012 Hasta 16/02/2012 Buscar

Total Ingresos **122,00**

Total Egresos **60,00**

TOTAL EN CAJA **62,00**

Fecha	Monto	Motivo	Periodo
07/02/2012	13,00	VIATICOS ESTUDIANTES DE LA UTA	2012
08/02/2012	5,00	OTROS	2012
09/02/2012	9,00	JAJAJ	2012
12/02/2012	10,00	ESTUDIANTE UTA	2012
12/02/2012	23,00	PAGO A TRABAJADORES	2012

Vista Previa Imprimir

Figura 6.56 Pantalla Informes Económicos

Pantalla visualizar informes económicos

Para visualizar e imprimir el informe se debe presionar el botón Vista Previa.



Figura 6.57 Pantalla visualizar informes económicos

Configuraciones

Pantalla Configurar IVA

Esta pantalla permite configurar el porcentaje del IVA.

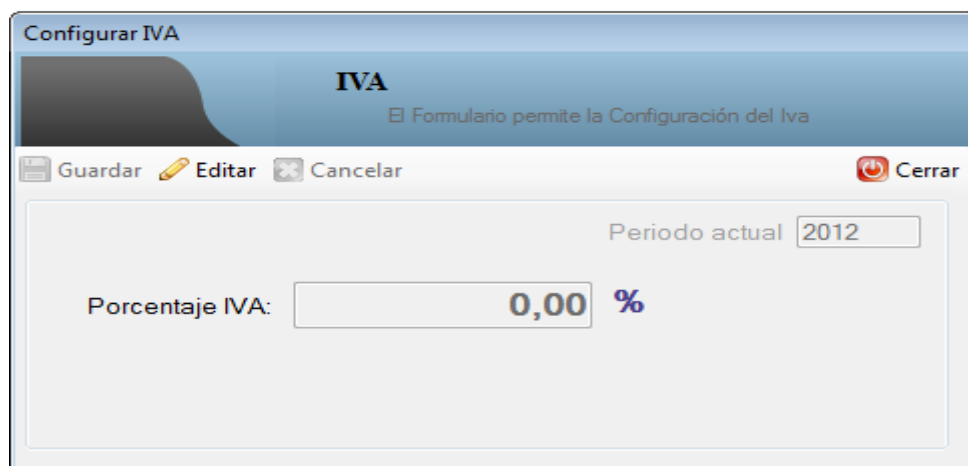


Figura 6.58 Pantalla configurar IVA.