



## **UNIVERSIDAD TÉCNICA DE AMBATO**

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

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E  
INFORMÁTICOS**

**Tema:**

---

“SISTEMA WEB INTEGRADO PARA LA GESTIÓN DE COBRANZA DE  
VALORES EN EL GOBIERNO PROVINCIAL DE TUNGURAHUA.”

---

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

Autor: Alberto Eduardo Pérez Bautista.

Tutor: Ing. Eduardo Chaso.

Ambato - Ecuador

Julio, 2012

## **APROBACIÓN DEL TUTOR**

En mi calidad de tutor del trabajo de investigación sobre el tema: “SISTEMA WEB INTEGRADO PARA LA GESTIÓN DE COBRANZA DE VALORES EN EL GOBIERNO PROVINCIAL DE TUNGURAHUA.”, del señor ALBERTO EDUARDO PÉREZ BAUTISTA, estudiante 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 de Tercer Nivel de la Universidad Técnica de Ambato.

Ambato julio, 2012

EL TUTOR

-----  
Ing. Eduardo Chaso.

## AUTORÍA

El presente trabajo de investigación titulado: “SISTEMA WEB INTEGRADO PARA LA GESTIÓN DE COBRANZA DE VALORES EN EL GOBIERNO PROVINCIAL DE TUNGURAHUA.”. 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

---

Alberto Eduardo Pérez Bautista.

CC: 1722249875

## APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Galo López y Ing. David Guevara, revisó y aprobó el Informe Final del trabajo de graduación titulado “Sistema web integrado para la gestión de cobranza de valores en el Gobierno Provincial de Tungurahua”, presentado por el señor Alberto Eduardo Pérez Bautista de acuerdo al Art. 17 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. David Guevara  
DOCENTE CALIFICADOR

Ing. Galo López  
DOCENTE CALIFICADOR

## **DEDICATORIA:**

Quiero dedicarle este trabajo a mi madre que siempre ha estado a mi lado brindándome su amor, dándome a cada instante palabras de aliento para poder cumplir con mis sueños y objetivos, A mi padre por ser un ejemplo a seguir.

A mis hermanas, tíos, primos, abuelos y amigos. Gracias por haber fomentado en mí el deseo de superación y el anhelo de triunfo en la vida.

A mi novia Silvana por confiar en mis decisiones, darme su apoyo, ánimo, comprensión y sobre todo por estar junto a mí en todo momento.

Alberto Eduardo Pérez Bautista.

## **AGRADECIMIENTO:**

A Dios, por protegerme y cuidar de mi familia, por darme la fuerza necesaria para enfrentarme a los retos que se me presentaron en mi vida estudiantil y así poder salir adelante.

A mis padres por su apoyo económico y moral a lo largo de mi carrera universitaria. A toda familia por estar a mi lado cuando necesite de su ayuda.

Mis maestros por compartir sus conocimientos, experiencias y forjar en mí el espíritu de triunfo.

A mi tutor, Ing. Eduardo Chaso por el apoyo y colaboración en el desarrollo de mi trabajo.

Alberto Eduardo Pérez Bautista.

## ÍNDICE

<b>CONTENIDO</b>	<b>PÁGINA</b>
Aprobación del tutor.....	i
Autoría.....	ii
Aprobación de la comisión calificadora.....	iii
Dedicatoria.....	iv
Agradecimiento.....	v
Índice.....	vi

## CAPÍTULO I

### EL PROBLEMA DE INVESTIGACIÓN

1.1 Tema de investigación.....	1
1.2 Planteamiento del problema.....	1
1.2.1 Contextualización.....	1
1.2.2 Análisis crítico.....	2
1.2.3 Prognosis.....	2
1.2.4 Formulación del problema.....	2
1.2.5 Preguntas directrices.....	2
1.2.6 Delimitación del problema.....	2
1.3 Justificación.....	3
1.4 Objetivos.....	3
1.4.1 Objetivo general.....	3
1.4.2 Objetivos específicos.....	3

## CAPÍTULO II

### MARCO TEÓRICO

2.1 Antecedentes investigativos.....	4
2.2 Fundamentación Legal.....	5
2.3 Categorías fundamentales.....	11
2.3.1 Seguridad Web.....	12

2.3.2 Software Libre.....	23
2.3.3 Sitios Web.....	24
2.3.4 Sistema Web Itegrado.....	26
2.3.5 Reportes.....	28
2.3.6 SQL (lenguaje de consulta estructurado).....	29
2.3.7 Base de Datos.....	31
2.3.8 Gestión de Cobranza de Valores.....	35
2.4 Hipótesis.....	36
2.5 Variables.....	36
2.5.1 Variable Independiente.- Sistema de web integrado.....	36
2.5.2 Variable Dependiente.- Gestión de cobranza de valores.....	36

### **CAPÍTULO III**

#### **METODOLOGÍA**

3.1 Enfoque.....	37
3.2 Modalidad básica de la investigación.....	37
3.3 Niveles o Tipos de investigación.....	38
3.4 Población y muestra.....	38
3.5 Operacionalización de variables.....	39
3.5.1 Variable independiente: Sistema Web Integrado.....	39
3.5.2 Variable dependiente: Gestión de cobranza de valores.....	41
3.6 Recolección de información.....	43
3.7 Procesamiento y análisis.....	43

### **CAPÍTULO IV**

#### **ANÁLISIS E INTERPRETACIÓN DE RESULTADOS**

4.1 Análisis e interpretación de los resultados.....	44
4.2 Análisis de los resultados de las encuestas.....	44



## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

5.1 Conclusiones.....	51
5.2 Recomendaciones.....	52

## **CAPITULO VI**

### **PROPUESTA**

6.1 Tema.....	53
6.2 Datos informativos.....	53
6.3 Antecedentes.....	53
6.4 Justificación.....	54
6.5 Objetivos.....	54
6.5.1 Objetivo general.....	54
6.5.2 Objetivos específicos.....	55
6.6 Análisis de factibilidad.....	55
6.6.1 Factibilidad operativa.....	55
6.6.2 Factibilidad económica.....	56
6.6.3 Factibilidad técnica.....	56
6.7 Fundamentación.....	56
6.7.1 PHP (Hypertext Pre-Processor).....	56
6.7.2 JavaScript.....	58
6.7.3 EXTJs.....	59
6.8 Metodología.....	61
6.9 Modelo operativo.....	62
6.9.1 Análisis del sistema.....	62
6.9.1.1 Análisis y requerimientos del sistema.....	62
6.9.1.2 Diagramas UML.....	63
6.9.1.2.1 Diagrama de casos de uso.....	63
6.9.1.2.2 Diagrama de estados.....	74
6.9.1.2.3 Diagrama de secuencias.....	78

6.9.1.2.4 Diagrama de despliegue.....	84
6.9.2 Diseño del sistema.....	84
6.9.2.1 Diseño de la base de datos.....	84
6.9.2.2 Diccionario de datos.....	86
6.9.2.3 Diseño de la interfaz.....	89
6.9.2.3.1 Diseño de salidas.....	90
6.9.2.3.2 Diseño de entradas.....	93
6.9.3 Implementación.....	95
6.9.3.1 Extracto de código fuente.....	95
6.9.4 Pruebas.....	117
6.9.4.1 Prueba Alfa.....	118
6.9.4.2 Prueba Beta.....	119
6.9.4.3 Pruebas de caja negra.....	120
6.9.4.4 Pruebas de particiones o clases de equivalencia.....	123
6.9.4.5 Pruebas de caja blanca.....	124
6.9.4.6 Pruebas del camino básico.....	124
6.9.4.7 Pruebas de Verificación y Validación.....	128
6.9.4.7.1 Verificación.....	128
6.9.4.7.2 Validación.....	128
6.9.5 Implantación.....	129
6.9.5.1 Implantación de carpetas y archivos necesarios.....	129
6.9.5.2 Instalación del motor de base de datos.....	130
6.9.5.3 Instalación del sistema de gestión de cobranza.....	134
6.9.5.4 Aplicación de pruebas en el sistema.....	135
6.9.5.5 Capacitación al personal sobre el uso y administración del sistema.....	136
6.9.5.6 Evaluaciones.....	136
6.10 Conclusiones y Recomendaciones.....	137
6.10.1 Conclusiones.....	137
6.10.2 Recomendaciones.....	138
Bibliografía.....	139
Información bibliográfica de libros.....	139
Información bibliográfica de páginas web.....	139

Glosario de términos.....	140
---------------------------	-----

## **ANEXOS**

<b>ANEXO 1:</b> Estructura del Cuestionario.....	143
<b>ANEXO 2:</b> Decreto presidencial para el uso de software libre en la república del Ecuador.....	145
<b>ANEXO 3:</b> Manual de usuario.....	147

## ÍNDICE DE FIGURAS

Figura 2.1 estructura del modelo en cascada.....	21
Figura 2.2 Niveles SSL .....	22
Figura 4.1 Pregunta 1.....	44
Figura 4.2 Pregunta 2.....	45
Figura 4.3 Pregunta 3.....	46
Figura 4.4 Pregunta 4.....	47
Figura 4.5 Pregunta 5.....	47
Figura 4.6 Pregunta 6.....	48
Figura 4.7 Pregunta 7.....	49
Figura 4.8 Pregunta 8.....	50
Figura 6.1 estructura del modelo en cascada.....	61
Figura 6.2. Diagrama de casos de uso Sistema de Gestión de Cobranza.....	64
Figura 6.3 Diagrama de estados tipo facturación.....	75
Figura 6.4 Diagrama de estados tonelaje.....	75
Figura 6.5. Diagrama de estados usuarios.....	75
Figura 6.6. Diagrama de estado cargos .....	76
Figura 6.7. Diagrama de estado departamento .....	76
Figura 6.8. Diagrama de estado contribuyente .....	77
Figura 6.9. Diagrama de estado vehículo .....	77
Figura 6.10. Diagrama de estado comprobante.....	77
Figura 6.11. Diagrama de secuencia ingreso al sistema.....	78
Figura 6.12. Diagrama de secuencia administración usuarios.....	79
Figura 6.13. Diagrama de secuencia administración de cargos.....	79
Figura 6.14. Diagrama de secuencia administración de departamentos.....	80
Figura 6.15. Diagrama de secuencia administración de contribuyentes.....	80
Figura 6.16. Diagrama de secuencia administración de vehículos.....	81
Figura 6.17. Diagrama de secuencia administración de comprobantes.....	81
Figura 6.18. Diagrama de secuencia Boletín Comprobante.....	82
Figura 6.19. Diagrama de secuencia salir del sistema.....	82
Figura 6.20. Diagrama de secuencia Tonelaje .....	83
Figura 6.21. Diagrama de secuencia salir del sistema.....	83

Figura 6.22. Diagrama de despliegue .....	84
Figura 6.23. Modelo relacional lógico.....	85
Figura 6.24. Pantalla inicio de sesión.....	90
Figura 6.25. Ventana inicial.....	91
Figura 6.26. Menú mantenimiento.....	92
Figura 6.27. Ventana Reportes.....	93
Figura 6.28. Manipulación de la información.....	94
Figura 6.29. Grafo de validación de la cédula.....	126
Figura 6.30. Control de eliminación de registros.....	128
Figura 6.31. Validación de la cedula del usuario .....	129
Figura 6.32. Validación ingreso sistema .....	129
Figura 6.33. Instalar Postgres.....	130
Figura 6.34. Directorio de instalación.....	131
Figura 6.35. Directorio de datos.....	131
Figura 6.36. Contraseña del usuario postgres .....	132
Figura 6.37. Puerto postgres.....	132
Figura 6.38. Configuración del idioma.....	133
Figura 6.39. Listo para instalar.....	133
Figura 6.40. Instalado .....	134
Figura A3.1 Pantalla de inicio de sesión.....	147
Figura A3.2 Menú administrador.....	148
Figura A3.3 Cambio de contraseña.....	148
Figura A3.4 Datos de departamentos.....	149
Figura A3.5 Mantenimiento Cargos.....	149
Figura A3.6 Forma mantenimiento usuario.....	150
Figura A3.7 Mantenimiento contribuyentes.....	151
Figura A3.8 Mantenimiento vehículo.....	151
Figura A3.9 Mantenimiento comprobantes.....	152
Figura A3.10 Reporte cierre diario.....	152
Figura A3.11 Reporte cierre fechas.....	153
Figura A3.12 Cerrar sesión.....	153

## ÍNDICE DE TABLAS

Tabla 3.1. Operacionalización de variable independiente.....	40
Tabla 3.2. Operacionalización de variable dependiente .....	42
Tabla 4.1. Tabla porcentual pregunta 1.....	44
Tabla 4.2. Tabla porcentual pregunta 2.....	45
Tabla 4.3. Tabla porcentual pregunta 3.....	46
Tabla 4.4. Tabla porcentual pregunta 4.....	46
Tabla 4.5. Tabla porcentual pregunta 5.....	47
Tabla 4.6. Tabla porcentual pregunta 6.....	48
Tabla 4.7. Tabla porcentual pregunta 7.....	49
Tabla 4.8. Tabla porcentual pregunta 8.....	49
Tabla 4.7. Tabla porcentual pregunta 7.....	49
Tabla 6.1. Especificación ingresar al sistema.....	65
Tabla 6.2 Especificación administración usuarios.....	66
Tabla 6.3 Especificación administración de cargos .....	67
Tabla 6.4 Especificación administración de departamentos.....	68
Tabla 6.5 Especificación Boletín Comprobante.....	69
Tabla 6.6 Especificación Modo Facturación.....	70
Tabla 6.7 Especificación Tonelaje.....	71
Tabla 6.8 Especificación Administración de Contribuyentes.....	72
Tabla 6.9 Especificación administración de vehículos.....	73
Tabla 6.10 Especificación administración de comprobantes.....	74
Tabla 6.11 Especificación Salir del Sistema.....	74
Tabla 6.12. Descripción de la tabla departamento.....	86
Tabla 6.13. Descripción de la tabla cargo.....	86
Tabla 6.14. Descripción de la tabla usuario.....	87
Tabla 6.15. Descripción de la tabla contribuyente.....	87
Tabla 6.16. Descripción de la tabla vehículo.....	87
Tabla 6.17. Descripción de la tabla comprobante.....	88
Tabla 6.18. Descripción de la tabla usuariosingresos.....	88
Tabla 6.19. Descripción de la tabla parametrostonelaje.....	88
Tabla 6.20. Descripción de la tabla parametromodofactura.....	89

Tabla 6.21. Descripción de la tabla boletincomprobantes.....	89
Tabla 6.22. Plan de Pruebas Alfa – Login.....	118
Tabla 6.23. Plan de Pruebas Alfa – InterfazPrincipal .....	118
Tabla 6.24. Plan de Pruebas Beta.....	119
Tabla 6.25. Tablas de clases de equivalencia tabla Departameto.....	124

## **RESUMEN EJECUTIVO**

En el Gobierno Provincial de Tungurahua, el departamento financiero tiene a su cargo la recaudación de valores económicos por concepto del rodaje vehicular en la provincia, con el fin de generar recursos para el mantenimiento de las vías intercantonales e interparroquiales, y otorgar el comprobante de pago que sirve a la ciudadanía como requisito para la matriculación vehicular en la provincia de Tungurahua.

Este tema de investigación surge como respuesta a la necesidad del Gobierno Provincial de Tungurahua de contar con un Sistema Informático que facilite y optimice el proceso de recaudación y almacenamiento de información.

Es importante el desarrollo del Sistema de Gestión de Cobranza ya que a través de él se podrá controlar de mejor manera los valores recaudados y comprobantes emitidos, la recuperación de la información será rápida y eficiente, eliminando la duplicidad de datos, existirá seguridad en la información almacenada, se realizará un control y seguimiento de los pagos que cada vehículo matriculado en la provincia tenga que obligatoriamente realizar a partir de su registro en la provincia, y por ende se conseguirá facilitar el trabajo a los recaudadores y mejorar el servicio a la ciudadanía.

Por la razón expuesta anteriormente a continuación se muestra el proceso de investigación que permitió el desarrollo de la solución informática para el Gobierno Provincial de Tungurahua.



## INTRODUCCIÓN

El propósito de la presente investigación es elaborar un sistema informático que satisfaga las necesidades del departamento financiero, área de cobranza del Gobierno Provincial de Tungurahua, con la tecnología y herramientas de desarrollo más adecuadas.

La investigación se ha elaborado de acuerdo a la organización de información que se detalla a continuación:

**En el capítulo I “EL PROBLEMA DE INVESTIGACIÓN”**, se identifica el problema a investigar, además se plantea la justificación y los objetivos.

**En el capítulo II “MARCO TEÓRICO”**, se presentan los antecedentes investigativos, la fundamentación legal, hipótesis y el señalamiento de las variables de la hipótesis.

**En el capítulo III “METODOLOGÍA”**, se determina la metodología de investigación a utilizar, el enfoque, la modalidad básica de la investigación, el tipo de investigación, la población y muestra.

**En el capítulo IV “ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS”**, se procede al análisis e interpretación de los resultados.

**En el capítulo V “CONCLUSIONES Y RECOMENDACIONES”**, se presenta las conclusiones y recomendaciones del trabajo desarrollado.

**En el capítulo VI “PROPUESTA”**, se presenta el desarrollo de la propuesta ante el problema investigado.

## **CAPITULO I**

### **EL PROBLEMA DE LA INVESTIGACION**

#### **1.1 Tema de la investigación**

Sistema web integrado para la gestión de cobranza de valores en el Gobierno Provincial de Tungurahua.

#### **1.2 Planteamiento del problema**

##### **1.2.1 Contextualización**

La mayoría de instituciones públicas del país, hace uso de algún sistema informático para automatizar procesos, pero también existen departamentos o áreas en las que el uso de hojas electrónicas, editores de texto, son la principal herramienta para administrar información, lo que en muchos casos da como consecuencia, pérdida de información, inconsistencia de datos, etc., afectando el desempeño laboral y generando ineficiencia en la atención al público.

En las instituciones públicas de la provincia de Tungurahua, existen departamentos o áreas, en las que se hace uso de hojas electrónicas, para el manejo de información importante, que debería ser administrada de mejor manera, para evitar los problemas que pudieran suscitarse.

En las instalaciones del Gobierno Provincial de la Provincia de Tungurahua, se cuenta con varios sistemas informáticos, que automatizan procesos, mejorando y garantizando el correcto funcionamiento de determinada área; sin embargo existen departamentos como el financiero, que para almacenar y administrar información de cobros, hace uso de hojas electrónicas, editores de texto, etc.

### **1.2.2 Análisis crítico**

La administración y almacenamiento en hojas electrónicas, de información de la gestión de cobranza de valores de la contribución para el mantenimiento de las vías intercantonales e interparroquiales de la provincia de Tungurahua, puede dar como consecuencia, pérdida de información de mucha importancia, inconsistencia en datos, ineficiencia en la atención al público, etc.

Para solucionar estos problemas, se debe desarrollar un sistema informático web, que almacene la información en una base de datos, muestre información confiable, brinde reportes, con esto se evitará la pérdida e inconsistencia de datos y se brindará una mejor atención al público.

### **1.2.3 Prognosis**

De continuar con este problema en el Gobierno Provincial de la provincia de Tungurahua, seguirá existiendo una inadecuada administración de la información, pérdida e inconsistencia de datos.

### **1.2.4 Formulación del problema**

¿Qué incidencias tiene el desarrollo de un sistema web integrado para la gestión de cobranza de valores, en el mantenimiento de las vías intercantonales e interparroquiales de la provincia de Tungurahua?

### **1.2.5 Preguntas Directrices**

¿Qué es un sistema web?

¿Qué es software libre?

¿Con que equipos cuenta la institución para alojar nuestro sistema web?

¿Cuáles son los requerimientos necesarios para implementar el sistema web?

### **1.2.6 Delimitación del problema**

El presente trabajo se desarrollará en las oficinas del Gobierno Provincial de la Provincia de Tungurahua, ubicado en el centro de la ciudad de Ambato en las

calles Av. Bolívar y Castillo, el tiempo estimado para el desarrollo del proyecto es de seis meses.

### **1.3 Justificación**

El estudio para el diseño del Sistema web integrado para la gestión de cobranza de valores de la contribución para el mantenimiento de las vías intercantonales e interparroquiales de la provincia de Tungurahua, se lo hace principalmente para tener un mayor control de este proceso, brindar mejor servicio al público.

La necesidad también de tener a nuestro alcance y de manera ordenada toda la información que el usuario requiera, sin que este proceso no signifique para las personas una actividad tediosa, con pérdida de tiempo.

Por otro lado, aplicar los conocimientos adquiridos en la carrera universitaria, permitirá diseñar e implementar un sistema web eficaz y eficiente, utilizando creatividad, dinamismo, entusiasmo, con un mejor desempeño en el desarrollo.

### **1.4 Objetivos**

#### **1.4.1 Objetivo General**

- Desarrollar e implantar un Sistema web integrado para la gestión de cobranza de valores.

#### **1.4.2 Objetivos Específicos**

- Analizar la información proporcionada por los responsables del proceso actual de gestión de cobranza de valores, de la contribución para el mantenimiento de las vías intercantonales e interparroquiales de la provincia de Tungurahua.
- Definir las mejores herramientas de software libre para el desarrollo del sistema web integrado propuesto.
- Estudiar las mejores técnicas de seguridad web, para su aplicación en el desarrollo de la aplicación.
- Integrar el sistema al portal web del Gobierno Provincial de Tungurahua, realizando las pruebas necesarias para su correcto funcionamiento.

## **CAPITULO II**

### **MARCO TEÓRICO**

#### **2.1 Antecedentes Investigativos**

En la Universidad Técnica de Ambato, se encontraron temas similares de desarrollo de un sistema web para gestión de cobros, como el actual proyecto, pero en instituciones diferentes.

En el Gobierno Provincial de la provincia de Tungurahua, se ha encontrado que se han desarrollado varios proyectos de investigación, en diferentes áreas; con respecto a el departamento financiero, también se han desarrollado aplicaciones para la automatización de procesos, pero ninguno que se ejecute en el internet.

En el departamento financiero del Gobierno Provincial de Tungurahua, en el área encargada de realizar los cobros de valores, de la contribución para el mantenimiento de las vías intercantonales e interparroquiales de la provincia, se realiza un control manual de la información de los pagos, se almacena en un archivo de Excel y se realizan otras tareas a partir de esta.

El presente trabajo se realizará en beneficio del departamento financiero del Gobierno Provincial de Tungurahua a través, del convenio entre El Gobierno Provincial de Tungurahua y la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

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

## **2.2 Fundamentación Legal**

### **CAPITULO I**

#### **Constitución y fines del Consejo Provincial**

**Art. 1.-** El Consejo Provincial es institución de derecho público, goza de autonomía y representa a la provincia.

Tiene personería jurídica, con capacidad para realizar los actos que fueren necesarios para el cumplimiento de sus fines, en la forma y condiciones que determinan la Constitución y las leyes.

Tiene personería jurídica y fundamentalmente, su misión es impulsar el desarrollo social, cultural y material de la provincia, con especial atención al sector rural, y colaborar con el Estado y las municipalidades en la respectiva circunscripción, para la realización armónica de los fines nacionales.

**Art. 2.-** El consejo provincial estará integrado por el prefecto o prefecta quien lo presidirá con voto dirimente, el viceprefecto o viceprefecta; por alcaldes o alcaldesas o concejales o concejalas en representación de los cantones; y, por representantes elegidos de entre quienes presidan las juntas parroquiales rurales, que se designarán observando las disposiciones previstas en esta ley.

**Nota:** Artículo sustituido por Ley No. 0, publicada en Registro Oficial Suplemento 36 de 29 de Septiembre del 2009.

**Art. 3.-** Cada cantón tendrá un representante en el consejo provincial, que será el alcalde o alcaldesa. En caso de que no pudiere asistir, ejercerá como su delegado o delegada ante el consejo provincial y con pleno poder de decisión, el concejal o concejala principal que el alcalde o alcaldesa designe, que será de carácter estable, respetando los principios de paridad de género e interculturalidad, donde fuere posible.

**Nota:** El artículo 7 de la Ley Orgánica de Elecciones elimina la Elección Indirecta y rige sobre el artículo 3 por el Principio de Jerarquía Normativa.

**Nota:** Artículo sustituido por Ley No. 0, publicada en Registro Oficial Suplemento 36 de 29 de Septiembre del 2009.

**Art. 4.-** La representación de los presidentes de las juntas parroquiales rurales, se integrará conforme las siguientes reglas:

a) En las provincias que tengan hasta 100.000 habitantes del área rural, el consejo provincial contará con tres presidentes o presidentas de las juntas parroquiales rurales;

b) En las provincias que tengan de 100.001 hasta 200.000 habitantes del área rural, el consejo provincial contará con cinco presidentes o presidentas de las juntas parroquiales rurales;

c) En las provincias que tengan más de 200.001 habitantes del sector rural, el consejo provincial contará con siete presidentes o presidentas de las juntas parroquiales rurales.

Para garantizar la alternabilidad, los representantes de los gobiernos autónomos descentralizados parroquiales rurales ejercerán su representación en el consejo provincial por medio período para el que fueron electos el prefecto o prefecta.

El Consejo Nacional Electoral, establecerá el número de representantes a ser elegidos por cada provincia, utilizando las proyecciones del Censo Nacional de Población, a la fecha de la convocatoria a la elección de estos representantes."

**Nota:** El artículo 7 de la Ley Orgánica de Elecciones elimina la Elección Indirecta y rige sobre el artículo 3 por el Principio de Jerarquía Normativa.

**Nota:** Artículo sustituido por Ley No. 0, publicada en Registro Oficial Suplemento 36 de 29 de Septiembre del 2009.

**Art. 5.-** Los alcaldes o alcaldesas, concejales o concejales, y los presidentes o presidentas de juntas parroquiales que en representación de sus cantones o parroquias integren el consejo provincial se denominarán consejeros provinciales.

**Nota:** Artículo sustituido por Ley No. 0, publicada en Registro Oficial Suplemento 36 de 29 de Septiembre del 2009.

**Art. 6.-** El Consejo Nacional Electoral, en un plazo máximo de diez días a partir de la posesión de los integrantes de las juntas parroquiales rurales, convocará a un colegio electoral conformado por los presidentes o presidentas de las juntas parroquiales rurales de cada provincia, para elegir de entre ellos y ellas a sus representantes principales y alternos al consejo provincial, en elección indirecta. Este procedimiento se realizará en la mitad del período para el que fue electo el o la prefecta. La provincia de Galápagos queda exceptuada de este procedimiento.

Los presidentes o presidentas de la juntas parroquiales rurales que integren cada consejo provincial deberán provenir de donde sea pertinente, de diferentes cantones, procurando la mayor representación territorial; y, en ningún caso un mismo presidente o presidenta podrá integrar el consejo por dos ocasiones consecutivas, con excepción de las provincias en donde, por el número de parroquias, no sea posible la alternabilidad. Para la elección deberá respetarse los principios de interculturalidad y paridad de género, en donde sea posible.

**Nota:** Artículo sustituido por Ley No. 0, publicada en Registro Oficial Suplemento 36 de 29 de Septiembre del 2009.

**Art. 7.-** Corresponde a los Consejos Provinciales:

- a) Propender al progreso de la provincia, en orden a robustecer el sentimiento de nacionalidad;
- b) Prestar servicios públicos de interés provincial directamente o en colaboración con los organismos del Estado o de las municipalidades;
- c) Realizar obras públicas de carácter provincial e interprovincial;
- d) Coordinar la acción de las municipalidades de la provincia, para fines de progreso común;
- e) Procurar el cabal cumplimiento y ejercicio de los servicios y de las obras públicas que se realicen en la provincia;
- f) Orientar las aspiraciones provinciales relacionadas con el desenvolvimiento económico, promoviendo la explotación y fomento de las fuentes de producción



agrícola, pecuaria, industrial y minera, para lo cual acordará los planes correspondientes, encuadrándolos dentro del Plan General de Desarrollo;

g) Fomentar la educación mediante la creación de escuelas, institutos técnicos, estaciones experimentales y de otros centros similares, de acuerdo con la ley;

g-1) Planificar, ejecutar, coordinar y evaluar, en el sector rural, programas sociales para la atención a niños de la calle, jóvenes, nutrición infantil, mujeres embarazadas, personas, con discapacidad, de la tercera edad, prevención y atención a la violencia doméstica. Para efectos de la ampliación y eficiencia de estos programas, las correspondientes entidades dependientes de la Función Ejecutiva encargadas de ejecutar programas y prestar servicios similares, a petición de los consejos provinciales obligatoriamente les transferirán sus funciones, atribuciones, responsabilidades y recursos, especialmente financieros internos y externos de conformidad con lo dispuesto en el Art. 18 de la Ley Especial de Descentralización del Estado y de Participación Social;

h) Atender y vigilar el estado sanitario de la provincia y propender a su mejoramiento, a través de una acción conjunta con los organismos estatales, con los concejos municipales, y con las juntas parroquiales de su jurisdicción;

i) Fomentar el turismo;

j) Vigilar que las rentas asignadas para las obras públicas provinciales se inviertan oportuna y correctamente;

k) Los consejos provinciales efectuarán su planificación siguiendo los principios de conservación, desarrollo y aprovechamiento sustentable de los recursos naturales; y,

l) Cumplir con los demás fines que les señalen la Constitución y las leyes.

**Nota:** (sic) Artículo reformado por Ley No. 000, publicada en Registro Oficial 116 de 2 de Julio del 2003.

## **CONCORDANCIAS:**

- LEY ORGANICA DE REGIMEN MUNICIPAL, CODIFICACION, Arts. **23, 24**

**Art. 8.-** Los litigios que se presentaren por la fijación de límites interprovinciales o intercantonales, serán conocidos por una comisión integrada por los siguientes miembros:

- a) Por el Presidente de la Corte Suprema de Justicia, quien la presidirá;
- b) Por el Decano de la Facultad de Jurisprudencia de la Universidad Central del Ecuador;
- c) Por el Director de la Sección de Historia y Geografía de la Casa de la Cultura Ecuatoriana;
- d) Por el Director del Instituto Geográfico Militar; y,
- e) Por un representante del Ministerio de Gobierno.

Las partes interesadas en el litigio presentarán sus reclamos en derecho ante el Presidente de la Corte Suprema de Justicia, quien notificará a los demás miembros de la Comisión, con el objeto de constituirla.

Constituida la Comisión, ésta y las partes actuarán de conformidad con el reglamento que, para el efecto, dictará el Ministerio de Gobierno.

En mérito de lo actuado, la Comisión elaborará un informe, que será sometido a la aprobación del Congreso Nacional que esté en funciones o próximo a reunirse, siempre que ante la Comisión no se hubiere dado solución al problema.

Cuando alguno de los miembros de la Comisión sea nativo de una de las provincias o cantones en litigio, será subrogado por quien corresponda.

## **Presidencia de la república**

### **Decreto uso de software libre**

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

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

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

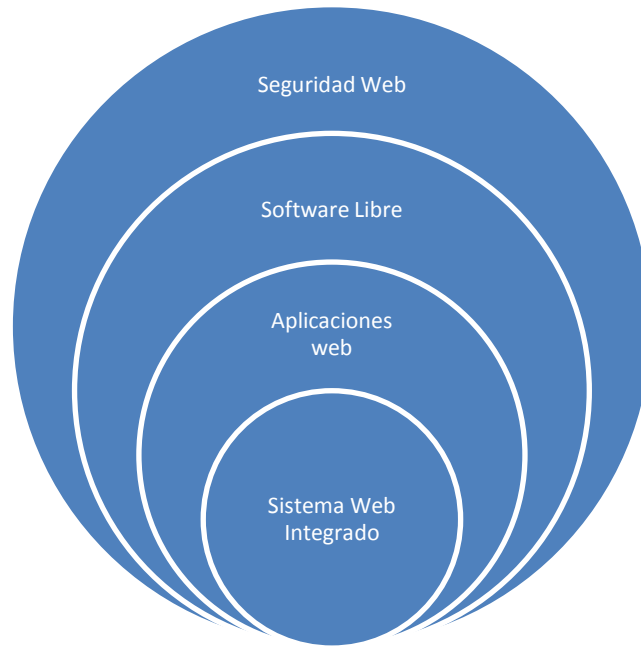
Estos programas de computación tienen las siguientes libertades:

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

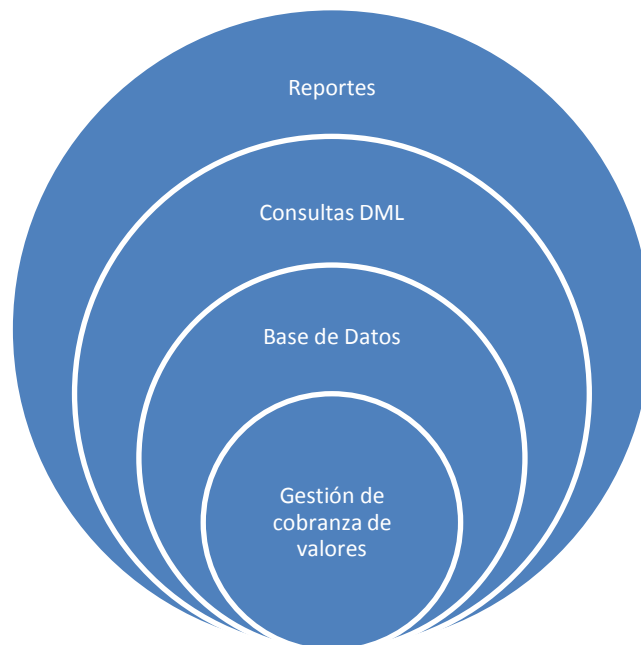
**Art. 3:** Las entidades de la administración pública central previa a la instalación del software libre en sus equipos, deberán verificar la existencia de capacidad técnica que brinde el soporte necesario para este tipo de software.

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

## 2.3 Categorías fundamentales



**VI**



**VD**

### 2.3.1 Seguridad Web

La seguridad, en informática como en otras áreas, se basa en la protección de activos. Estos activos pueden ser elementos tan tangibles como un servidor o una base de datos, o pueden ser la reputación de una empresa.

Generalmente podemos evaluar la seguridad de un activo en base a tres aspectos principales que no necesitan explicación: integridad, disponibilidad, confidencialidad.

Estos tres aspectos a su vez dependen de otros tres elementos principales que engloban prácticamente todos los distintos controles que se pueden establecer en un sistema informático:

**Autenticación:** los clientes de nuestras aplicaciones o servicios deben ser identificados de forma única, sean usuarios finales, otros servicios o computadoras externas.

**Autorización:** no solo es necesario saber quiénes acceden a nuestros activos, también es necesario establecer que es lo que pueden hacer con ellos. Un nivel de autorización dado determina qué tipo de operaciones o transacciones puede efectuar un cliente dado sobre un recurso dado.

**Registro y Auditoría:** luego de efectuada una operación, es importante que esta sea registrada adecuadamente, en particular es esencial si queremos evitar el repudio de transacciones efectuada por un cliente.

Si bien cuando se habla de la seguridad de aplicaciones web se deben considerar no sólo las amenazas externas a la institución sino también las internas (administradores malintencionados, usuarios que provocan accidentes, etc.).

*Seguridad Web. (Sebastián Lopez, 2006). Extraído el 15 de Junio del 2011 desde <http://www.monografias.com/trabajos75/seguridad-desarrollo-aplicaciones-web/seguridad-desarrollo-aplicaciones-web.shtml>*

## **Principios Básicos de Seguridad**

**Validación de la entrada y salida de información.** La entrada y salida de información es el principal mecanismo que dispone un atacante para enviar o recibir código malicioso contra el sistema.

Por tanto, siempre debe verificarse que cualquier dato entrante o saliente es apropiado y en el formato que se espera. Las características de estos datos deben estar predefinidas y debe verificarse en todas las ocasiones.

**Diseños simples.** Los mecanismos de seguridad deben diseñarse para que sean los más sencillos posibles, huyendo de sofisticaciones que compliquen excesivamente la vida a los usuarios. Si los pasos necesarios para proteger de forma adecuada una función o modulo son muy complejos, la probabilidad de que estos pasos no se ejecuten de forma adecuada es muy elevada.

**Verificación de privilegios.** Los sistemas deben diseñarse para que funcionen con los menos privilegios posibles. Igualmente, es importante que los procesos únicamente dispongan de los privilegios necesarios para desarrollar su función, de forma que queden compartimentados.

**Ofrecer la mínima información.** Ante una situación de error o una validación negativa, los mecanismos de seguridad deben diseñarse para que faciliten la mínima información posible. De la misma forma, estos mecanismos deben estar diseñados para que una vez denegada una operación, cualquier operación posterior sea igualmente denegada.

En el desarrollo de aplicaciones web, un aspecto de vital importancia a tomar en consideración es la seguridad, la creación, desarrollo y administración de las aplicaciones web es una tarea de mucha responsabilidad, ya que esta va a ser una herramienta para determinada empresa, institución, etc. Y es aquí donde toma lugar la seguridad que se implemente para su correcto funcionamiento.

*Principios Básicos de Seguridad. (Gabriel Arellano, 2008). Extraído el 15 de Junio del 2011 desde <http://www.slideshare.net/areche/seguridad-de-aplicaciones-web-presentation>*

### **2.3.1.1 Guías de seguridad**

#### **2.3.1.1.1 Seguridad en el Cliente**

##### **Código móvil**

Código que circula por la red y se ejecuta en una máquina remota. Aparece incrustado en un documento HTML. Un cliente de correo o un navegador que carguen el documento lo ejecutarán en la máquina cliente

Potencia la funcionalidad de los documentos HTML pero entraña riesgos de seguridad. Un código móvil puede obtener información acerca de un sistema o un usuario y enviarla a una máquina remota

Puede estar incrustado directamente en el documento, caso de los lenguajes de script como JavaScript y VBScript

Hay cuatro tipos básicos de código móvil:

1. Lenguajes de macro como Visual Basic for Applications (VBA)

Es un lenguaje de macro propio de Microsoft Office, aunque otras aplicaciones lo han adoptado

Permite el acceso a todas las funciones de la aplicación, incluyendo el acceso a disco

La macro está incrustada en el documento

2. Scripts como JavaScript y VBScript

JavaScript.

Fue diseñado para añadir interactividad a una página web

Un código JavaScript tiene acceso únicamente a la información contenida en la página en la que está incrustado

Es bastante seguro. Algunos problemas del pasado han estado relacionados con implementaciones de JavaScript por parte de Microsoft y Netscape. La madurez de la tecnología ha permitido solucionarlos

VBScript.

Sólo funciona con Internet Explorer y Outlook, por lo que no es tan popular como JavaScript

Ofrece una funcionalidad similar pero con una notable excepción: puede interactuar con los controles ActiveX instalados en la máquina

### 3. Applets de Java

Normalmente no pueden leer ni escribir datos en el disco local ni comunicarse con otro recurso de la red salvo el servidor del que procede, lo cual garantiza que no tendrán comportamiento malicioso

### 4. Controles ActiveX

Son la respuesta de Microsoft a las applets de Java, sólo funcionan básicamente con Internet Explorer y Outlook. La seguridad de los controles ActiveX se basa en los certificados digitales. Los controles están firmados digitalmente para garantizar su autenticidad

Un método de protección común es tener siempre actualizado el software

#### **2.3.1.1.2 Seguridad en el Servidor**

El desarrollo de una aplicación web requiere de una serie de herramientas: servidores web, servidores de aplicaciones, servidores de bases de datos, lenguajes de servidor, etc.

#### **Servidor Web**

Proporciona muchos servicios y es muy probable que algunos de ellos no sean necesarios para el funcionamiento de la aplicación web. En tal caso es conveniente deshabilitarlos, para ello el servidor dispone de múltiples opciones de configuración que es conveniente adaptar a las circunstancias concretas de la aplicación web.

Es muy conveniente revisar periódicamente los ficheros de log (*access\_log* y *error\_log* en Apache) para detectar posibles ataques al servidor



## **Servidor de Bases de Datos**

Proporciona acceso a bases de datos, fundamental en toda aplicación web importante. Riesgos:

- Descubrimiento de información acerca de los datos de conexión al servidor (usuario y clave), información sensible almacenada en la base de datos (tarjetas de crédito...) o información sobre la estructura de la base de datos
- Modificación de las instrucciones SQL enviadas al servidor, construidas de forma dinámica a partir de datos recibidos del usuario y por tanto potencialmente peligrosos (Inyección SQL)
- Acceso no autorizado a información restringida
- Hay que vigilar la configuración por defecto del servidor, que puede incluir bases de datos y usuarios predefinidos que conviene eliminar

En el código de la aplicación hay que tener, entre otras, las siguientes precauciones:

- Validar las instrucciones SQL antes de enviarlas al servidor
- No revelar información sobre la base de datos en los mensajes de error (esquema, naturaleza de los datos almacenados, fragmentos SQL)
- Proteger el código donde aparezca información sensible para el acceso al servidor

## **Lenguajes de servidor**

ASP, JSP, PHP, etc.

Aumentan enormemente la potencia de los documentos HTML al permitir la comunicación con aplicaciones residentes en el servidor, y muy especialmente con servidores de bases de datos.

Esta potencialidad conlleva riesgos. Hay que revisar a fondo la configuración para eliminar funcionalidades no utilizadas y seguir prácticas adecuadas de programación, sobre todo en funciones con vulnerabilidades conocidas.

Hay que proteger el código fuente para evitar que pueda ser visualizado, especialmente cuando contiene información sensible como pueden ser los datos de conexión al servidor de bases de datos. Una medida razonable consiste en sacar el código fuente sensible fuera de la raíz de la web

### **2.3.1.1.3 Seguridad en la Aplicación**

#### **Control de acceso**

Un aspecto muy importante de una aplicación web es el control de acceso de los usuarios a zonas restringidas de la aplicación

#### **Autenticación basada en la aplicación**

La propia aplicación puede implementar un mecanismo de autenticación que implica la presentación de un formulario para que el usuario introduzca sus credenciales y el uso de una base de datos para verificar la corrección de éstas

Es más costosa pero más flexible ya que permite establecer diferentes permisos y niveles de acceso en función del usuario que solicita la autenticación

#### **Sesiones**

Una vez que el usuario se ha autenticado introduciendo su nombre de usuario y su clave, es preciso mantener esta autenticación en cada conexión subsiguiente

Para evitar tener que mostrar nuevamente la ventana de autenticación se recurre habitualmente al uso de sesiones, un mecanismo que permite mantener el estado entre diferentes peticiones HTTP

El mecanismo es el siguiente:

Una vez autenticado, al usuario se le asigna un identificador de sesión, este identificador acompañará invisiblemente a cada petición del usuario, con lo cual se garantizará que la petición proviene de un usuario previamente autenticado

El identificador de sesión se suele almacenar en la propia máquina del cliente, mediante una *cookie*. La gestión de las sesiones es responsabilidad del programador. Normalmente los lenguajes de servidor disponen de funciones diseñadas específicamente para ello.

### **Validación de datos de entrada**

El problema más frecuente que presentan las aplicaciones web es no validar correctamente los datos de entrada, esto da lugar a algunas de las vulnerabilidades más importantes de las aplicaciones, como la Inyección SQL, el *Cross-Site Scripting* y el *Buffer Overflow*

### **Inyección SQL**

Consiste en inyectar un mandato dentro de una consulta SQL. Sea la consulta:

```
$consulta = "SELECT titulo FROM libros WHERE codigo = $codigo";
```

Siendo, \$codigo un valor introducido desde un formulario. Si el valor es '23' la consulta será:

```
SELECT titulo FROM libros WHERE codigo = 23
```

Si el valor es '23; DROP TABLE users' la consulta es:

```
SELECT titulo FROM libros WHERE codigo = 23; DROP TABLE users
```

que destruiría la tabla de usuarios de una Base de Datos.

Sea ahora el siguiente código muy habitual en una aplicación Web:

```
$consulta = "SELECT id FROM usuarios WHERE username = '$username' AND password = '$password'";
```

Si se introducen los valores juan como username y jj.ssii como password, la consulta queda:

```
SELECT id FROM usuarios WHERE username = 'juan' AND password = 'jj.ssii'
```

### **La inyección SQL puede utilizarse para:**

- Cambiar valores de las consultas
- Concatenar varias consultas
- Añadir llamadas a función y procedimientos almacenados a una consulta

Para evitar la inyección SQL es muy importante validar los valores que se han de enviar en los formularios.

### **Inyección HTML (*Cross-site Scripting*)**

Consiste en insertar en un texto (p.ej. un mensaje de un foro) código malicioso (p.ej. JavaScript). Cuando otro usuario visualice el texto el código se ejecutará en su máquina. Por ejemplo, si se inserta el texto:

```
¿Una galleta?<script>alert(document.cookie)</script>
```

Cuando un usuario lo visualice aparecerá su *cookie* en una ventana. Esto no es grave ya que cada usuario visualiza su propia *cookie*, pero si se modifica así:

```
<script>document.write('')</script>
```

dejará la *cookie* del usuario en el log del servidor del atacante, que podría hacerse con la sesión

Hay varios tipos de cosas que se pueden insertar en el código HTML de esta manera:

Marcas HTML, como <SCRIPT>, <A>, <IMG> o <IFRAME>. El efecto se produce cuando el texto se visualiza en el navegador de otro usuario

Eventos, como ONCLICK, asociados habitualmente a elementos de formulario

Para evitar este ataque es conveniente filtrar todos los caracteres que tienen un significado especial en HTML como “, &, < y >. Los lenguajes disponen de funciones específicas para ello.

### **Buffer Overflow**

Este ataque consiste en corromper la pila de ejecución de una aplicación enviando unos datos de entrada especialmente preparados con tal fin. El objetivo es conseguir la ejecución de un código enviado por el atacante y tomar el control de la máquina

Estas vulnerabilidades no son fáciles de detectar y, de hacerse, son muy difíciles de explotar, pueden estar presentes en las herramientas (como el servidor web) o bibliotecas externas, siendo en tal caso conocidas públicamente y por tanto más peligrosas. La única protección contra ellas consiste en tener actualizadas todas las herramientas

#### **2.3.1.1.4 Programación segura.**

Para evitar o al menos disminuir las vulnerabilidades de una aplicación web es muy importante seguir unas correctas prácticas de programación. Veamos algunas de las más importantes:

- Inicialización de variables
- En general es recomendable inicializar todas las variables antes de usarlas
- Gestión de errores
- Los mensajes de error son una fuente de información muy importante para los atacantes. Pueden proporcionar información sensible que les permita refinar sus ataques
- En un entorno de producción debe evitarse la aparición de mensajes de aviso o error
- Protección de información
- Toda información sensible debe almacenarse por separado del programa que la utiliza y preferentemente en un directorio situado fuera del árbol de directorios de la Web para evitar que pueda ser accedida por su URL. De

esta forma, aunque se revele el código fuente de los programas, no se mostrará información que comprometa al sistema

- Debe evitarse utilizar en el código comentarios que den demasiados detalles acerca del funcionamiento del programa. Puede ser conveniente eliminarlos en la versión de producción de la aplicación

#### 2.3.1.1.4 Seguridad en la Comunicación

SSL (*Secure Socket Layer*) es un protocolo para asegurar el transporte de datos entre el cliente y el servidor web. Diseñado inicialmente por Netscape, hoy día es soportado por la mayoría de los servidores web

Podemos definir una aplicación segura como aquella que cifra sus comunicaciones con el navegador que la visita, es decir, tanto el cliente como el servidor se comunican de una forma segura.

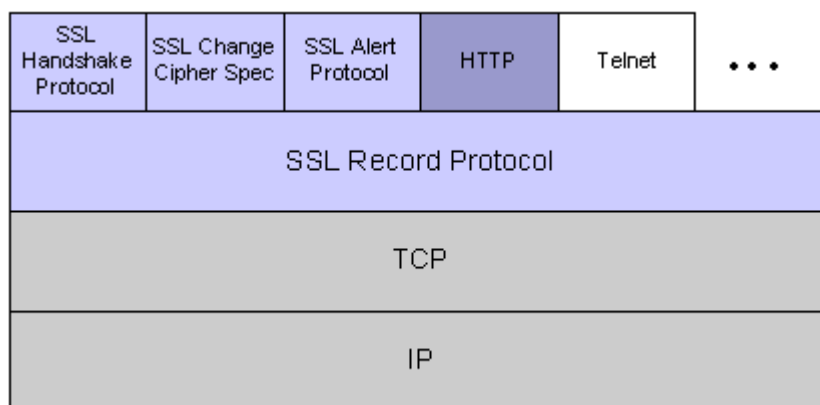
Esto se consigue mediante el uso del protocolo SSL que permite la autenticación de servidores, la codificación de datos y la integridad de los mensajes. Podemos reconocer una conexión HTTP sobre SSL porque aparece el prefijo 'https' en lugar de 'http' en la URL



**Figura 2.1** Seguridad en la Comunicación

SSL no es un protocolo simple sino que tiene dos niveles de protocolos

- El protocolo Record proporciona servicios de seguridad básica a varios protocolos de nivel más alto, entre ellos HTTP



**Figura 2.2** Niveles SSL

SSL proporciona una comunicación segura entre cliente y servidor permitiendo la autenticación mutua, el uso de firmas digitales y garantizando la privacidad mediante encriptación. Una sesión SSL se establece según una secuencia de operaciones

### **HTTPS y cómo funciona**

El protocolo de Transferencia de Hiper-Texto (HTTPS) es la versión segura de el http (Hyper Text Transfer Protocol) que todos conocemos y utilizamos habitualmente. La diferencia es que, con HTTPS podemos desarrollar actividades ecommerce, ya que permite realizar transacciones de forma segura.

En los navegadores comunes, como Firefox o Explorer, cuando estamos empleando un protocolo https podemos ver el icono de un candado, que aparece en la barra principal de nuestro navegador. Además, en la barra de direcciones podremos ver que “http://” será sustituido por “https://”.

¿Y cómo funciona la conexión exactamente? ¿Por qué es más segura? Básicamente, lo que ocurre es que la página Web codifica la sesión con certificado digital. De este modo, el usuario tiene ciertas garantías de que la información que envíe desde dicha página no podrá ser interceptada y utilizada por terceros.

Estos certificados de seguridad son conocidos como SSL. Cuando estos estan instalados en la pagina Web veremos el candado del que hablabamos anteriormente. Por otro lado, si estan instalados Certificados de Validacion Extendida, los usuarios, ademas del candado, podremos ver que la barra de URL's del navegador toma un fondo verdoso. (siempre que estemos utilizando las ultimas versiones de Firefox, Explorer o Opera).

*Guias de Seguridad. (2011). Extraido el 15 de Junio del 2011 desde <http://www.lsi.us.es/~quivir> Dpto. Lenguajes y Sistemas Informaticos Universidad de Sevilla*

*HTTPS y como funciona. (s.f). Extraido el 15 de Junio del 2011 desde <http://www.linuxhispano.net/2011/02/21/configurar-soporte-https-en-apache/>*

### **2.3.2 Software Libre**

El software libre es la denominacion 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.

- La libertad de ejecutar el programa, para cualquier proposito (libertad 0).
- La libertad de estudiar como trabaja el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al codigo fuente es una condicion necesaria para ello.
- La libertad de redistribuir copias para que pueda ayudar al projimo (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (la 3<sup>a</sup> libertad). Si lo hace, puede dar a toda la comunidad una oportunidad de beneficiarse de sus cambios. El acceso al codigo fuente es una condicion necesaria para ello.

Un programa es software libre si los usuarios tienen todas esas libertades. Entonces, debera ser libre de redistribuir copias, tanto con o sin modificaciones,



ya sea gratis o cobrando una tarifa por distribución, a cualquiera en cualquier parte.

*Software Libre. (s.f). Extraído el 15 de Junio del 2011 desde <http://www.hispalinux.es/SoftwareLibre>*

### **2.3.2.1 ¿Por qué utilizar software libre?**

Cada día es más importante el papel de las tecnologías de la información en la sociedad actual. Las comunicaciones, la economía, la educación, el flujo de información, incluso la vida doméstica depende de los programas de ordenador o software.

El ejemplo positivo es Internet: un sistema informático global en el que generalmente nadie gobierna, ni limita lo que se publica. El resultado: libertad de expresión, libertad de comunicación, impacto económico positivo a nivel global, y un largo etcétera.

El Software Libre viene a solucionar estas cuestiones éticas de una forma útil y eficaz.

**'Software libre'** no significa 'no comercial'. Un programa libre debe estar disponible para uso comercial, desarrollo comercial y distribución comercial. El desarrollo comercial del software libre ha dejado de ser inusual; el software comercial libre es muy importante.

El software libre uno de los aspectos más importantes en el desarrollo de aplicaciones web, ya que se puede hacer uso de herramientas muy poderosas, y sin costo alguno, esto beneficia tanto a los desarrolladores como a los usuarios, que serán parte de un eficiente sistema.

*Software Libre. (2011). Extraído el 15 de Junio del 2011 desde <http://www.hispalinux.es/SoftwareLibre>*

### **2.3.3 Sitios Web**

Un sitio web es un conjunto organizado y coherente de páginas web (generalmente archivos en formato html, php, cgi, etc.) y objetos (gráficos,

animaciones, sonidos, etc.). Su temática puede ser muy variada. A través de un sitio web podemos ofrecer, informar, publicitar o vender contenidos, productos y servicios al resto del mundo.

Para que un sitio web pueda ser visitado por otras personas es necesario que se encuentre alojado en un servidor. Se trata de una computadora conectada a la World Wide Web con espacio en disco y conectividad suficiente para albergar sitios y servirlos al resto de la comunidad de usuarios de Internet a través de direcciones IP o nombres de dominio.

*Un sitio web es un conjunto organizado y coherente de páginas web. (s.f.).  
Extraído el 15 de Junio del 2011 desde <http://www.masadelante.com/faqs/sitio-web>*

Cada sitio web tiene una página de inicio (en inglés Home Page), que es el primer documento que ve el usuario cuando entra en el sitio web poniendo el nombre del dominio de ese sitio web en un navegador. El sitio normalmente tiene otros documentos (páginas web) adicionales. Cada sitio pertenece y es gestionado y por un individuo, una compañía o una organización.

Un sitio web dinámico es uno que puede tener cambios frecuentes en la información. Cuando el servidor web recibe una petición para una determinada página de un sitio web, la página se genera automáticamente por el software, como respuesta directa a la petición de la página.

Hay un amplio abanico de sistemas de software, como el lenguaje de programación PHP, Active Server Pages (ASP), y Java Server Pages (JSP) que están disponibles para generar sistemas de sitios web dinámicos. Los sitios dinámicos a menudo incluyen contenido que se recupera de una o más bases de datos o usando tecnologías basadas en XML como por ejemplo el RSS.

Hay plugins disponibles para navegadores, que se usan para mostrar contenido activo como Flash, Silverlight, Shockwave o applets, escritos en Java.

Los sitios web son un conjunto varios elementos que nos ayudan a dar una buena presentación de nuestros diseños o ideas. Este tema es muy amplio y cada día hay

nuevos modelos de páginas muy profesionales, la innovación constante y la investigación nos pueden ayudar a desarrollar mejor sitios web cada día, el uso de varias herramientas que harán q nuestros sitios sean más completos es un aspecto a tomar en cuenta.

*Cada sitio web tiene una página de inicio. (2011.). Extraído el 15 de Junio del 2011 desde [http://es.wikipedia.org/wiki/Sitio\\_web](http://es.wikipedia.org/wiki/Sitio_web)*

#### **2.3.4 Sistema Web Integrado**

Los sistemas desarrollados en plataformas Web (Aplicaciones Web), tienen marcadas diferencias con otros tipos de sistemas, lo que lo hacen muy beneficioso tanto para las empresas que lo utilizan, como para los usuarios que operan en el sistema.

Las aplicaciones web se han convertido en pocos años en complejos sistemas con interfaces de usuario cada vez más parecidas a las aplicaciones de escritorio.

En la ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

En los últimos años, la rápida expansión de Internet y del uso de intranets corporativas ha supuesto una transformación en las necesidades de información de las organizaciones.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales.

Ventajas:

- Las aplicaciones web requieren poco o nada de espacio en disco. Además suelen ser livianas.

- No requieren que los usuarios las actualicen, eso es implementado del lado del servidor.
- Proveen gran compatibilidad entre plataformas (portabilidad), dado que operan en un navegador web.

Desventajas:

- Muchas veces requieren una conexión a internet para funcionar, si la misma se interrumpe, no es posible utilizarla más. De todas maneras, en ocasiones, pueden ser descargadas e instaladas localmente para su uso offline.
- La aplicación web desaparece si así lo requiere el desarrollador o si el mismo se extingue. Las aplicaciones tradicionales, en general, pueden seguir usándose en esos casos.
- El usuario, en general, no tiene libertad de elegir la versión de la aplicación web que quiere usar. Un usuario podría preferir usar una versión más antigua, hasta que la nueva sea probada.
- En teoría, el desarrollador de la aplicación web puede rastrear cualquier actividad que el usuario realice.

*Sistema Web Integrado. (2010.). Extraído el 15 de Junio del 2011 desde [http://es.wikipedia.org/wiki/Aplicaci%C3%B3n\\_web](http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web)*

Las aplicaciones web en la actualidad, son de vital importancia y de mucha acogida por la flexibilidad que ofrece esta tecnología, gracias a esto los usuarios tienen muchos más beneficios que en las aplicaciones de escritorio. No hay que dejar de lado las ventajas y desventajas que estas aplicaciones pueden tener y el uso que se les puede dar.

El hecho de poder integrar una aplicación web a un sitio web, es muy importante en la actualidad, ya que tendremos estos aspectos trabajando en forma simultanea y/o acceder en cualquier momento que el usuario requiera a una de estas ya mencionadas herramientas.

*Las aplicaciones web en la actualidad. (2010). Extraído el 15 de Junio del 2011 desde <http://www.alegsa.com.ar/Dic/aplicacion%20web.php>*

### **2.3.5 Reportes**

Muchas veces nos encontramos realizando sistemas de información que se dedican a capturar datos pero que también necesitan una manera de procesarlos y mostrarlos. Para esta tarea entran en juego los famosos Reportes que no son más que objetos que entregan información en un formato particular y que permiten realizar ciertas operaciones como imprimirlos, enviarlos por email, guardarlos a un archivo, etc.

Es importante mencionar que los datos almacenados son útiles en la misma medida que se puedan convertir en información para las personas que los necesitan. También es importante subrayar que la plataforma tecnológica que utilizemos debe poder tener facilidades para convertir los datos en información y poder entregarlos a los usuarios de forma que sean útiles.

Los reportes son una manera excelente de organizar y presentar los datos de una base de datos. Los informes permiten aplicar formato a los datos dando lugar a un diseño atractivo e informativo en la pantalla o en las copias impresas.

1. Los informes pueden darles mayor utilidad a los datos.
2. Los informes pueden tener diferentes niveles de complejidad, desde una simple lista (como un informe sobre el estado de un proyecto o un informe semanal de ventas) hasta un catálogo detallado de productos.
3. Si bien los informes permiten embellecer los datos, agrupar y presentarlos de muchas maneras distintas, no alteran los datos subyacentes en las tablas de la base de datos.

Los reportes es uno de los aspectos más relevantes cuando de desarrollo de software se trata, el presentar información ordenada, precisa, coherente es una ayuda importante para los usuarios de las aplicaciones, esto convierte a los procesos en tareas más eficaces y eficientes. Se considerar como un punto muy

importante al momento de desarrollar aplicaciones, ya que en cualquier momento del uso de estas, se va a requerir la presentación de información en reportes.

*Reportes. (s.f.). Extraído el 15 de Junio del 2011 desde <http://weblogs.asp.net/joseguay/archive/2008/06/26/construyendo-reportes-para-aplicaciones-web-parte-1.aspx>*

### **2.3.6 SQL (lenguaje de consulta estructurado)**

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés structured query language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Una de sus características es efectuar consultas con el fin de recuperar de una forma sencilla información de interés de una base de datos, así como también hacer cambios sobre ella.

#### **2.3.6.1 Características Generales**

El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones en éstos últimos.

Es un lenguaje declarativo de "alto nivel" o "de no procedimiento", que gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros, y no a registros individuales, permite una alta productividad en codificación y la orientación a objetos. El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agragado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

*SQL. (Lenguaje de Consulta Estructurado) (s.f.). Extraído el 15 de Junio del 2011 desde <http://es.wikipedia.org/wiki/SQL>*

#### **2.3.6.2 Lenguaje de definición de datos DDL (Data Definition Language)**

El lenguaje de definición de datos (en inglés Data Definition Language, o DDL), es el que se encarga de la modificación de la estructura de los objetos de la base

de datos. Existen cuatro operaciones básicas: CREATE, ALTER, DROP y TRUNCATE.

### **Create**

Este comando crea un objeto dentro de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier otro objeto que el motor de la base de datos soporte.

### **Alter**

Este comando permite modificar la estructura de un objeto. Se pueden agregar/quitar campos a una tabla, modificar el tipo de un campo, agregar/quitar índices a una tabla, modificar un trigger, etc.

### **Drop**

Este comando elimina un objeto de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier otro objeto que el motor de la base de datos soporte. Se puede combinar con la sentencia ALTER.

### **Truncate**

Este comando trunca todo el contenido de una tabla. La ventaja sobre el comando DROP, es que si se quiere borrar todo el contenido de la tabla, es mucho más rápido, especialmente si la tabla es muy grande. La desventaja es que TRUNCATE sólo sirve cuando se quiere eliminar absolutamente todos los registros, ya que no se permite la cláusula WHERE.

### **2.3.6.3 Lenguaje de manipulación de datos DML (Data Manipulation Language)**

Un lenguaje de manipulación de datos (Data Manipulation Language, o DML en inglés) es un que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.

### **Insert**

Una sentencia INSERT de SQL agrega uno o más registros a una (y sólo una) tabla en una base de datos relacional.

## **Update**

Una sentencia UPDATE de SQL es utilizada para modificar los valores de un conjunto de registros existentes en una tabla.

## **Delete**

Una sentencia DELETE de SQL borra uno o más registros existentes en una tabla. Debido a la diversidad de lenguajes y de bases de datos existentes, la manera de comunicar entre unos y otras sería realmente complicada a gestionar de no ser por la existencia de estándares que nos permiten el realizar las operaciones básicas de una forma universal.

El SQL es un lenguaje que nos permite acceder a las bases de datos con el fin de realizar operaciones ya sean de creación, mantenimiento, consulta, etc. Hay que comprender la importancia de este lenguaje cuando de aplicaciones con bases de datos se habla.

*Introducción a SQL. (Claudio Casares s.f.). Extraído el 15 de Junio del 2011 desde <http://www.maestrosdelweb.com/editorial/tutsql1/>*

### **2.3.7 Base de Datos**

Una base de datos o banco de datos (en ocasiones abreviada con la sigla BD o con la abreviatura b. d.) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

Existen programas denominados sistemas gestores de bases de datos, abreviado 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.



Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro.

### **2.3.7.1 Características**

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

### **2.3.7.2 Ventajas de las bases de datos**

#### **Control sobre la redundancia de datos**

En los sistemas de bases de datos todos estos ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos.

### **Consistencia de datos**

Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente.

### **Mejora en la integridad de datos**

La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargar de mantenerlas.

### **Mejora en la seguridad**

La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros.

### **Mejora en la accesibilidad a los datos:**

Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.

### **2.3.7.3 Desventajas de las bases de datos**

#### **Complejidad**

Los SGBD son conjuntos de programas que pueden llegar a ser complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder realizar un buen uso de ellos.

### **Vulnerable a los fallos**

El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse. Es por ello que deben tenerse copias de seguridad (Backup).

*Bases de Datos. (2011) Extraído el 15 de Junio del 2011 desde [http://es.wikipedia.org/wiki/Base\\_de\\_datos](http://es.wikipedia.org/wiki/Base_de_datos)*

### **2.3.7.4 Modelo entidad-relación**

Los diagramas o modelos entidad-relación (denominado por su siglas, ERD “Diagram Entity relationship”) son una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información, sus inter-relaciones y propiedades.

Una **base de datos relacional** es una base de datos que cumple con el modelo relacional, el cual es el modelo más utilizado en la actualidad para implementar bases de datos ya planificadas. Permiten establecer interconexiones (relaciones) entre los datos (que están guardados en tablas), y a través de dichas conexiones relacionar los datos de ambas tablas

### **Características**

- Una base de datos relacional se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre ni registro.
- Cada tabla es a su vez un conjunto de registros (filas y columnas).
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y ajenas (o foráneas).
- Las claves primarias son la clave principal de un registro dentro de una tabla y éstas deben cumplir con la integridad de datos.

- Las claves ajenas se colocan en la tabla hija, contienen el mismo valor que la clave primaria del registro padre; por medio de éstas se hacen las relaciones

Se puede decir que las bases de datos, son uno de los elementos sino el mas indispensable en el desarrollo de aplicaciones, ya que aquí se almacena toda la información que va a procesar nuestro sistema, hay que dar mucha importancia ya que de aquí parte las actividades o procesos que realizará nuestra aplicación.

En el ámbito web, toma mayor importancia ya que la seguridad pasa a tomar mayor importancia, debido a que nuestra base de datos se encuentra expuesta a varios riesgos que se adquieren en el internet.

*Modelo Entidad – Relación. s.f.). Extraído el 15 de Junio del <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>*

### **2.3.8 Gestión de Cobranza de Valores**

El proceso de gestión de cobranza de valores comprende etapas claramente diferenciadas que pueden ser tratadas en forma independiente y que se pueden clasificar en los siguientes segmentos:

#### **Distribución de la factura.**

Gestión y activación de cobranza.

Recolección de valores.

Gestión de tesorería.

#### **Distribución de la Factura.**

Se propone esta etapa, teniendo en cuenta la problemática existente en las empresas, respecto a la tarea de administrar un proceso de facturación y sus documentos asociados - nota de crédito y débito etc., el cual se compone en algunos casos de grandes gastos, como así también de problemas de control interno y de logística.

#### **Ventajas**

- Disminución de costos administrativos.

- Disminución de costos financieros.
- Erradicación de costos operativos en el área de Atención al Cliente.

### **Gestión y Activación de Cobranza.**

En forma práctica el sistema de gestión, consiste en obtener mediante acuerdo con el cliente, el día y hora de la cobranza, así también un control no solo de la fecha de pago de las facturas, sino también, el seguimiento de pagos del cliente.

### **Gestión y recolección de valores.**

Operativamente, aportará agilidad e información en tiempo real de todas aquellas gestiones que se hayan decidido insertar en el presente proceso.

### **Gestión de Tesorería.**

Una vez verificado el pago, la acreditación, se llevará a cabo su debida imputación en la contabilidad de la empresa mediante una interfaz que permita trasladar mediante sistematización contable, la imputación de los pagos de las facturas a la contabilidad general de la empresa.

*Gestión de Cobranza. (s.f.). Extraído el 15 de Junio del [http://www.emacsa.com.ar/ordinarias\\_es.htm](http://www.emacsa.com.ar/ordinarias_es.htm)*

## **2.4 Hipótesis**

La integración de un sistema web, optimizará la gestión de cobranza de valores para la administración de la información de la contribución para el mantenimiento de las vías intercantonales en interprovinciales en el Gobierno Provincial de Tungurahua.

## **2.5 Variables**

### **2.5.1 Variable Independiente**

Sistema Web Integrado.

### **2.5.2 Variable Dependiente**

Gestión de cobranza de valores.

## **CAPITULO III**

### **METODOLOGÍA**

#### **3.1 Enfoque**

El enfoque de la investigación es eminentemente cuantitativa ya que el investigador conoce, analiza y toma las decisiones más óptimas con las técnicas adecuadas para solucionar el problema y la institución nos proporciona la información necesaria; es importante conocer la información que proporcione la institución respecto al problema. La investigación dentro de la institución fue realizada por el técnico para solucionar el problema que fue proporcionado por la población de la misma institución interesada en solucionarlo.

#### **3.2 Modalidad básica de la Investigación**

##### **3.2.1 Investigación de Campo**

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

##### **3.2.2 Investigación Documental – Bibliográfica**

Esta modalidad permitió conocer, comparar, ampliar, profundizar y deducir diferentes enfoques, teorías, conceptualizaciones y criterios de diversos autores para el diseño del sistema para dar mejor control y administración de información, basándose en documentos (fuentes primarias), libros así también como en Internet

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

### **3.2.3 Proyecto Factible**

Se realizó una propuesta en base al desarrollo de un sistema web, para la mejor administración de la información del proceso de cobranza de valores de la contribución vial, que es un modelo práctico que permitió solucionar los problemas detectados en el Gobierno Provincial de Tungurahua previo el diagnóstico realizado con anterioridad al transcurso de la investigación y sustentación en el marco teórico.

### **3.3 Niveles o Tipos de Investigación**

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

### **3.4 Población y Muestra**

El presente trabajo de investigación se lo realizó en el Gobierno Provincial de Tungurahua, con una población de 6 personas las cuales están directamente involucradas con el proceso de gestión de cobranza.

Se trabajó con todo el universo investigativo considerando que la población es pequeña.





		<p><b>Confiabilidad</b></p> <p>¿La información almacenada de los contribuyentes y valores recaudados es confiable?</p> <p>Si <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	<p>Encuestas</p>
		<p><b>Reportes</b></p> <p>¿El sistema o mecanismo utilizado actualmente en el área de gestión de cobranza de valores de la contribución vial de la provincia, permite la generación de reportes oportunos?</p> <p>Si <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	

**Tabla 3.1** Operacionalización de variable independiente



			<p>¿Cree que el uso del Sistema Web optimice tiempos?</p> <p>Si <input type="checkbox"/></p> <p>No <input type="checkbox"/></p>	Encuestas
			<p>¿En cuanto beneficiará a los recaudadores la implementación del sistema web?</p> <p>Mucho <input type="checkbox"/></p> <p>Poco <input type="checkbox"/></p> <p>Nada <input type="checkbox"/></p>	Encuestas

**Tabla 3.2.** Operacionalización de variable dependiente

### **3.6 Recolección de información**

#### **3.6.1 Plan de recolección de información**

La recolección de información se realizó mediante encuestas realizadas al personal encargado del área de cobranza de valores de la contribución vial para el mantenimiento de las vías intercantonales e interprovinciales del Gobierno Provincial de Tungurahua, y al departamento financiero de la misma institución.

La estructura del cuestionario se incluye en los anexos como anexo 1

### **3.7 Procesamiento y análisis**

#### **3.7.1 Procesamiento y análisis de la información**

El análisis de la información se realizó mediante la interpretación de los datos recolectados, los cuales al ser procesados permitieron obtener un informe en base a sus resultados, lo que permitió plantear conclusiones y recomendaciones para dar solución al problema trazado.

#### **3.7.2 Plan de análisis e interpretación de los resultados**

Los resultados obtenidos a través de las encuestas realizadas al personal encargado del departamento de gestión de cobranza de valores de la contribución vial para el mantenimiento de las vías intercantonales e interprovinciales del Gobierno Provincial de Tungurahua se los representó mediante la utilización de gráficos estadísticos, los cuales sirvieron para dar solución al problema planteado, por último se elaboró una síntesis general para la elaboración de las conclusiones y recomendaciones.

## CAPITULO IV

### ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

#### 4.1 Análisis e interpretación de los resultados

Para la realización del análisis e interpretación de resultados, es importante recalcar que se aplicó las encuestas a 6 personas que están directamente involucradas en el área de recaudación y personal del departamento de sistemas del Gobierno Provincial de Tungurahua directamente involucrados con el proceso de gestión de cobranza como se mencionó en el capítulo III.

#### 4.2 Análisis de los resultados de las encuestas

##### 1. ¿De qué manera se maneja la gestión de cobranza de valores de la contribucion vial de la Provincia?

RESPUESTA	CANTIDAD	PORCENTAJE
Manualmente	0	0.0
Sistema Informático	0	0.0
Excel	6	100.0
Otros	0	0.0
Total	6	100.00

Tabla 4.1. Tabla porcentual pregunta 1

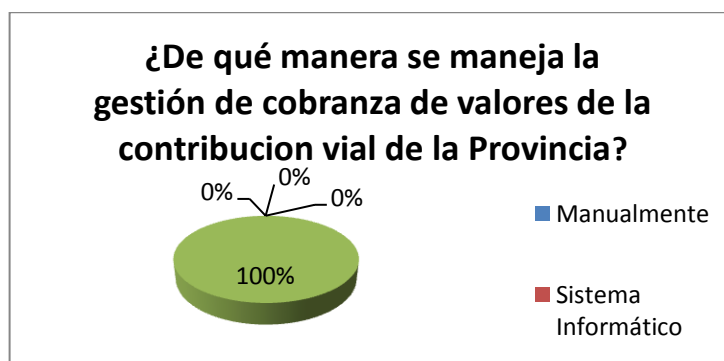


Figura 4.1. Pregunta 1

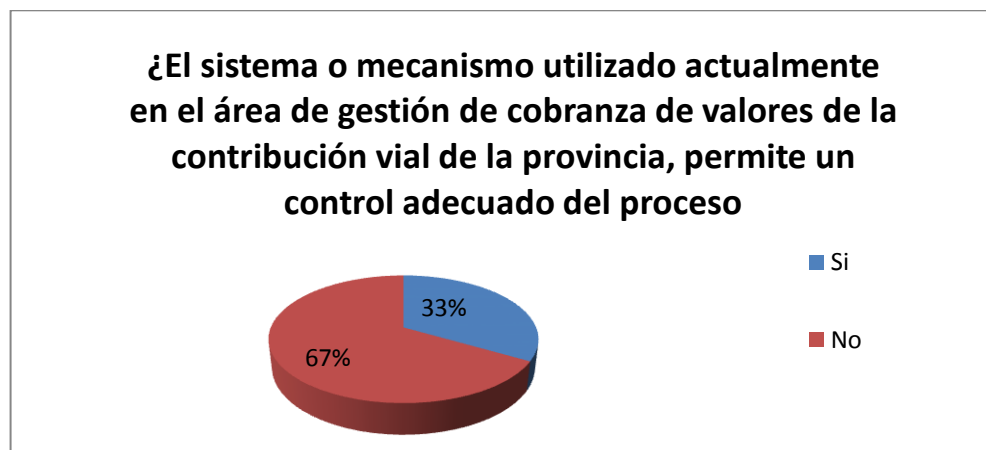
**Interpretación:** La gráfica refleja que el 100% de las personas encuestadas respondieron que manejan la información de la recaudación de valores por concepto de la contribución vial mediante Excel.

**Análisis:** Se demuestra que el proceso de gestión de cobranza en su totalidad se lo está realizando manualmente.

**2. ¿El sistema o mecanismo utilizado actualmente en el área de gestión de cobranza de valores de la contribución vial de la provincia, permite un control adecuado del proceso de valores a los contribuyentes?**

RESPUESTA	CANTIDAD	PORCENTAJE
Si	2	33,33%
No	4	66,67%
Total	6	100,00%

**Tabla 4.2.** Tabla porcentual pregunta 2



**Figura 4.2.** Pregunta 2

**Interpretación:** La gráfica refleja que el 67% de las personas encuestadas respondieron que no se tiene un control adecuado del proceso y el 33% respondió que si.

**Análisis:** Se concluye que la técnica de control del proceso de cobranza no permite un control eficiente de la información.

3. ¿El sistema o mecanismo utilizado actualmente en el área de gestión de cobranza de valores de la contribución vial de la provincia, permite la generación de reportes oportunos?

RESPUESTA	CANTIDAD	PORCENTAJE
Si	1	16,67%
No	5	83,33%
Total	6	100,00%

Tabla 4.3. Tabla porcentual pregunta 3

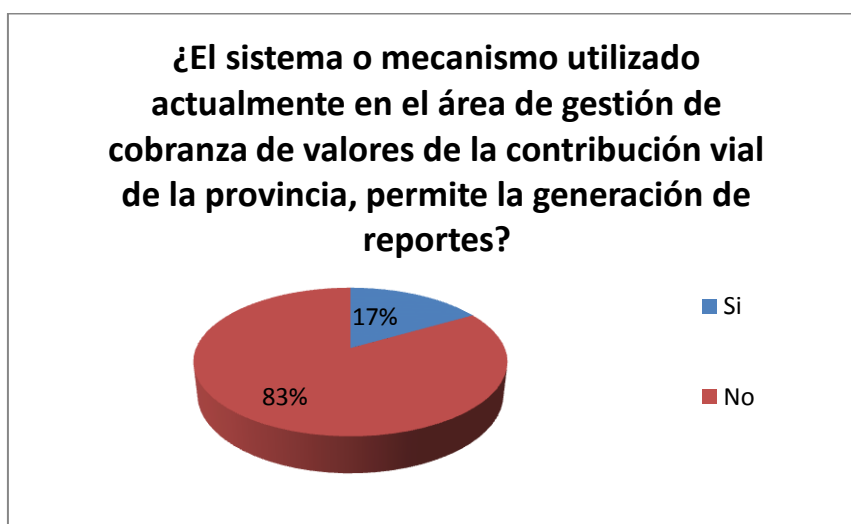


Figura 4.3. Pregunta 3

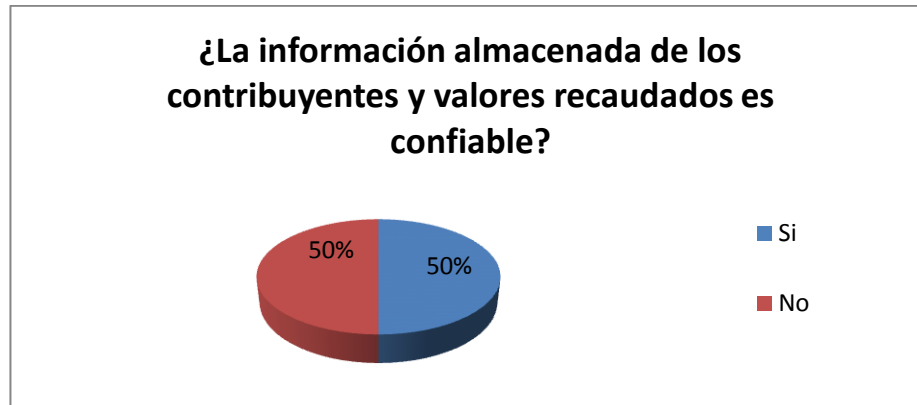
**Interpretación:** La gráfica refleja que el 83% de las personas encuestadas respondieron que el sistema o mecanismo utilizado actualmente no permite la generación de reportes oportunos y el 17% respondió que si permite.

**Análisis:** La mayoría de las personas encuestadas manifiesta que el mecanismo actual de gestión de cobranza no genera reportes oportunos.

4.- ¿La información almacenada de los contribuyentes y valores recaudados es confiable?

RESPUESTA	CANTIDAD	PORCENTAJE
Si	3	50,00%
No	3	50,00%
Total	6	100,00%

Tabla 4.4. Tabla porcentual pregunta 4



**Figura 4.4.** Pregunta 4

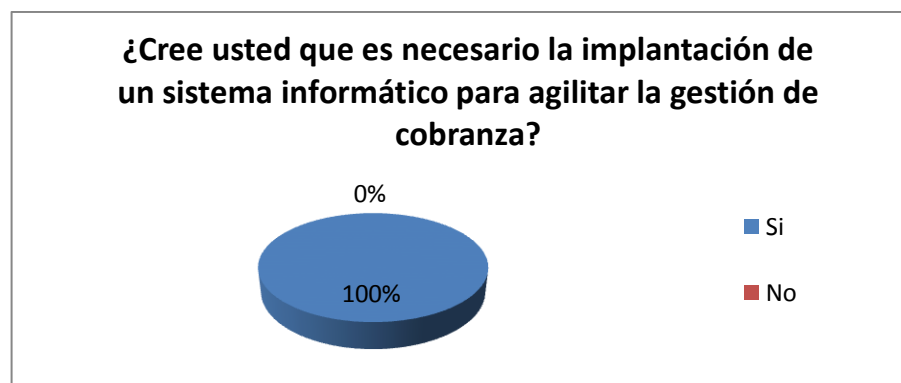
**Interpretación:** La gráfica refleja que el 50% de las personas encuestadas respondieron que la información almacenada de los contribuyentes y valores recaudados si es confiable, mientras que el otro 50% respondió que no.

**Análisis:** Concluimos que no se tiene un criterio seguro de si los datos generados en el procesos de cobranza es confiable.

**5.- ¿Cree usted que es necesario la implantación de un sistema informático para agilizar la gestión de cobranza?**

RESPUESTA	CANTIDAD	PORCENTAJE
Si	6	100,00%
No	0	0,00%
Total	6	100,00%

**Tabla 4.5.** Tabla porcentual pregunta 5



**Figura 4.5.** Pregunta 5



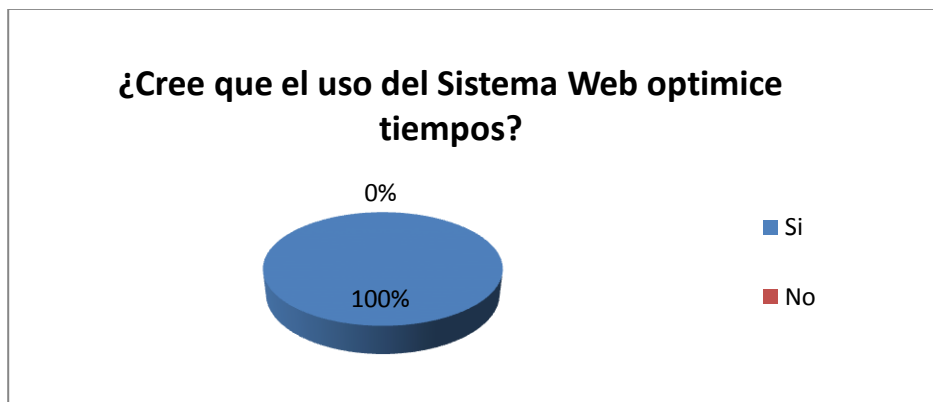
**Interpretación:** La gráfica refleja que el 100% de las personas encuestadas respondió que si es necesario la implantación de un sistema informático para agilizar la gestión de cobranza.

**Análisis:** La totalidad de las personas ecuestadas coinciden que se necesita de un sistemas informático para el proceso de cobranza.

#### 6.- ¿Cree que el uso del Sistema Web optimice tiempos?

RESPUESTA	CANTIDAD	PORCENTAJE
Si	6	100,00%
No	0	0,00%
Total	6	100,00%

**Tabla 4.6.** Tabla porcentual pregunta 6



**Figura 4.6.** Pregunta 6

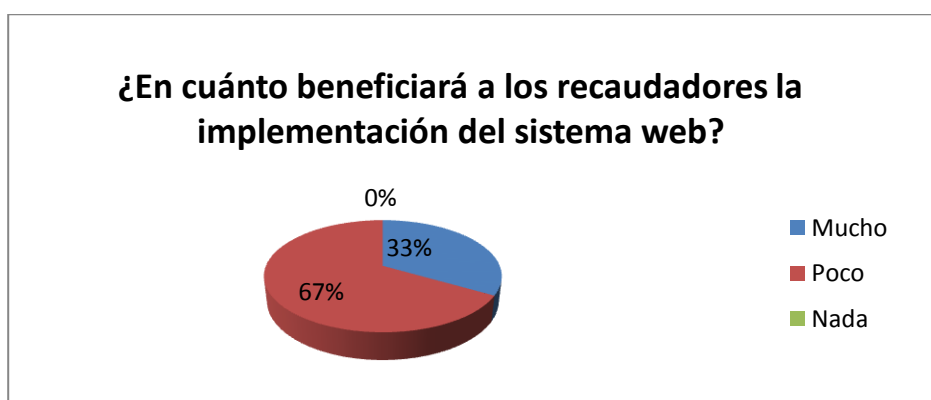
**Interpretación:** La gráfica refleja que el 100% de las personas encuestadas respondió que la implantación y uso de un sistema informático si optimizará tiempos.

**Análisis:** La totalidad de las personas ecuestadas coinciden que el uso de un sistema informático ayudará a optimizar el tiempo empleado en el proceso de cobranza.

**7.- ¿En cuánto beneficiará a los recaudadores la implementación del sistema web?**

RESPUESTA	CANTIDAD	PORCENTAJE
Mucho	2	33,33%
Poco	4	66,67%
Nada	0	0,00%
Total	6	100,00%

**Tabla 4.7.** Tabla porcentual pregunta 7



**Figura 4.7.** Pregunta 7

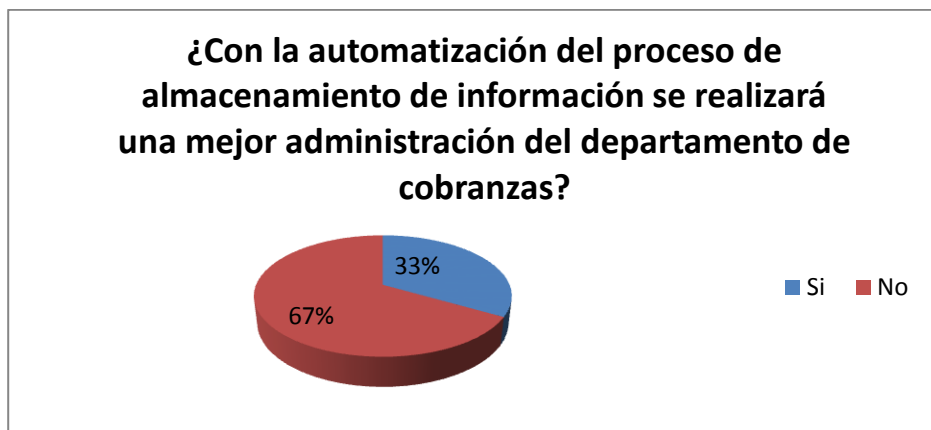
**Interpretación:** La gráfica refleja que el 67% de las personas encuestadas respondió que la implantación del sistema web beneficiará en mucho el poco de recaudación, mientras que el 33% dijo que el beneficiará en mucho.

**Análisis:** La mayoría de las personas ecuestadas desconocen el beneficio del uso de un sistema informático para automatizar algún proceso, motivo por el cual respondieron que será poco el beneficio.

**8.- ¿Con la automatización del proceso de almacenamiento de información se realizará una mejor administración del departamento de cobranzas?**

RESPUESTA	CANTIDAD	PORCENTAJE
Si	2	33,33%
No	4	66,67%
Total	6	100,00%

**Tabla 4.8.** Tabla porcentual pregunta 8



**Figura 4.8.** Pregunta 8

**Interpretación:** La gráfica refleja que el 67% de las personas encuestadas respondió que no se mejorará la administración del proceso de cobranza y el 33% dijo que si.

**Análisis:** Se concluye que la mayoría de personas encuestadas no confían en el uso de un sistema informático y no conocen de los beneficios del mismo.

## CONCLUSIÓN

Una vez procesados y analizados los datos resultantes de las encuestas efectuadas a las personas que trabajan en la Gobierno Provincial De Tungurahua, se llegó a determinar que la forma actual del proceso de cobranza no brinda confianza y dificulta la administración de la información, ya que mencionada institución no cuenta con un registro electrónico para almacenar los datos que se generan en el proceso de cobranza de la contribución vial, dando como consecuencia inconsistencia, duplicidad y/o pérdida de los mismos.

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1 Conclusiones**

- La forma en la que se está manejando la gestión de cobranza de valores en el Gobierno Provincial de Tungurahua es manual, esta forma no permite llevar una correcta administración de la información de los contribuyentes y de los valores recaudados.
- El personal de recaudación de valores de la contribución vial ha dado a conocer su inconformidad por la forma en que se ejecuta el proceso de cobranza en la actualidad y creen conveniente poner en marcha la implementación de un sistema informático.
- Con la implantación de un sistema informático se evitará las inconsistencias y duplicidad de la información al momento realizar el proceso de recaudación.
- Debido a que se trata de una institución pública el sistema deberá ser desarrollado en base a Software Libre, por medio del cual se escogió como lenguaje de desarrollo PHP y como SGBD a PostgreSQL ya que es un motor de base de datos confiable, potente y de gran eficiencia a la hora de recuperar información.

## 5.2 Recomendaciones

- Se recomienda desarrollar e implantar el sistema informático en el área de cobranza de valores como solución al problema investigado.
- Se recomienda analizar las herramientas seleccionadas para el desarrollo del sistema revisando las características como: licencias, soporte de sistemas operativos, soporte de servidores Web, soporte de gestores de base de datos, entre otros.
- Se recomienda incluir técnicas de seguridad web en el desarrollo del sistema.
- Los análisis de requerimientos y avances de desarrollo deberán presentarse de manera oportuna para que así luego se eviten problemas por mala interpretación de resultados.

## CAPÍTULO VI

### PROPUESTA

#### 6.1 Tema

SISTEMA WEB INTEGRADO PARA LA GESTIÓN DE COBRANZA DE VALORES EN EL GOBIERNO PROVINCIAL DE TUNGURAHUA.

#### 6.2 Datos informativos

<b>Institución:</b>	Gobierno Provincial de Tungurahua
<b>Ciudad:</b>	Ambato.
<b>Investigador:</b>	Alberto Eduardo Pérez Bautista.
<b>Tiempo:</b>	El presente proyecto será ejecutado entre el mes de Abril del 2011 hasta Abril del 2012.
<b>Tutor:</b>	Ing. Eduardo Chaso.

#### 6.3 Antecedentes

La implantación del Sistema Web surge por la necesidad de controlar la información de las personas que realizan las contribuciones para el mantenimiento vial de la provincia, y con esto mejorar la administración de esta información, evitando así inconsistencia y pérdida de datos.

El Gobierno Provincial en la mayoría de sus departamentos aún tiene varios procesos que se los ejecuta de forma manual, lo cual ocasiona en la mayoría de casos, molestia en los trabajadores de esta institución, no es la excepción el área de gestión de cobranza de valores de las contribuciones para el mantenimiento vial, que no cuenta con apoyo de ninguna herramienta informática que facilite la administración de la información, en beneficio de la población tungurahuesa.

Considerando que la seguridad, fiabilidad y fácil recuperación de la información se diseñará la base de datos en el SGBD PostgreSQL, para la parte de desarrollo Web se estableció el lenguaje de programación PHP y el framework ExtJS.

## **6.4 Justificación**

El mecanismo utilizado para la administración de la información de los contribuyentes y valores recaudados de la contribución para el mantenimiento vial de la provincia en el Gobierno Provincial de Tungurahua, no ayuda a llevar un control de datos, y tampoco a dar un mejor servicio a la población, por esta razón se decidió desarrollar e implantar un sistema web para la automatización de este proceso.

Además, la elaboración de la presente propuesta es factible y viable para utilizarla como instrumento de cambio y mejoramiento para la entidad, a la vez que permitirá al investigador adquirir conocimientos de nuevas e innovadoras tecnologías utilizadas en el desarrollo de aplicaciones web y más aun al ser estas libres.

El Software Libre al convertirse en una política tecnológica de estado proporciona oportunidades para profesionales en el campo del desarrollo Web, ya que son herramientas de fácil adquisición y gran soporte.

Por último es de gran ayuda conocer herramientas tanto propietarias como libres para poder desenvolverse en el ámbito laboral puesto que las dos opciones son útiles pero como en el mundo informático todo cambia se debe buscar la mejor opción tanto para el desarrollador como para el que lo necesita.

## **6.5 Objetivos**

### **6.5.1 Objetivo general**

Desarrollar e implementar un Sistema Web integrado utilizando herramientas de open source, para la gestión de cobranza de valores de la contribución vial en el Gobierno Provincial de Tungurahua.

### **6.5.2 Objetivos específicos**

- Determinar las necesidades del área de cobranza de valores de la contribución vial en el Gobierno Provincial de Tungurahua, para definir el funcionamiento que tendrá el sistema web a desarrollarse.
- Utilizar técnicas de seguridad web en el desarrollo del sistema, para proteger la integridad, disponibilidad y confidencialidad de la información.
- Desarrollar el sistema web en base a los requerimientos obtenidos, utilizando la tecnología PHP y PostgreSql.
- Utilizar el Framework Extjs para diseñar la interfaz del Sistema
- Realizar las pruebas necesarias para garantizar el óptimo funcionamiento del Sistema.

## **6.6 Análisis de factibilidad**

### **6.6.1 Factibilidad operativa**

El sistema web desarrollado, cuenta con interfaces sencillas y amigables para el manejo de todas sus funcionalidades; los usuarios del sistema, no requieren de una capacitación extensa para poder utilizarlo, ya que este provee un menú de administración lo que facilita el manejo de todas las funcionalidades.

El sistema web permite el ingreso, modificación y/o eliminación de información de los contribuyentes y sus vehículos, así mismo realiza el cálculo de valores a ser cobrados por la contribución del mantenimiento vial; de esta forma a los administradores del sistema, y encargados del área financiera podrán obtener la información clara, precisa y confiable para tomar decisiones rápidas y oportunas.

El sistema cuenta con los siguientes tipos de usuario:

Administrador

Tesorero(a)

Usuario (Recaudadores)



### **6.6.2 Factibilidad económica**

Es proyecto es factible ya que aporta significativamente al departamento financiero del Gobierno Provincial de Tungurahua y el sistema se lo desarrolló con herramientas de software libre, que no tienen ningún costo para la adquisición del lenguaje y SGBD seleccionados.

### **6.6.3 Factibilidad técnica**

Para la creación del sistema web, el Gobierno Provincial de Tungurahua cuenta con las herramientas necesarias, como el Apache Web Server, PostgreSQL un motor de Base de Datos potente, PHP como lenguaje de programación y el dominio para realizar pruebas en línea.

#### **Software**

- Lenguaje de programación PHP.
- PostgreSQL como motor de base de datos.
- Extjs para diseño de la interfaz.

#### **Hardware**

El Gobierno Provincial de Tungurahua, al ser una institución del estado, cuenta con las herramientas necesarias para la implantación del Sistema Web, aspecto que facilitará el desarrollo del mismo.

## **6.7 Fundamentación**

### **6.7.1 PHP (Hypertext Pre-Processor).**

PHP, cuyas siglas responden a un acrónimo recursivo (PHP: hypertext preprocessor), es un lenguaje sencillo, de sintaxis cómoda y similar a la de otros lenguajes como Perl, C y C++. Es rápido, interpretado, orientado a objetos y multiplataforma. Para él se encuentra disponible una multitud de librerías. PHP es un lenguaje ideal tanto para aprender a desarrollar aplicaciones web como para desarrollar aplicaciones web complejas. PHP añade a todo eso la ventaja de que el

intérprete de PHP, los diversos módulos y gran cantidad de librerías desarrolladas para PHP son de código libre, con lo que el programador de PHP dispone de un impresionante arsenal de herramientas libres para desarrollar aplicaciones.

Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. Un sitio con páginas dinámicas es el que permite interactuar con el visitante, de modo que cada usuario que visita la página vea la información modificada para requisitos articulares. Las aplicaciones dinámicas para el Web son frecuentes en los sitios comerciales (e-commerce), donde el contenido visualizado se genera de la información alcanzada en una base de datos u otra fuente externa.

*PHP. (s.f.) Extraído el 10 de Agosto del 2011 desde <http://www.programacion.com/php>*

#### **6.7.1.1 Cómo funciona**

Para funcionar Php necesita su motor de scripting (script engine) que ejecuta las partes en código antes de que el servidor de la red envíe la página al usuario.

Si solicitamos a nuestro servidor una página PHP, éste envía dicha página al intérprete de PHP que la ejecuta (de hecho, no se trata más que de un programa) y devuelve el resultado (generalmente HTML) al servidor web, el cual, a su vez, se lo enviará al cliente

#### **6.7.1.2 Acceso a Base de Datos**

PHP proporciona métodos para acceder a un gran número de sistemas de bases de datos (mySQL, PostgreSQL, Oracle, etc.). Esta funcionalidad es imprescindible para el desarrollo de aplicaciones web complejas.

PHP también ofrece la integración con las varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa desde generar documentos en pdf hasta analizar código XML.

PHP ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil

de mantener y ponerse al día que el código comparables en otros lenguajes. Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores.

*Como Funciona – Acceso a Datos. (2011) Extraído el 10 de Agosto del 2011 desde <http://es.wikipedia.org/wiki/PHP>*

### **6.7.2 JavaScript**

Es un lenguaje interpretado orientado a las páginas web, con una sintaxis semejante a la del lenguaje Java. Se utiliza en páginas web HTML, para realizar tareas y operaciones en el marco de la aplicación cliente.

Muchos confunden el Javascript con el Java pero ambos lenguajes son diferentes y tienes sus características singulares. Javascript tiene la ventaja de ser incorporado en cualquier página web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado.

Java por su parte tiene como principal característica ser un lenguaje independiente de la plataforma. Se puede crear todo tipo de programa que puede ser ejecutado en cualquier ordenador del mercado: Linux, Windows, Apple, etc. Debido a sus características también es muy utilizado para internet.

JavaScript sirve principalmente para mejorar la gestión de la interfaz cliente/servidor. Un script JavaScript insertado en un documento HTML permite reconocer y tratar localmente, es decir, en el cliente, los eventos generados por el usuario. Estos eventos pueden ser el recorrido del propio documento HTML o la gestión de un formulario.

#### **6.7.2.1 Características de JavaScript**

- Es interpretado (no compilado) por el cliente, es decir, directamente del programa fuente se pasa a la ejecución de dicho programa, con lo que al contrario que los lenguajes compilados no se genera ni código objeto ni ejecutable para cada máquina en el que se quiera ejecutar dicho programa.

- Su código se integra en las páginas HTML, incluido en las propias páginas.
- No es necesario declarar los tipos de variables que van a utilizarse ya que como se verá más adelante, JavaScript realiza una conversión automática de tipos.
- Las referencias a objetos se comprueban en tiempo de ejecución. Esto es consecuencia de que JavaScript no es un lenguaje compilado.

*JavaScript. (s.f.) Extraído el 10 de Agosto del 2011 desde <http://www.efectosjavascript.com/javascript.html>*

### **6.7.3 EXTjs**

ExtJS es una librería Javascript que permite construir aplicaciones complejas en Internet. Esta librería incluye:

- Componentes UI del alto performance y personalizables.
- Modelo de componentes extensibles.
- Un API fácil de usar.
- Licencias Open source y comerciales.

Extjs comenzó como una extensión de la poderosa librería YUI (Yahoo User Interface). Pero Extjs no es otra librería JavaScript más, de hecho es un framework que además puede trabajar conjuntamente con otras librerías JavaScript utilizando adaptadores.

Extjs se debe utilizar en sitios web que requieran un alto nivel de interacción con el usuario, lo cual es algo más complejo que un típico sitio web. Un sitio web que requiera procesos y un flujo de trabajo sería un ejemplo perfecto.

Extjs contribuye al desarrollo de aplicaciones en los siguientes aspectos:

- Provee compatibilidad con todos los browsers que es transparente al desarrollador.
- Hace posible implementar complejos componentes en una página web.

- Ofrece comunicación optimizada con el servidor sin tener que hacer recarga de la página. Es decir implementa Ajax en tiempo real.
- Nos permite elegir entre desarrollo tradicional y el desarrollo utilizando Ajax
- Interfaces manejadas a través de eventos.

*EXTjs. (s.f.) Extraído el 10 de Agosto del 2011 desde [http://www.ecured.cu/index.php/Sencha\\_Ext\\_JS](http://www.ecured.cu/index.php/Sencha_Ext_JS)*

### **6.7.3.1 AJAX**

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

*AJAX. (2011) Extraído el 10 de Agosto del 2011 desde <http://es.wikipedia.org/wiki/AJAX>*

### **6.7.3.2 AJAX Y JSON**

Ajax obtuvo parte de su nombre por su interacción inicial con XML. Pero poco a poco la comunidad de desarrolladores se fue dando cuenta que se debía agilizar el transporte de datos con algo diferente y más óptimo que XML. Es así que se llegó a crear JSON que significa Javascript Object Notation. JSON permite el transporte de datos desde y hacia el cliente liberando a las aplicaciones de las etiquetas y encabezados propios del XML. Otra ventaja añadida es que JSON se traduce directamente a objetos de javascript, lo cual mejora los tiempos de respuesta y procesamiento

*AJAX Y JSON. (s.f) Extraído el 10 de Agosto del 2011 desde <http://www.sindikos.com/tag/Json>*

## 6.8 Metodología

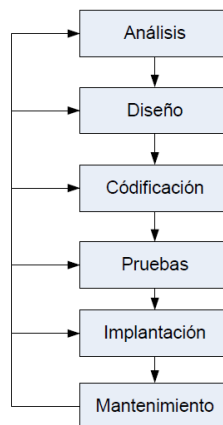
El proyecto se desarrolló con la metodología de desarrollo denominada **modelo en cascada**, la misma que consiste en el desarrollo de software en fases.

Se utiliza en los proyectos donde se conocen exactamente todos los requerimientos. Consiste en pasar por cada una de las etapas (requerimientos, análisis, diseño, implementación, pruebas) de desarrollo, pero no al mismo tiempo, sólo cuando definitivamente termine la etapa de requerimientos puede seguir a la de análisis y sucede de igual forma para el resto de las etapas.

### Ventajas de la Metodología Modelo de Cascada:

- Modelo y planificación fácil y sencillos
- La calidad del producto resultante es alta.
- Produce mejores resultados en proyectos donde los requerimientos del software son claramente entendidos.

### Estructura del modelo en cascada



**Figura 6.1** Estructura del modelo en cascada

Para el análisis del sistema se utilizó la herramienta UML (Lenguaje Unificado de Modelado) el cual es un conjunto de notificaciones y diagramas gráficos para modelar sistemas orientados a objetos; dando diferentes perspectivas a un sistema y logrando especificar, visualizar, construir y documentar un sistema de software.

## **6.9 Modelo operativo**

### **6.9.1 Análisis del Sistema**

#### **6.9.1.1 Análisis y requerimientos del sistema**

El departamento de cobranzas, no cuenta con un mecanismo de control de información de la contribución vial para la provincia, y mucho menos de los contribuyentes y sus aportaciones económicas, por lo cual se decide implantar un sistema web que facilite dicho proceso.

Luego de haber analizado los requerimientos que solicita el departamento de cobranzas se determinó que el sistema debe contar con:

- Un módulo para el ingreso, modificación, eliminación y control de: comprobantes de pago, contribuyentes, vehículos y usuarios del sistema. Recordando que en un comprobante se emite los datos del vehiculo y del contribuyente, así como también año al que corresponde la contribución, hora y fecha en la que se realizó el pago, y el valor recaudado.
- Un módulo que administre el personal que trabaja en el departamento financiero, así mismo como los cargos que ejercen.
- Un patrón que gestione el control de los años no cancelados por los contribuyentes.
- Control de la generación e impresión de los comprobantes de pago.
- Parametrización del tipo de facturación.
- Parametrización del costo para un rango de tonelaje
- Asignación de boletines en rangos parametrizables, a recaudadores del departamento financiero.
- Generación de reportes de: cierre de caja diarios, consulta de comprobantes por fechas, consulta de pagos por placa, consulta de usuarios, departamentos, cargos, vehículos.

- Además, el sistema debe contar con interfaces sencillas y amigables para su manipulación.

El Gobierno Provincial de Tungurahua cuenta con un Sitio Web en el cual se implantó e integró el sistema web desarrollado, cuenta con una base de datos PostgreSQL con tamaño suficiente para almacenar la información del sistema web sin generar ningún conflicto a la hora de la manipulación, extracción y almacenamiento de los datos.

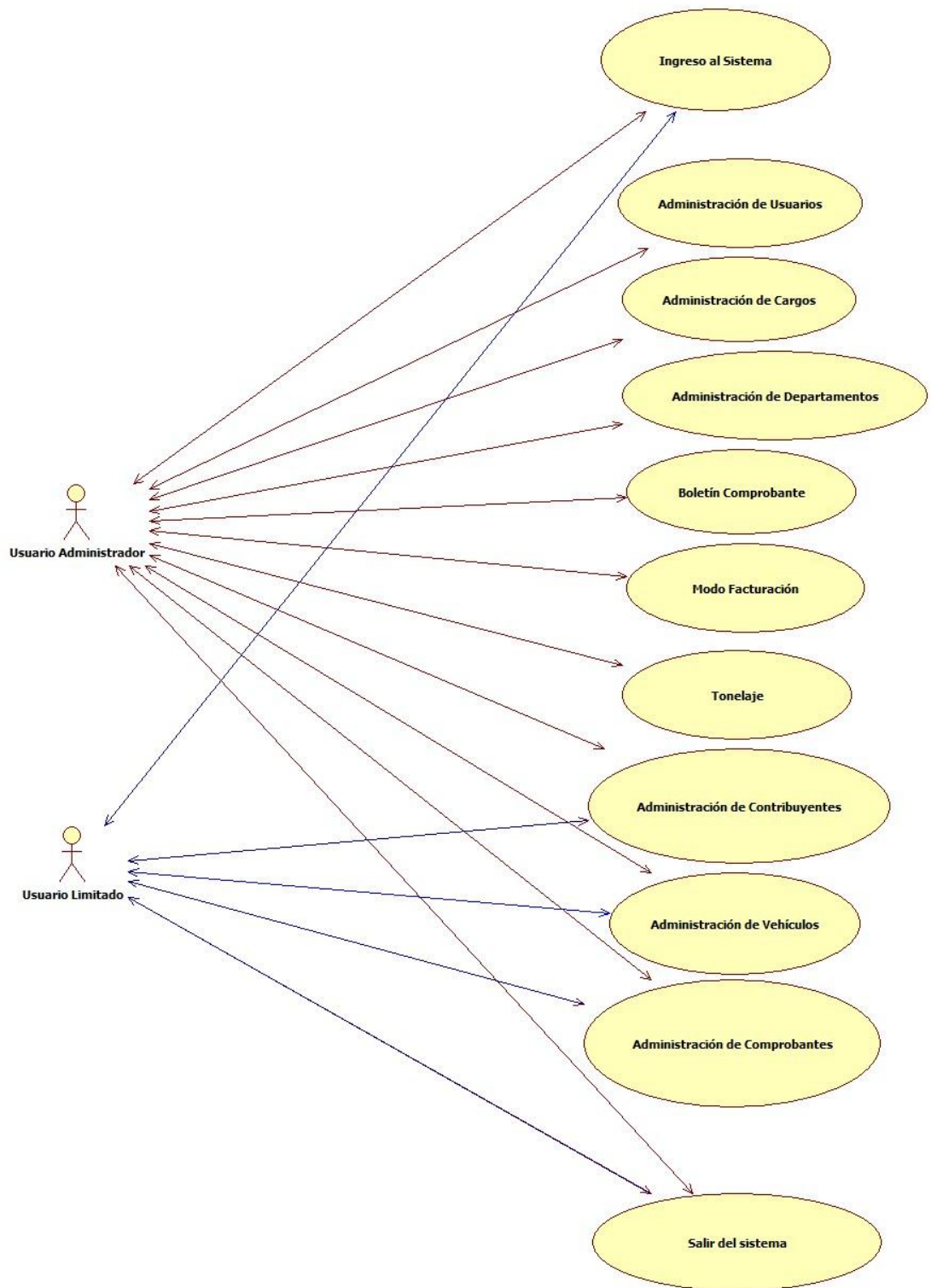
### **6.9.1.2 Diagramas UML**

UML (Lenguaje unificado de modelado) está consolidado como el lenguaje estándar en el análisis y diseño de software; facilita varios tipos de diagramas, los que nos permiten describir los requisitos, funcionalidad, y otros conceptos relativos al proyecto de desarrollo del sistema web, por esto se procedió a utilizarlo previo al proceso intensivo de escribir código.

#### **6.9.1.2.1 Diagrama de casos de uso**

Es una técnica para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje, es una técnica para captura de requisitos. Describe cómo se usa el sistema, partiendo desde el punto de vista del usuario final. Esto da una buena pauta para conocer más a fondo los requisitos que deberá tener el sistema a desarrollar.





**Figura 6.2.** Diagrama de casos de uso Sistema de Gestión de Cobranza

## Especificación de casos de uso

	<b>Caso de uso: Ingresar al sistema</b>
<b>Actores:</b>	Usuario administrador Usuario limitado
<b>Descripción:</b>	Validar a un usuario ya registrado para el uso del sistema.
<b>Precondiciones:</b>	El usuario debe estar registrado en el sistema
<b>Pos condiciones</b>	Validación realizada con éxito
<b>Flujo normal de eventos:</b>	<b>Validar</b> <ol style="list-style-type: none"> <li>1. El usuario ingresa el nombre de usuario y contraseña</li> <li>2. El sistema valida los datos ingresados por el usuario.</li> <li>3. Una vez validado el usuario, el sistema muestra el menú de opciones</li> </ol>
<b>Flujos alternos:</b>	<b>usuario no registrado</b> <ol style="list-style-type: none"> <li>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.</li> </ol>

**Tabla 6.1.** Especificación ingresar al sistema

	<b>Caso de uso: Administración de Usuarios</b>
<b>Actores:</b>	Usuario administrador
<b>Descripción:</b>	Añadir, modificar, eliminar usuarios del sistema
<b>Precondiciones:</b>	El usuario debe estar registrado como administrador
<b>Pos condiciones</b>	Resultados almacenados con éxito
<b>Flujo normal de eventos:</b>	<b>Nuevo</b> <ol style="list-style-type: none"> <li>1. El usuario escoge la opción de Administrar usuarios.</li> <li>2. El sistema muestra las opciones disponibles Nuevo, Guardar, Modificar, Eliminar, Cancelar, Añadir cargo.</li> <li>3. El usuario escoge la opción Nuevo.</li> <li>4. El sistema muestra los campos para ingresar la información de los usuarios.</li> <li>5. El usuario ingresa la información.</li> <li>6. El sistema verifica el ingreso correcto de información.</li> </ol>
<b>Flujos alternos:</b>	<b>Modificar</b> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción modificar.</li> </ol>

	<ol style="list-style-type: none"> <li>2. El sistema muestra los campos con información del usuario a modificar.</li> <li>3. El sistema muestra un mensaje donde se indica si el usuario está seguro de la modificación.</li> <li>4. El usuario modifica la información.</li> <li>5. El sistema verifica el ingreso correcto de información.</li> </ol> <p><b>Eliminar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción eliminar.</li> <li>2. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación</li> <li>3. El usuario toma una decisión.</li> <li>4. El sistema realiza la operación indicada.</li> </ol> <p><b>Cancelar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción cancelar.</li> <li>2. El sistema realiza la operación indicada.</li> </ol> <p><b>Añadir cargo</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción añadir cargo.</li> <li>2. El sistema muestra un formulario para agregar un nuevo cargo.</li> </ol>
--	---

**Tabla 6.2** Especificación administración de usuarios

<b>Caso de uso: Administración de Cargos</b>	
<b>Actores:</b>	Usuario administrador
<b>Descripción:</b>	Añadir, modificar, eliminar cargos
<b>Precondiciones:</b>	El usuario debe estar registrado como administrador
<b>Pos condiciones</b>	Resultados almacenados con éxito
<b>Flujo normal de eventos:</b>	<p><b>Nuevo</b></p> <ol style="list-style-type: none"> <li>1. El usuario escoge la opción de cargos.</li> <li>2. El sistema muestra las opciones disponibles Nuevo, Guardar, Modificar, Eliminar, Cancelar, Añadir Departamento.</li> <li>3. El usuario escoge la opción Nuevo.</li> <li>4. El sistema muestra los campos para ingresar la información del cargo.</li> <li>5. El usuario ingresa la información.</li> </ol>

	6. El sistema verifica el ingreso correcto de información.
<b>Flujos alternos:</b>	<p><b>Modificar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción modificar.</li> <li>2. El sistema muestra los campos con información del cargo a modificar.</li> <li>3. El sistema muestra un mensaje donde se indica si el usuario está seguro de la modificación.</li> <li>4. El usuario modifica la información.</li> <li>5. El sistema verifica el ingreso correcto de información.</li> </ol> <p><b>Eliminar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción eliminar.</li> <li>2. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación</li> <li>3. El usuario toma una decisión.</li> <li>4. El sistema realiza la operación indicada.</li> </ol> <p><b>Cancelar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción cancelar.</li> <li>2. El sistema realiza la operación indicada.</li> </ol> <p><b>Añadir departamento</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción añadir departamento.</li> <li>2. El sistema muestra un formulario para agregar un nuevo departamento.</li> </ol>

**Tabla 6.3** Especificación administración de cargos

	<b>Caso de uso: Administración de Departamentos</b>
<b>Actores:</b>	Usuario administrador
<b>Descripción:</b>	Añadir, modificar, eliminar departamentos
<b>Precondiciones:</b>	El usuario debe estar registrado como administrador
<b>Pos condiciones</b>	Resultados almacenados con éxito
<b>Flujo normal de eventos:</b>	<p><b>Nuevo</b></p> <ol style="list-style-type: none"> <li>1. El usuario escoge la opción de departamentos.</li> <li>2. El sistema muestra las opciones disponibles Nuevo, Guardar, Modificar, Eliminar, Cancelar.</li> <li>3. El usuario escoge la opción Nuevo.</li> </ol>

	<ol style="list-style-type: none"> <li>4. El sistema muestra los campos para ingresar la información del departamento.</li> <li>5. El usuario ingresa la información.</li> <li>6. El sistema verifica el ingreso correcto de información.</li> </ol>
<b>Flujos alternos:</b>	<p><b>Modificar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción modificar.</li> <li>2. El sistema muestra los campos con información del departamento a modificar.</li> <li>3. El sistema muestra un mensaje donde se indica si el usuario está seguro de la modificación.</li> <li>4. El usuario modifica la información.</li> <li>5. El sistema verifica el ingreso correcto de información.</li> </ol> <p><b>Eliminar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción eliminar.</li> <li>2. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación</li> <li>3. El usuario toma una decisión.</li> <li>4. El sistema realiza la operación indicada.</li> </ol> <p><b>Cancelar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción cancelar.</li> <li>2. El sistema realiza la operación indicada.</li> </ol>

**Tabla 6.4** Especificación administración de departamentos

	<b>Caso de uso: Boletín Comprobante</b>
<b>Actores:</b>	Usuario administrador
<b>Descripción:</b>	Añadir, modificar Boletines
<b>Precondiciones:</b>	El usuario debe estar registrado como administrador
<b>Pos condiciones</b>	Resultados almacenados con éxito
<b>Flujo normal de eventos:</b>	<p><b>Nuevo</b></p> <ol style="list-style-type: none"> <li>1. El usuario escoge la opción Asignar Boletín.</li> <li>2. El sistema muestra las opciones disponibles Nuevo, Guardar, Modificar, Cancelar.</li> <li>3. El usuario escoge la opción Nuevo.</li> <li>4. El sistema muestra los campos para ingresar el rango de los boletines y el usuario al que se le</li> </ol>

	<p>asignará el mismo.</p> <ol style="list-style-type: none"> <li>5. El usuario ingresa la información.</li> <li>6. El sistema verifica el ingreso correcto de información.</li> </ol>
<b>Flujos alternos:</b>	<p><b>Modificar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción modificar.</li> <li>2. El sistema muestra los campos con información del boletín a modificar.</li> <li>3. El sistema muestra un mensaje donde se indica si el usuario está seguro de la modificación.</li> <li>4. El usuario modifica la información.</li> <li>5. El sistema verifica el ingreso correcto de información.</li> </ol> <p><b>Cancelar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción cancelar.</li> <li>2. El sistema realiza la operación indicada.</li> </ol>

**Tabla 6.5** Especificación Boletín Comprobante

	<b>Caso de uso: Modo Facturación</b>
<b>Actores:</b>	Usuario administrador
<b>Descripción:</b>	Cambiar Modo Facturación
<b>Precondiciones:</b>	El usuario debe estar registrado como administrador
<b>Pos condiciones</b>	Resultados almacenados con éxito
<b>Flujo normal de eventos:</b>	<p><b>Nuevo</b></p> <ol style="list-style-type: none"> <li>7. El usuario escoge la opción Elegir Tipo Facturación.</li> <li>8. El sistema muestra las opciones disponibles Cambiar, Guardar, Cancelar.</li> <li>9. El usuario escoge la opción Cambiar.</li> <li>10. El sistema muestra el tipo facturación con el que trabaja actualmente el sistema, presenta la opción de escoger el otro tipo de facturación.</li> <li>11. El usuario escoge el tipo de facturación.</li> <li>12. El sistema verifica el ingreso correcto de información.</li> </ol>

<b>Flujos alternos:</b>	<p><b>Cancelar</b></p> <ol style="list-style-type: none"> <li>3. En el paso 3 del Flujo Normal, si el usuario escoge la opción cancelar.</li> <li>4. El sistema realiza la operación indicada.</li> </ol>
-------------------------	---

**Tabla 6.6** Especificación Modo Facturación

	<b>Caso de uso: Tonelaje</b>
<b>Actores:</b>	Usuario administrador
<b>Descripción:</b>	Añadir, modificar, eliminar contribuyentes
<b>Precondiciones:</b>	El usuario debe estar registrado como administrador
<b>Pos condiciones</b>	Resultados almacenados con éxito
<b>Flujo normal de eventos:</b>	<p><b>Nuevo</b></p> <ol style="list-style-type: none"> <li>1. El usuario escoge la opción Parametros Tonelaje.</li> <li>2. El sistema muestra las opciones disponibles Nuevo, Guardar, Modificar, Eliminar, Cancelar.</li> <li>3. El usuario escoge la opción Nuevo.</li> <li>4. El sistema muestra los campos para ingresar la el rango de valores de tonelaje y el valor que tendrá el mismo.</li> <li>5. El usuario ingresa la información.</li> <li>6. El sistema verifica el ingreso correcto de información.</li> </ol>
<b>Flujos alternos:</b>	<p><b>Modificar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción modificar.</li> <li>2. El sistema muestra los campos con información del rango de tonelaje a modificar.</li> <li>3. El sistema muestra un mensaje donde se indica si el usuario está seguro de la modificación.</li> <li>4. El usuario modifica la información.</li> <li>5. El sistema verifica el ingreso correcto de información.</li> </ol> <p><b>Eliminar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción eliminar.</li> <li>2. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación</li> <li>3. El usuario toma una decisión.</li> <li>4. El sistema realiza la operación indicada.</li> </ol> <p><b>Cancelar</b></p>

	<ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción cancelar.</li> <li>2. El sistema realiza la operación indicada.</li> </ol>
--	--

**Tabla 6.7** Especificación Tonelaje

	<b>Caso de uso: Administración contribuyentes</b>
<b>Actores:</b>	Usuario administrador Usuario limitado
<b>Descripción:</b>	Añadir, modificar, eliminar contribuyentes
<b>Precondiciones:</b>	El usuario debe estar registrado
<b>Pos condiciones</b>	Resultados almacenados con éxito
<b>Flujo normal de eventos:</b>	<p><b>Nuevo</b></p> <ol style="list-style-type: none"> <li>1. El usuario escoge la opción de contribuyentes.</li> <li>2. El sistema muestra las opciones disponibles Nuevo, Guardar, Modificar, Eliminar, Cancelar.</li> <li>3. El usuario escoge la opción Nuevo.</li> <li>4. El sistema muestra los campos para ingresar la información del contribuyente.</li> <li>5. El usuario ingresa la información.</li> <li>6. El sistema verifica el ingreso correcto de información.</li> </ol>
<b>Flujos alternos:</b>	<p><b>Modificar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción modificar.</li> <li>2. El sistema muestra los campos con información del contribuyente a modificar.</li> <li>3. El sistema muestra un mensaje donde se indica si el usuario está seguro de la modificación.</li> <li>4. El usuario modifica la información.</li> <li>5. El sistema verifica el ingreso correcto de información.</li> </ol> <p><b>Eliminar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción eliminar.</li> <li>2. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación</li> <li>3. El usuario toma una decisión.</li> <li>4. El sistema realiza la operación indicada.</li> </ol>



	<p><b>Cancelar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción cancelar.</li> <li>2. El sistema realiza la operación indicada.</li> </ol>
--	---

**Tabla 6.8** Especificación Administración de Contribuyentes

	<b>Caso de uso: Administración de vehículos</b>
<b>Actores:</b>	Usuario administrador Usuario limitado
<b>Descripción:</b>	Añadir, modificar, eliminar vehículos
<b>Precondiciones:</b>	El usuario debe estar registrado
<b>Pos condiciones</b>	Resultados almacenados con éxito
<b>Flujo normal de eventos:</b>	<p><b>Nuevo</b></p> <ol style="list-style-type: none"> <li>1. El usuario escoge la opción de vehículos.</li> <li>2. El sistema muestra las opciones disponibles Nuevo, Guardar, Modificar, Eliminar, Cancelar, Añadir Contribuyente.</li> <li>3. El usuario escoge la opción Nuevo.</li> <li>4. El sistema muestra los campos para ingresar la información del vehículo.</li> <li>5. El usuario ingresa la información.</li> <li>6. El sistema verifica el ingreso correcto de información.</li> </ol>
<b>Flujos alternos:</b>	<p><b>Modificar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción modificar.</li> <li>2. El sistema muestra los campos con información del vehículo a modificar.</li> <li>3. El sistema muestra un mensaje donde se indica si el usuario está seguro de la modificación.</li> <li>4. El usuario modifica la información.</li> <li>5. El sistema verifica el ingreso correcto de información.</li> </ol> <p><b>Eliminar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción eliminar.</li> <li>2. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación</li> <li>3. El usuario toma una decisión.</li> <li>4. El sistema realiza la operación indicada.</li> </ol> <p><b>Cancelar</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción cancelar.</li> </ol>

	<p>2. El sistema realiza la operación indicada.</p> <p><b>Añadir Contribuyente</b></p> <ol style="list-style-type: none"> <li>1. En el paso 3 del Flujo Normal, si el usuario escoge la opción añadir contribuyente.</li> <li>2. El sistema muestra un formulario para agregar un nuevo contribuyente.</li> </ol>
--	---

**Tabla 6.9** Especificación administración de vehículos

	<b>Caso de uso: Administración de comprobantes</b>
<b>Actores:</b>	<p>Usuario administrador</p> <p>Usuario limitado</p>
<b>Descripción:</b>	Añadir, modificar, eliminar comprobantes
<b>Precondiciones:</b>	<p>El usuario debe estar registrado.</p> <p>El sistema debe tener asignado usuarios recaudadores.</p> <p>El sistema debe tener configurado un parámetro de tonelaje</p>
<b>Pos condiciones</b>	Resultados almacenados con éxito
<b>Flujo normal de eventos:</b>	<p><b>Nuevo</b></p> <ol style="list-style-type: none"> <li>1. El usuario escoge la opción de comprobantes.</li> <li>2. El sistema muestra las opciones disponibles Nuevo, Guardar, Modificar, Eliminar, Cancelar, Añadir Vehículo.</li> <li>3. El usuario escoge la opción Nuevo.</li> <li>4. El sistema muestra los campos para ingresar la información del comprobante.</li> <li>5. El usuario ingresa la información.</li> <li>6. El sistema verifica el ingreso correcto de información.</li> </ol>
<b>Flujos alternos:</b>	<p><b>Modificar</b></p> <ol style="list-style-type: none"> <li>6. En el paso 3 del Flujo Normal, si el usuario escoge la opción modificar.</li> <li>7. El sistema muestra los campos con información del comprobante a modificar.</li> <li>8. El sistema muestra un mensaje donde se indica si el usuario está seguro de la modificación.</li> <li>9. El usuario modifica la información.</li> <li>10. El sistema verifica el ingreso correcto de información.</li> </ol> <p><b>Eliminar</b></p> <ol style="list-style-type: none"> <li>5. En el paso 3 del Flujo Normal, si el usuario escoge la opción eliminar.</li> <li>6. El sistema muestra un mensaje donde se indica</li> </ol>

	<p>si el usuario está seguro de la eliminación</p> <ol style="list-style-type: none"> <li>7. El usuario toma una decisión.</li> <li>8. El sistema realiza la operación indicada.</li> </ol> <p><b>Cancelar</b></p> <ol style="list-style-type: none"> <li>3. En el paso 3 del Flujo Normal, si el usuario escoge la opción cancelar.</li> <li>4. El sistema realiza la operación indicada.</li> </ol> <p><b>Añadir Vehículo</b></p> <ol style="list-style-type: none"> <li>7. En el paso 3 del Flujo Normal, si el usuario escoge la opción añadir vehículo.</li> <li>8. El sistema muestra un formulario para agregar un nuevo vehículo.</li> </ol>
--	--

**Tabla 6.10** Especificación administración de comprobantes

	<b>Caso de uso: Salir sistema</b>
<b>Actores:</b>	Usuario administrador Usuario limitado
<b>Descripción:</b>	El usuario sale del sistema.
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• El usuario debe estar registrado.</li> <li>• El usuario debe estar con la sesión iniciada en el sistema.</li> </ul>
<b>Pos condiciones</b>	Operación realizada con éxito.
<b>Flujo normal de eventos:</b>	<p><b>Validar</b></p> <ol style="list-style-type: none"> <li>1. El usuario escoge la opción cerrar sesión.</li> <li>2. El sistema realiza la operación indicada.</li> </ol>

**Tabla 6.11** Especificación Salir del Sistema

#### 6.9.1.2.2 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.

## Tipo Facturación

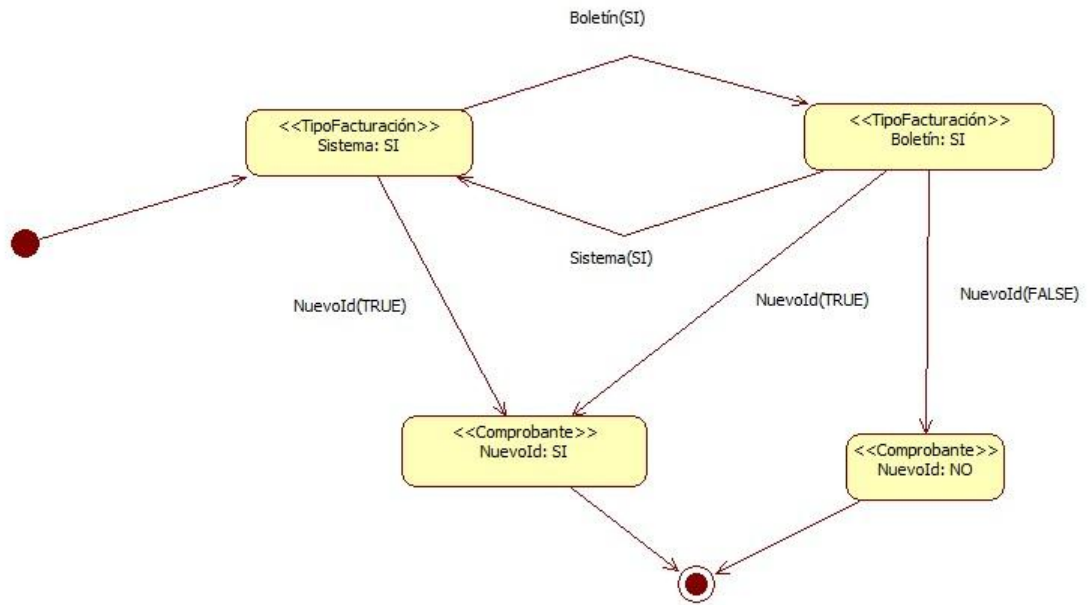


Figura 6.3 Diagrama de estados tipo facturación

## Tonelaje

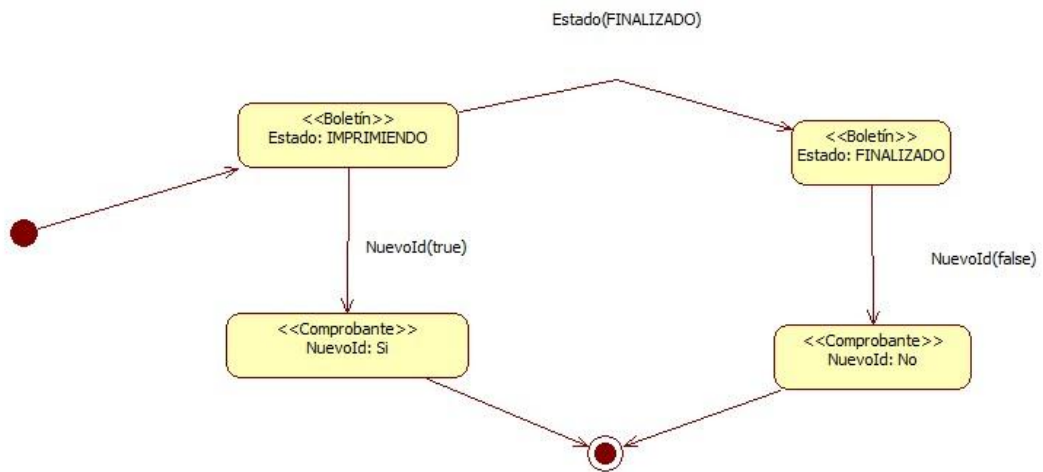
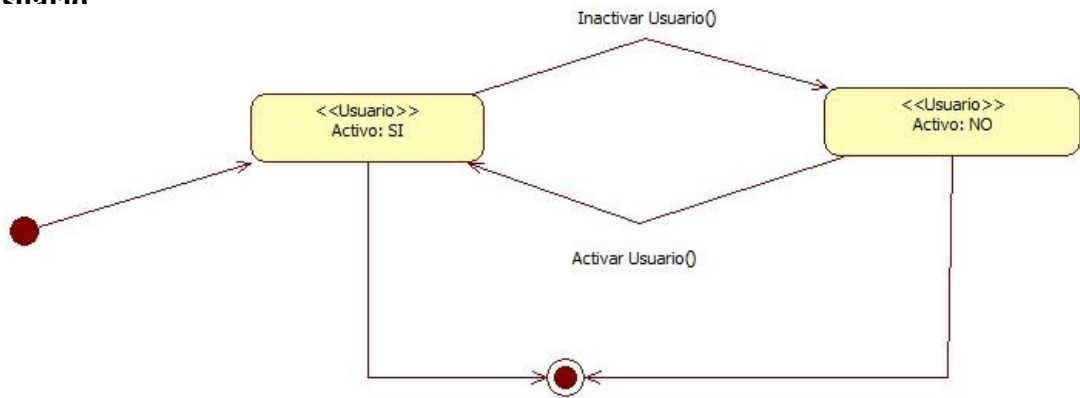


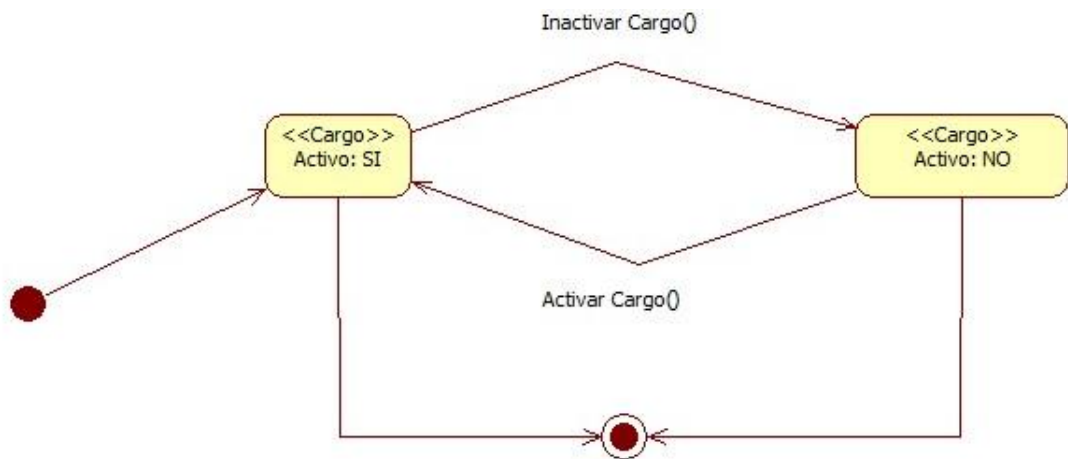
Figura 6.4 Diagrama de estados tonelaje

**Usuario**



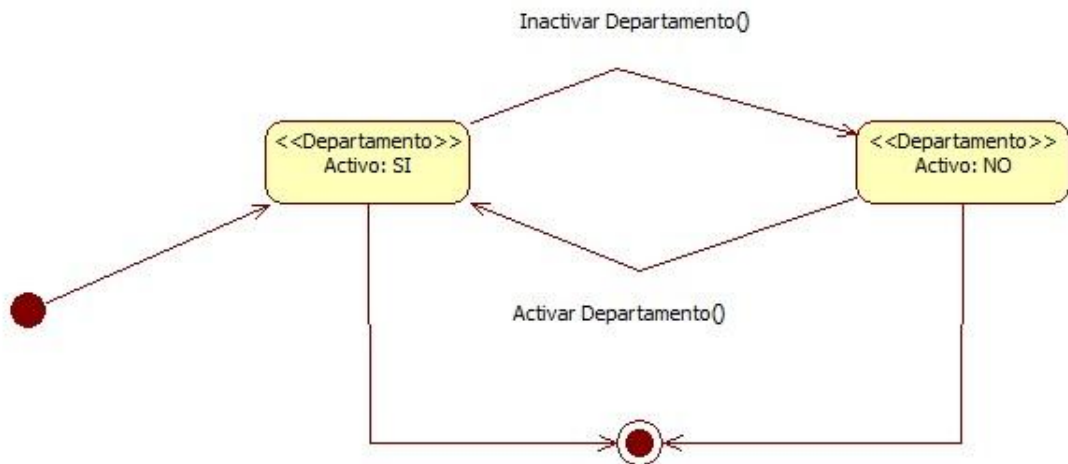
**Figura 6.5** Diagrama de estados usuarios

**Cargo**



**Figura 6.6** Diagrama de estado cargos

**Departamento**



**Figura 6.7** Diagrama de estado departamento

## Contribuyente

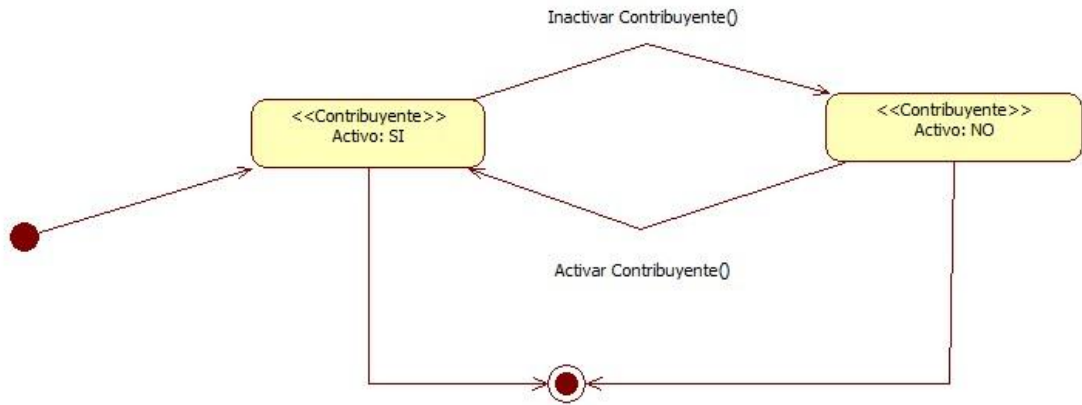


Figura 6.8 Diagrama de estado contribuyente

## Vehículo

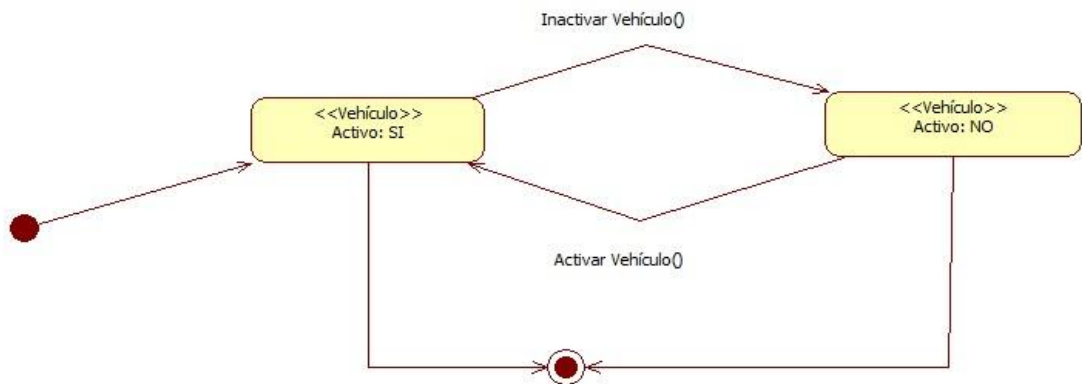


Figura 6.9 Diagrama de estado vehículo

## Comprobante

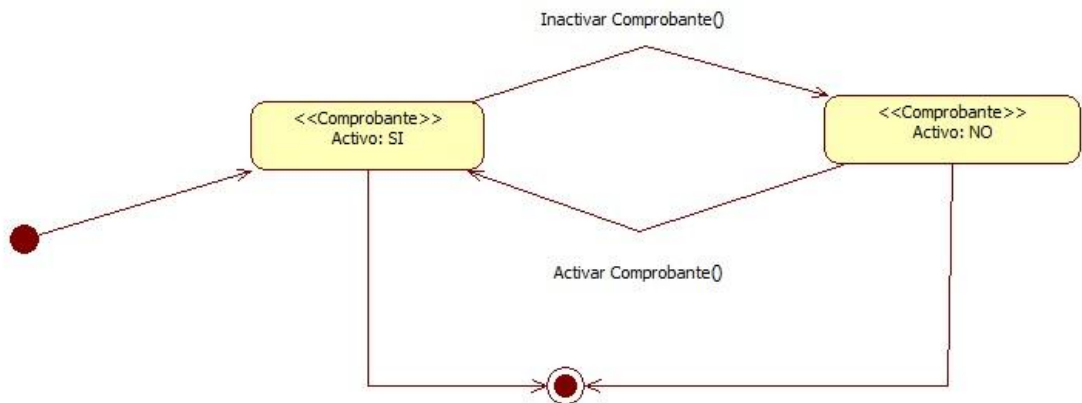


Figura 6.10 Diagrama de estado comprobante

### 6.9.1.2.3 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.

#### Ingresar al sistema

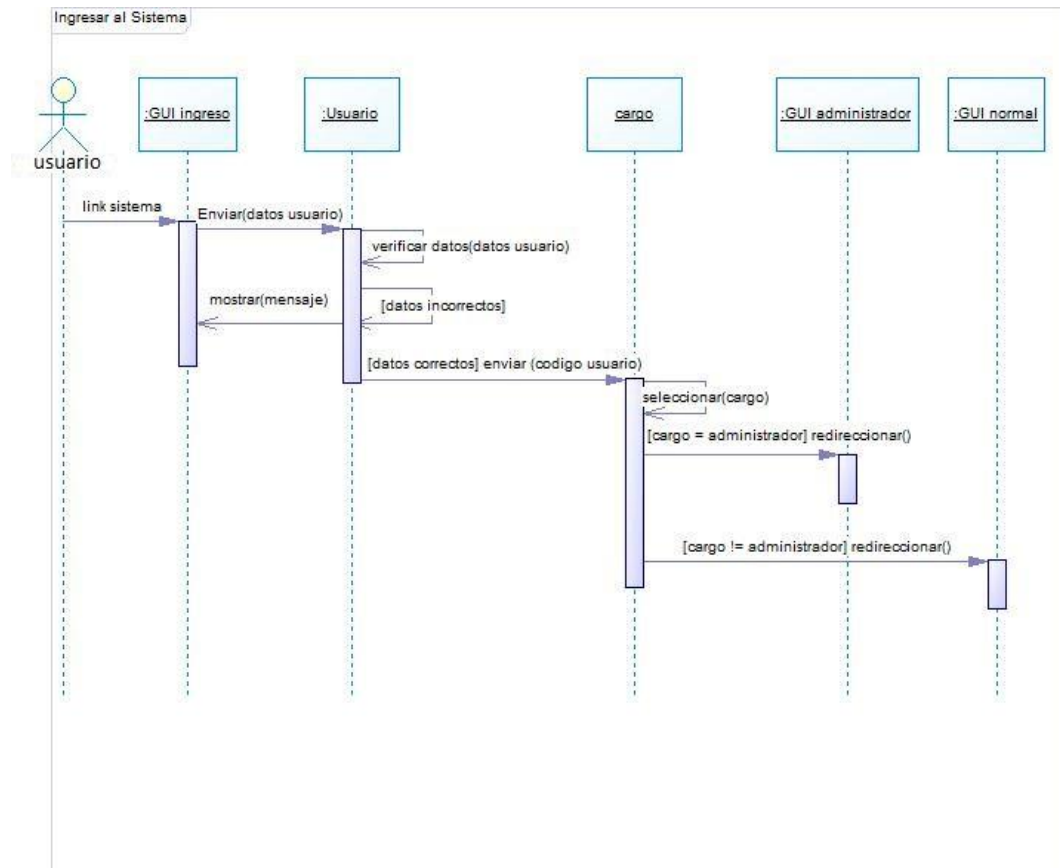


Figura 6.11 Diagrama de secuencia ingreso al sistema

## Administración de Usuarios

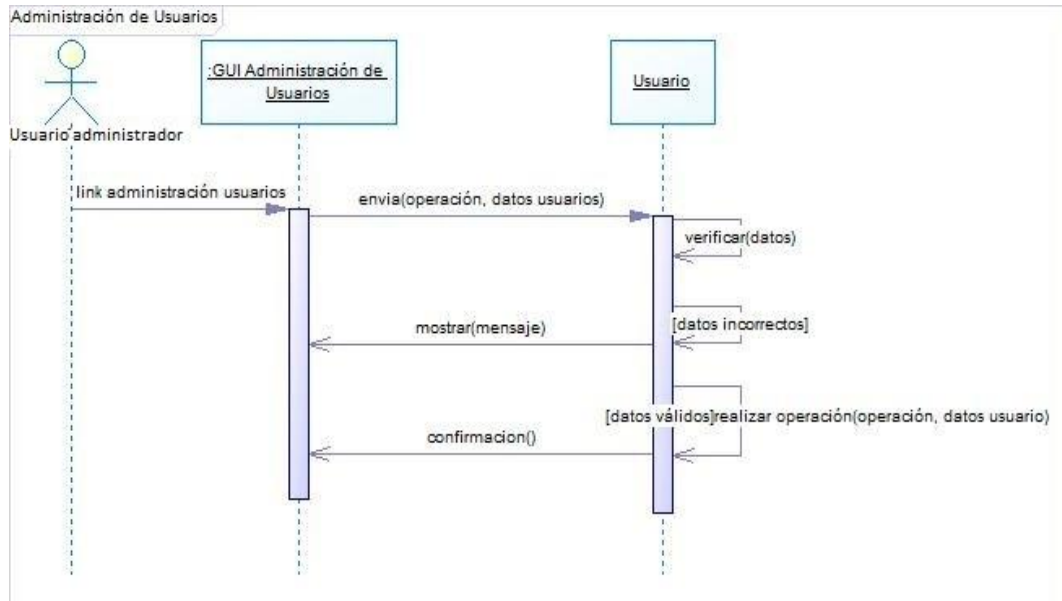


Figura 6.12 Diagrama de secuencia administración usuarios

## Cargos

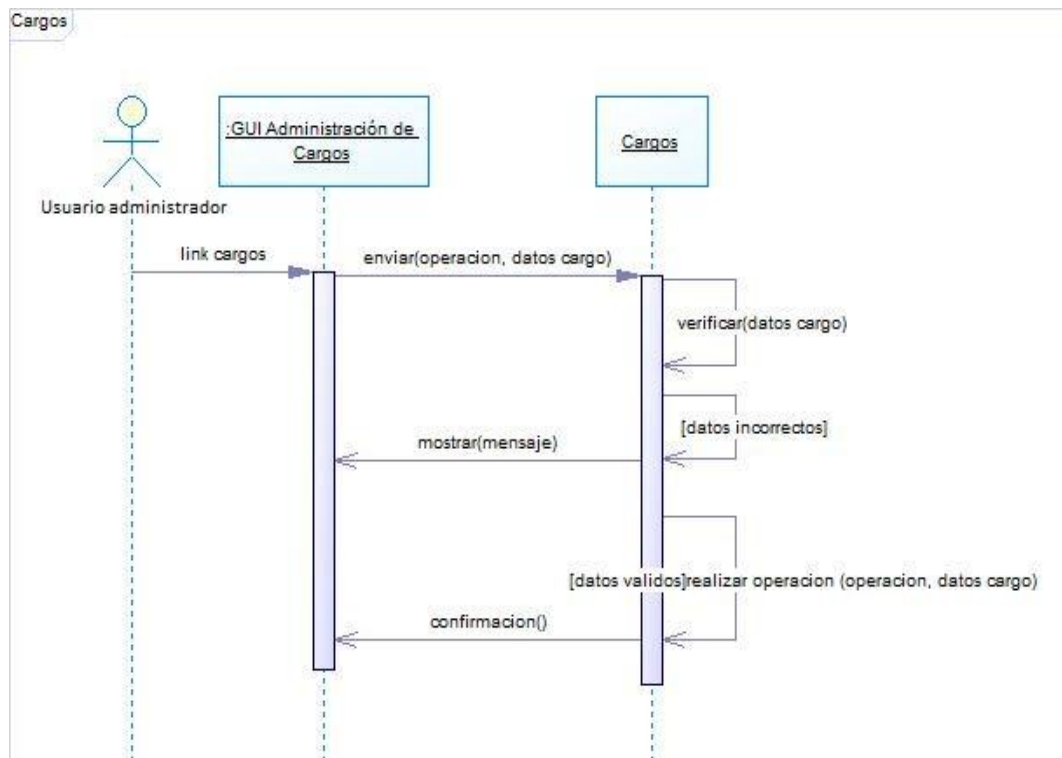


Figura 6.13 Diagrama de secuencia administración de cargos



## Departamentos

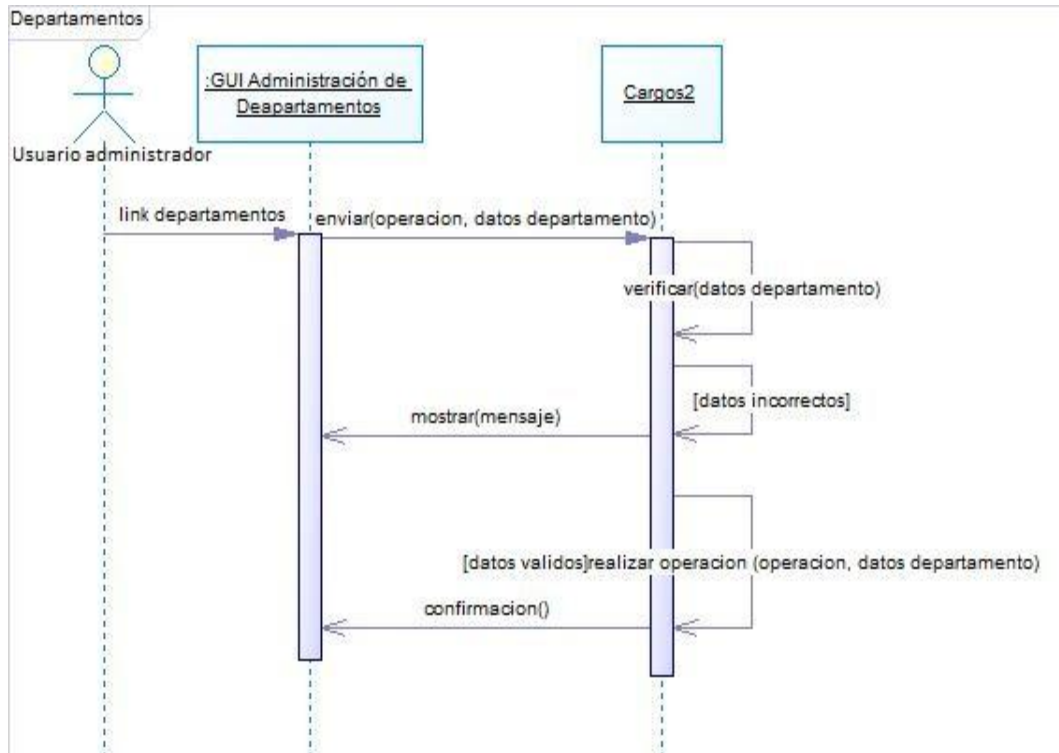


Figura 6.14 Diagrama de secuencia administración de departamentos

## Administración de contribuyentes

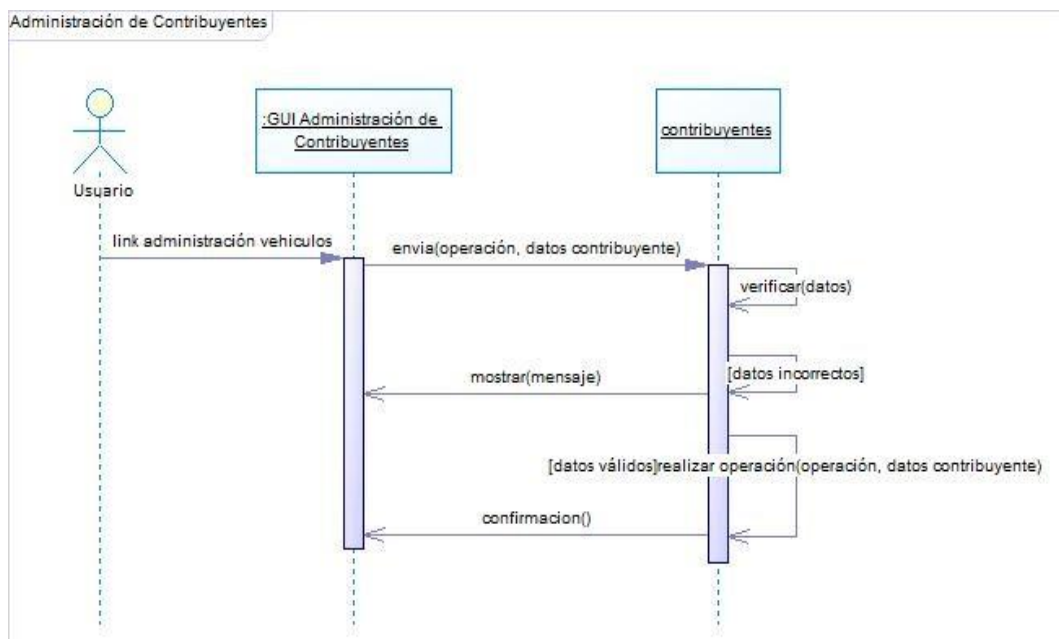


Figura 6.15 Diagrama de secuencia administración de contribuyentes

## Administración de vehículos

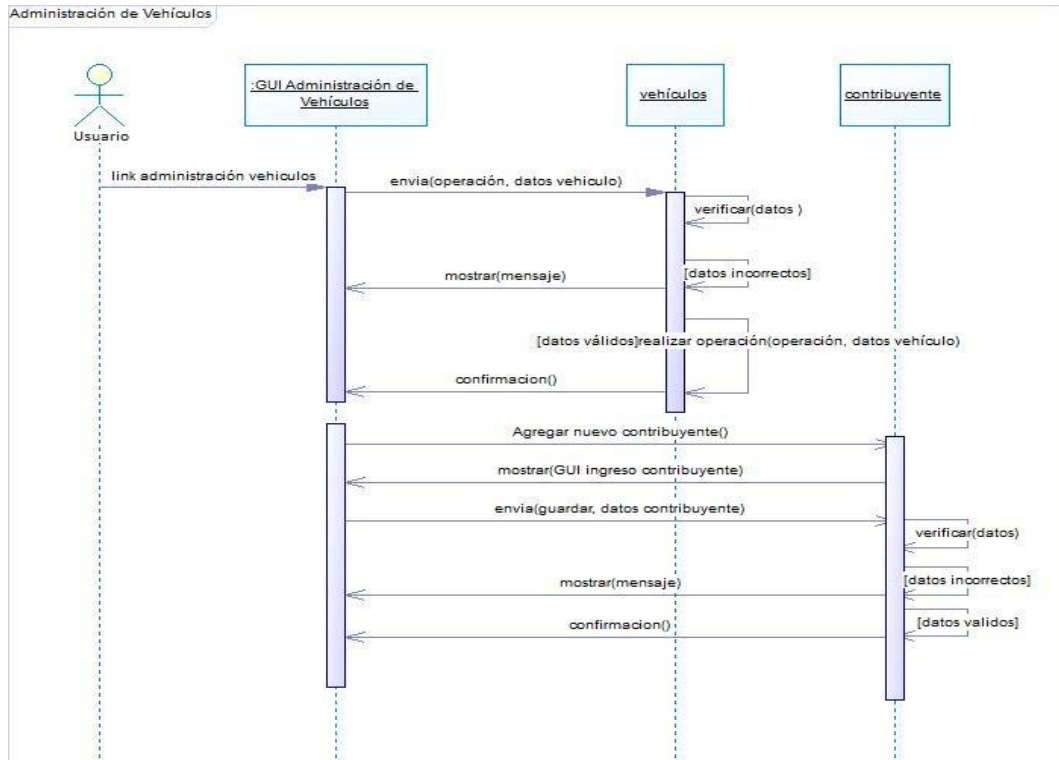


Figura 6.16 Diagrama de secuencia administración de vehículos

## Administración de Comprobantes

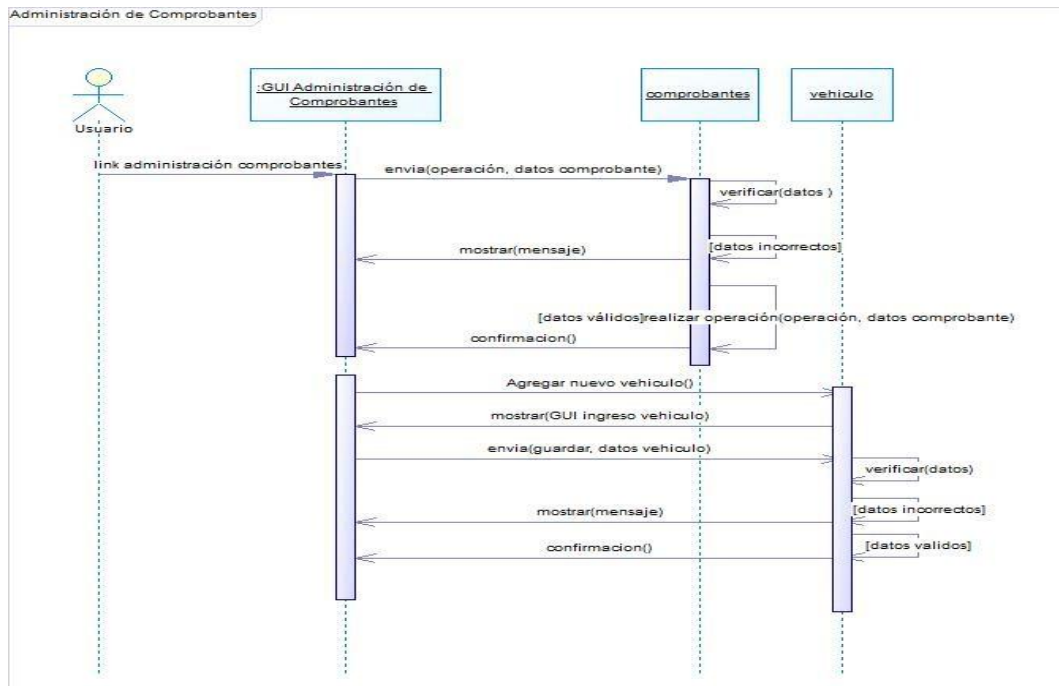


Figura 6.17 Diagrama de secuencia administración de comprobantes

## Boletín Comprobantes

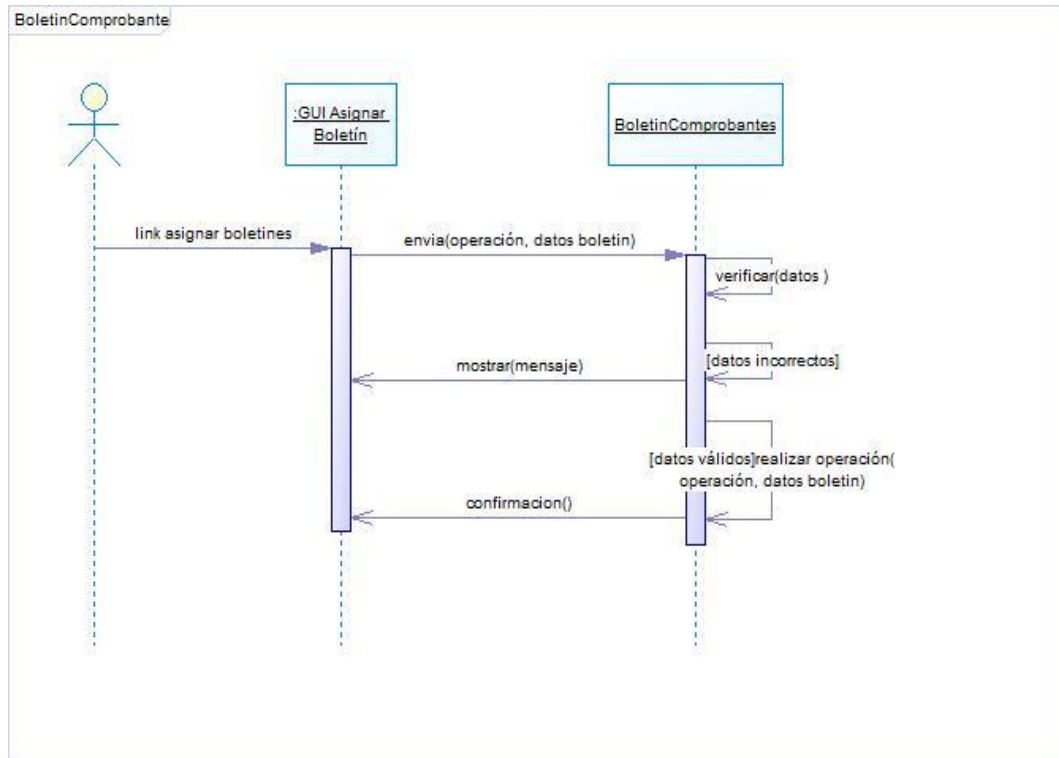


Figura 6.18 Diagrama de secuencia Boletín Comprobante

## Modo Facturación

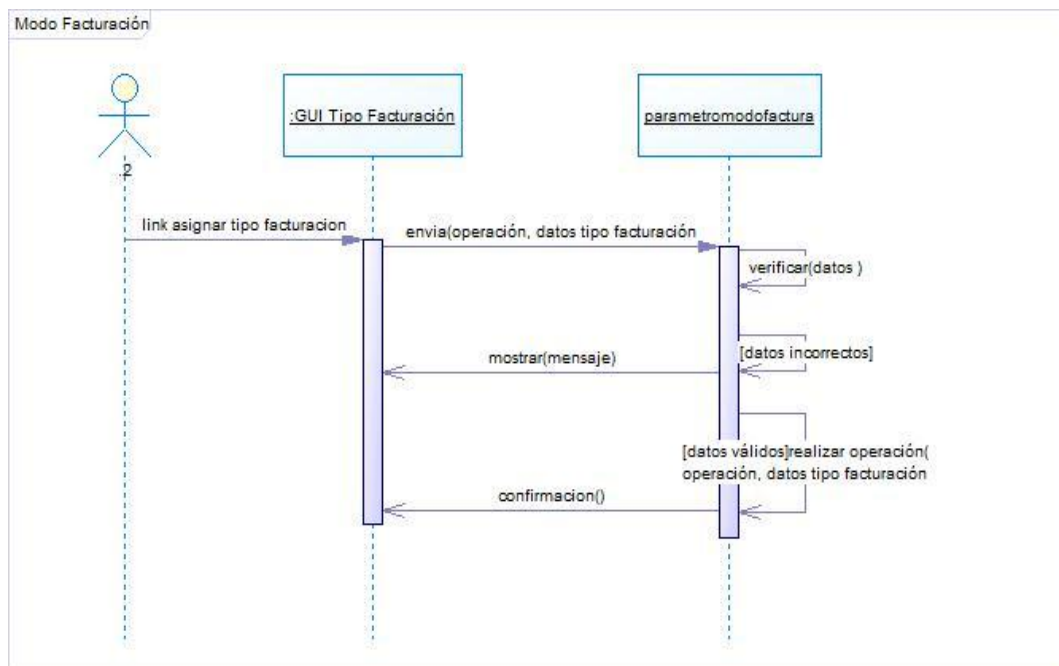


Figura 6.19 Diagrama de secuencia Modo Facturación

## Tonelaje

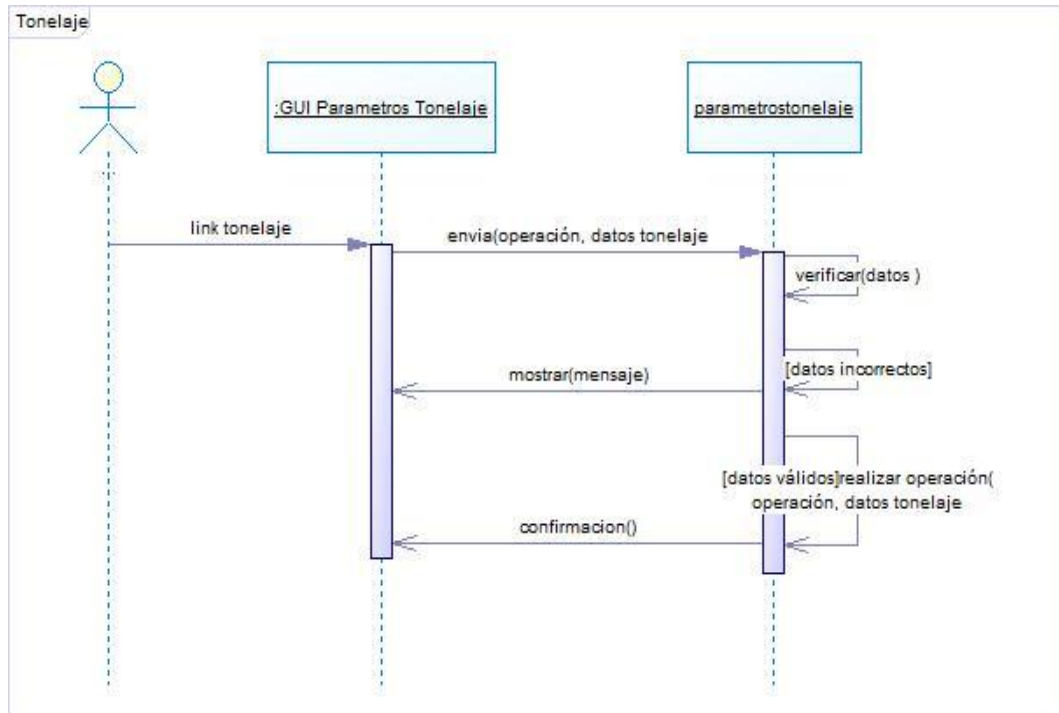


Figura 6.20 Diagrama de secuencia Tonelaje

## Salir del sistema

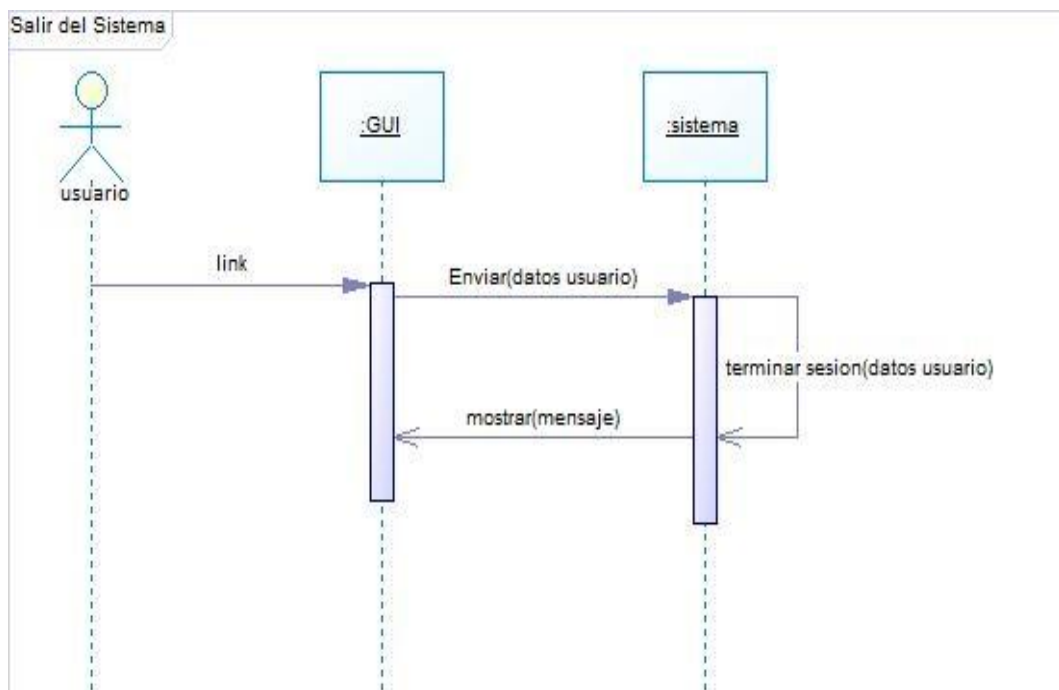
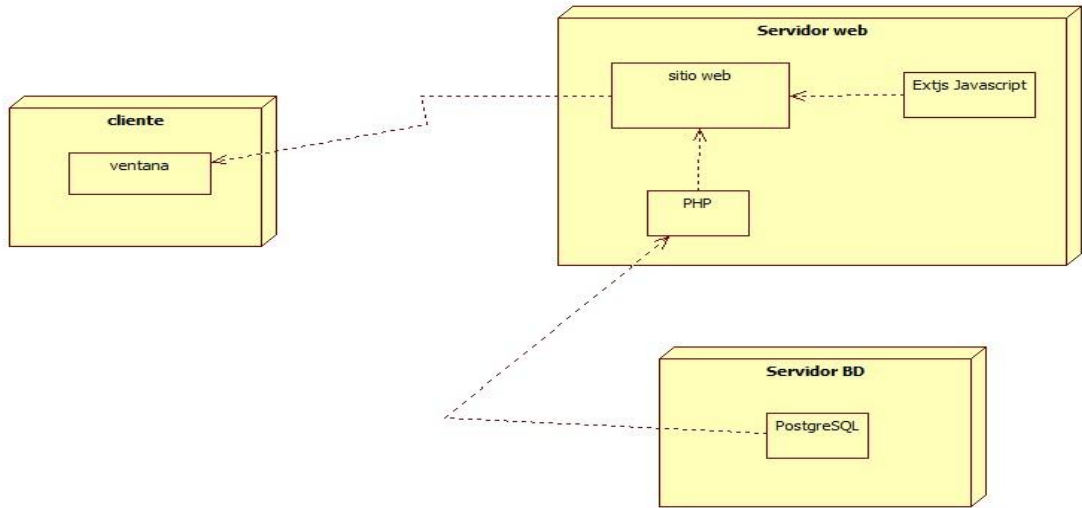


Figura 6.21 Diagrama de secuencia salir del sistema

#### 6.9.1.2.4 Diagrama de despliegue

Los diagramas de despliegue se usan para modelar las relaciones físicas entre los componentes hardware y software en el sistema.



**Figura 6.22** Diagrama de despliegue

### 6.9.2 Diseño del sistema

#### 6.9.2.1 Diseño de la base de datos

Al ser una aplicación PHP con base de datos PostgreSQL, luego del análisis minucioso del proceso de gestión de cobranza de valores por concepto de la contribución vial en el Gobierno Provincial de Tungurahua se ha modelado la base de datos de la siguiente manera.

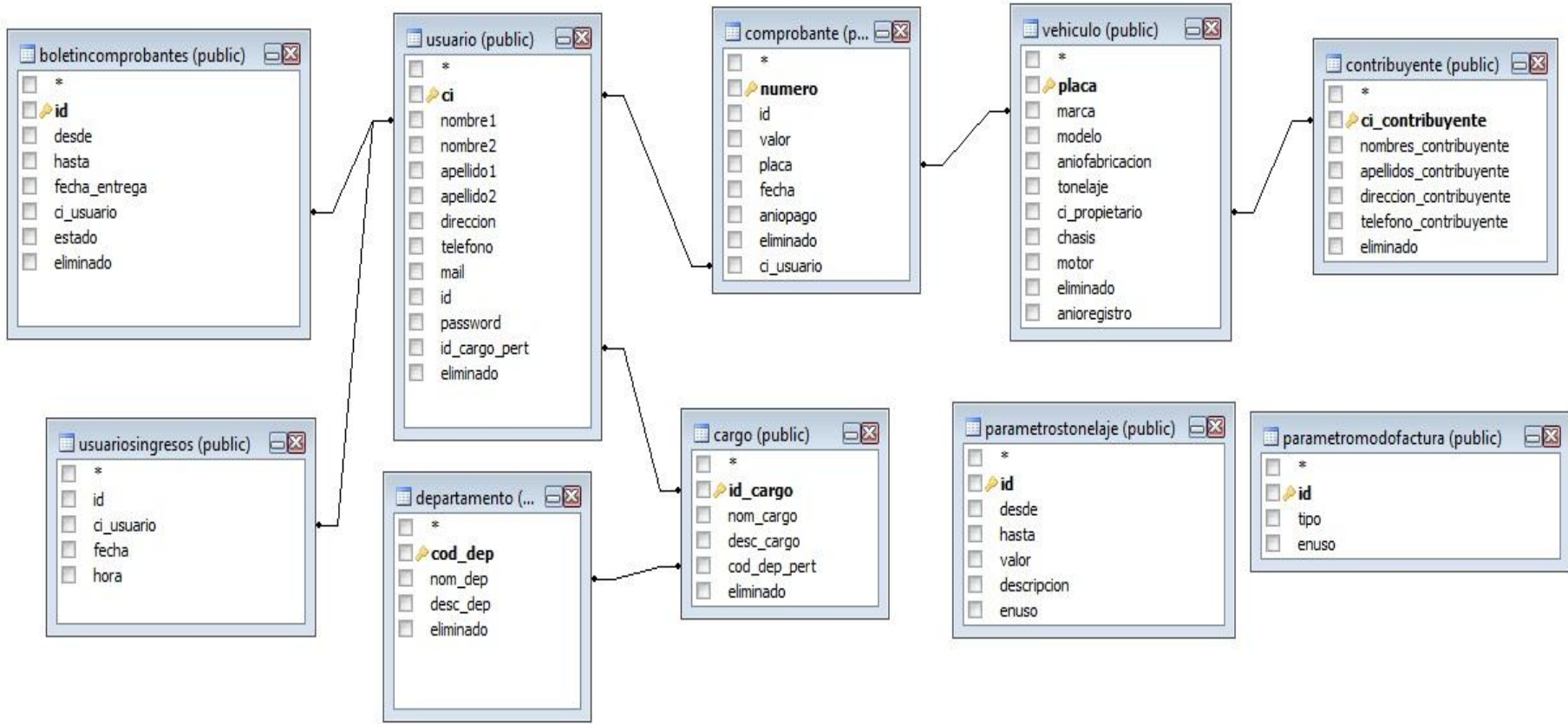


Figura 6.23 Modelo relacional lógico.

### 6.9.2.2 Diccionario de datos

TABLA: departamento			
ALMACENA INFORMACION DE LOS DEPARTAMENTOS			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
cod_dep	character varying(10)	Código del departamento	Primary key
nom_dep	character varying(20)	Nombre del departamento	Not null, UNIQUE
desc_dep	character varying(50)	Descripción del nombre del departamento	Null
Eliminado	Boolean	Valor para mostrar o no la información en el sistema	NOT NULL DEFAULT false

**Tabla 6.12.** Descripción de la tabla departamento

TABLA: cargo			
ALMACENA INFORMACION DE LOS CAGOS			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
id_cargo	character varying(10)	Id del cargo	Primary key
nom_cargo	character varying(25)	Nombre del cargo	Not null, UNIQUE
desc_cargo	character varying(50)	Descripción del nombre del cargo	Null
cod_dep_pert	character varying(10)	Código del departamento al que pertenece	Foreign key (cod_dep de la tabla departamento)
eliminado	Boolean	Valor para mostrar o no la información en el sistema	NOT NULL DEFAULT false

**Tabla 6.13.** Descripción de la tabla cargo

TABLA: usuario			
ALMACENA INFORMACION DE LOS USUARIOS			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
Ci	character(10)	CI del usuario	Primary key
nombre1	character varying(15)	Primer nombre del usuario	Not null
nombre2	character varying(15)	Segundo nombre del usuario	Not null
apellido1	character varying(15)	Primer apellido del usuario	Not null
apellido2	character varying(15)	Segundo apellido del usuario	Not null
direccion	character varying(50)	Dirección del usuario	Not null
Teléfono	character(9)	Teléfono del usuario	NULL
Mail	character varying(50)	e-mail del usuario	NULL
Id	character varying	nombre de login de usuario	Not null

password	character varying	contraseña de usuario	Not null
id_cargo_pert	character varying(10)	Id del cargo que tiene el usuario	Foreign key (id_cargo de la tabla cargo)
eliminado	Boolean	Valor para mostrar o no la información en el sistema	NOT NULL DEFAULT false

**Tabla 6.14.** Descripción de la tabla usuario

<b>TABLA: contribuyente</b>			
ALMACENA INFORMACION DE LOS CONTRIBUYENTES			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>	<b>RESTRICCIÓN</b>
ci_contribuyente	character(10)	CI del usuario	Primary key
nombres_contribuyente	character varying(25)	Nombres del contribuyente	Not null
apellidos_contribuyente	character varying(25)	Apellido del contribuyente	Not null
direccion_contribuyente	character varying(50)	Dirección del contribuyente	Not null
telefono_contribuyente	character(9)	Teléfono del contribuyente	NULL
Eliminado	Boolean	Valor para mostrar o no la información en el sistema	NOT NULL DEFAULT false

**Tabla 6.15.** Descripción de la tabla contribuyente

<b>TABLA: vehiculo</b>			
ALMACENA INFORMACION DE LOS VEHICULOS			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>	<b>RESTRICCIÓN</b>
Placa	character varying(7)	Placa del vehículo	Primary key
Marca	character varying(20)	Marca del vehículo	NULL
Modelo	character varying(20)	Modelo del vehículo	NULL
aniofabricacion	character(4)	Año fabricación del vehículo	NULL
Tonelaje	Numeric	Tonelaje del vehículo	NULL
ci_propietario	character(10)	CI del propietario del vehículo	Foreign key (ci_contribuyente de la tabla contribuyente)
Chasis	character varying(15)	Número de Chasis del vehículo	NULL
Motor	character varying(15)	Número de Marca del vehículo	NULL
eliminado	Boolean	Valor para mostrar o no la información en el sistema	NOT NULL DEFAULT false

**Tabla 6.16.** Descripción de la tabla vehículo



<b>TABLA: comprobante</b>			
ALMACENA INFORMACION DE LOS COMPROBANTES			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>	<b>RESTRICCIÓN</b>
Numero	Bigint	Número del comprobante	Primary key
Valor	Integer	Valor del comprobante	NULL
Placa	character varying(7)	Placa del vehículo pago	Foreign key (placa de la tabla vehiculo)
Tonelaje	Numeric	Tonelaje del vehículo	NULL
Aniopago	character(4)	Año al que corresponde el pago	Foreign key (ci_contribuyente de la tabla contribuyente)
Eliminado	Boolean	Valor para mostrar o no la información en el sistema	NOT NULL DEFAULT false

**Tabla 6.17.** Descripción de la tabla comprobante

<b>TABLA: usuariosingresos</b>			
ALMACENA INFORMACION DE LOS INGRESOS DE LOS USUARIOS AL SISTEMA			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>	<b>RESTRICCIÓN</b>
Id	Bigserial	Id de los registros	Primary key
ci_usuario	character(10)	Cédula del usuario	NOT NULL
Fecha	Date	Fecha de ingreso	NOT NULL
Hora	Time	Hora de ingreso	NOT NULL

**Tabla 6.18.** Descripción de la tabla usuariosingresos

<b>TABLA: parametrostonelaje</b>			
ALMACENA INFORMACION DE LOS PARAMETROS TONELAJE			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>	<b>RESTRICCIÓN</b>
id	bigserial	Id de los registros	Primary key
desde	real	Valor desde del rango	NOT NULL
hasta	real	Valor hasta del rango	NOT NULL
valor	double	Valor para el rango	NOT NULL
descripcion	character varying(50)	Descripcion para el rango	
enuso	boolean	Valor para mostrar o no la información en el sistema	NOT NULL

**Tabla 6.19.** Descripción de la tabla parametrostonelaje

<b>TABLA: parametromodofactura</b>			
ALMACENA INFORMACION DE LOS PARAMETROS DEL TIPO FACTURACION			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>	<b>RESTRICCIÓN</b>
id	Bigserial	Id de los registros	Primary key
tipo	character varying(50)	Tipo de facturación	NOT NULL
enuso	Boolean	Valor para mostrar o no la información en el sistema	NOT NULL

**Tabla 6.20.** Descripción de la tabla parametromodofactura

<b>TABLA: boletincomprobantes</b>			
ALMACENA INFORMACION DE LOS PARAMETROS TONELAJE			
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>	<b>RESTRICCIÓN</b>
id	Bigserial	Id de los registros	Primary key
desde	Bigint	Valor desde del rango de boletines	NOT NULL
hasta	Bigint	Valor hasta del rango de boletines	NOT NULL
fecha_entrega	Date	Fecha en la que se entrego el boletín	NOT NULL
ci_usuario	character(10)	Usuario	Foreign key (ci de la tabla usuario)
estado	character varying(50)		NOT NULL
enuso	Boolean	Valor para mostrar o no la información en el sistema	NOT NULL

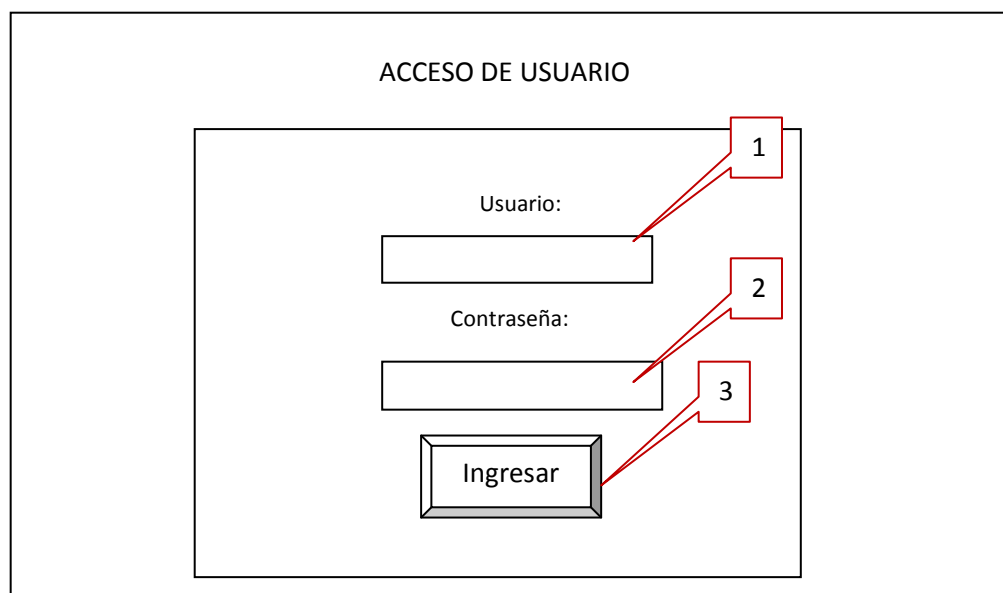
**Tabla 6.21.** Descripción de la tabla boletincomprobantes

### 6.9.2.3 Diseño de la interfaz

Se presenta una interfaz amigable, la misma que permite utilizar las funciones del sistema de una manera fácil y sencilla.

#### **Pantalla inicio de sesión**

En esta página web la persona que desee ingresar al sistema de gestión de cobranza deberá ingresar su usuario y contraseña obtenidos previamente a ser registrados por el administrador del sistema, estos datos serán validados y permitirán su acceso al sistema.



**Figura 6.24.** Pantalla inicio de sesión

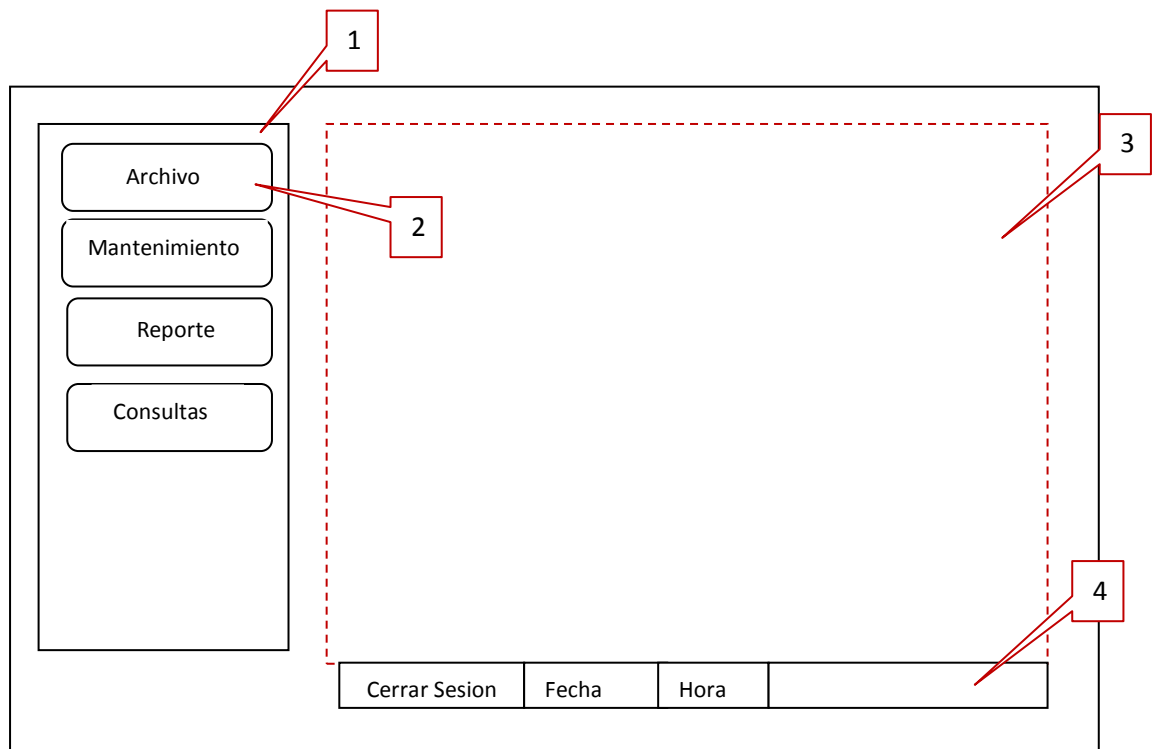
1. Caja de texto que permite el ingreso del nombre de usuario
2. Caja de texto que permite el ingreso de la contraseña
3. Botón Ingresar: valida los parámetros ingresados en las cajas de texto y posteriormente si los datos son correctos re-direcciona al Sistema de Gestión de Cobranza.

#### **6.9.2.3.1 Diseño de salidas**

En esta fase se procedió a diseñar las interfaces necesarias que permitirán al sistema informático entregar información a usuarios.

#### **Página principal del sistema de gestión de cobranza**

Cuando el usuario acaba de iniciar su sesión se le presenta la siguiente página con las opciones correspondientes:



**Figura 6.25.** Ventana inicial

1. Menú principal. Se podrá escoger 5 opciones como: Archivo, Mantenimiento, Reportes, Consultas.
2. Menú Archivo. Resultante de la selección “Archivo”, en éste tendrá la opción de cambiar su contraseña.
3. Página contenido. Espacio destinado para la gestión de información. La información mostrada dependerá directamente de las opciones marcadas en los menús.
4. ToolBar que muestra botón para cerrar sesión, fecha y hora del sistema.

Para la gestión de cobranza de valores y administración de la información de contribuyentes, el sistema cuenta con:

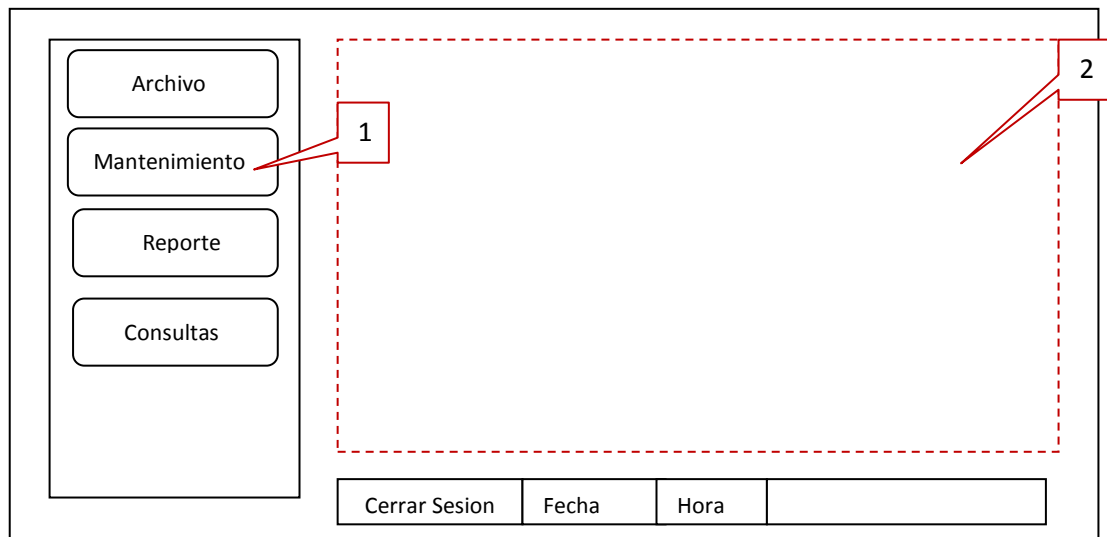
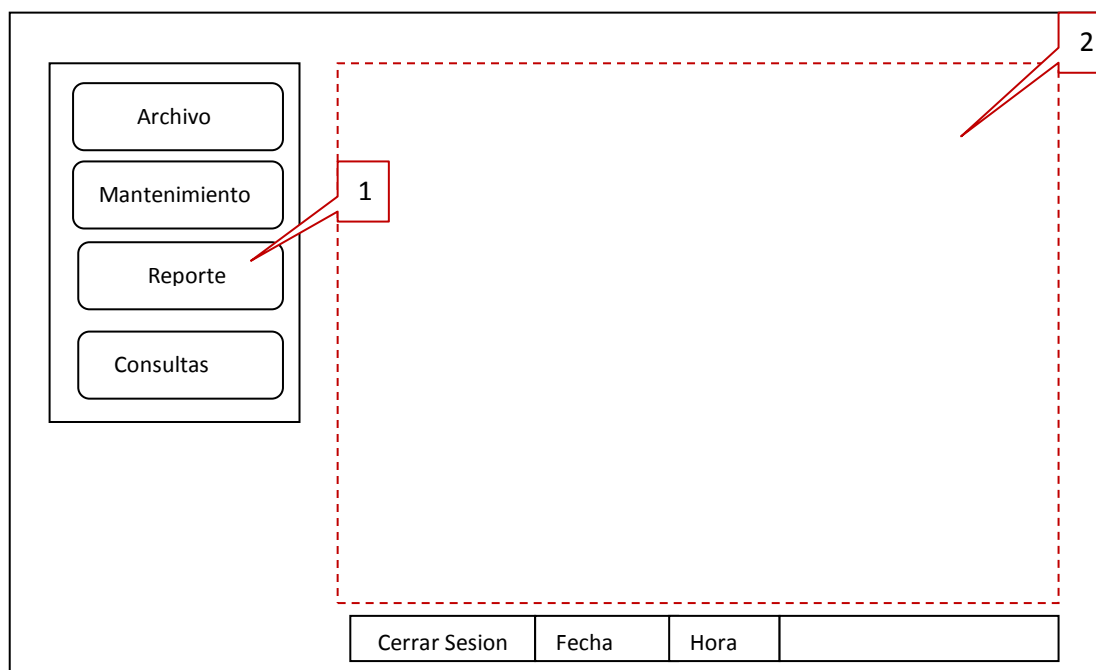


Figura 6.26. **Menú mantenimiento**

1. Menú mantenimiento. Resultante de la selección “Mantenimiento” en éste se podrá escoger 3 opciones como: Contribuyente, Vehículo, Comprobante.
  - a. Contribuyentes. Muestra información de los contribuyentes ya ingresados en el sistema
  - b. Vehículos. Muestra información de los vehículos registrados en el sistema.
  - c. Comprobante. Muestra información de los comprobantes ya generados por el sistema.
2. Ventana de contenido. Espacio destinado para la gestión de información. La información mostrada dependerá directamente de las opciones marcadas en los menús.

### **Generación de Reportes.**

Para la administración de cada uno de los Reportes el sistema cuenta con:



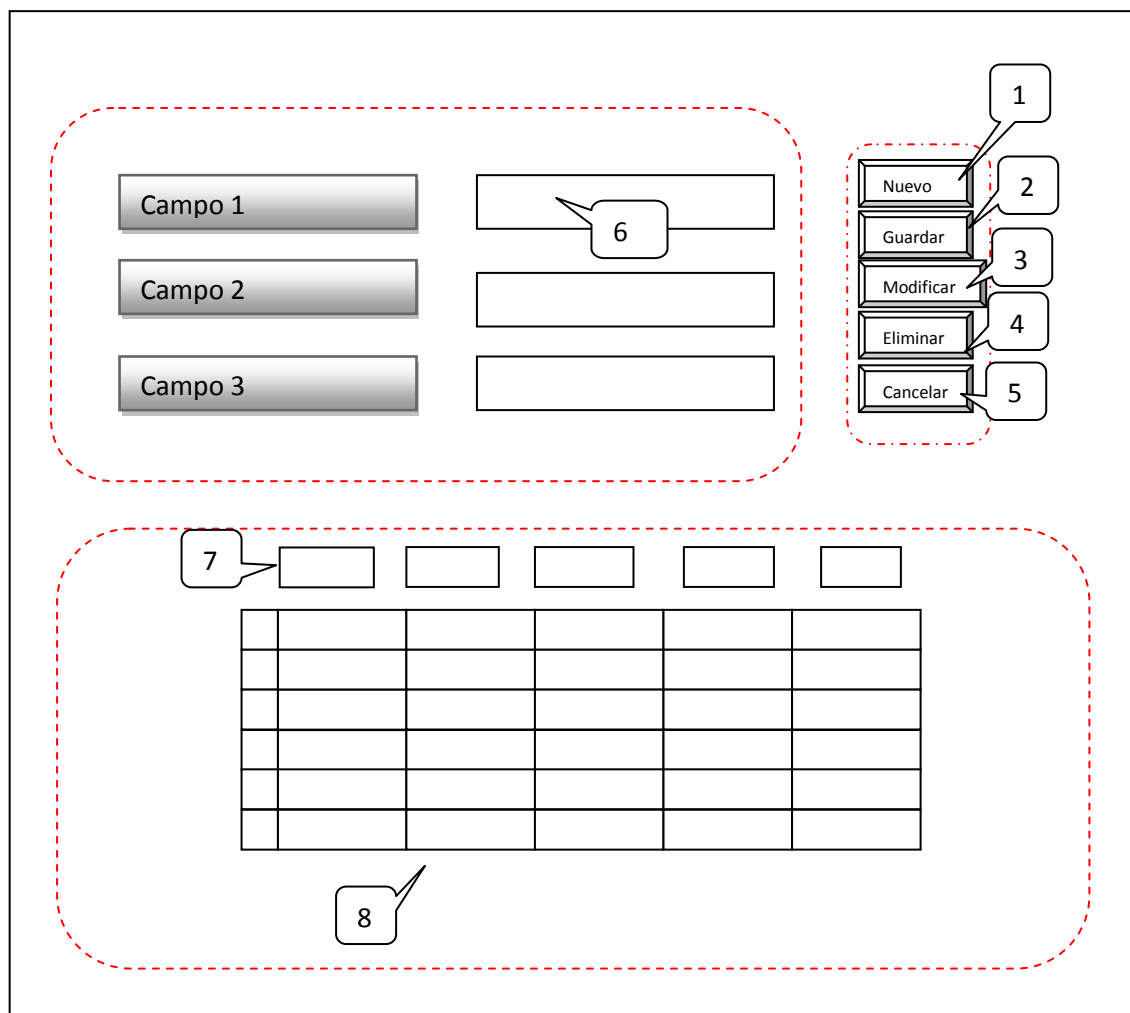
**Figura 6.27.** Ventana Reportes

1. Menú reportes. Resultante de la selección “Reportes” en éste se podrá escoger 2 opciones como: Cierre Diario, Consultar por fechas.
  - a. Cierre Diario. Muestra un reporte con la información de los comprobantes generados en todo el día, con los valores y el total recaudado.
  - b. Consulta por fechas. Muestra un reporte de los comprbantes y los valores generados en un rango de fecha ingresados por el usuario.
2. Ventana de contenido. Espacio destinado para la gestión de información. La información mostrada dependerá directamente de las opciones marcadas en los menús.

#### **6.9.2.3.2 Diseño de entradas**

#### **Manipulación de información**

Esta es la descripción general para las ventanas de manipulación de datos:



**Figura 6.28.** Manipulación de la información

1. Botón Nuevo. Habilita los campos para poder ingresar la información deseada.
2. Botón Guardar. Guarda los cambios que se haya realizado.
3. Botón Modificar. Habilita los campos que pueden ser manipulados para actualizarlos una vez que ha seleccionado con anterioridad.
4. Botón Eliminar. Elimina el registro que ha seleccionado con anterioridad.
5. Botón Cancelar. Cancela cualquier operación que esté realizando bloqueando los campos.
6. Controles para ingreso y visualización de datos.
7. Controles para filtrar información cargada por la página.
8. Grilla que muestra información de la Base de Datos.

## 6.9.3 Implementación

### 6.9.3.1 Extracto de código fuente

```
<?php
class ConectarBD
{
private $Servidor="localhost";
private $Usuario="postgres";
private $Clave="sa";
private $BD="SGCBD";
private $port="5432";
function ConectarBD(){
}

function conectar()
{
    $BaseDato = pg_connect("host=$this->Servidor port=$this->port dbname=$this->BD user=$this->Usuario password=$this->Clave");
    return $BaseDato;
}

function cadenaConexion()
{
    $cadenaConexion = "host=$this->Servidor port=$this->port dbname=$this->BD user=$this->Usuario password=$this->Clave";
    return $cadenaConexion;
}
}
?>
```

### Código para iniciar sesión

```
<?php
include_once("accesoBD/conectarBD.php");
$usr= $_POST["usuario"];
$password = $_POST["password"];
$con = new conectarBD;
if (!$con->conectar())
{
    echo '{"success": false, "errors":{"reason": "Error al intentar conectarse"}}';
}
else
{
    $conexion = pg_connect($con->cadenaConexion());
    $sql1 = "select ci,id_cargo_pert from usuario where id = '$usr' AND password = '$password'";
    $ejecutal = pg_send_query($conexion, $sql1); //devuelve true o false si la consulta se ejecuta o no
    if($ejecutal)
    {
        $res1 = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
        $cant = pg_num_rows($res1); //OBTIENE LA CANTIDAD DE REGISTROS DE LA CONSULTA
        if($cant>0)
        {
            session_start();
            $usuarioConectado = $_SESSION['user'] = $usr;
            $estado = $_SESSION['estado'] = "logueado"; // Coloco la variable de sesión 'estado'
            while ($row1 = pg_fetch_row($res1))
            {

```



```

        $_SESSION['ci'] = $cedula = $row1[0];
        $_tipoUsuario = $row1[1];
    }
    if($_tipoUsuario=="CARGO1")
    {
        $url="sistemaAdministrador.php";
        $_SESSION['usuario'] = "administrador";
    }
    else
    {
        $url="sistemaUsuario.php";
        $_SESSION['usuario'] = "usuario";
    }
    echo '{"success": true, "login":{"web": "'.$_usuarioConectado.'"}, "estado":{"web": "'.$_estado.'"}, "llevar":{"web": "'.$_url.'"},
"ci":{"web": "'.$_cedula.'"}}';
    }
    else
    {
        echo '{"success": false, "errors":{"reason": "Usuario o contraseña incorrecta"}}';
    }
}
}
}
?>

```

## Código para mantenimiento de contribuyentes

```

<?php
session_start();
if($_SESSION['estado'] == "logueado")
{
    include_once("../accesoBD/conectarBD.php");
    //recibe el formulario
    $Ci = $_POST['txtNuevaCiContribuyente'];
    $Nombre = $_POST['txtNuevoNombreContribuyente'];
    $Apellidos = $_POST['txtNuevosApellidosContribuyente'];
    $Direccion = $_POST['txtNuevaDireccionContribuyente'];
    $Telefono = $_POST['txtNuevoTelefonoContribuyente'];
    $task = '';
    if (isset($_POST['task'])) {
        $task = $_POST['task'];
    }
    //conectarse a la base de datos
    $con = new conectarBD;
    if (!$con->conectar())
    {
        echo '{"success": false, "errors":{"reason": "Error al intentar conectarse"}}';
    }
    else
    {
        $conexion = pg_connect ($con->cadenaConexion());
        pg_setclientencoding( 'latin1' );//definir el lenguaje de la conexion

        switch($task){
            case "GRABAR":
                //VEMOS Q la CI NO SE ENCUENTRE EN LOS ELIMINADOS PARA QUE NO DE ERROR X NOMBRE UNICO
                $sqlVerificaEliminado = "select ci_contribuyente from contribuyente where ci_contribuyente= '$Ci' AND
eliminado='TRUE'";
                $ejecutaVerificaEliminado = pg_send_query($conexion, $sqlVerificaEliminado);
                if($ejecutaVerificaEliminado)
                {
                    $resultVerificaEliminado = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
                    $contar = pg_num_rows($resultVerificaEliminado);
                    if($contar<1)
                    {
                        //INSERTO
                        $comando_insert = " INSERT INTO contribuyente (ci_contribuyente, nombres_contribuyente,
apellidos_contribuyente, direccion_contribuyente, telefono_contribuyente)
VALUES('$Ci','$Nombre', '$Apellidos','$Direccion','$Telefono') ";
                        $ejecuta = pg_send_query($conexion, $comando_insert); //devuelve true o false si la consulta se
ejecuta o no

                        if($ejecuta)
                        {
                            $res = pg_get_result($conexion);
                            if(pg_result_error($res)== false)
                            {

```

```

        echo '{"success":true, "message":{"reason": "Los datos se ingresaron correctamente"}}';
    }
    else
    {
        $mensaje = pg_result_error($res);
        echo '{"success":false, "errors":{"reason": "Número de cédula ya existente"}}';
    }
}
else
{
    echo "NO SE EJECUTO LA CONSULTA";
}
}
else
{
    $comando_update = "Update contribuyente set eliminado = 'FALSE' Where ci_contribuyente='%Ci'";
    $ejecutal = pg_send_query($conexion, $comando_update); //devuelve true o false si la consulta
se ejecuta o no
    if($ejecutal)
    {
        $res1 = pg_get_result($conexion);
        if(pg_result_error($res1)== false)
        {
            //ACTUALIZA nuevo
            $res1 = pg_get_result($conexion);

            if(pg_result_error($res1)== false)
            {
                echo '{"success":true, "message":{"reason": "El número de cédula ingresado ya
estaba registrado en el sistema, verifique los datos del contribuyente"}}';
            }
            else
            {
                $mensaje = pg_result_error($res1);
                echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con
el administrador del sistema"}}';
            }
        }
        //fin insertar nuevo
    }
    else
    {
        echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el
administrador del sistema"}}';
    }
}
else
{
    echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el
administrador del sistema"}}';
}
}
}

break;

case "MODIFICAR":
    $comando_update = "Update contribuyente Set
nombres_contribuyente='%Nombres',apellidos_contribuyente='%Apellidos',direccion_contribuyente='%Direccion', telefono_contribuyente='%Telefono'
Where ci_contribuyente='%Ci'";
    $ejecutal = pg_send_query($conexion, $comando_update); //devuelve true o false si la consulta se ejecuta o no
    if($ejecutal)
    {
        $res1 = pg_get_result($conexion);
        if(pg_result_error($res1)== false)
        {
            echo '{"success":true, "message":{"reason": "Los datos se actualizaron correctamente"}}';
        }
        else
        {
            $mensaje = pg_result_error($res1);
            echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el administrador del sistema"}}';
        }
    }
}
else
{
    echo "NO SE EJECUTO LA CONSULTA";
}

break;

case "ELIMINAR":
    //VERIFICO QUE NO TENGA DATOS RELACIONADOS
    $sqlverificaeliminar = "select ci_propietario from vehiculo where ci_propietario='%Ci' ";
    $ejecutaverificaeliminar = pg_send_query($conexion, $sqlverificaeliminar); //devuelve true o false si la consulta se
ejecuta o no
    if($ejecutaverificaeliminar)
    {
        $resultverificaeliminar = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
        $contar = pg_num_rows($resultverificaeliminar);
        if($contar<1)
        {
            $comando_delete = "Update contribuyente set eliminado = 'TRUE' Where ci_contribuyente='%Ci'";
            $ejecutal = pg_send_query($conexion, $comando_delete); //devuelve true o false si la consulta se ejecuta o no
            if($ejecutal)
            {
                $res1 = pg_get_result($conexion);
                if(pg_result_error($res1)== false)
                {

```

```

        echo '{"success":true, "message":{"reason": "El registro fue eliminado"}}';
    }
    else
    {
        $mensaje = pg_result_error($res1);
        echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el administrador del
sistema"}}';
    }
}
else
{
    echo "NO SE EJECUTO LA CONSULTA";
}
}
else
{
    echo '{"success":false, "errors":{"reason": "No se puede eliminar tiene datos relacionados"}}';
}
}
break;
default:
    // generar y ejecutar el query
    $sql = "SELECT ci_contribuyente, nombres_contribuyente, apellidos_contribuyente, direccion_contribuyente,
direccion_contribuyente,telefono_contribuyente FROM contribuyente WHERE eliminado='FALSE'";
    $ejecuta1 = pg_send_query($conexion, $sql); //devuelve true o false si la consulta se ejecuta o no

    if($ejecuta1)
    {
        //definir un array para poner los datos
        $arr= array();
        $res1 = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
        $rows1 = pg_num_rows($res1); //OBTIENE LA CANTIDAD DE REGISTROS DE LA CONSULTA
        for ( $i = 0 ; $i <= $rows1 ; $i ++ )
        {
            $row= pg_fetch_object($res1);
            $arr[] = $row;
        }
        echo '{"success:true,rows:'.json_encode($arr).'}';
    }
}
}
}
else {
header("Location: login.html");
}
?>

```

## Código para mantenimiento de contribuyentes

```

<?php
session_start();
if($_SESSION['estado'] == "logueado")
{
include_once("../accesoBD/conectarBD.php");
//recibe el formulario
$Placa = $_POST['txtPlaca'];
$Marca = $_POST['txtMarca'];
$Modelo = $_POST['txtModelo'];
$Chasis = $_POST['txtChasis'];
$Motor = $_POST['txtMotor'];
$Anio = $_POST['cboAnioVehiculo'];
$Tonelaje = $_POST['txtTonelaje'];
$Ci_Propietario = $_POST['cboPropietario'];
$task = '';
if ( isset($_POST['task']) ){
    $task = $_POST['task'];
}
//conectarse a la base de datos
$con = new conectarBD;
if (!$con->conectar())
{
    echo '{"success": false, "errors":{"reason": "Error al intentar conectarse"}}';
}
else

```

```

{
    $conexion = pg_connect ($con->cadenaConexion());
    pg_setclientencoding( 'latin1' );//definir el lenguaje de la conexion
    switch($task){
    case "GRABAR":
        ////////////////////////////////////////VERIFICA QUE LA PLACA NO ESTE EN LOS ELIMINADOS
        $sqlVerificaEliminado = "select placa from vehiculo where placa= '$Placa' AND eliminado='TRUE'";
        $ejecutaVerificaEliminado = pg_send_query($conexion, $sqlVerificaEliminado); //devuelve true o false si la consulta se
        ejecuta o no
        if($ejecutaVerificaEliminado)
        {
            $resultVerificaEliminado = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
            $contar1 = pg_num_rows($resultVerificaEliminado);
            if($contar1<1)
            {
                ////////////////////////////////////INSERTO
                $comando_insert = "INSERT INTO vehiculo (placa, marca, modelo, anio,fabricacion, tonELAJE,ci_propietario,
                chasis,motor ) VALUES('$Placa','$Marca', '$Modelo','$Anio','$TonELAJE','$Ci_Propietario','$Chasis','$Motor') ";
                $ejecuta = pg_send_query($conexion, $comando_insert); //devuelve true o false si la consulta se ejecuta o no
                if($ejecuta)
                {
                    $res = pg_get_result($conexion);
                    if(pg_result_error($res)== false)
                    {
                        {
                            echo '{"success":true, "message":{"reason": "Los datos se ingresaron correctamente"}}';
                        }
                        else
                        {
                            $mensaje = pg_result_error($res);
                            echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el administrador del sistema"}}';
                        }
                    }
                    else
                    {
                        echo "NO SE EJECUTO LA CONSULTA";
                    }
                }
                else
                {
                    //////si esq si esta en los eliminados solo le activo
                    $comando_update = "Update vehiculo Set eliminado='FALSE' Where placa= '$Placa'";
                    $ejecuta1 = pg_send_query($conexion, $comando_update); //devuelve true o false si la consulta se ejecuta
                    o no
                    if($ejecuta1)
                    {
                        $res1 = pg_get_result($conexion);
                        if(pg_result_error($res1)== false)
                        {
                            echo '{"success":true, "message":{"reason": "La placa ya esta registrada en el
                            sistema, se procedio a reactivarla"}}';
                        }
                        else
                        {
                            $mensaje = pg_result_error($res1);
                            echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el
                            administrador del sistema"}}';
                        }
                    }
                }
            }
            else
            {
                echo "NO SE EJECUTO LA CONSULTA";
            }
        }
        break;
    case "MODIFICAR":
        ////////////////////////////////////////MODIFICA
        $comando_update = "Update vehiculo Set marca='$Marca', modelo='$Modelo', aniofabricacion='$Anio', tonELAJE =
        $TonELAJE. ci propietario ='$Ci Propietario'. chasis='$Chasis'. motor='$Motor' Where placa='$Placa'";
        $ejecuta1 = pg_send_query($conexion, $comando_update); //devuelve true o false si la consulta se ejecuta o no
        if($ejecuta1)
        {
            $res1 = pg_get_result($conexion);
            if(pg_result_error($res1)== false)
            {
                echo '{"success":true, "message":{"reason": "Los datos se actualizaron correctamente"}}';
            }
            else
            {
                $mensaje = pg_result_error($res1);
                echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el administrador del
                sistema"}}';
            }
        }
        else
        {
            echo "NO SE EJECUTO LA CONSULTA";
        }
        break;
    case "ELIMINAR":
        ////////////////////////////////////////verificamos que el registro a eliminar no este relacionado con COMPROBANTES

```

```

$qlverificaeliminar = "select placa from comprobante where placa= '$Placa'";
$ejecutaverificaeliminar = pg_send_query($conexion, $qlverificaeliminar); //devuelve true o false si la
consulta se ejecuta o no
if($ejecutaverificaeliminar)
{
    $resultverificaeliminar = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
    $contar = pg_num_rows($resultverificaeliminar);
    if($contar<1)
    {
        //ELIMINAMOS
        $comando_delete = "Update vehiculo set eliminado = 'TRUE' WHERE placa='$Placa'";
        $ejecutal = pg_send_query($conexion, $comando_delete); //devuelve true o false si la consulta
se ejecuta o no
        if($ejecutal)
        {
            $res = pg_get_result($conexion);
            if(pg_result_error($res)== false)
            {
                echo '{"success":true, "message":{"reason": "El registro fue eliminado"}}';
                //echo '{"success":true, "message":{"reason": " ".$comando_delete."}';
            }
            else
            {
                $mensaje = pg_result_error($res);

                echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el
administrador del sistema"}}';
            }
        }
        else
        {
            echo "NO SE EJECUTO LA CONSULTA";
        }
    }
    else
    {
        echo '{"success":false, "errors":{"reason": "No se puede eliminar tiene datos relacionados"}}';
    }
}
break;
case "VERIFICARUNICO":
$comando_verificar = "select placa from vehiculo Where eliminado = 'FALSE'";
$ejecutal = pg_send_query($conexion, $sql); //devuelve true o false si la consulta se ejecuta o no
if($ejecutal)
{
    $result = pg_query($conexion, $comando_verificar); //devuelve A query result resource on success o
FALSE en caso de error.
    $cant = pg_num_rows($result);
    if (!$result)
    {
        echo "An error occured.\n";
        exit;
    }
    else
    {
        if ($cant > 0)
        {
            for ($i=0; $i<$cant; $i++)
            {
                $verConfig = pg_fetch_array($result);
                $CargaConfig[$i] = $verConfig["placa"];
            }

            if (in_array($_POST['verificanombre'], $CargaConfig)) {
                echo 0;
            } else {
                echo 1;
            }
        }
    }
}
break;

echo "An error occured.\n";
exit;
else
{
    if ($cant > 0)
    {
        for ($i=0; $i<$cant; $i++)
        {
            $verConfig = pg_fetch_array($result);
            $CargaConfig[$i] = $verConfig["placa"];
        }

        if (in_array($_POST['verificanombre'], $CargaConfig)) {
            echo 0;
        } else {
            echo 1;
        }
    }
}
break;

```

```

case "VERIFICARUNICOALMODIFICAR":
    $actual = $_POST['verificanombre'];
    $comando_verificar = "select placa from vehiculo Where eliminado = 'FALSE' and placa != '$actual'";
    $ejecutal = pg_send_query($conexion, $sql); //devuelve true o false si la consulta se ejecuta o no
    if($ejecutal)
    {
        $result = pg_query($conexion, $comando_verificar); //devuelve A query result resource on success o
        $cant = pg_num_rows($result);

        if (!$result)
        {
            echo "An error occurred.\n";
            exit;
        }
        else
        {
            if ($cant > 0)
            {
                for ($i=0; $i<$cant; $i++)
                {
                    $verConfig = pg_fetch_array($result);

                    $CargaConfig[$i] = $verConfig["placa"];

                    if (in_array($_POST['verificanombre'], $CargaConfig)) {
                        echo 0;
                    } else {
                        echo 1;
                    }
                }
            }
        }
    }
    break;
case "MODELO":
    ////////////////////////////////////////////////////
    function toJSON($array)
    {
        $data=array(); //data es el array
        $i=0;
        $total = count($array);
        foreach($array as $key=>$value){

            array_push($data,array('value'=>$i+1,'label'=>$value));
        }
        return json_encode(array('total'=>$total,'data'=>$data));
    }

    $anioactual = date('Y');
    $intervalo= $anioactual-1960+2;
    for ($i=0; $i < $intervalo; $i++)
    {
        $anios[$i] = 1960 + $i;
    }

    echo toJSON($anios);

break;
default:
    $sql = "SELECT v.placa, v.marca, v.modelo, v.aniofabricacion, v.tonelaje, v.ci_propietario, v.chasis, v.motor,
c.nombres_contribuyente, c.apellidos_contribuyente
FROM vehiculo v, contribuyente c
WHERE v.eliminado = 'FALSE'
AND v.ci_propietario = c.ci_contribuyente
";
    $ejecutal = pg_send_query($conexion, $sql); //devuelve true o false si la consulta se ejecuta o no

    if($ejecutal)
    {
        //definir un array para poner los datos
        $arr= array();
        $res1 = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
        $rows1 = pg_num_rows($res1); //OBTIENE LA CANTIDAD DE REGISTROS DE LA CONSULTA
        for ( $i = 0 ; $i < $rows1 ; $i ++ )
        {
            $row= pg_fetch_object($res1);
            $arr[] = $row;
        }
        echo '{"success:true,rows:".json_encode($arr).'}';
    }
}

}

else {
header("Location: login.html");
}
?>

```

## Código para mantenimiento de comprobantes

```
<?php
session_start();
if($_SESSION['estado'] == "logueado")
{
include_once("../accesoBD/conectarBD.php");
//recibe el formulario
$Valor = $_POST['ValorPago'];
$Placa = $_POST['cboPlacas'];
$AnioPago = $_POST['ValorAnioPago'];
$Numero = $_POST['Numero'];
$task = '';
$Fecha = date("d-m-Y");
if (isset($_POST['task'])){
$task = $_POST['task'];
}

$con = new conectarBD;
if (!$con->conectar())
{
echo '{"success": false, "errors":{"reason": "Error al intentar conectarse"}}';
}
else
{
$conexion = pg_connect ($con->cadenaConexion());
pg_setclientencoding( 'latin1' );//definir el lenguaje de la conexion
switch($task){
case "GRABAR":
$bandera=0;
$anioactual = date('Y');

$comando = "SELECT max(aniopago) as ultimo FROM comprobante WHERE placa = '$Placa'";
$ejecutal = pg_send_query($conexion, $comando); //devuelve true o false si la consulta se ejecuta o no
if($ejecutal)
{
$result1 = pg_query($conexion, $comando); //devuelve A query result resource on success o FALSE en caso
de error.

$cant1 = pg_num_rows($result1);
if ($cant1 > 0)
{

for ($i=0; $i<$cant1; $i++)
{
$verConfig = pg_fetch_array($result1);
$cargaConfig[$i] = $verConfig["ultimo"];
}
$ultimopagado = $cargaConfig[0] + 1;
if($ultimopagado == 1)
{
//no hago nada xq el anio pago va a ser un valor valido
$bandera = 0;
}
else
{
if($ultimopagado-1 == $anioactual)
{
//no hago nada xq el anio pago va a ser un valor valido
$bandera = 0;
}
else
{
if($ultimopagado-1 > $anioactual)
{
//ver q hacer xq va aniopago va a tener un valor NINGUN VALOR PENDIENTE
$bandera = 2;
}
else
{
if($AnioPago>$ultimopagado)
{
$bandera = 1;
}
else
{
$bandera = 0;
}
}
}
}
}
}
}
}
}
```





```

$ejecutal = pg_send_query($conexion, $comando_delete); //devuelve true o false si la consulta se ejecuta o no
if($ejecutal)
{
    $res1 = pg_get_result($conexion);
    if(pg_result_error($res1)== false)
    {
        echo '{"success":true, "message":{"reason": "El registro fue eliminado"}}';
    }
    else
    {
        $mensaje = pg_result_error($res1);
        echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el administrador del
sistema"}}';
    }
}
else
{
    echo "NO SE EJECUTO LA CONSULTA";
}

break;
case "ULTIMOPAGADO":
////////////////////////////////////
function toJSON($array)
{
    $data=array(); //data es el array
    $i=0;
    $total = count($array);
    foreach($array as $key=>$value){
        array_push($data,array('value'=>$i++, 'label'=>$value));
    }
    return json_encode(array('total'=>$total, 'data'=>$data));
}

////////////////////////////////////
$placaseleccionada = $_POST['placaseleccionada'];

$anioactual = date('Y');
$comando = "SELECT max(aniopago) as ultimo FROM comprobante WHERE placa = '$placaseleccionada' AND eliminado = 'FALSE'";

$ejecutal = pg_send_query($conexion, $comando); //devuelve true o false si la consulta se ejecuta o no
if($ejecutal)
{
    $result = pg_query($conexion, $comando); //devuelve A query result resource on success o FALSE en caso de error.
    $cant = pg_num_rows($result);
    if ($cant > 0)
    {
        for ($i=0; $i<$cant; $i++)
        {
            {
                $verConfig = pg_fetch_array($result);
                $CargaConfig[$i] = $verConfig["ultimo"];
            }
            $ultimopagado = $CargaConfig[0] + 1;
            //COMPARO SI EL RESULTADO NO ESTA VACIO
            if($ultimopagado == 1)
            {
                $aniosxpagar[0] = $anioactual;
                echo toJSON($aniosxpagar);
            }
            else
            {
                if($ultimopagado-1 == $anioactual)
                {
                    $aniosxpagar[0] = $anioactual + 1;
                    echo toJSON($aniosxpagar);
                }
                else
                {
                    if($ultimopagado > $anioactual)
                    {
                        $aniosxpagar[0] = "NADA PENDIENTE";
                        echo toJSON($aniosxpagar);
                    }
                }
            }
        }
    }
}

```



```

        for (i=0; i<$contar; i++)
        {
            $verConfig = pg_fetch_array($resultocontar);
            $CargaConfig[$i] = $verConfig["cod_dep"];
        }
        $UltimoCodigo = $CargaConfig[0];
        $numero = substr($UltimoCodigo, 3, strlen($UltimoCodigo));
        $numero = $numero + 1;
        $Codigo = "DEP". $numero;
    }
    else
    {
        $Codigo= "DEP1";
    }
}

////////////////////// VEMOS Q EL NOMBRE NO SE ENCUENTRE EN LOS ELIMINADOS PARA QUE NO DE ERROR X NOMBRE UNICO

$sqlVerificaEliminado = "select nom_dep from departamento where nom_dep= '$Nombre' AND eliminado='TRUE'";
$ejecutaVerificaEliminado = pg_send_query($conexion, $sqlVerificaEliminado); //devuelve true o false si la consulta se ejecuta
o no
if($ejecutaVerificaEliminado)
{
    $resultVerificaEliminado = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
    $contar = pg_num_rows($resultVerificaEliminado);
    if($contar<1)
    {
        //inserto
        $comando_insert = "INSERT INTO departamento (cod_dep, nom_dep, desc_dep) VALUES('$Codigo','$Nombre','$Desc')";
        $ejecuta = pg_send_query($conexion, $comando_insert); //devuelve true o false si la consulta se ejecuta o no
        if($ejecuta)
        {
            $res = pg_get_result($conexion);
            if(pg_result_error($res)== false)
            {

                echo '{"success":true, "message":{"reason": "Los datos se ingresaron correctamente"}}';
            }
            else
            {
                $mensaje = pg_result_error($res);
                echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el administrador del sistema"}}';
            }
        }
        else
        {
            echo "NO SE EJECUTO LA CONSULTA";
        }
    }
    else
    {
        $comando_update = "Update departamento Set desc_dep='$Desc', eliminado='FALSE' Where nom_dep= '$Nombre'";
        $ejecuta1 = pg_send_query($conexion, $comando_update); //devuelve true o false si la consulta se ejecuta o no
        if($ejecuta1)
        {
            $res1 = pg_get_result($conexion);
            if(pg_result_error($res1)== false)
            {
                echo '{"success":true, "message":{"reason": "Los datos se ingresaron correctamente"}}';
            }
            else
            {
                $mensaje = pg_result_error($res1);
                echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el administrador del
sistema"}}';
            }
        }
        else
        {
            echo "NO SE EJECUTO LA CONSULTA";
        }
    }
}

```

```

    }
  }
}

break;
case "MODIFICAR":
// VEMOS Q EL NOMBRE NO SE ENCUENTRE EN LOS ELIMINADOS PARA QUE NO DE ERROR X NOMBRE UNICO
$sqlVerificaEliminado = "select nom_dep from departamento where nom_dep= '$Nombre' AND eliminado='TRUE'";
$ejecutaVerificaEliminado = pg_send_query($conexion, $sqlVerificaEliminado); //devuelve true o false si la consulta se ejecuta
o no

if($ejecutaVerificaEliminado)
{
$resultVerificaEliminado = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
$contar = pg_num_rows($resultVerificaEliminado);

if($contar<1)
{
$Codigo = $_POST['txtCodigoDepartamento'];
$comando_update = "Update departamento Set nom_dep='$Nombre', desc_dep='$Desc' Where cod_dep='$Codigo'";
$ejecuta1 = pg_send_query($conexion, $comando_update); //devuelve true o false si la consulta se ejecuta o no
if($ejecuta1)
{
$res1 = pg_get_result($conexion);
if(pg_result_error($res1)== false)
{
echo '{"success":true, "message":{"reason": "Los datos se actualizaron correctamente"}}';
}
else
{
}

{
$mensaje = pg_result_error($res1);
echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el administrador del
sistema"}}';
}
}
else
{
echo "NO SE EJECUTO LA CONSULTA";
}
}
else
{
echo '{"success":false, "errors":{"reason": "Nombre no válido, por favor ingrese uno diferente"}}';
}
}
}

break;
case "ELIMINAR":
$Codigo = $_POST['txtCodigoDepartamento'];
$sqlverificaeliminar = "select id_cargo from cargo where cod_dep_pert= '$Codigo'";
$ejecutaverificaeliminar = pg_send_query($conexion, $sqlverificaeliminar); //devuelve true o false si la consulta se ejecuta o no

if($ejecutaverificaeliminar)
{
$resultverificaeliminar = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
$contar = pg_num_rows($resultverificaeliminar);

if($contar<1)
{
$comando_delete = "Update departamento Set eliminado='TRUE' Where cod_dep='$Codigo'";
$ejecuta1 = pg_send_query($conexion, $comando_delete); //devuelve true o false si la consulta se ejecuta o no

```

```

        if($ejecutal)
        {
            $res1 = pg_get_result($conexion);
            if(pg_result_error($res1)== false)
            {
                echo '{"success":true, "message":{"reason": "El registro fue eliminado"}}';
            }
            else
            {
                $mensaje = pg_result_error($res1);
                echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el administrador del
sistema"}}';
            }
        }
        else
        {
            echo "NO SE EJECUTO LA CONSULTA";
        }
    }
    else
    {
        echo '{"success":false, "errors":{"reason": "No se puede eliminar tiene datos relacionados"}}';
    }
}
break;
case "VERIFICARUNICO":

$comando_verificar = "select nom_dep from departamento Where eliminado = 'FALSE'";
$ejecutal = pg_send_query($conexion, $comando_verificar); //devuelve true o false si la consulta se ejecuta o no

if($ejecutal)
{
    $result = pg_query($conexion, $comando_verificar); //devuelve A query result resource on success o FALSE en caso de
error.

    $cant = pg_num_rows($result);

    if (!$result)
    {
        echo "An error occured.\n";
        exit;
    }

    else
    {
        if ($cant > 0)
        {
            for ($i=0; $i<$cant; $i++)
            {
                $verConfig = pg_fetch_array($result);
                $CargaConfig[$i] = $verConfig["nom_dep"];

                if (in_array($_POST['verificanombre'], $CargaConfig)) {
                    echo 0;
                } else {
                    echo 1;
                }
            }
        }
    }
}
break;
case "VERIFICARUNICOMODIFICAR":
    $actual = $_POST['verificanombre'];
    $comando_verificar = "select nom_dep from departamento Where eliminado = 'FALSE' and nom_dep != '$actual'";

```

```

$ejecutal = pg_send_query($conexion, $sql); //devuelve true o false si la consulta se ejecuta o no
if($ejecutal)
{
    $result = pg_query($conexion, $comando_verificar); //devuelve A query result resource on success o
FALSE en caso de error.
    $cant = pg_num_rows($result);

    if (!$result)
    {
        echo "An error ocurred.\n";
        exit;
    }

    else
    {
        if ($cant > 0)
        {
            for ($i=0; $i<$cant; $i++)
            {
                $verConfig = pg_fetch_array($result);
                $CargoConfig[$i] = $verConfig["nom_dep"];
            }

            if (in_array($_POST['verificanombre'], $CargoConfig)) {
                echo 0;
            } else {
                echo 1;
            }
        }
    }
}

break;
case "NOMBREDEPARTAMENTO":
    $Codigo = $_POST['codigocargo'];
    $sql = "SELECT d.nom_dep
FROM cargo c, departamento d
WHERE c.id_cargo = '$Codigo'
AND c.cod_dep_pert = d.cod_dep ";
$ejecutal = pg_send_query($conexion, $sql); //devuelve true o false si la consulta se ejecuta
o no
if($ejecutal)
{
    $res1 = pg_get_result($conexion);
    while ($row = pg_fetch_row($res1)) {
        $arr[] = $row;
    }
    echo '{"rows":'.json_encode($arr).'}';
}
else
{
    echo '{"success":false, "message":{"reason": "Hubo un error"}}';
}

break;
default:
$sql = "SELECT cod_dep,nom_dep, desc_dep
FROM departamento
WHERE eliminado = 'FALSE'
";
$ejecutal = pg_send_query($conexion, $sql); //devuelve true o false si la consulta se ejecuta o no
if($ejecutal)
{
    $arr= array();
    $res1 = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA

    $rows1 = pg_num_rows($res1); //OBTIENE LA CANTIDAD DE REGISTROS DE LA CONSULTA
    for ( $i = 0 ; $i < $rows1 ; $i ++ )
    {
        $row= pg_fetch_object($res1);
        $arr[] = $row;
    }
    echo '{"rows":'.json_encode($arr).'}';
}

}

}

else {
header("Location: login.html");
}
?>

```

## Código para mantenimiento de cargos

```

<?php
session_start();
if($_SESSION['estado'] == "logueado")
{
    include_once("../accesoBD/conectarBD.php");
    //recibe el formulario
    $Nombre = $_POST['txtNombreCargo'];
    $Desc = $_POST['txtDescripcionCargo'];

    $task = '';
    if (isset($_POST['task'])){
        $task = $_POST['task'];
    }
    //conectarse a la base de datos
    $con = new conectarBD;
    if (!$con->conectar())
    {
        echo '{"success": false, "errors":{"reason": "Error al intentar conectarse"}}';
    }
    else
    {
        $conexion = pg_connect ($con->cadenaConexion());
        pg_setclientencoding( 'latin1' );//definir el lenguaje de la conexion
        switch($task){
            case "GRABAR":
                //consulto el codigo del departamento al que va a pertenecer el combo
                $NombreDepartamento = $_POST['cboDepartamento'];
                $sqlConsultarCodigo = "select cod_dep from departamento where nom_dep ='$NombreDepartamento'";
                $ejecutaConsultarCodigo = pg_send_query($conexion, $sqlConsultarCodigo); //devuelve true o false si la consulta se
                ejecuta o no
                if($ejecutaConsultarCodigo)
                {
                    $resultConsultarCodigo = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
                    while ($row1 = pg_fetch_row($resultConsultarCodigo))
                    {
                        $Cod_Depa_Pert = $row1[0];
                    }
                }
                //genero automaticamente el codigo del cargo
                $sqlcontar = "select max(id_cargo) as id_cargo from cargo";
                $ejecutacontar = pg_send_query($conexion, $sqlcontar); //devuelve true o false si la consulta se ejecuta o
                no
                if($ejecutacontar)
                {
                    $resultcontar = pg_query($conexion, $sqlcontar);
                    $contar = pg_num_rows($resultcontar);
                    if ($contar > 0)
                    {
                        for ($i=0; $i<$contar; $i++)
                        {
                            $verConfig = pg_fetch_array($resultcontar);
                            $CargaConfig[$i] = $verConfig["id_cargo"];
                        }
                        $UltimoCodigo = $CargaConfig[0];
                        $numero = substr($UltimoCodigo, 5, strlen($UltimoCodigo));
                        $numero = $numero + 1;
                        $sid = "CARGO". $numero;
                    }
                    else
                    {
                        $sid = "CARGO1";
                    }
                }
                // VEMOS Q EL NOMBRE NO SE ENCUENTRE EN LOS ELIMINADOS PARA QUE NO DE ERROR X NOMBRE UNICO
                $sqlVerificaEliminado = "select nom_cargo from CARGO where nom_cargo= '$Nombre' AND eliminado='TRUE'";
                $ejecutaVerificaEliminado = pg_send_query($conexion, $sqlVerificaEliminado); //devuelve true o false si la consulta se
                ejecuta o no
                if($ejecutaVerificaEliminado)
                {
                    $resultVerificaEliminado = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
                    $contar1 = pg_num_rows($resultVerificaEliminado);
                    if($contar1<1)

```

```

        {
            //inserto
            $comando_insert = "INSERT INTO cargo (id_cargo, nom_cargo, desc_cargo, cod_dep_pert)
VALUES('$id','$Nombre','$Desc','$Cod_Depa_Pert)";
            $ejecuta = pg_send_query($conexion, $comando_insert); //devuelve true o false si la consulta se ejecuta o
no
            if($ejecuta)
            {
                $res = pg_get_result($conexion);
                if(pg_result_error($res) == false)
                {
                    echo '{"success":true, "message":{"reason": "Los datos se ingresaron correctamente"}}';
                }
                else
                {
                    $mensaje = pg_result_error($res);
                    echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el administrador
del sistema"}}';
                    //echo '{"success":false, "errors":{"reason": "'. $res.'"}}';
                }
            }
            else
            {
                echo "NO SE EJECUTO LA CONSULTA";
            }
            //fin inserto
        }
        else //si esq si esta en los eliminados le activo
        {
            $comando_update = "Update cargo Set desc_cargo='$Desc', cod_dep_pert='$Cod_Depa_Pert', eliminado='FALSE'
Where nom_cargo= '$Nombre'";
            $ejecuta1 = pg_send_query($conexion, $comando_update); //devuelve true o false si la consulta se ejecuta
o no
            if($ejecuta1)
            {
                $res1 = pg_get_result($conexion);

                if(pg_result_error($res1) == false)
                {
                    echo '{"success":true, "message":{"reason": "Los datos se ingresaron
correctamente"}}';
                }
                else
                {
                    $mensaje = pg_result_error($res1);
                    echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el
administrador del sistema"}}';
                }
            }
            else
            {
                echo "NO SE EJECUTO LA CONSULTA";
            }
        }
    }
}
break;
case "MODIFICAR":
    // VEMOS Q EL NOMBRE NO SE ENCUENTRE EN LOS ELIMINADOS PARA QUE NO DE ERROR X NOMBRE UNICO
    $sqlVerificaEliminado = "select nom_cargo from cargo where nom_cargo= '$Nombre' AND eliminado='TRUE'";
    $ejecutaVerificaEliminado = pg_send_query($conexion, $sqlVerificaEliminado);
    if($ejecutaVerificaEliminado)
    {
        $resultVerificaEliminado = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
        $contar = pg_num_rows($resultVerificaEliminado);
        if($contar < 1)
        {
            $NombreDepartamento = $_POST['cboDepartamento'];
            $sqlConsultarCodigo = "select cod_dep from departamento where nom_dep = '$NombreDepartamento'";
            $ejecutaConsultarCodigo = pg_send_query($conexion, $sqlConsultarCodigo); //devuelve true o
false si la consulta se ejecuta o no
            if($ejecutaConsultarCodigo)
            {
                $resultConsultarCodigo = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA

```



```

CONSULTIA

while ($row1 = pg_fetch_row($resultConsultarCodigo))
{
    $Cod_Depa_Pert = $row1[0];
}
$id=$_POST['txtCodigoCargo'];
$comando_update = "Update cargo Set nom_cargo='$Nombre', desc_cargo='$Desc',
cod_dep_pert='$Cod_Depa_Pert' Where id_cargo='$id'";
$ejecuta1 = pg_send_query($conexion, $comando_update); //devuelve true o false si la consulta
se ejecuta o no
if($ejecuta1)
{
    $res1 = pg_get_result($conexion);
    if(pg_result_error($res1)== false)
    {
        echo '{"success":true, "message":{"reason": "Los datos se actualizaron correctamente"}}';
    }
    else
    {
        $mensaje = pg_result_error($res1);
        echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el
administrador del sistema"}}';
    }
}
else
{
    echo "NO SE EJECUTO LA CONSULTA";
}
}
else
{
    echo '{"success":false, "errors":{"reason": "Nombre no válido, por favor ingrese uno diferente"}}';
}
}

break;
case "ELIMINAR":
//Verificamos que el registro a eliminar no este relacionado con usuarios
$Codigo = $_POST['txtCodigoCargo'];
$sqlverificaeliminar = "select ci from usuario where id_cargo_pert= '$Codigo'";
$ejecutaverificaeliminar = pg_send_query($conexion, $sqlverificaeliminar); //devuelve true o false si la
consulta se ejecuta o no
if($ejecutaverificaeliminar)
{
    $resultverificaeliminar = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
    $contar = pg_num_rows($resultverificaeliminar);
    if($contar<1)
    {
        $comando_delete = "Update cargo Set eliminado='TRUE' Where id_cargo='$Codigo'";
        $ejecuta1 = pg_send_query($conexion, $comando_delete); //devuelve true o false si la consulta
se ejecuta o no
if($ejecuta1)
{
    $res1 = pg_get_result($conexion);
    if(pg_result_error($res1)== false)
    {
        echo '{"success":true, "message":{"reason": "El registro fue eliminado"}}';
    }
    else
    {
        $mensaje = pg_result_error($res1);
        echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el
administrador del sistema"}}';
    }
}
else
{
    echo "NO SE EJECUTO LA CONSULTA";
}
}
}
else
{
}
}
}

```



## Código para mantenimiento de usuarios

```
<?php
session_start();
if($_SESSION['estado'] == "logueado")
{
include_once("../accesoBD/conectarBD.php");
//recibe el formulario
$Ci = $_POST['txtCi'];
$Nombre1 = $_POST['txtNombre1'];
$Nombre2 = $_POST['txtNombre2'];
$Apellido1 = $_POST['txtApellido1'];
$Apellido2 = $_POST['txtApellido2'];
$Direccion = $_POST['txtDireccion'];
$Telefono = $_POST['txtTelefono'];
>Email = $_POST['txtEmail'];
$Fecha_nacimiento = $_POST['txtFecha'];
$Cargo = $_POST['cboCargo'];
$Usuario2 = $_POST['txtUsuario'];
>Password = $_POST['txtPassword'];
$task = '';
if (isset($_POST['task'])) {
$task = $_POST['task'];
}
//conectarse a la base de datos
$con = new conectarBD;
if (!$con->conectar())
{
echo '{"success": false, "errors":{"reason": "Error al intentar conectarse"}}';
}
else
{
$conexion = pg_connect ($con->cadenaConexion());
pg_setclientencoding( 'latin1' );//definir el lenguaje de la conexion
switch($task){
case "GRABAR":
// VEMOS Q EL NOMBRE NO SE ENCUENTRE EN LOS ELIMINADOS PARA QUE NO DE ERROR X NOMBRE UNICO
$sqlVerificaEliminado = "select id from usuario where id= '$Usuario2' AND eliminado='TRUE'";

$ejecutaVerificaEliminado = pg_send_query($conexion, $sqlVerificaEliminado);
if($ejecutaVerificaEliminado)
{
$resultVerificaEliminado = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
$contar = pg_num_rows($resultVerificaEliminado);
if($contar<1)
{
$CodigoCargo = $_POST['codigocargo'];
$comando_insert = " INSERT INTO usuario (ci, nombre1, nombre2, apellido1, apellido2 ,direccion,
telefono, mail, id, password,id_cargo_pert) VALUES('$Ci','$Nombre1','$Nombre2', '$Apellido1',
'$Apellido2','$Direccion','$Telefono','$Email','$Usuario2','$Password','$CodigoCargo') ";

$ejecuta = pg_send_query($conexion, $comando_insert); //devuelve true o false si la consulta se
ejecuta o no
if($ejecuta)
{
$res = pg_get_result($conexion);
if(pg_result_error($res)== false)
{
echo '{"success":true, "message":{"reason": "Los datos se ingresaron correctamente"}}';
//echo '{"success":true, "message":{"reason": "'.$CodigoCargo.'"}}';
}
else
{
echo '{"success":false, "errors":{"reason": "Error de BD, ci ya registrada, o consulte
con el administrador"}}';
}
}
else
{
echo "NO SE EJECUTO LA CONSULTA";
}
}
else
{
$comando_update = "Update usuario Set id='$Ci' Where id='$Usuario2'";

```



```

//actualizo
comando_update = "Update usuario Set direccion='$Direccion', telefono='$Telefono',
mail='$Email', id='$Usuario2', password='$Password', id_cargo_pert= '$id_Cargo' Where ci='$Ci'";
sejecutal = pg_send_query($conexion, $comando_update); //devuelve true o false si la consulta
se ejecuta o no
if($sejecutal)
{
$res1 = pg_get_result($conexion);
if(pg_result_error($res1)== false)
{
echo '{"success":true, "message":{"reason": "Los datos se actualizaron correctamente"}}
';
}
else
{
echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el
administrador del sistema"}}';
}
}
else
{
echo "NO SE EJECUTO LA CONSULTA";
}
}
else
{
echo '{"success":false, "errors":{"reason": "Nombre de login no valido, por favor ingrese uno
diferente"}}';
}
}
break;
case "ELIMINAR":
comando_delete = "Delete from usuario Where ci='$Ci'";
sejecutal = pg_send_query($conexion, $comando_delete); //devuelve true o false si la consulta se ejecuta o no
if($sejecutal)
{
$res1 = pg_get_result($conexion);
if(pg_result_error($res1)== false)
{
echo '{"success":true, "message":{"reason": "El registro fue eliminado"}}';
}
else
{
$mensaje = pg_result_error($res1);
echo '{"success":false, "errors":{"reason": "Se produjo un error consultar con el administrador del
sistema"}}';
}
}
else
{
echo "NO SE EJECUTO LA CONSULTA";
}
}
break;
case "VERIFICARUNICO":
comando_verificar = "select id from usuario Where eliminado = 'FALSE'";
sejecutal = pg_send_query($conexion, $sql); //devuelve true o false si la consulta se ejecuta o no
if($sejecutal)
{
$result = pg_query($conexion, $comando_verificar); //devuelve  query result resource on success o
FALSE en caso de error.
$scant = pg_num_rows($result);
if (!$result)
{
echo "An error occurred.\n";
exit;
}
else
{
if ($scant > 0)
{
for ($i=0; $i<$scant; $i++)
{

```

```

        $verConfig = pg_fetch_array($result);
        $CargaConfig[$i] = $verConfig["id"];
    }
    if (in_array($_POST['verificanombre'], $CargaConfig)) {
        echo 0;
    } else {
        echo 1;
    }
}
}
}
break;
case "VERIFICARUNICOALMODIFICAR":
    $actual = $_POST['verificanombre'];
    $comando_verificar = "select id from usuario Where eliminado = 'FALSE' and id != '$actual'";
    $ejecutal = pg_send_query($conexion, $sql); //devuelve true o false si la consulta se ejecuta o no
    if($ejecutal)
    {
        $result = pg_query($conexion, $comando_verificar); //devuelve A query result resource on success o
        FALSE en caso de error.
        $cant = pg_num_rows($result);
        if (!$result)
        {
            echo "An error ocured.\n";
            exit;
        }
        else
        {
            if ($cant > 0)
            {
                for ($i=0; $i<$cant; $i++)
                {
                    $verConfig = pg_fetch_array($result);
                    $CargaConfig[$i] = $verConfig["id"];
                }
                if (in_array($_POST['verificanombre'], $CargaConfig)) {
                    echo 0;
                } else {
                    echo 1;
                }
            }
        }
    }
}
break;
default:
    // generar y ejecutar el query
    $sql = "SELECT u.ci, u.nombre1, u.nombre2, u.apellido1, u.apellido2, u.direccion, u.telefono, u.mail, u.id, u.password,
    c.nom_cargo,d.nom_dep
    FROM usuario u, cargo c, departamento d
    WHERE u.id_cargo_pert = c.id_cargo
    AND c.cod_dep_pert = d.cod_dep
    AND u.eliminado='FALSE'
    ";
    $ejecutal = pg_send_query($conexion, $sql); //devuelve true o false si la consulta se ejecuta o no
    if($ejecutal)
    {
        //definir un array para poner los datos
        $arr= array();
        $res1 = pg_get_result($conexion); //OBTIENE EL RESULTADO DE LA CONSULTA
        $rows1 = pg_num_rows($res1); //OBTIENE LA CANTIDAD DE REGISTROS DE LA CONSULTA
        for ( $i = 0 ; $i < $rows1 ; $i ++ )
        {
            $row= pg_fetch_object($res1);
            $arr[] = $row;
        }
        echo '{"success:true,rows:'.json_encode($arr).'}';
    }
}
}
}
else {
header("Location: login.html");
}
?>

```

## 6.9.4 Pruebas

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

El objetivo específico de la fase de pruebas es encontrar cuantos más errores, mejor.

Una vez concluido con el desarrollo del sistema informático, 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 de Cobranza.

### 6.9.4.1 Prueba Alfa

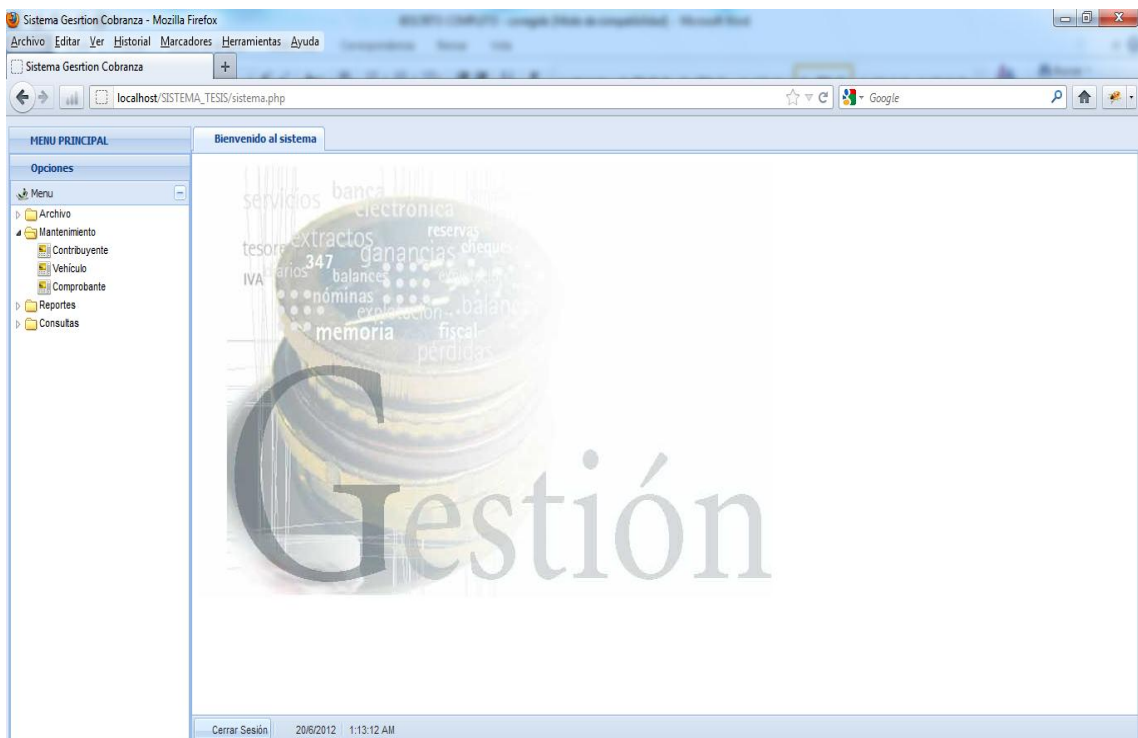
OBJETO	FUNCION	OBJETIVO	RESULTADO
<b>Función</b>	login	Iniciar la sesión	Se inicia la sesión ✓

[Tabla 6.22. Plan de Pruebas Alfa – Login



OBJETO	FUNCION	OBJETIVO	RESULTADO
<b>Función</b>	CargarInterfazPrincipal	Visualizar interfaz sistema	Interfaz del sistema ✓

Tabla 6.23. Plan de Pruebas Alfa – InterfazPrincipal



La prueba alfa se usa de forma natural con el desarrollador como observador del usuario, en el caso anterior se realizó con éxito la prueba alfa en relación al inicio de sesión del sistema y presentación de la interfaz.

#### 6.9.4.2 Prueba Beta

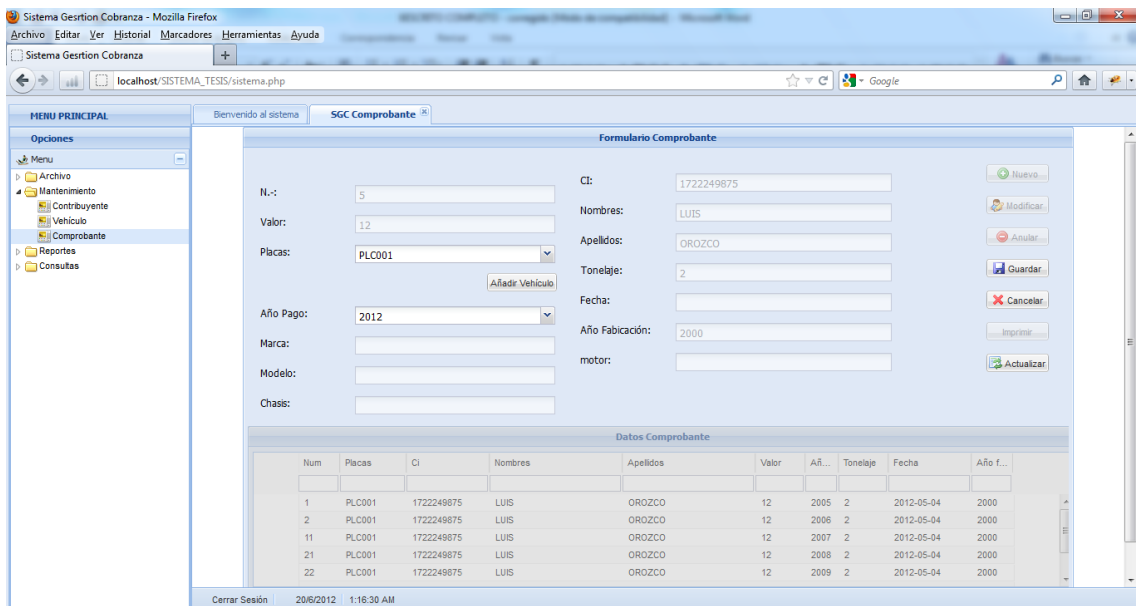
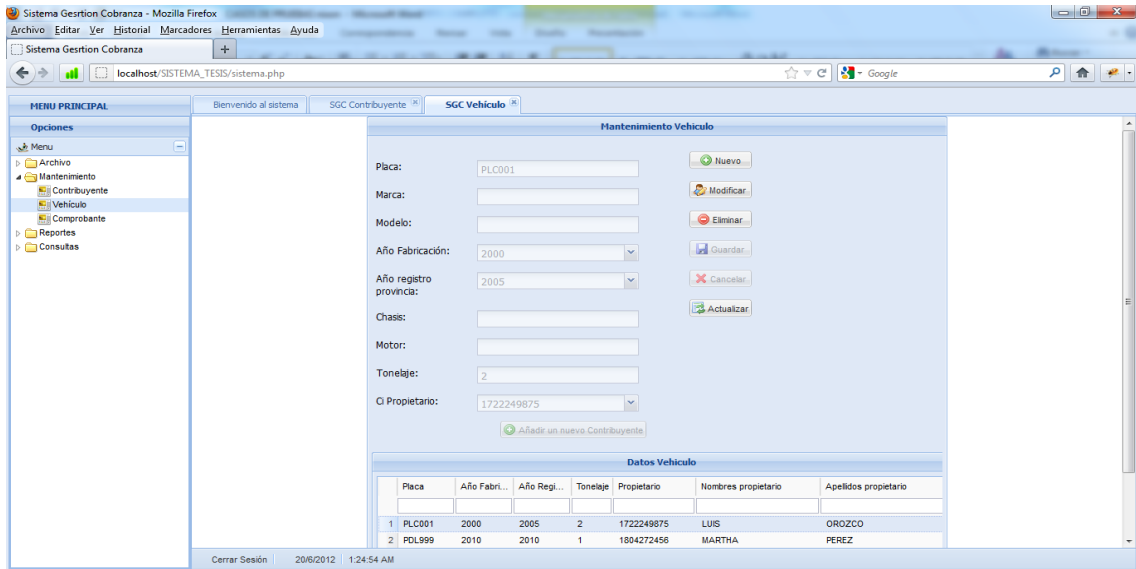
Se realizan pruebas a los usuarios que sean recaudadores, cada uno de ellos comprobo la funcionalidad de las tareas.

Usuario Recaudador	
<b>ESTRATEGIA PRUEBA</b>	Generación de Comprobantes
<b>TIPO DE PRUEBA</b>	Manual
<b>DESCRIPCION</b>	Se realiza el proceso de registro de contribuyentes, vehículos y finalmente la emisión del comprobante de pago.
<b>DESCRIPCION TÉCNICA</b>	Observación y Comprobación
	Ingreso y almacenamiento de datos Válidos.
	Generación de Comprobante con datos válidos y relacionados
<b>RESULTADOS ESPERADOS</b>	SI
<b>RESULTADOS OBTENIDOS</b>	Comprobante registrado

**Tabla 6.24.** Plan de Pruebas Beta

CI	Nombres	Apellidos	Dirección	Teléfono
172249875	LUIS	OROZCO	PILAAARO	098776555
1804272456	MARTHA	PEREZ	HUACHI	090887766
0501566202	ALBERTO	PEREZ	PILAAARO	090098989



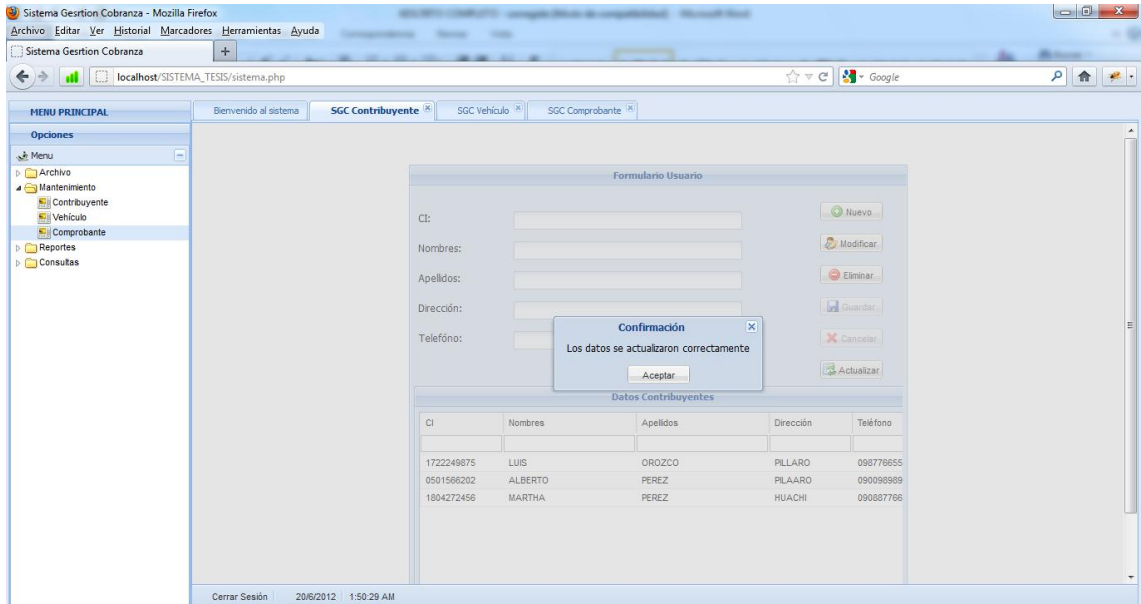


### 6.9.4.3 Pruebas de caja negra

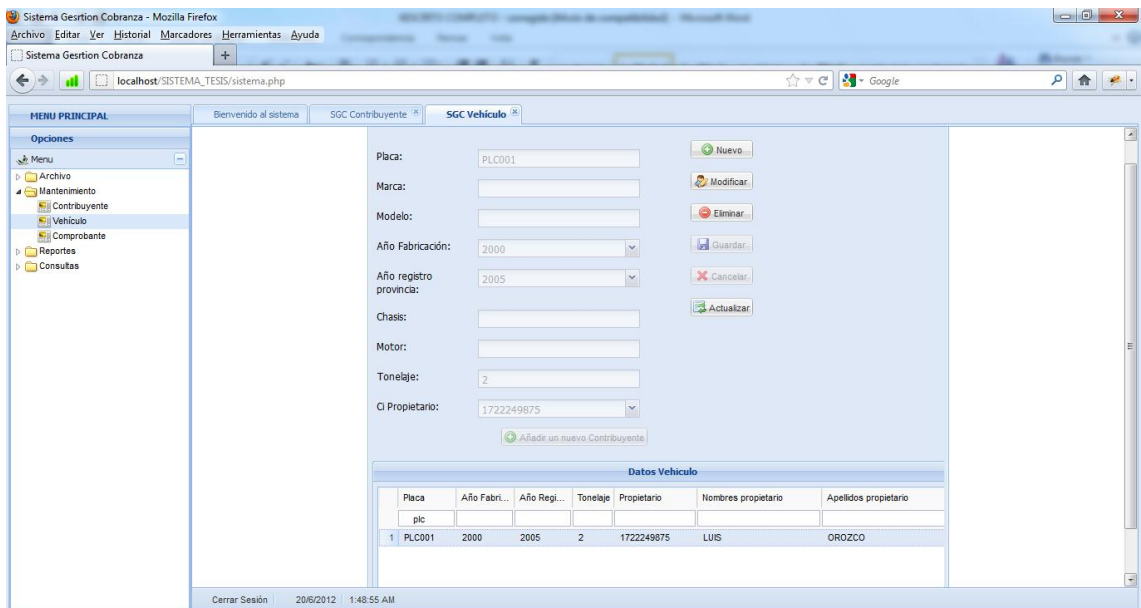
En este tipo de pruebas no se considera la codificación dentro de sus parámetros a evaluar, es decir, no está basado en el conocimiento interno del sistema. Estas pruebas se enfocan en la funcionalidad y requerimientos especificados del software.

Esta prueba se llevo a cabo sobre la interfaz del sistema web, por lo tanto queda demostrado que son completamente indiferentes del comportamiento interno del sistema.

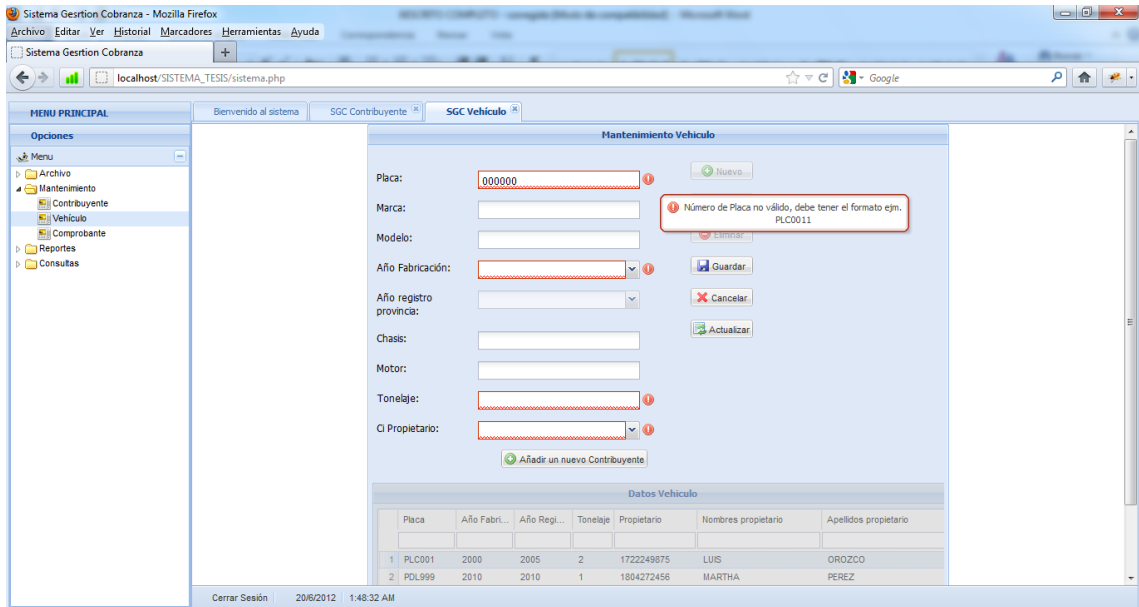
Las pruebas de caja negra para las administraciones se basan en la verificación contra la base de datos de las operaciones que se realice en la interfaz (inserción, eliminación, modificación), para el caso de la inserción que los datos sean coherentes y del tipo indicado.



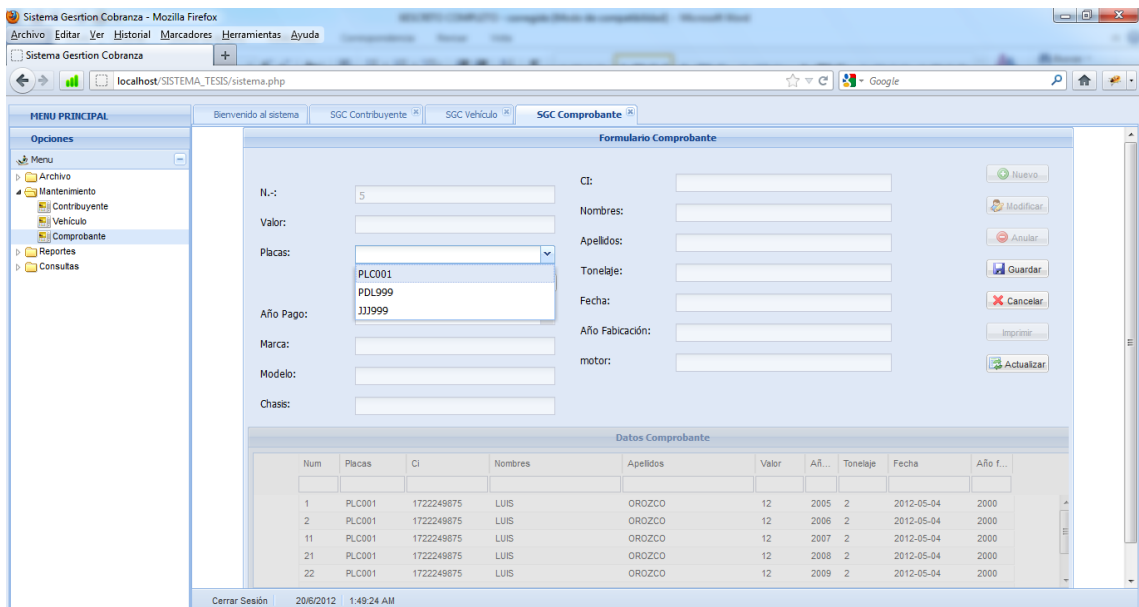
Que las búsquedas devuelvan los valores solicitados.



Validación de cajas de texto ingreso de caracteres específicos.



Habilitación de controles según el proceso.



Con estas pruebas se pudo demostrar que:

- El ingreso al sistema es fácil y sencillo
- Las funciones que realiza el sistema son claras y precisas.
- El sistema es de fácil navegación e interacción con el usuario.
- Las interfaces del sistema no permiten confusión en su manipulación.

- La información se almacena de una manera confiable y fiable, la recuperación de la misma es inmediata.
- Se evitó la duplicidad y redundancia de datos.
- Se mantiene la integridad del sistema a lo largo de su utilización.

Para obtener estas pruebas lo que se hizo es probar reiteradamente las entradas y salidas de los datos; es decir se evaluó la manera en que el sistema presenta los datos al usuario.

#### **6.9.4.4 Prueba de particiones o clases de equivalencia**

Clases de equivalencia método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba.

Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Típicamente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica.

Para generar los casos de prueba, se considera la técnica de Análisis de Valores Límite (AVL). Es una técnica de diseño de casos de prueba que completa a la partición equivalente. En lugar de seleccionar cualquier elemento de una clase de equivalencia, el AVL lleva a la elección de casos de prueba en los extremos de la clase. En lugar de centrarse solamente en las condiciones de entrada, el AVL obtiene casos de prueba también para el campo de salida.

Departamento			
Condición de Entrada	Tipo	Clase de Equivalencia Válida	Clase de Equivalencia No Válida
cod_dep	Valor	Cadena de 6 a 10 caracteres (1)	<6 (2) >10 caracteres (3)
nom_dep	Valor	Cadena de 3 a 20 caracteres (4)	< 3 caracteres (5) > 20 caracteres(6)
desc_dep	Valor	Cualquier cadena de caracteres alfanuméricos de 20 caracteres (7)	>20 dígitos caracteres (8)
eliminado	Valores Lógicos	Si (9) No(10)	No ha seleccionado (11)

**Tabla 6.25.** Tablas de clases de equivalencia tabla Departameto

### Casos de prueba con valores límites

#### Caso válido

- DEP100,CONTABILIDAD, CONTADORES, NO
- No. (1)(4)(7)(10)

#### Caso no válido

- 0, D, OOOOOOOOOO11111111111,
- No. (2)(5)(8)(11)

### 6.9.4.5 Pruebas de caja blanca

Las pruebas de caja blanca permiten examinar la estructura del programa, para esto se diseñaron casos de prueba para examinar la lógica del programa, además para evaluar la estructura de control del diseño procedimental es decir:

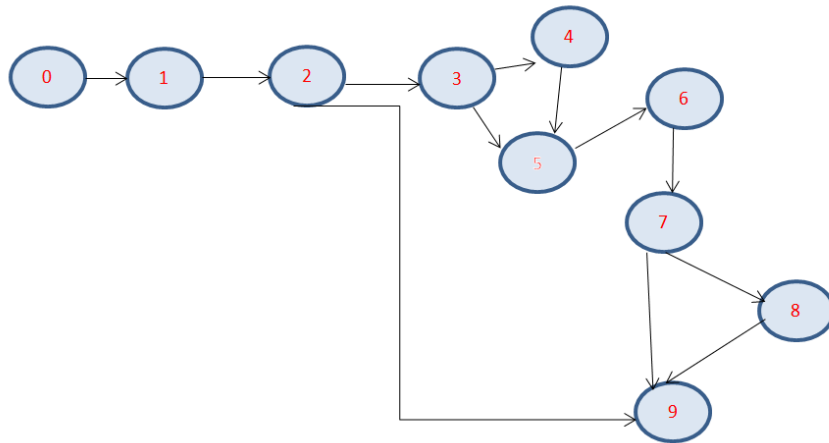
- Se evaluaron las decisiones lógicas.
- Se evaluaron las respuestas de los posibles errores a cometer.

- Se evaluaron las validaciones y controles existentes.
- Se evaluaron todos los bucles.

#### 6.9.4.6 Prueba del camino básico

El método del camino básico (propuesto por McCabe) permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez.

session_start(); if(\$_SESSION['estado'] == "logueado") {	0
\$cedula = \$_POST["cedula"]; //conectarse a la base de datos \$num = strlen(\$cedula); \$array = array(); for(\$i = 0; \$i < \$num; \$i++) { \$character = \$cedula[\$i]; array_push(\$array,\$character); }	1
if ( \$num == 10 ) {	2
\$total = 0; \$digito = (\$array[9]*1); for( \$i=0; \$i < (\$num-1); \$i++ ) {	3
\$mult = 0; if ( ( \$i%2 ) != 0 ) { \$total = \$total + ( \$array[\$i] * 1 ); } else	4
{ \$mult = \$array[\$i] * 2; if ( \$mult > 9 ) \$total = \$total + ( \$mult - 9 ); else \$total = \$total + \$mult; } }	5
\$decena = \$total / 10; \$decena = floor( \$decena ); \$decena = ( \$decena + 1 ) * 10; \$final = ( \$decena - \$total );	6
if ( ( \$final == 10 && \$digito == 0 )    ( \$final == \$digito ) ) { echo 1; }	7
else { echo 0; }	8
else { echo 1; }	9



**Figura 6.29.-** Grafo de validación de la cédula

### Complejidad Ciclomática

$$V(G) = A \text{ (aristas)} - N \text{ (nodos)} + 2$$

$$V(G) = 12 - 10 + 2 = 4$$

### Caminos Básicos

#1: 0-1-2-9

#2: 0-1-2-3-4-5-6-7-9

#3: 0-1-2-3-5-6-7-9

#4: 0-1-2-3-4-5-6-7-8-9

### Comprobación de la prueba del camino básico.

Se tomará como referencia el camino #2:

Ingresa una cadena de caracteres para la cedula (1805906179001). Nodo0

Se inicializa las variables auxiliares. Nodo1

Comprobar que la longitud de la cadena ingresada,

```

(for($i = 0; $i < $num; $i++) {
    $character = $cedula[$i];
    array_push($array,$character);
}). Corresponda a una cedula Nodo1
  
```

Si la cadena de caracteres, es menor diferente de 10 retorna falso para indicar que no es una cédula. Caso contrario sacamos el último dígito de la cédula para comparar. *Nodo2*

```
$total = 0;
```

```
$digito = ($array[9]*1);
```

*Creamos un ciclo y empezamos a sacar los porcentajes. *Nodo3*.*

*Llenamos en una variable para comparar al final dependiendo de las condiciones*

*Nodo5. *Nodo5**

```
for( $i=0; $i < ($num-1); $i++ )  
{  
    $mult = 0;  
    if ( ( $i%2 ) != 0 )  
    {  
        $total = $total + ( $array[$i] * 1 );  
    }  
    else  
    {  
        $mult = $array[$i] * 2;  
        if ( $mult > 9 )  
            $total = $total + ( $mult - 9 );  
        else  
            $total = $total + $mult;  
    }  
}
```

Calculamos los totales para comparar si es una cédula válida o no. *Nodo 6*

```
$decena = $total / 10;
```

```
$decena = floor( $decena );
```

```
$decena = ( $decena + 1 ) * 10;
```

```
$final = ( $decena - $total );
```

Determinamos si es válida o no dependiendo las condiciones del valor total y devolvemos una respuesta. *Nodo7*

```
if ( ( $final == 10 && $digito == 0 ) || ( $final == $digito ) )  
{  
    echo 1;  
}
```



```

else
{
echo 0;
}

```

## 6.9.4.7 Pruebas de Verificación y Validación

### 6.9.4.7.1 Verificación

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

### Eliminación

Se debe controlar que la eliminación se efectue siempre y cuando no tenga relaciones que afecten a la consistencia de datos..

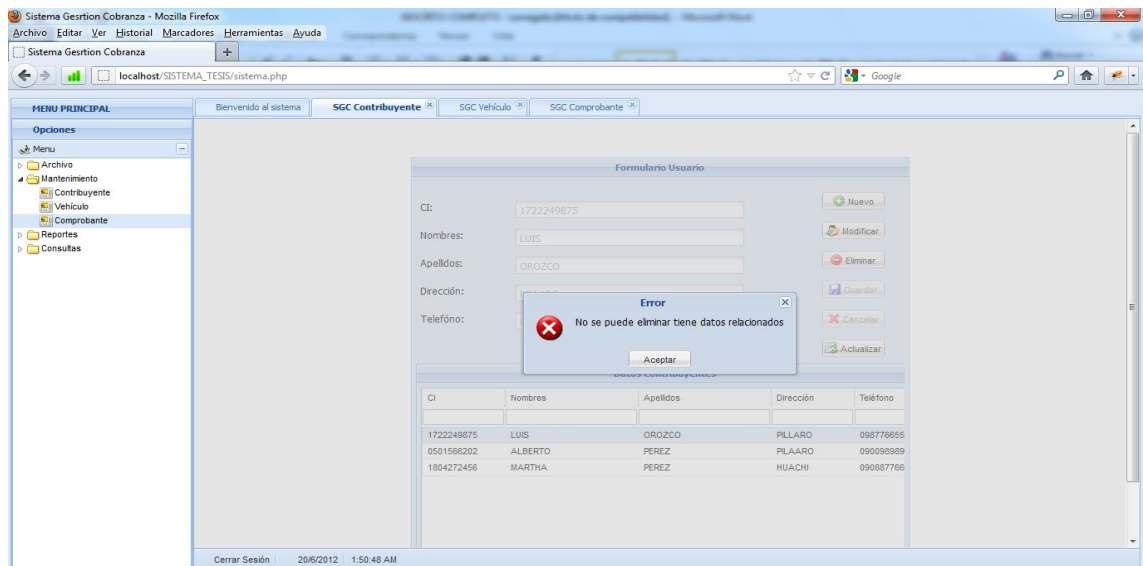


Figura 6.30.- Control de eliminación de registros

### 6.9.4.7.2 Validación

Esta prueba verificó si el Sitio cumple con las expectativas del cliente.

## VALIDACIÓN DE DATOS

El sistema valida el ingreso de datos como:

### Validación de la cedula de Usuario:

Al ingresar en número de cedula incorrecto aparecerá el siguiente mensaje.

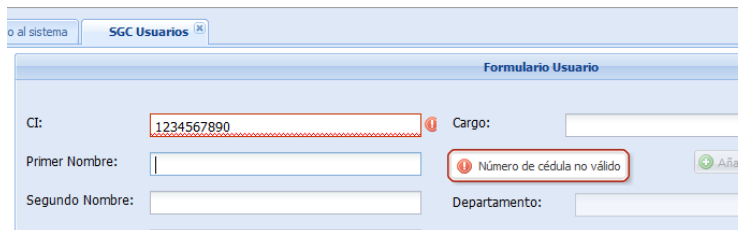


Figura 6.31.- Validación de la cedula del usuario

### Validación de Ingreso de usuario.

Al ingresar o digitar de forma errónea el nombre del usuario y/o la contraseña para ingresar al sistema.

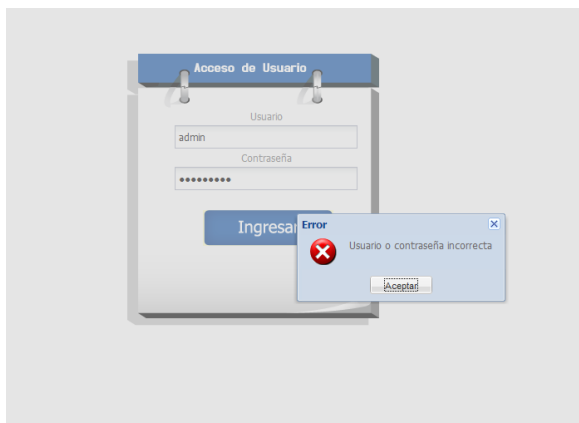


Figura 6.32.- Validación ingreso sistema

## 6.9.5 Implantación

El proceso de implantación de un sistema informático es uno de los últimos pasos en el ciclo de vida del software, y el producto de largas horas de trabajo.

### 6.9.5.1 Implantación de carpetas y archivos necesarios

El Gobierno Provincial de Tungurahua gracias al contar con le dominio correspondiente y el servidor para alojar los archivos necesarios para el funcionamiento del sistema web, el mismo que se puso en la intranet de la

institución que tiene salida a Internet para las pruebas necesarias antes de la publicación y difusión del sistema.

### 6.9.5.2 Instalación del motor de base de datos

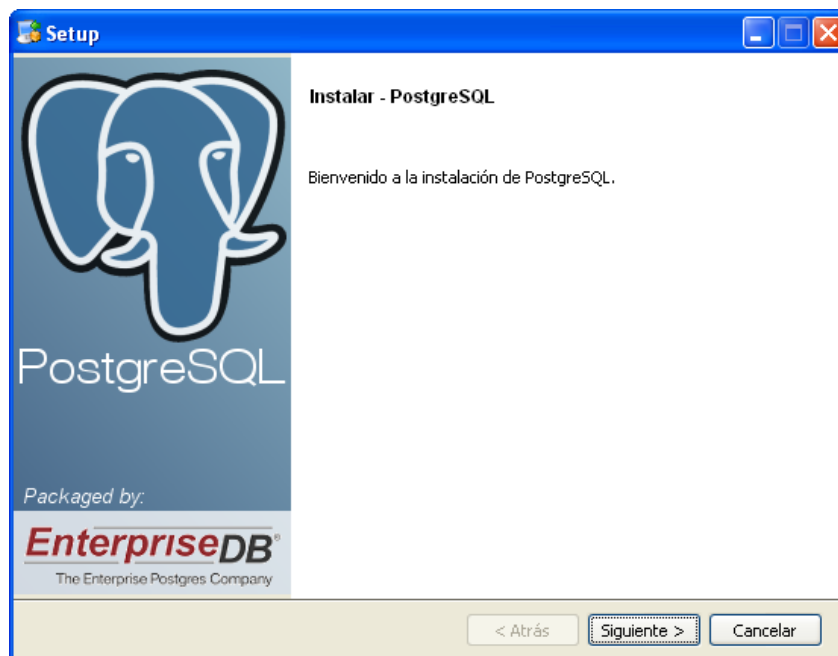
El sistema de gestión de cobranza trabaja con PostgreSQL para la parte de Base de datos motivo por el cual se procedió a instalarlo en el servidor web.

Como primer paso se efectuó la descarga del paquete llamado:

postgresql-8.4.2-1-windows.exe

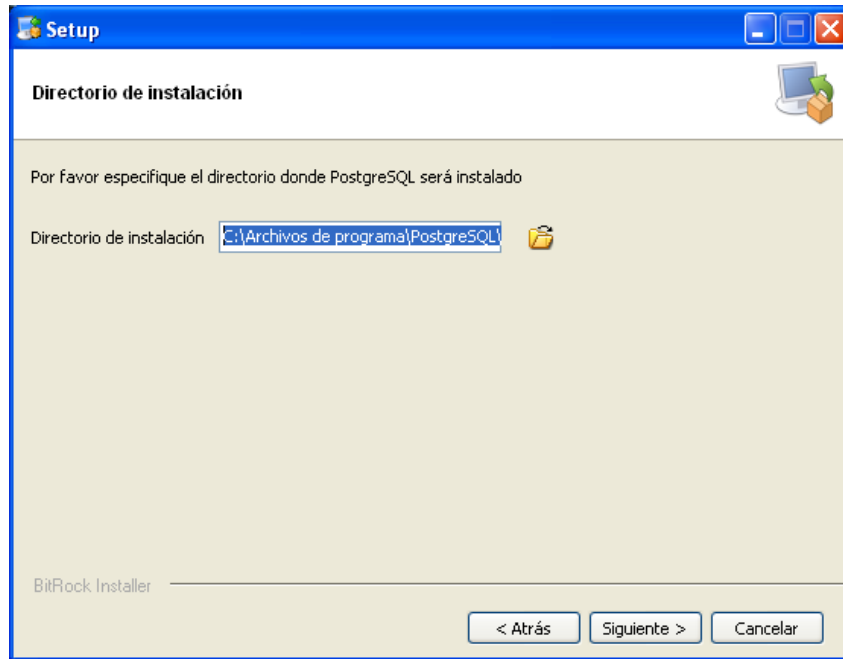
Luego se procedió a realizar su ejecución en el sistema.

Una vez ejecutado el se generó un asistente grafico para seguir con la instalación.



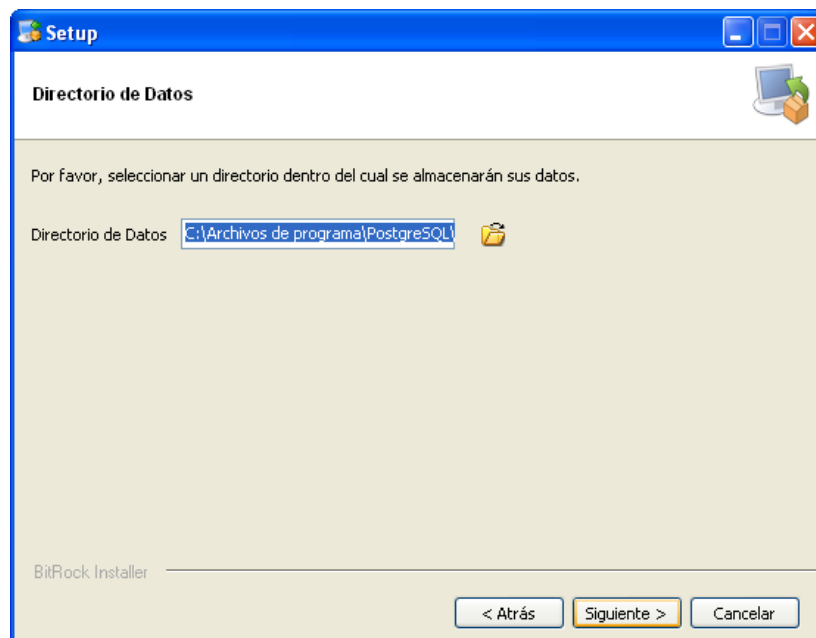
**Figura 6.33.** Instalar Postgres

En este paso seleccionamos la dirección de instalación de PostgreSQL



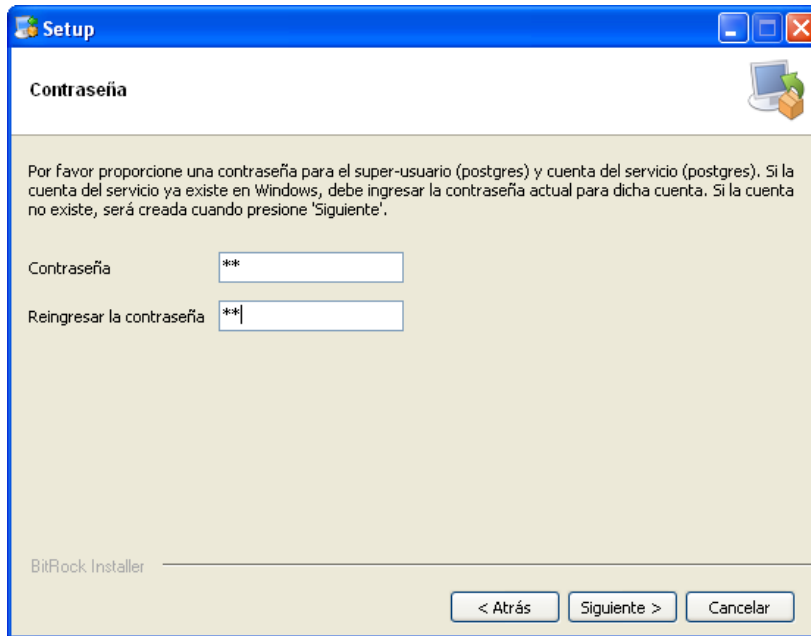
**Figura 6.34.** Directorio de instalación

A continuación seleccionamos el directorio dentro del cual se almacenarán los datos



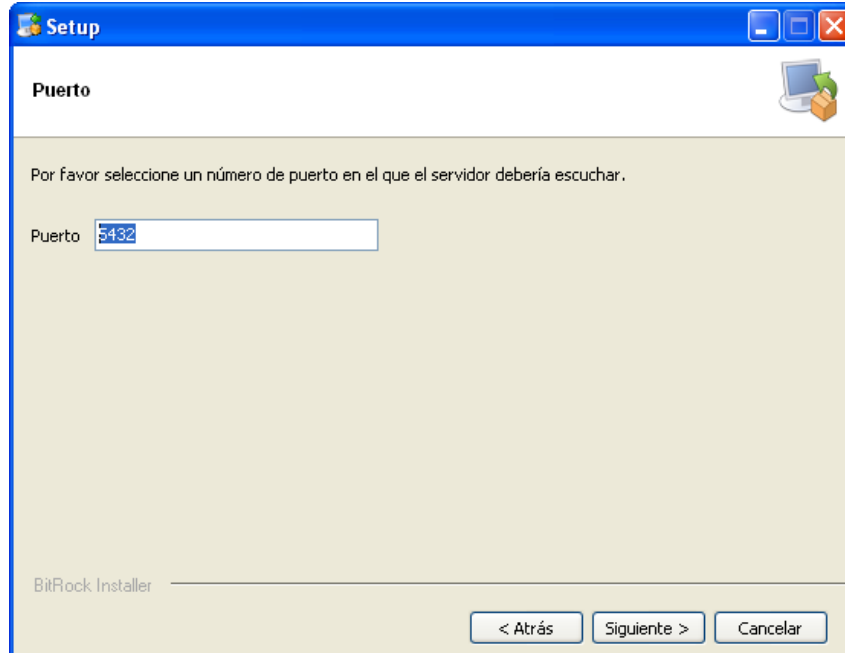
**Figura 6.35.** Directorio de datos

Como paso siguiente se procedió a ingresar la contraseña del usuario postgres



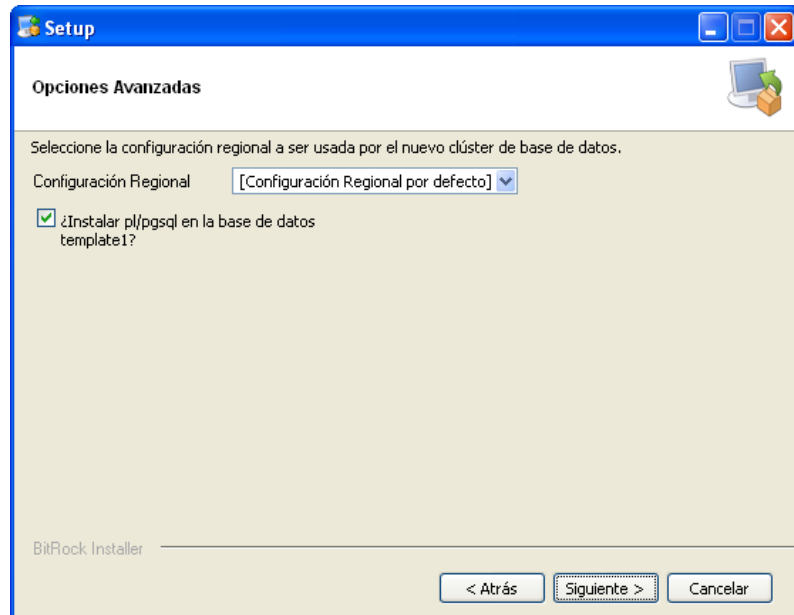
**Figura 6.36.** Contraseña del usuario postgres

En este paso ingresamos el número de puerto en el que el servidor debería escuchar



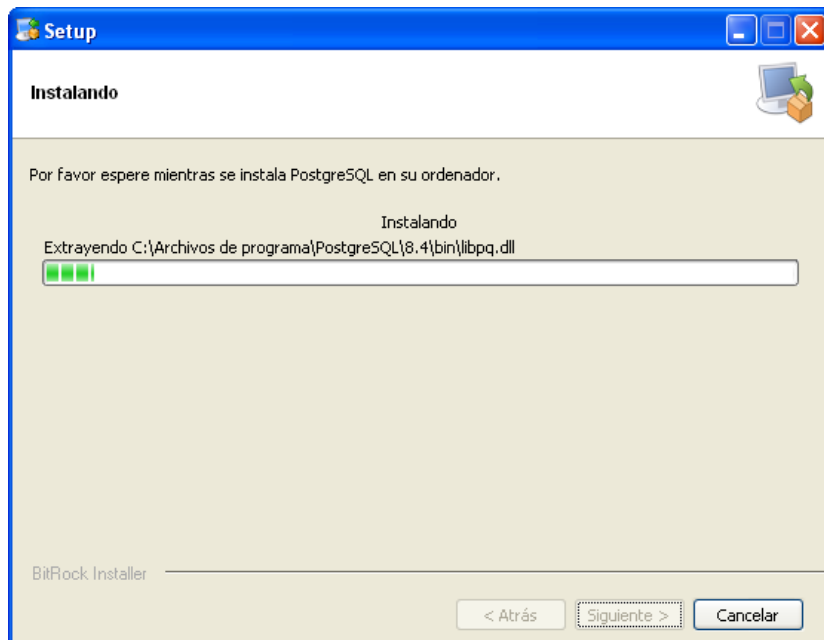
**Figura 6.37.** Puerto postgres.

Configuramos el idioma

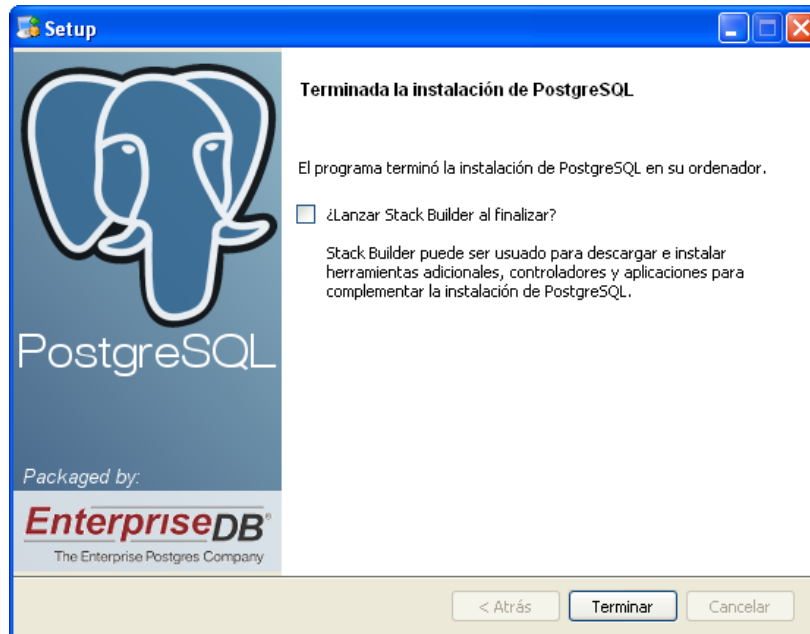


**Figura 6.38.** Configuración del idioma.

Como último paso ejecutamos la instalación en el servidor



**Figura 6.39.** Listo para instalar.



**Figura 6.40.** Instalado

### 6.9.5.3 Instalación del sistema de gestión de cobranza

#### Primer paso: Base de datos

Se procedió a crear la base de datos que utilizará el Sistema en PostgreSQL con el nombre “SGCBD”, a continuación mediante la opción restaurar, seleccionaremos el archivo que contiene el script de la base de datos y lo subimos, al finalizar aparecerán las tablas, funciones y triggers a ser utilizados por el sistema.

Al crear la base de datos, también se crea también 3 datos fundamentales como son:

- “DEPARTAMENTO DE SISTEMAS” en la tabla departamento
- El cargo “ADMINISTRADOR” en la tabla cargos
- Un usuario que tiene el cargo de administrador en la tabla usuarios

### **Segundo paso: Subir archivos al servidor**

Instalar la herramienta wamp server, la cual permite instalar PHP y apache web server en el servidor, software que es necesario para que se corra el sistema en el ambiente web cliente servidor.

Subir al servidor web completamente los archivos del sistema, en el directorio www del servidor configurado, para evitar errores en la utilización.

Poner online la herramienta wamp server para la publicación del directorio donde tenemos almacenado nuestro sistema.

### **Tercer paso: Configuraciones.**

Este paso es fundamental para el funcionamiento del sistema. Para que el sistema se conecte con la base de datos creada se necesita configurar el archivo conectaBD.php ubicado en el directorio accesoBD del sistema, para ello utilizamos cualquier editor de texto, y modificar el archivo de configuración de PHP para descomentar la línea que permite conectar el PostgreSQL y PHP.

En la parte de la función **pg\_connect** existen 5 parametros que están definidos como constantes los cuales deben ser modificados dependiendo las necesidades.

#### **6.9.5.4 Aplicación de pruebas en el sistema**

Luego de que el sistema fue implantado en el servidor web de la institución, se procedió a verificar que su funcionamiento sea el mismo al ejecutado y aprobado por el director del departamento de sistemas del Gobierno Provincial de Tungurahua en el ambiente de desarrollo obteniéndose los siguientes resultados:

- El diseño de la interfaz del sistema no tuvo ningún inconveniente, las imágenes, colores y estilos funcionaron correctamente en los diferentes navegadores existentes.
- La gestión de información de la base de datos a través del sistema no generó ningún error.



- Se verificó el funcionamiento correcto de cada uno de los formularios web pertenecientes al sistema.

Posteriormente el sistema fue sometido a un periodo de prueba en el ambiente de recaudación de valores correspondientes a la contribución vial del departamento financiero del Gobierno Provincial de Tungurahua, con la finalidad de constatar y evaluar su correcto funcionamiento en escenarios reales.

#### **6.9.5.5 Capacitación al personal sobre el uso y administración del sistema.**

A pesar de que el sistema informático fue realizado a medida y existió una constante comunicación con el personal del departamento en cada fase de desarrollo del software, se realizó una charla de capacitación al director de sistemas sobre el funcionamiento y administración del mismo, y con juntamente con él se ejecutó una conferencia donde se informó al resto del personal las bondades, características y la forma correcta de utilización del sistema informático, con la finalidad de aprovechar al máximo los recursos que éste nos ofrece.

Otra forma de preparación y guía para la utilización del sistema informático fue la laboración del manual de usuario, el mismo que permitirá resolver dudas que se presenten en el futuro, sin la presencia del desarrollador del software.

#### **6.9.5.6 Evaluaciones**

- En cuanto a la parte técnica, el sistema se acoplo adecuadamente al servidor del Gobierno Provincial (Base de datos y sistema web).
- En el lapso que el sistema estuvo a prueba en el ambiente real del proceso de cobranza, se pudo encontrar y corregir errores que pasaron desapercibidos en fases anteriores, factor que permitió cumplir con el objetivo de implantar el sistema en el departamento.
- El sistema implantado es de calidad, porque a través de él se logró satisfacer las expectativas y necesidades del departamento con respecto a la gestión de cobranza

## 6.10 Conclusiones y Recomendaciones

### 6.10.1 Conclusiones

- Se desarrolló e implantó un sistema web de gestión de cobranza de valores que permitió dar solución al manejo inadecuado de la información de contribuciones viales en el departamento financiero del Gobierno Provincial de Tungurahua.
- Mediante la utilización de Diagramas UML, se comprendió de mejor manera, la forma que interactúa el usuario con el sistema.
- En cuanto al desarrollo web con PHP, es más que un lenguaje de programación es una tecnología de servidor con poderosas herramientas que facilitan el trabajo a los desarrolladores.
- La base de datos creada en PostgreSQL cuenta con todas las características para brindar seguridad, y confianza en el almacenamiento de información del sistema.
- ExtJs demostró ser un poderoso framework capaz de realizar interfaces difíciles o casi imposibles con otras tecnologías como JSF , RichFaces , etc. y lo que es más importante encapsulando aspectos cruciales como la funcionalidad multibrowser , los accesos asíncronos , la transformación de objetos de servidor a objetos javascript , lo mejor de lo mejor de librerías tales como: jquery , mootools, prototype ,etc ; ofreciendo además un mecanismo de conexión con otras librerías AJAX mediante el uso de adaptadores.
- La interfaz del sistema es de fácil navegación y tiene búsquedas inteligentes, para que el usuario pueda interactuar de la mejor manera con el sistema y recupere datos fácilmente.
- El desarrollo del sistema fortaleció los conocimientos adquiridos en el transcurso de la vida estudiantil en la Universidad Técnica de Ambato, incrementando la capacidad de auto educación e investigación para cumplir con los objetivos planteados.

### 6.10.2 Recomendaciones

- Se recomienda designar a una persona capacitada para el mantenimiento y control del sistema informático y base de datos, para asegurar un mejor desempeño y el correcto funcionamiento del mismo.
- Realizar respaldos periódicos de la base de datos para salvaguardar la información de la institución.
- Implementar un servidor web dedicado, que cumpla todas las especificaciones de seguridad web implementadas en el proyecto.
- Es recomendable utilizar navegadores como Firefox desde la versión 2.5 en adelante para la administración del sistema informático, teniendo en cuenta el soporte JavaScript ya que el sistema está desarrollado en una parte considerable con este lenguaje de programación.
- El desarrollo del proyecto se benefició de las tecnologías open source las cuales permitieron un desarrollo ágil en la fase de construcción. En relación a la arquitectura, se puede destacar la versatilidad y poder de PHP en el lado del servidor; mientras que en el lado del cliente la robustez de Extjs aseguran el desarrollo de aplicaciones RIA (Rich Internet Applications) exitosas, por lo cual se recomienda su uso.

## **Bibliografía**

### **Información bibliográfica de libros**

- JESUS GARCIA, Extjs in Action, Manning Publishing; First Edition, 2010.
- STEVE BLADES, Learning Extjs, Packt Publishing, First Edition, 2009.
- TIM CONVERSE, Biblia del PHP5 y Mysql , First Edition , John Wiley and Sons , 2005.
- POWERS David, The Essential Guide to Dreamweaver CS3 with CSS, Ajax, and PHP, 2007, ISBN: 978-1-59059-859-7
- Secure PHP Development Mohammed J. Kabir - Ed. Wiley Publishing.

### **Información bibliográfica de páginas web**

- “Ejemplos básicos para programar con php.” Disponible en: <http://www.webtaller.com/construccion/lenguajes/info/lecciones/php/>
- “Ejemplos de cómo crear un pdf en una página web” Disponible en: [http://www.webtaller.com/construccion/lenguajes/php/lecciones/crear\\_documento\\_s\\_pdf\\_usando\\_php.php](http://www.webtaller.com/construccion/lenguajes/php/lecciones/crear_documento_s_pdf_usando_php.php).
- “PHP Security – Open Source Convention” Disponible en: <http://shiflett.org/talks/oscon2004/php-security>
- “Open Web Application Security Project” Disponible en: <http://www.owasp.org/>

## Glosario de términos

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

**SQL:** (Structured Query Language), es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.

**Php.-** (Hypertext Pre-Processor). Lenguaje de programación usado generalmente en la creación de contenidos para sitios web. Es un lenguaje interpretado especialmente usado para crear contenido dinámico web y aplicaciones para servidores.

**UML:** (Unified Modeling Language), es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

**Open Source:** Es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones morales y/o filosóficas las cuales destacan en el llamado software libre.

**Interfaz Amigable,** Calidad de una interfaz de programa que por su forma de interactuar con el usuario es considerada de fácil uso.

**AJAX,** Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.

**JSON**, es un formato ligero de intercambio de datos que desde hace algún tiempo ha comenzado a usarse de una forma tan popular como se uso en su tiempo el XML. JSON es la abreviación de JavaScript Object Notación – Notación de Objetos de JavaScript – También es subconjunto del estándar ECMA 262 publicado en diciembre de 1999. El formato de JSON es ampliamente reconocido por una gran variedad de lenguajes como Java, PHP, JavaScript, C++, C# entre otros, la lista completa se puede consultar desde el sitio web de JSON.

**EXTJS**, es una librería Javascript ligera y de alto rendimiento que permite construir aplicaciones complejas en Internet; compatible con la mayoría de navegadores para crear páginas web y aplicaciones dinámicas.

# **ANEXOS**

## ANEXO 1: Estructura del Cuestionario

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

#### **CARRERA DE INGENIERÍA EN SISTEMAS**

Cuestionario dirigido al personal que labora en el departamento de cobranza de valores de la contribución vial del Gobierno Provincial de Tungurahua.

Sirvase a contestar el cuestionario de la manera más honesta posible, puesto que la información servirá exclusivamente para mejorar la productividad del departamento.

#### **INSTRUCCIONES:**

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

**1. ¿De qué manera se maneja la gestión de cobranza de valores de la contribución vial de la Provincia?**

- Manualmente
- Sistema Informático
- Excel
- Otros

**2. ¿El sistema o mecanismo utilizado actualmente en el área de gestión de cobranza de valores de la contribución vial de la provincia, permite un control adecuado del proceso de valores a los contribuyentes?**

- Si
- No

**3. ¿El sistema o mecanismo utilizado actualmente en el área de gestión de cobranza de valores de la contribución vial de la provincia, permite la generación de reportes oportunos?**

- Si
- No



**4.- ¿La información almacenada de los contribuyentes y valores recaudados es confiable?**

- Si
- No

**5.- ¿Cree usted que es necesario la implantación de un sistema web para agilizar la gestión de cobranza?**

- Si
- No

**6.- ¿Cree que el uso del Sistema Web optimice tiempos?**

- Si
- No

**7.- ¿En cuanto beneficiará a los recaudadores la implementación del sistema web?**

- Mucho
- Poco
- Nada

**8.- ¿Con la automatización del proceso de almacenamiento de información se realizará una mejor administración del departamento de cobranzas?**

- Si
- No

## **ANEXO 2: Decreto presidencial para el uso de software libre en la república del Ecuador**

**Artículo 1.-** Establecer como política pública para las Entidades de la Administración Pública Central la utilización de Software Libre en sus sistemas y equipamientos informáticos.

**Artículo 2.-** Se entiende por Software Libre, a los programas de computación que se pueden utilizar y distribuir sin restricción alguna, que permitan su acceso a los códigos fuentes y que sus aplicaciones. puedan ser mejoradas. Estos programas de computación tienen las siguientes libertades:

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

**Artículo 3.-** Las entidades de la Administración Pública Central previa a la instalación del software libre en sus equipos, deberán verificar la existencia de capacidad técnica que brinde el soporte necesario para el uso de éste tipo de software.

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

En este caso, se concibe como seguridad nacional, las garantías para la supervivencia de la colectividad y la defensa del patrimonio nacional. 161

Para efectos de este decreto se entiende por un punto de no retorno, cuando el sistema o proyecto informático se encuentre en cualquiera des estas condiciones:

- a) Sistema en producción funcionando satisfactoriamente y que un análisis de costo – beneficio muestre que no es razonable ni conveniente una migración a Software Libre.

b) Proyecto en estado de desarrollo y que un análisis de costo – beneficio muestre que no es conveniente modificar el proyecto y utilizar Software Libre.

Periódicamente se evaluarán los sistemas informáticos que utilizan software propietario con la finalidad de migrarlos a Software Libre.

**Artículo 5.-** Tanto para software libre como software propietario, siempre y cuando se satisfagan los requerimientos, se debe preferir las soluciones en este orden:

- a) Nacionales que permitan autonomía y soberanía tecnológica.
- b) Regionales con componente nacional.
- c) Regionales con proveedores nacionales.
- d) Internacionales con componente nacional.
- e) Internacionales con proveedores nacionales.
- f) Internacionales.

**Artículo 6.-** La Subsecretaría de Informática como órgano regulador y ejecutor de las políticas y proyectos informáticos en las entidades del Gobierno Central deberá realizar el control y seguimiento de éste Decreto.

Para todas las evaluaciones constantes en este decreto la Subsecretaría de Informática establecerá los parámetros y metodologías obligatorias.

**Artículo 7.-** Encárguese de la ejecución de este decreto a los señores Ministros Coordinadores y el señor Secretario General de la Administración Pública y Comunicación. 162

Dado en el Palacio Nacional en la ciudad de San Francisco de Quito, Distrito Metropolitano, el día de hoy 10 de abril del 2008.

**RAFAEL CORREA DELGADO**

**PRESIDENTE CONSTITUCIONAL DE LA REPÚBLICA**

### ANEXO 3: Manual de usuario

Para poder ingresar al sistema se debe ingresar a la dirección **www.xxxx.com**, ingresando a la dirección se redirecciona a la página de login de usuario.

Todas las páginas para los usuarios visitantes contendrán lo siguiente:

- Menú Principal en la parte izquierda del navegador, ahí están los vínculos para los principales servicios del sistema web.
- Ventana de contenido. Espacio destinado para la gestión de información. La información mostrada dependerá directamente de las opciones marcadas en los menús.
- ToolBar inferior con los datos de fecha, hora y la opción de cerrar sesión de usuario.

#### **Pantalla de inicio de sesión.**

Cuando se accede al sistema la primera pantalla muestra un formulario en el cual se debe ingresar el nombre de usuario y contraseña.

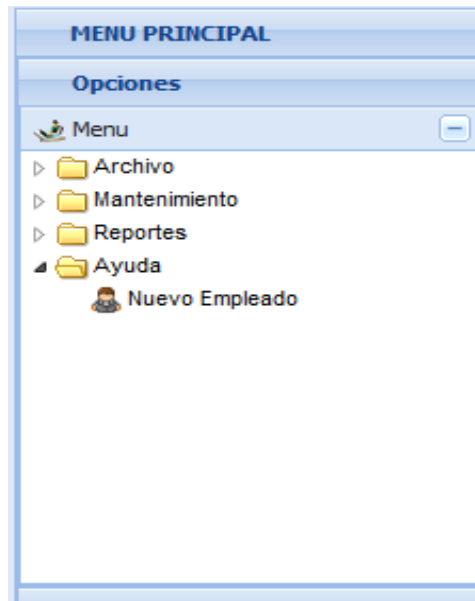
The image shows a login form titled "Acceso de Usuario" (User Access) presented as a document on a clipboard. The form contains two input fields: "Usuario" (User) and "Contraseña" (Password). Below these fields is a blue button labeled "Ingresar" (Login). The entire form is set against a light gray background with a subtle drop shadow.

**Figura A3.1** Pantalla de inicio de sesión

En esta página el usuario que desea entrar al sistema deberá ingresar el nombre de usuario y su contraseña datos que serán validados y permitirán su acceso.

## Menú administrador

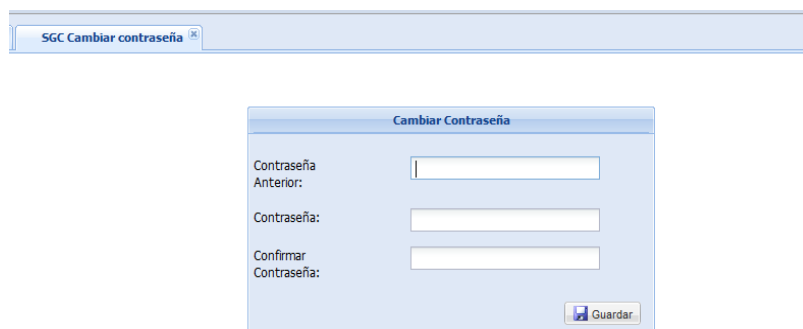
Cuando el usuario se encuentre habilitado aparecerá un menú con las siguientes opciones: Archivo-Mantenimiento-Reportes-Ayuda:



**Figura A3.2** Menú administrador

## Pantalla Cambio de contraseña

Seleccionando esta opción aparecerán los textos para ingresar la nueva contraseña.



**Figura A3.3** Cambio de contraseña

Los botones que se muestra son el de guardar. Guardamos los datos una vez ingresadas las claves y que todos los datos estén correctos.

## Pantalla mantenimiento departamentos

Esta página contiene la información básica de los departamentos del Gobierno Provincial, existen botones como nuevo, guardar, modificar, eliminar, cancelar.

The screenshot shows a web application interface for department management. On the left is a navigation menu with options like 'Departamentos', 'Cargos', 'Usuarios', etc. The main area contains a 'Formulario Departamentos' with input fields for 'Código', 'Nombre', and 'Descripción', and buttons for 'Nuevo', 'Guardar', 'Modificar', 'Eliminar', and 'Cancelar'. Below the form is a table titled 'Datos Departamento' with columns for 'Código Departamento', 'Nombre Departamento', and 'Descripción'. The table contains one entry: 'DEP1', 'SISTEMAS', and 'DEPARTAMENTO DE SISTEMAS DEL GOBIERNO PROVINCIAL...'. The footer shows 'Cerrar Sesión | 07/3/2012 | 8:41:56 PM'.

**Figura A3.4** Datos de departamentos

En esta pantalla podemos agregar un registro con la opción nuevo, a la vez podemos eliminar y modificar un registro.

## Pantalla mantenimiento cargos

Esta página contiene la información básica de los cargos que serán asignados a los usuarios del sistema, existen botones como nuevo, guardar, modificar, eliminar, cancelar.

The screenshot shows a web application interface for job management. On the left is a navigation menu with options like 'Cargos' and 'Usuarios'. The main area contains a 'Formulario Cargo' with input fields for 'Código Cargo', 'Nombre Cargo', 'Descripción Cargo', and 'Departamento', and buttons for 'Nuevo', 'Guardar', 'Modificar', 'Eliminar', and 'Cancelar'. Below the form is a table titled 'Datos Cargo' with columns for 'Código Cargo', 'Nombre Cargo', 'Descripción', and 'Departamento'. The table contains one entry: 'CARGO1', 'ADMINISTRADOR', 'ADMINISTRADOR DEL SISTEMA', and 'SISTEMAS'. The footer shows 'Cerrar Sesión | 07/3/2012 | 8:51:54 PM'.

**Figura A3.5** Mantenimiento Cargos

En esta pantalla podemos modificar cualquier registro de los cargos como son insertar, eliminar y modificar cargos existentes en los diferentes departamentos del Gobierno Provincial.

### **Pantalla mantenimiento usuarios**

Esta página contiene la información básica de los usuarios que serán resgistrados en el sistema, existen botones como nuevo, guardar, modificar, eliminar, cancelar.

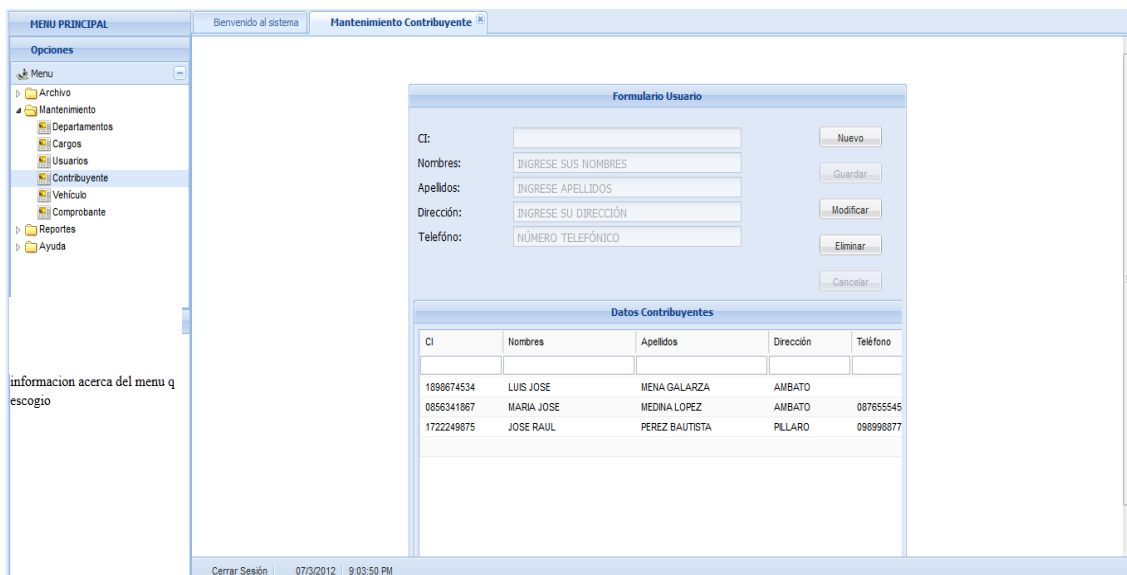
CI	Primer Nombre	Segundo No...	Primer Apell...	Segundo A...	Dirección	Teléfono	E-mail
1722249875	ALBERTO	EDUARDO	PEREZ	BAUTISTA	ZAMBA	032421820	btoperes@hotma

**Figura A3.6** Forma mantenimiento usuario

En esta pantalla podemos manipular los datos de los usuarios existentes en el sistema ya sea agregando, eliminando o modificando un empleado.

### **Pantalla mantenimiento contribuyentes**

Esta página contiene la información de los contribuyentes a los que se les hará el seguimiento de los pagos, existen botones como nuevo, guardar, modificar, eliminar, cancelar.

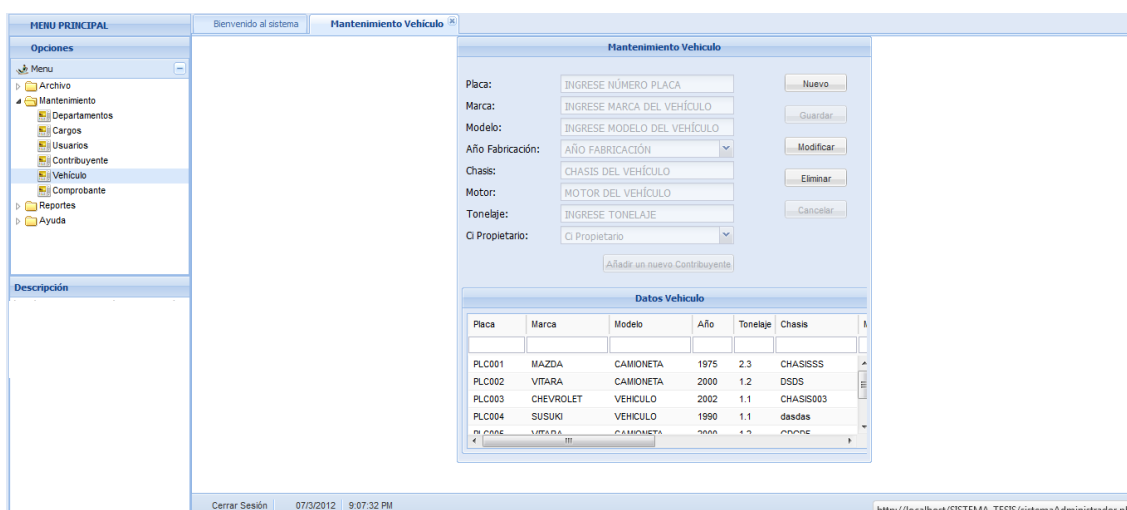


**Figura A3.7** Mantenimiento contribuyentes

En esta pantalla podemos manipular los datos de los contribuyentes existentes en el sistema ya sea agregando, eliminando o modificando los datos.

### Pantalla mantenimiento vehículos

Esta página contiene la información de los vehículos ingresados en el sistema, existen botones como nuevo, guardar, modificar, eliminar, cancelar.



**Figura A3.8** Mantenimiento vehículo

En esta pantalla podemos manipular los datos de los vehículos registrados en el sistema ya sea agregando, eliminando o modificando los datos.



## Pantalla mantenimiento comprobantes

Esta página contiene la información de los comprobantes generados con el sistema, existen botones como nuevo, guardar, modificar, eliminar, cancelar.

The screenshot shows a web application interface for 'Mantenimiento Comprobante'. On the left is a 'MENU PRINCIPAL' sidebar with 'Comprobante' selected. The main area is divided into two sections: 'Formulario Comprobante' and 'Datos Comprobante'. The form contains fields for 'Número Comprobante', 'Placas', 'Año Pago', 'Marca', 'Modelo', 'Chasis', 'CI', 'Nombres', 'Apellidos', 'Tonelaje', 'Fecha', 'Año Fabricación', and 'motor'. Below the form is a table with the following data:

Num	Placas	CI	Nombres	Apellidos	Valor	Añ...	Marc
77	PLC005	0856341967	MARIA JOSE	MEDINA LOPEZ	12	2011	VI
79	PLC003	0856341967	MARIA JOSE	MEDINA LOPEZ	12	2006	Ch
80	PLC005	0856341967	MARIA JOSE	MEDINA LOPEZ	12	2012	VI
84	PLC003	0856341967	MARIA JOSE	MEDINA LOPEZ	12	2007	Ch
81	PLC004	1008274634	LUIS JOSE	MENA CALADIA	10	2010	VI

Figura A3.9 Mantenimiento comprobantes

En esta pantalla podemos manipular los datos de los comprobantes generados con el sistema ya sea agregando, eliminando o modificando los datos.

## Pantalla reporte cierre diario

Esta página mostrará un reporte con los datos de los comprobantes generados en el día, realizando el cálculo del cierre diario de caja y permitiendo imprimir a formato pdf.

The screenshot shows a web application interface for 'Mantenimiento Cierre Diario'. On the left is a 'MENU PRINCIPAL' sidebar with 'Reportes' selected. The main area is mostly blank, with a 'Cierre Diario' window open in the center. The window has an 'IMPRIMIR' button and a table with the following columns: 'Numero', 'Valor', and 'Placa'. The table is currently empty. At the bottom of the window, it says 'Page 1 of 1' and 'No Registros'. The status bar at the bottom of the application shows 'Cerrar Sesión | 07/3/2012 | 9:16:28 PM' and the URL 'http://localhost/SISTEMA\_TESIS/Reportes/reporteCierreDiario.php#'

Figura A3.10 Reporte cierre diario

## Pantalla reporte cierre diario

Esta página mostrará un reporte con los datos de los comprobantes generados en un rango de fechas ingresados previamente por el usuario, además se nos muestra la opción de imprimir dicha consulta.

Numero	Valor	Placa
Fecha: 2012-01-31		
1 77	\$12.00	PLC005
2 79	\$12.00	PLC003
3 80	\$12.00	PLC005
4 84	\$12.00	PLC003
5 83	\$23.00	PLC005
(5 Comprobantes)	\$71.00	
Fecha: 2012-02-01		
6 85	\$12.00	PLC001
7 86	\$12.00	PLC004
(2 Comprobantes)	\$24.00	

Figura A3.11 Reporte Fechas

## Cerrar sesión

En el toolbar inferior tenemos la opción para cerrar la sesión de usuario activa.

Confirmar  
¿Seguro que desea cerrar sesión?  
Sí No

Figura A3.12 Cerrar sesión