



**UNIVERSIDAD TÉCNICA DE AMBATO**

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

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

**TEMA:**

---

**“MÓDULO DE GESTIÓN DE CLIENTES Y SEGURIDADES BAJO  
ARQUITECTURA SOA PARA EL SISTEMA ADMINISTRATIVO  
CONTABLE DE LA EMPRESA BESIXPLUS CIA LTDA”.**

---

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

**AUTOR: LUIS DAMIÁN LAURA PAUCAR**

**TUTOR: ING. M.SC. GALO LÓPEZ**

**AMBATO – ECUADOR**

**Febrero 2012**

## APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de investigación sobre el tema: “**MÓDULO DE GESTIÓN DE CLIENTES Y SEGURIDADES BAJO ARQUITECTURA SOA PARA EL SISTEMA ADMINISTRATIVO CONTABLE DE LA EMPRESA BESIXPLUS CIA LTDA**”, del señor Luis Damián Laura Paucar, estudiante de la Carrera de Ingeniería en Sistemas Informáticos y Computacionales, 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 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 febrero 22, 2012

EL TUTOR

-----  
Ing. M.Sc.: Galo López

## AUTORÍA

El presente trabajo de investigación titulado “**MÓDULO DE GESTIÓN DE CLIENTES Y SEGURIDADES BAJO ARQUITECTURA SOA PARA EL SISTEMA ADMINISTRATIVO CONTABLE DE LA EMPRESA BESIXPLUS CIA**”. 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 febrero 22, 2012

---

Luis Damián Laura Paucar  
CC: 1804161949

## APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. M.Sc. Franklin Mayorga, Dra. Anita Larrea, revisaron y aprobaron el Informe Final del trabajo de graduación titulado **“MÓDULO DE GESTIÓN DE CLIENTES Y SEGURIDADES BAJO ARQUITECTURA SOA PARA EL SISTEMA ADMINISTRATIVO CONTABLE DE LA EMPRESA BESIXPLUS CIA”**, presentado por el señor Luis Damián Laura Paucar de acuerdo al Art. 17 del Reglamento de Graduación para Obtener el Título Terminal de Tercer Nivel de la Universidad Técnica de Ambato.

Ing. M.Sc. Oswaldo Paredes  
PRESIDENTE DEL TRIBUNAL

Ing. M.Sc. Franklin Mayorga  
MIEMBRO DEL TRIBUNAL

Dra. Anita Larrea  
MIEMBRO DEL TRIBUNAL

## DEDICATORIA

*Dedico este proyecto de tesis a Dios y a mis padres. A Dios porque ha estado conmigo a cada paso que doy, cuidándome y dándome fortaleza para continuar, a mis padres, quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento.*

*Luis Damián Laura Paucar*

## AGRADECIMIENTO

*En primer lugar a Dios por haberme dado la salud y las fuerzas para poder culminar una de las metas más importantes de mi vida.*

*A mis padres Olmedo y Rosa por haber confiado en mis capacidades, por haberme guiado siempre por el buen camino y por haberme dado todo lo necesario para poder llegar hacia mi meta.*

*A mis hermanos Javier y Gabriela por haberme brindado todo su apoyo y confianza.*

*A mi esposa Angeles y a mi hija Emily por haberme dado toda la felicidad y por hacer de nuestro hogar lo primordial.*

*Gracias al Ing. Galo López por haberme brindado su apoyo durante todo el tiempo que duro mi tesis y por haberme instruido con todos sus conocimientos durante toda mi etapa de estudiante.*

*Luis Damián Laura Paucar*

## ÍNDICE GENERAL

CONTENIDO	PÁGINA
Carátula .....	i
Aprobación del tutor .....	ii
Autoría .....	iii
Aprobación de la comisión calificadora.....	iv
Dedicatoria .....	v
Agradecimiento .....	vi
Índice General .....	vii
Índice de Figuras.....	xiv
Índice de Tablas.....	xvii
Resumen de ejecutivo .....	xx
Introducción.....	xxi

### CAPÍTULO I

#### EL PROBLEMA DE INVESTIGACIÓN

1.1 Planteamiento del problema .....	1
1.1.1 Contextualización.....	1
1.1.2 Análisis crítico .....	2
1.1.3 Prognosis .....	2
1.2 Formulación del problema.....	3
1.2.1 Preguntas directrices .....	3
1.2.2 Delimitación del problema .....	3
1.3 Justificación .....	3
1.4 Objetivos.....	4
1.4.1 Objetivo General.....	4

1.4.2 Objetivos Específicos .....	4
-----------------------------------	---

## **CAPÍTULO II**

### **MARCO TEÓRICO**

2.1 Antecedentes Investigativos .....	5
2.2 Fundamentación legal .....	5
2.3 Categorías Fundamentales.....	7
2.3.1 Servidor de Base de Datos.....	8
2.3.1.1 Base de Datos.....	8
2.3.1.2 Gestor de Base de Datos.....	8
2.3.1.2.1 PostgreSQL.....	8
2.3.1.2.2 MySQL .....	9
2.3.1.2.3 Firebird .....	9
2.3.2 Servidores .....	10
2.3.2.1 Servidor de Aplicación .....	10
2.3.2.1.1 Servidores de Aplicación JEE.....	10
2.3.2.2 Servidor de Reportes .....	11
2.3.2.2.1 Jasper Server .....	11
2.3.2.2.2 JasperReport.....	11
2.3.3. Plataforma JEE.....	12
2.3.3.1 Lenguaje de Programación Java .....	12
2.3.3.2 Entornos de desarrollo integrados para Java.....	13
2.3.3.2.1 Eclipse .....	13
2.3.3.2.2 NetBeans.....	13
2.3.4 Servicios Web .....	13
2.3.4.1 Tecnologías que se usan en Web Services .....	14
2.3.4.1.1 SOAP .....	14



2.3.4.1.2 WSDL: Web Service Description Language .....	15
2.3.4.1.3 UDDI .....	15
2.3.5 Arquitectura SOA .....	15
2.3.5.1 Capas que define SOA.....	18
2.3.5.2 Elementos de la Arquitectura SOA.....	19
2.3.6 Aplicación Cliente.....	19
2.3.6.1 Lenguajes de Programación del lado del cliente.....	19
2.3.6.1.1 HTML.....	19
2.3.6.1.2 JavaScript.....	20
2.3.6.1.3 Tecnologías XML.....	21
2.3.6.2 Framework JavaScript .....	21
2.3.7 Gestión de Clientes .....	22
2.3.7.1 Funcionalidades .....	23
2.3.8 Seguridades.....	24
2.3.8.1 Seguridades en la base de datos .....	24
2.3.8.2 Seguridad a nivel de Web .....	24
2.3.8.2.1 Seguridad en las tecnologías de Información .....	24
2.3.9 Besixplus CIA LTDA.....	25
2.3.10 Administración Contable.....	26
2.3.10.1 Documentos Comerciales .....	26
2.3.10.1.1 Tipos de Documentos .....	26
2.4 Hipótesis.....	27
2.5 Señalamiento de variables de la hipótesis .....	27
2.5.1 Variable Independiente.....	27
2.5.2 Variable Dependiente.....	27

### CAPÍTULO III

## MARCO METODOLÓGICO

3.1 Enfoque .....	28
3.2 Modalidad básica de la investigación .....	28
3.2.1 Investigación de campo .....	28
3.2.2 Investigación documental - bibliográfica .....	28
3.2.3 Proyecto factible .....	29
3.3 Nivel o tipo de investigación .....	29
3.4 Población y muestra .....	29
3.4.1 Población .....	29
3.4.2 Muestra .....	29
3.5 Operacionalización de las variables .....	29
3.5.1 Variable Independiente .....	30
3.5.2 Variable Dependiente .....	32
3.6 Recolección de la información .....	33
3.6.1 Plan de recolección de información .....	33
3.6.2 Procesamiento y análisis de la información .....	33

## CAPÍTULO IV

### ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1 Análisis e interpretación de los resultados .....	34
4.1.1 Análisis de los resultados de la entrevista .....	34
4.2 Verificación de la Hipótesis .....	43
4.2.1 Combinación de frecuencias .....	44
4.2.2 Frecuencias esperadas .....	44
4.2.3 Modelo Lógico .....	45
4.2.4 Nivel de significancia y reglas de decisión .....	45

4.2.4.1 Grado de Libertad.....	45
4.2.4.2 Grado de significancia .....	45

**CAPÍTULO V**  
**CONCLUSIONES Y RECOMENDACIONES**

5.1 Conclusiones.....	48
5.2 Recomendaciones .....	49

**CAPÍTULO VI**  
**PROPUESTA**

6.1 Tema.....	50
6.2 Datos informativos.....	50
6.3 Antecedentes.....	50
6.4 Justificación.....	51
6.5 Objetivos.....	52
6.5.1 Objetivo General.....	52
6.5.2 Objetivos Específicos.....	52
6.6 Análisis de factibilidad.....	52
6.6.1 Factibilidad operativa.....	52
6.6.2 Factibilidad económica.....	53
6.6.3 Factibilidad técnica .....	53
6.7 Fundamentación.....	55
6.8 Metodología.....	57
6.9 Modelo Operativo .....	57
6.9.1 Análisis del Sistema .....	57

6.9.1.1 Análisis del Sistema actual .....	57
6.9.1.2 Requerimientos del sistema .....	58
6.9.1.3 Diagramas UML.....	59
6.9.1.3.1 Diagramas de caso de Uso .....	59
6.9.1.3.2 Diagramas de Clases.....	75
6.9.1.3.3 Diagramas de Secuencia .....	76
6.9.1.3.3.1 Diagrama de secuencia Inicio de Sesión.....	76
6.9.1.3.3.2 Diagrama de secuencia Administración de tablas de configuración ...	77
6.9.1.3.3.3 Diagramas de secuencia Administración de Clientes.....	79
6.9.1.3.3.4 Diagramas de secuencia Administración de Créditos .....	81
6.9.1.3.3.5 Diagramas de secuencia de Administración de seguridades.....	82
6.9.1.3.4 Diagramas de actividad.....	84
6.9.1.3.4.1 Diagrama de actividad Administración tablas de configuración .....	84
6.9.1.3.4.2 Diagrama de actividad Administración de créditos.....	84
6.9.1.3.4.3 Diagrama de actividad Control de Sesiones de Usuario.....	85
6.9.1.3.5 Diagrama de despliegues .....	85
6.9.2 Diseño del Sistema.....	86
6.9.2.1 Diseño de Base de datos .....	86
6.9.2.2 Diccionario de Datos .....	88
6.9.2.3 Script de la base de datos.....	98
6.9.2.4 Diseño de la interfaz.....	104
6.9.2.4.1 Diseño de entradas y salida.....	104
6.9.3 Implementación.....	113
6.9.4 Implantación .....	139
6.9.5 Cronograma de implantación.....	140
6.9.6 Pruebas .....	140

6.9.6.1 Pruebas de caja blanca.....	140
6.9.6.1.1 Pruebas del camino básico.....	143
6.9.6.2 Pruebas de caja negra.....	145
6.10 Conclusiones y recomendaciones.....	146
6.10.1 Conclusiones.....	146
6.10.2 Recomendaciones.....	147
6.11 Bibliografía.....	147
6.11.1 Fuentes Bibliográficas.....	147
6.11.2 Linkografía.....	147
Glosario de Términos.....	151

**ANEXOS**

## ÍNDICE DE FIGURAS

<b>Figura 2.1.</b> Inclusión interrelacional de la variable Independiente.....	7
<b>Figura 2.2.</b> Inclusión interrelacional de la variable Dependiente .....	7
<b>Figura 2.3.</b> Servicios Web .....	13
<b>Figura 2.4.</b> Comunicación SOAP .....	15
<b>Figura 2.5.</b> Esquema básico SOA.....	16
<b>Figura 2.6.</b> Capas de una aplicación SOA.....	18
<b>Figura 4.1.</b> Gráfico pregunta 1 .....	35
<b>Figura 4.3.</b> Gráfico pregunta 3 .....	36
<b>Figura 4.4.</b> Gráfico pregunta 4 .....	37
<b>Figura 4.5.</b> Gráfico pregunta 5 .....	38
<b>Figura 4.6.</b> Gráfico pregunta 6 .....	39
<b>Figura 4.7.</b> Gráfico pregunta 7 .....	40
<b>Figura 4.8.</b> Gráfico pregunta 8 .....	41
<b>Figura 4.9.</b> Gráfico pregunta 9 .....	42
<b>Figura 4.10.</b> Gráfico pregunta 10.....	43
<b>Figura 6.1.</b> Diagrama de caso de uso de Gestión de Clientes y Seguridades.....	59
<b>Figura 6.2.</b> Diagrama de caso de uso Administración de configuración .....	60
<b>Figura 6.3.</b> Diagrama de caso de uso Administración de Clientes.....	64
<b>Figura 6.4.</b> Diagrama de caso de uso Administración de Créditos .....	69
<b>Figura 6.5.</b> Diagrama de caso de uso Administración de Seguridades.....	72
<b>Figura 6.6.</b> Diagrama de clases.....	75
<b>Figura 6.7.</b> Diagrama de secuencia Iniciar sesión .....	76
<b>Figura 6.8.</b> Diagrama de secuencia ingresar datos de configuración .....	77
<b>Figura 6.9.</b> Diagrama de secuencia modificación de datos de configuración.....	77

<b>Figura 6.10.</b> Diagrama de secuencia eliminar datos de configuración .....	78
<b>Figura 6.11.</b> Diagrama de secuencia visualizar datos de configuración .....	78
<b>Figura 6.12.</b> Diagrama de secuencia ingreso de datos de clientes .....	79
<b>Figura 6.13.</b> Diagrama de secuencia modificación de datos de clientes.....	79
<b>Figura 6.14.</b> Diagrama de secuencia eliminar cliente.....	80
<b>Figura 6.15.</b> Diagrama de secuencia visualizar datos de clientes.....	80
<b>Figura 6.16.</b> Diagrama de secuencia registrar venta a crédito .....	81
<b>Figura 6.17.</b> Diagrama de secuencia cobro de cuotas de un crédito.....	81
<b>Figura 6.18.</b> Diagrama de secuencia administración de cuentas de usuario .....	82
<b>Figura 6.19.</b> Diagrama de secuencia administración de roles de usuario .....	82
<b>Figura 6.20.</b> Diagrama de secuencia administración de perfiles de usuario .....	83
<b>Figura 6.21.</b> Diagrama de secuencia administración de sesiones de usuario.....	83
<b>Figura 6.22.</b> Diagrama de actividad Administración de tablas de configuración	84
<b>Figura 6.23.</b> Diagrama de actividad Administración de créditos .....	84
<b>Figura 6.24.</b> Diagrama de actividad registro de sesiones de usuario.....	85
<b>Figura 6.25.</b> Diagrama de despliegue del módulo de Gestión de Clientes y Seguridades .....	85
<b>Figura 6.26.</b> Diseño de la base de datos del módulo de clientes y seguridades ...	87
<b>Figura 6.27.</b> Prototipo de la pantalla de validación de usuario .....	105
<b>Figura 6.28.</b> Pantalla de validación de usuario.....	105
<b>Figura 6.29.</b> Pantalla principal del sistema .....	106
<b>Figura 6.30.</b> Prototipo de la pantalla de listado de provincias .....	106
<b>Figura 6.31.</b> Pantalla del listado de provincias.....	107
<b>Figura 6.32.</b> Prototipo de la pantalla de ingreso de las agencias.....	108
<b>Figura 6.33.</b> Pantalla de ingreso de las agencias .....	108
<b>Figura 6.34.</b> Prototipo de la pantalla de ingreso de empleados.....	109

<b>Figura 6.35.</b> Pantalla de ingreso de empleados .....	109
<b>Figura 6.36.</b> Prototipo de la pantalla de ingreso de clientes.....	110
<b>Figura 6.37.</b> Pantalla de ingreso de clientes .....	110
<b>Figura 6.38.</b> Almacén de claves y entradas asimétricas para el servidor de seguridades .....	114
<b>Figura 6.39.</b> Confidencialidad en la comunicación .....	115
<b>Figura 6.40.</b> Cronograma de implantación.....	141
<b>Figura 6.41.</b> Validación de campos en el formulario.....	141
<b>Figura 6.42.</b> Verificación de almacenamiento en la base de datos.....	142
<b>Figura 6.43.</b> Mensaje de advertencia .....	143
<b>Figura 6.44.</b> Grafo Validación de cédula/RUC .....	144



## ÍNDICE DE TABLAS

<b>Tabla 3.1.</b> Operacionalización de la variable independiente .....	30
<b>Tabla 3.2.</b> Operacionalización de la variable dependiente .....	32
<b>Tabla 4.1.</b> Cuadro porcentual pregunta 1 .....	34
<b>Tabla 4.3.</b> Cuadro porcentual pregunta 3 .....	36
<b>Tabla 4.4.</b> Cuadro porcentual pregunta 4 .....	36
<b>Tabla 4.5.</b> Cuadro porcentual pregunta 5 .....	37
<b>Tabla 4.6.</b> Cuadro porcentual pregunta 6 .....	38
<b>Tabla 4.7.</b> Cuadro porcentual pregunta 7 .....	39
<b>Tabla 4.8.</b> Cuadro porcentual pregunta 8 .....	40
<b>Tabla 4.9.</b> Cuadro porcentual pregunta 9 .....	41
<b>Tabla 4.10.</b> Cuadro porcentual pregunta 10 .....	42
<b>Tabla 4.2.1.</b> Combinación de frecuencias .....	44
<b>Tabla 4.2.2.</b> Frecuencias esperadas .....	44
<b>Tabla 4.2.3.</b> Distribución de chi-cuadrado .....	46
<b>Tabla 4.2.3.1.</b> Cálculo de chi-cuadrado.....	47
<b>Tabla 6.1.</b> Caso de uso de Inicio de Sesión .....	60
<b>Tabla 6.2.</b> Caso de uso Ingreso de tablas de configuración .....	61
<b>Tabla 6.3.</b> Caso de uso modificación de registros de las tablas de configuración	62
<b>Tabla 6.4.</b> Caso de uso eliminación de datos de configuración.....	62
<b>Tabla 6.5.</b> Caso de uso visualización de datos de las tablas de configuración.....	63
<b>Tabla 6.6.</b> Caso de uso Generar reporte de datos de configuración.....	63
<b>Tabla 6.7.</b> Caso de uso imprimir reporte sobre tablas de configuración.....	64
<b>Tabla 6.8.</b> Caso de uso ingresar cliente.....	65
<b>Tabla 6.9.</b> Caso de uso modificar cliente .....	66

<b>Tabla 6.10.</b> Caso de uso eliminar cliente .....	66
<b>Tabla 6.11.</b> Caso de uso visualizar datos del cliente.....	67
<b>Tabla 6.12.</b> Caso de uso generar reportes sobre los clientes .....	67
<b>Tabla 6.13.</b> Caso de uso imprimir reporte de cliente .....	68
<b>Tabla 6.14.</b> Caso de uso gestionar trabajos con clientes .....	68
<b>Tabla 6.15.</b> Caso de uso registrar venta a crédito .....	69
<b>Tabla 6.16.</b> Caso de uso Cobro de cuotas .....	70
<b>Tabla 6.17.</b> Caso de uso cambiar el estado al crédito .....	71
<b>Tabla 6.18.</b> Caso de uso generar reporte de crédito.....	71
<b>Tabla 6.19.</b> Caso de uso Administrar cuentas de usuario.....	72
<b>Tabla 6.20.</b> Caso de uso Administrar roles .....	73
<b>Tabla 6.21.</b> Caso de uso Administrar perfiles y privilegios .....	74
<b>Tabla 6.22.</b> Caso de uso Administrar sesiones de usuario .....	74
<b>Tabla 6.23.</b> Descripción de la tabla Configuración .....	88
<b>Tabla 6.24.</b> Descripción de la tabla provincia .....	88
<b>Tabla 6.25.</b> Descripción de la tabla cantón .....	89
<b>Tabla 6.26.</b> Descripción de la tabla parroquia .....	89
<b>Tabla 6.27.</b> Descripción de la tabla agencia.....	90
<b>Tabla 6.28.</b> Descripción de la tabla departamento.....	91
<b>Tabla 6.29.</b> Descripción de la tabla empleado.....	92
<b>Tabla 6.30.</b> Descripción de la tabla Cliente.....	93
<b>Tabla 6.31.</b> Descripción de la tabla tipo cuota .....	93
<b>Tabla 6.32.</b> Descripción de la tabla crédito .....	94
<b>Tabla 6.33.</b> Descripción de la tabla cuota .....	94
<b>Tabla 6.34.</b> Descripción de la tabla cobro.....	95
<b>Tabla 6.35.</b> Descripción de la tabla usuario .....	96

<b>Tabla 6.36.</b> Descripción de la tabla rol .....	96
<b>Tabla 6.37.</b> Descripción de la tabla perfil .....	96
<b>Tabla 6.38.</b> Descripción de la tabla sesión .....	97
<b>Tabla 6.39.</b> Descripción de la tabla Objeto .....	97
<b>Tabla 6.40.</b> Descripción de la tabla Objeto rol.....	98

## RESUMEN EJECUTIVO

Mientras avanza el tiempo la tecnología no se queda atrás, es por esta razón que ciertas entidades públicas o privadas optan por adquirir algún tipo de software para la administración de su negocio, para así poder facilitar el manejo de la información evitando pérdida o alteración en la misma

En la mayoría de entidades optan por adquirir software libre ya que el software comercial no esta a su alcance, el software libre posee la ventaja de que al ser adquirido este puede ser usado, copiado, estudiado, modificado, y redistribuido libremente facilitando así el mantenimiento y la actualización del software.

La empresa BESIXPLUS CIA LTDA lleva la información manualmente y al momento de realizar alguna transacción suele ocurrir varios inconvenientes como alteración, datos erróneos, duplicidad de la información, entre otros.

El software que se ha desarrollado para la empresa BESIXPLUS CIA LTDA está siendo implementado en un sitio web en donde todos los empleados de dicha empresa pueden acceder al sistema desde cualquier punto de las instalaciones de la misma.

El presente trabajo denominado “MÓDULO DE GESTIÓN DE CLIENTES Y SEGURIDADES BAJO ARQUITECTURA SOA PARA EL SISTEMA ADMINISTRATIVO CONTABLE DE LA EMPRESA BESIXPLUS CIA LTDA”, ayudará a llevar un control adecuado sobre la información de los clientes y además garantizará la integridad de la información con el manejo de seguridades.

## INTRODUCCIÓN

Lo principal para la creación de este módulo es tener bien en cuenta los requerimientos del usuario ya que según ello podemos definir las herramientas y la plataforma de desarrollo.

El informe final esta dividido en capítulos en los cuales tendremos detallado cada uno de los pasos que se siguieron para la elaboración del proyecto, adicionalmente hemos incluido gráficos para una buena comprensión de este contenido.

**En el Capítulo I** denominado “PROBLEMA”, en donde se identifica el problema a investigar, además se plantea la justificación y los objetivos.

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

**En el Capítulo III** denominado “METODOLOGÍA”, en donde 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** denominado “ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS”, en donde se procede a realizar el análisis e interpretación de los resultados para así poder comprobar nuestra hipótesis.

**En el capítulo V** denominado “CONCLUSIONES Y RECOMENDACIONES”, en la cual se presentan las conclusiones y recomendaciones referentes al proyecto.

**En el capítulo VI** denominado “PROPUESTA”, en la cual detallamos todo el desarrollo del módulo propuesto para ello utilizamos: diagramas de modelado, diseño y pruebas.

Adicional a esto presentamos como anexos entrevistas, el manual de instalación y manual de usuario.

## **CAPÍTULO I**

### **EL PROBLEMA DE INVESTIGACIÓN**

#### **1.1 Planteamiento del problema**

##### **1.1.1 Contextualización**

Actualmente en el Ecuador varias de las empresas públicas y privadas cuentan con sistemas para la administración de clientes pero en muchos de los casos los procesos no son administrados de la mejor manera, provocando así la pérdida de clientes como a su vez pérdida de grandes cantidades económicas, llegando así a la quiebra de las empresas.

En la provincia de Tungurahua varias empresas pierden credibilidad con sus clientes debido a que los sistemas implementados manipulan la información de una manera insegura, existen casos en que las empresas poseen sistemas para la administración de clientes pero que no cumplen con todas sus expectativas, y los niveles de seguridad al acceso a la información es inestable.

En la empresa BESIXPLUS CIA LTDA de la ciudad de Ambato se ha podido observar que para la administración de clientes se dispone de un sistema CRM muy complejo, el cual posee características no referentes a la empresa, tornándose en un inconveniente para el manejo de la información con los mismos; tampoco existe un proceso automatizado para la realización de reportes sobre las transacciones de cada uno de los clientes.

### **1.1.2 Análisis crítico**

En la empresa BESIXPLUS CIA LTDA utilizan un sistema para el control de clientes CRM, el cual no cumple con todas las necesidades que tiene la empresa para la administración de los mismos, provocando que la información no sea almacenada de una forma adecuada.

La información de los clientes se la almacena en una base de datos externa, la cual no está relacionada específicamente con toda la información referente a la empresa, provocando pérdidas o duplicidad en la información.

Al momento de realizarse una transacción con algún tipo de cliente todos los procesos se los realiza manualmente, los datos pueden ser procesados de una manera inestable, provocando resultados erróneos en el manejo de la contabilidad de la empresa.

Todo el proceso de la contabilidad se lo hace de una forma manual, y cuando se desea realizar algún tipo de reporte sobre saldos de clientes o cuentas por cobrar, el proceso manual tarda mucho tiempo, y en mucho de los casos no toda la información coincide.

El sistema utilizado actualmente para el manejo de los clientes no cumple con todas las expectativas de la empresa, las seguridades del sistema no son las adecuadas ya que cualquier usuario q ingresa al sistema puede manipular todo tipo de información de cualquier cliente.

### **1.1.3 Prognosis**

Al no integrar el Módulo de Gestión de Clientes y Seguridades en el Sistema Administrativo Contable de la empresa BESIXPLUS CIA LTDA se tendrá problemas al momento de realizar algún tipo de transacción con los clientes, no se podrá realizar un estado de cuentas sobre todos los saldos pendientes de los clientes lo cual puede afectar a toda la contabilidad de la empresa, por lo tanto se ha tomado la decisión de desarrollar dicho módulo con las últimas tecnologías existentes en el mercado de desarrollo.

## **1.2 Formulación del problema**

¿Qué incidencia tendría el desarrollo e integración del Módulo de Gestión de Clientes y Seguridades bajo la arquitectura SOA para el Sistema Administrativo Contable de la empresa BESIXPLUS CIA LTDA?

### **1.2.1 Preguntas directrices**

¿Cuáles son los requerimientos necesarios para el desarrollo del Módulo?

¿Qué es la arquitectura SOA?

¿Cuál es el beneficio de utilizar la arquitectura SOA en el desarrollo del Módulo?

¿Qué tipo de herramientas y bajo que plataforma se debe desarrollar el Módulo de Gestión de Clientes y Seguridades?

¿Qué tipo de motor de Base de Datos es la más óptima?

¿Qué nivel de seguridades se proporcionará para el acceso al sistema?

### **1.2.2 Delimitación del problema**

El desarrollo e implementación del Módulo de Gestión de Clientes y Seguridades para el Sistema Administrativo Contable se realizará en la empresa BESIXPLUS CIA LTDA, que está ubicada en la provincia de Tungurahua, en el cantón Ambato en las calles Seymour 013 y Tortuga, desde Agosto del 2010 hasta Febrero del 2011.

## **1.3 Justificación**

En la actualidad existe una gran variedad de sistemas orientados a la administración de clientes ya sean libres o bajo licencias, pero estos sistemas no cumplen con todas las expectativas de cada una de las empresas, tornándose en un inconveniente para el manejo de la información.

La creación de este Módulo es de suma importancia en la empresa debido a que este facilitará la organización y ejecución de cada uno de los procesos que



intervienen para la administración de clientes; así como también evitará la pérdida o el robo de la información debido a que se implementarán varios niveles de seguridad.

Este proyecto se desarrollará con el fin de que la empresa BESIXPLUS CIA LTDA pueda gestionar fácilmente las relaciones con sus clientes, proveedores y otros socios de negocio, llevando así toda la contabilidad de una forma ordenada y segura.

## **1.4 Objetivos**

### **1.4.1 Objetivo General**

- Desarrollar e integrar el Módulo de Gestión de Clientes y seguridades bajo la arquitectura SOA para el sistema administrativo contable de la empresa BESIXPLUS CIA LTDA.

### **1.4.2 Objetivos Específicos**

- Analizar los procesos de Gestión de Clientes y Seguridades a fin de establecer los requisitos del Módulo.
- Investigar en qué consiste la arquitectura SOA en comparación a las diferentes arquitecturas de programación.
- Diseñar el modelo relacional de la base de datos a utilizarse en el Módulo de Gestión de Clientes y Seguridades.
- Desarrollar el Módulo de Gestión de Clientes y Seguridades bajo arquitectura SOA, utilizando la plataforma JEE e integrarlo al Sistema Administrativo Contable.
- Proveer y consumir información para los diferentes módulos que integran el Sistema Administrativo Contable.

## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1 Antecedentes Investigativos

En la empresa BESIXPLUS CIA LTDA no se han realizado temas relacionados con el desarrollo sobre gestión de clientes y seguridades para la administración contable, así como tampoco ha sido abordado por otros investigadores, constituyéndolo en un tema exclusivo, no obstante una vez revisado los archivos de la facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato, se encontró que existen temas sobre gestión de clientes, pero en especial se resalta el siguiente tema: “ANÁLISIS DE SEGURIDADES E IMPLEMENTACIÓN DE HERRAMIENTAS Y APLICACIONES PARA EL PORTAL WEB DE DIFUSIÓN Y PROMOCIÓN WWW.AQUI3D.COM” realizado por Zambrano López Rosa Jacqueline, el cual posee similitud con el actual proyecto que se encuentra en desarrollo, para destacar lo significativo de este proyecto es que servirá de soporte para realizar investigaciones que oriente al correcto desarrollo del módulo de gestión de clientes y seguridades para el sistema administrativo contable de la empresa BESIXPLUS CIA LTDA de la ciudad de Ambato.

#### 2.2 Fundamentación Legal

Estatutos de la compañía BESIXPLUS CIA. LTDA

**CAPITULO PRIMERO.-** de la compañía, su denominación, objeto social, duración, domicilio, nacionalidad, transformación, fusión y escisión.-**Artículo primero.- Denominación:** La compañía se denominará **BESIXPLUS CIA.**

**LTDA.-Artículo Segundo.- Objeto Social:** La compañía tendrá por objeto social: **a)** La importación y exportación, de toda clase de productos bienes y servicios computacionales e informáticos; **b)** Desarrollo y comercialización de productos de software en representación de compañías extranjeras en el Ecuador. **c)** La compra, venta y distribución de equipos de cómputo- **d)** Diseño, construcción y mantenimiento de redes de transmisión de información **e)** Compra venta de dominios y hosting **f)** La compañía no podrá constituirse en garante o fiadora de obligaciones distintas de las suyas propias sin que medie autorización escrita previa y expresa de la Junta General de Socios. Para el cumplimiento de su objeto, la Compañía podrá intervenir como socio o accionista en la formación de toda clase de sociedades o compañías, aportar capital a las mismas o adquirir, tener y poseer acciones, obligaciones o participaciones en otras compañías. En general, la Compañía podrá realizar toda clase de actos, contratos y operaciones permitidas por las leyes ecuatorianas, que sean acordes con su objeto y necesarias y convenientes para su cumplimiento.

Actualmente la empresa BESISXPLUS CIA LTDA no cuenta con leyes que rigen la gestión de clientes.

### 2.3 Categorías Fundamentales

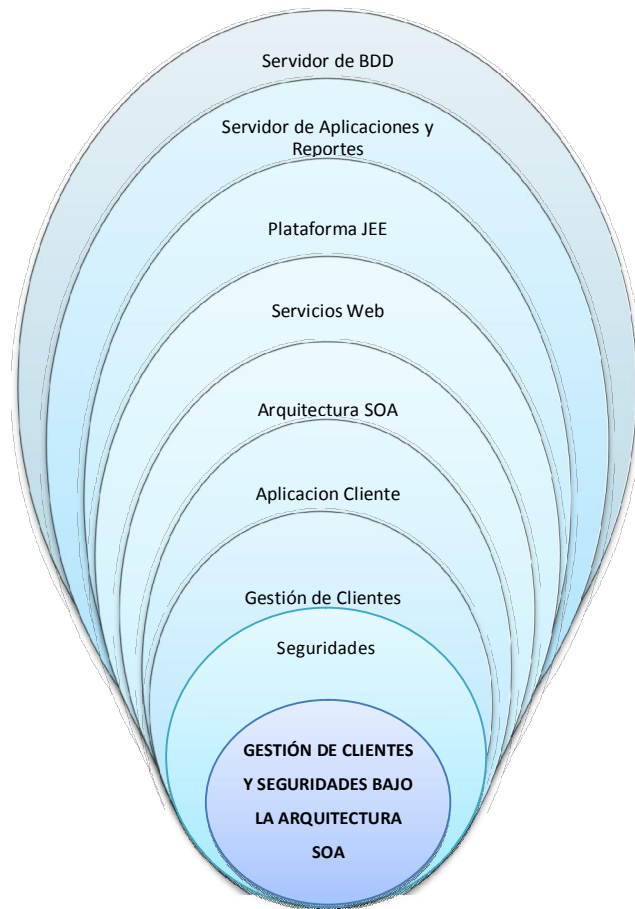


Figura 2.1. Inclusión interrelacionadas de la variable independiente



Figura2.2. Inclusión interrelacionadas de la variable dependiente

### **2.3.1 Servidor de Base de Datos**

Un servidor de Base de Datos es capaz de manejar grandes y complejos volúmenes de datos, al tiempo que requieren compartir la información con un conjunto de clientes (que pueden ser tanto aplicaciones como usuarios) de una manera segura. Ante este enfoque, un sistema gestor de bases de datos (SGBD) deberá ofrecer soluciones de forma fiable, rentable y de alto rendimiento. A estas tres características, le debemos añadir una más: debe proporcionar servicios de forma global y, en la medida de lo posible, independientemente de la plataforma. {2}

#### **2.3.1.1 Base de Datos**

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

#### **2.3.1.2 Gestor de Base de Datos**

Son un tipo de software muy específico cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. {3}

Entre los sistemas gestores de base de datos tenemos versiones libres, gratuitas y versiones comerciales, en nuestro caso nos centraremos más en gestores libres tales como son:

PostgreSQL, MySQL, Firebird, etc {4}

##### **2.3.1.2.1 PostgreSQL**

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD.

Como muchos otros proyectos Open Source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo.

Dicha comunidad es denominada el PGDG (PostgreSQL Global DevelopmentGroup). {5}

#### **2.3.1.2.2 MySQL**

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desde enero de 2008 una subsidiaria de Sun Microsystems desarrolla MySQL como software libre en un esquema de licenciamiento dual. {6}

MySQL es un gestor de base de datos multiplataforma tanto para Windows como para Linux.

#### **2.3.1.2.3 Firebird**

Firebird es un sistema de administración de base de datos relacional (o RDBMS) (Lenguaje consultas: SQL) de código abierto, basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en el año 2000. Su código fue reescrito de C a C++. El proyecto se desarrolla activamente, el 18 de abril de 2008 fue liberada la versión 2.1 y el 26 de diciembre de 2009 fue liberada la versión 2.5.0 RC1.

#### **Características**

- Es multiplataforma, y actualmente puede ejecutarse en los sistemas operativos : Linux, HP-UX, FreeBSD, Mac OS, Solaris y Microsoft Windows.
- Ejecutable pequeño, con requerimientos de hardware bajos.
- Arquitectura Cliente/Servidor sobre protocolo TCP/IP y otros. {7}

## 2.3.2 Servidores

### 2.3.2.1 Servidor de Aplicaciones

Un servidor de aplicaciones generalmente gestiona la mayor parte (o la totalidad) de las funciones de lógica de negocio y de acceso a los datos de la aplicación. Los principales beneficios de la aplicación de la tecnología de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones. {8}

#### 2.3.2.1.1 Servidores de aplicación JEE

##### **JOnAS**

Jonas es un servidor de aplicaciones J2EE de código abierto implementado en Java. JOnAS forma parte de la iniciativa de código abierto de ObjectWeb, la cual fue lanzada en colaboración con varios socios, entre los que se encuentran Bull, France Télécom e INRIA. {9}

##### **Jboss**

JBoss AS es el primer servidor de aplicaciones de código abierto, preparado para la producción y certificado J2EE 1.4, disponible en el mercado, ofreciendo una plataforma de alto rendimiento para aplicaciones de e-business. Combinando una arquitectura orientada a servicios revolucionaria con una licencia de código abierto, JBoss AS puede ser descargado, utilizado, incrustado y distribuido sin restricciones por la licencia. Por este motivo es la plataforma más popular de middleware para desarrolladores, vendedores independientes de software y, también, para grandes empresas.

Las características destacadas de JBoss incluyen:

- Producto de licencia de código abierto sin coste adicional.
- Cumple los estándares.

- Confiable a nivel de empresa
- Orientado a arquitectura de servicios.
- Servicios del middleware para cualquier objeto de Java. {10}

## **GlassFish**

GlassFish es un servidor de aplicaciones desarrollado por Sun Microsystems que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. La versión comercial es denominada SunGlassFish Enterprise Server. Es gratuito y de código libre, se distribuye bajo un licenciamiento dual a través de la licencia CDDL y la GNU GPL. {11}

### **2.3.2.2 Servidor de Reportes**

#### **2.3.2.2.1 Jasper Server**

Software libre desarrollado en java para la generación de informes, consulta y análisis de datos. JasperServer funciona individualmente o integrado con otras aplicaciones. Proporciona el análisis OLAP, permitiendo definir consultas MDX sobre los mismos y la visualización de los resultados en gráficos editables. Es utilizado para realizar minería de datos, en donde dicha información es previamente desconocida y podrá resultar útil para algún proceso. En otras palabras, la minería de datos prepara, sondea y explora los datos para ver la información oculta en ellos {12}

#### **2.3.2.2.2 JasperReport**

Está completamente escrita en Java y se puede utilizar en una gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones Web, para generar contenido dinámico.

Para la generación de informes se utiliza la herramienta JasperReports, que permite la generación de informes con una variedad de formatos, como puede ser PDF, HTML, XLS, CSV y XML.



## **iReport**

iReport es un diseñador visual de código libre para JasperReports escrito en Java. Es un programa que ayuda a los usuarios y desarrolladores que usan la librería JasperReports para diseñar reportes visualmente. A través de una interfaz amigable y simple de usar, iReport provee las funciones más importantes para crear reportes amenos en poco tiempo. {13}

### **2.3.3 Plataforma JEE**

Java Platform, Enterprise Edition o Java EE (anteriormente conocido como Java 2 Platform, Enterprise Edition o J2EE hasta la versión 1.4), es una plataforma de programación parte de la Plataforma Java para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una especificación. Similar a otras especificaciones del Java CommunityProcess, Java EE es también considerada informalmente como un estándar debido a que los suministradores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java EE; estandarizado por The Java CommunityProcess / JCP. {14}

#### **2.3.3.1 Lenguaje de Programación Java**

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

La programación en Java, permite el desarrollo de aplicaciones bajo el esquema de Cliente Servidor, como de aplicaciones distribuidas, lo que lo hace capaz de conectar

dos o más computadoras u ordenadores, ejecutando tareas simultáneamente, y de esta forma logra distribuir el trabajo a realizar. {15}

### 2.3.3.2 Entornos de desarrollo integrados para Java.

#### 2.3.3.2.1 Eclipse

Eclipse es una plataforma de desarrollo Open Source basada en Java. Es un desarrollo de IBM cuyo código fuente fue puesto a disposición de los usuarios. En sí mismo Eclipse es un marco y un conjunto de servicios para construir un entorno de desarrollo a partir de componentes conectados (plug-in). {16}

Eclipse dispone de un Editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (wizards) para creación de proyectos, clases, tests, etc., y refactorización.

#### 2.3.3.2.2 NetBeans

NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. {17}

### 2.3.4 Servicios Web

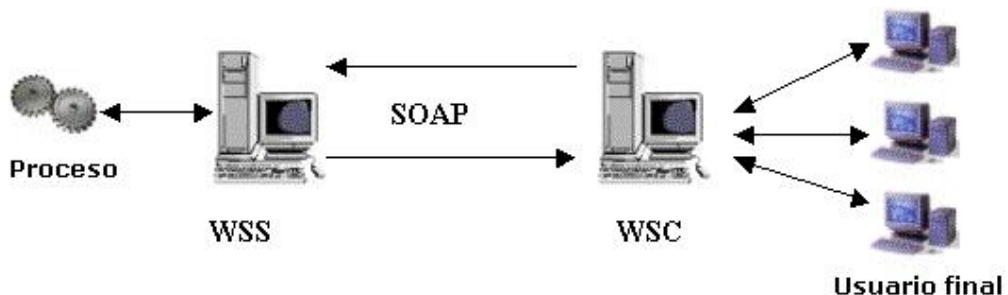


Figura 2.3. Servicios Web

Los Web Services permite la comunicación entre aplicaciones o componentes de aplicaciones de forma estándar a través de protocolos comunes (como http) y de manera independiente al lenguaje de programación, plataforma de implantación, formato de presentación o sistema operativo. Un Web Service es un contenedor que encapsula funciones específicas y hace que estas funciones puedan ser utilizadas en otros servidores.

Algunas ventajas que presentan los Web Services son:

- Son programables
- Están basados en XML, que es un lenguaje abierto
- Son auto descriptivos
- Pueden buscar registros de otros Web services {18}

#### **2.3.4.1 Tecnologías que se usan en WebServices**

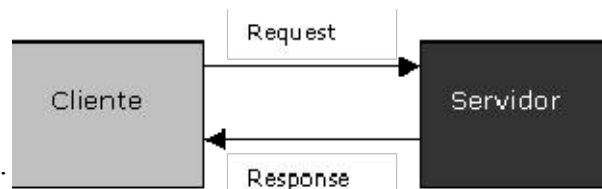
##### **2.3.4.1.1 SOAP**

SOAP es un protocolo para el intercambio de información en un ambiente descentralizado y distribuido. Es el protocolo más utilizado para realizar el intercambio de información en el modelo de web services. {19}

Está basado en XML y potencialmente puede ser utilizado en combinación con una variedad de protocolos de comunicación, siendo el más utilizado HTTP. Por lo tanto se utiliza HTTP para transportar la información, y XML para representar la misma.

#### **EL MODELO DE COMUNICACIÓN DE SOAP**

El modelo de comunicación de SOAP es muy similar al de HTTP. Un cliente hace un requerimiento (request), el servidor que está escuchando los requerimientos lo atiende y responde (response) brindando la información solicitada o enviando un mensaje de error en caso de que el requerimiento no haya sido válido. {19}



**Figura 2.4.** Comunicación SOAP

#### 2.3.4.1.2 WSDL: Web Services Description Language

WSDL es un lenguaje basado en XML que se utiliza para describir un Web Services. Ha sido suministrado por la W3C por estandarización.

Un archivo con formato WSDL provee información de los distintos métodos (operaciones) que el Web Services brinda, muestra cómo accederlos y que formatos deben de tener los mensajes que se envían y se reciben. Es como un contrato entre el proveedor del servicio y el cliente, en el cual el proveedor se compromete a brindar ciertos servicios solo si el cliente envía un requerimiento con determinado formato.

En forma resumida podríamos decir que un archivo WSDL describe lo siguiente:

- Mensajes que el servicio espera y mensajes que el servicio responde.
- Protocolos que el servicio soporta.
- A donde mandar los mensajes. {19}

#### 2.3.4.1.3 UDDI

Protocolo para publicar la información de los servicios Web. Permite comprobar qué servicios web están disponibles.

UDDI utiliza WSDL para describir las interfaces de los Web Services. {20}

#### 2.3.5 Arquitectura SOA

El acrónimo SOA proviene del inglés Service-OrientedArchitecture. Se trata de un modelo de arquitectura que caracteriza el procedimiento para crear y usar los diversos

procesos, reunidos en forma de servicios, que configuran un determinado Proceso de Negocio (Un proceso de negocio se puede ver como un conjunto estructurado de tareas, que contribuyen colectivamente a lograr los objetivos de una organización.)

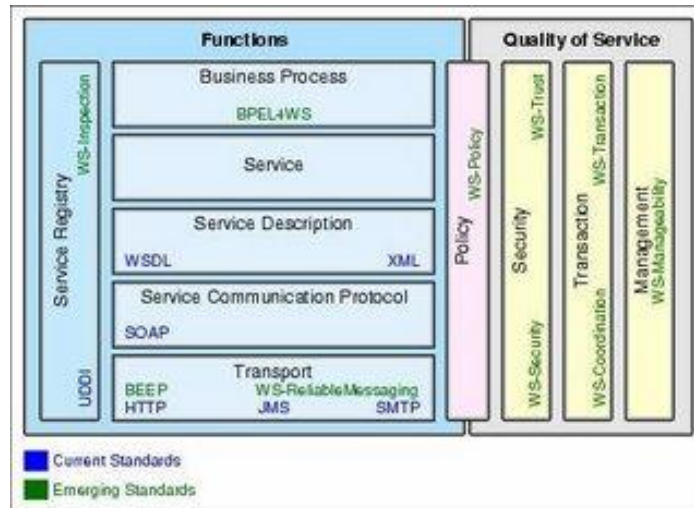


Figura 2.5. Esquema básico SOA

Como se puede observar, una SOA está formada por un conjunto de Funciones y por la Calidad del Servicio.

Las Funciones están formadas por:

- **Transporte:** Mecanismo utilizado para trasladar las peticiones desde el cliente, hasta el proveedor del servicio, y viceversa.
- **Protocolo de comunicación:** Es el sistema de comunicación entre el cliente y el proveedor de servicios.
- **Descripción del servicio:** Es un esquema utilizado para describir qué servicio es, como se le puede invocar, y cuáles son los datos necesarios para realizar su invocación.
- **Servicio:** Es la implementación del servicio.

- **Proceso de negocio:** Es una colección de servicios, invocados en una determinada secuencia, con un conjunto particular de reglas para satisfacer un requisito de negocio.
- **Registro de servicios:** Es un repositorio de servicios y datos, usado por los proveedores de servicio y publicar los servicios, y para los clientes, donde buscarlos.

La calidad del servicio por:

- **Política:** Son un conjunto de reglas bajo las cuales, un proveedor de servicio hace que el servicio esté disponible para los clientes (*WS-Policy*).
- **Seguridad:** Son un conjunto de reglas que podrían ser aplicadas en la identificación, autorización y control de acceso a los servicios, por parte del cliente (*WS-Security*).
- **Transacción:** Conjunto de atributos que podrían ser aplicados sobre un grupo de servicios para devolver un conjunto de datos consistentes (*WS-Transaction, WS-Coordination*).
- **Gestión:** Conjunto de atributos que podrían ser aplicados para gestionar los servicios proporcionados (*WS-Manageability*). {21}

La Arquitectura SOA establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es mediante Servicios Web, una tecnología basada en estándares e independiente de la plataforma, con la que SOA puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar esta funcionalidad en forma modular. {22}

Esta arquitectura define y proporciona la infraestructura necesaria para que el intercambio de información y la participación en los procesos de negocio se lleve a cabo con total independencia de la plataforma hardware-software sobre la que trabajan: sistema operativo, lenguaje de programación, características de los equipos, etc. {23}

### 2.3.5.1 Capas que define SOA



Figura 2.6. Capas de una aplicación SOA

Una aplicación SOA la podemos dividir en tres capas. La capa de **recepción de peticiones** (servicios controladores), la capa de **tareas** (servicios de negocio) la capa de **lógica reutilizables** (servicios de utilidad). {24}

SOA define las siguientes **capas de software**:

- Aplicaciones básicas.- Sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad.
- De exposición de funcionalidades.- Donde las funcionalidades de la capa aplicativa son expuestas en forma de servicios (servicios web).

- De integración de servicios.- Facilitan el intercambio de datos entre elementos de la capa aplicativa orientada a procesos empresariales internos o en colaboración.
- De composición de procesos.- Que define el proceso en términos del negocio y sus necesidades, y que varía en función del negocio.
- De entrega.- donde los servicios son desplegados a los usuarios finales. {25}

### **2.3.5.2 Elementos de la arquitectura SOA**

- Funciones:
  - Transporte:
  - Protocolo de comunicación de servicios:
  - Descripción de servicio:
  - Servicio:
  - Procesos de Negocio:
  - Registro de Servicios:
- Calidad de Servicio:
  - Política:
  - Seguridad:
  - Transacciones:
  - Administración: {23}

### **2.3.6 Aplicación Cliente**

#### **2.3.6.1 Lenguajes de Programación del lado del cliente**

##### **2.3.6.1.1 HTML**

El lenguaje llamado HTML es un lenguaje estático para el desarrollo de sitios web, el cual indica al navegador donde colocar cada texto, cada imagen o cada video y la forma que tendrán estos al ser colocados en la página.



**Ventajas:**

- ✓ Sencillo que permite describir hipertexto.
- ✓ No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web.
- ✓ Archivos pequeños.
- ✓ Despliegue rápido.
- ✓ Lenguaje de fácil aprendizaje.
- ✓ Lo admiten todos los exploradores.

**Desventajas:**

- ✓ Lenguaje estático.
- ✓ La interpretación de cada navegador puede ser diferente.
- ✓ El diseño es más lento.
- ✓ Las etiquetas son muy limitadas.

**2.3.6.1.2 Javascript**

Este es un lenguaje interpretado, no requiere compilación. Fue creado por BrendanEich en la empresa Netscape Communications. Utilizado principalmente en páginas web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código Javascript.

**Ventajas:**

- ✓ Lenguaje de scripting seguro y fiable.
- ✓ Los script tienen capacidades limitadas, por razones de seguridad.
- ✓ El código Javascript se ejecuta en el cliente.

**Desventajas:**

- ✓ Código visible por cualquier usuario.
- ✓ El código debe descargarse completamente.

- ✓ Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS). {26}

#### **2.3.6.1.3 Tecnologías XML**

XML, con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. {27}

#### **2.3.6.2 Framework JavaScript**

##### **Prototype**

Prototype es una librería JavaScript Orientada a Objetos, creada en marzo de 2005 por Sam Stephenson.

##### **Características:**

- Prototype tiene como objetivo facilitar el desarrollo de aplicaciones web dinámicas.
- Es interpretado del lado del cliente por el navegador web, es parte de la vista.
- Incluye facilidades para el manejo del DOM, eventos, elementos, arreglos, formularios, AJAX, JSON, etc.
- Se acopla muy bien a Scriptaculous, librería de efectos creada por Thomas Fuchs en junio de 2005. {28}

##### **Ext**

Ext JS es una librería JavaScript ligera y de alto rendimiento compatible con la mayoría de navegadores que nos permite crear páginas e interfaces web dinámicas. {29}

## **JQuery**

jQuery es una biblioteca o framework de Javascript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. {30}

## **GWT**

GWT o Google Web Toolkit es un framework creado por Google que permite ocultar la complejidad de varios aspectos de la tecnología AJAX. Es compatible con varios navegadores, lo cual es notorio ya que cada navegador suele necesitar código específico para lograr un front-end correcto en una aplicación web. El concepto de Google Web Toolkit es bastante sencillo, básicamente lo que se debe hacer es crear el código en Java usando cualquier entorno de desarrollo (IDE) de Java y el compilador lo traducirá a HTML y JavaScript. {31}

## **YUI**

La Biblioteca YUI es un conjunto de utilidades y controles, escrito con JavaScript y CSS, para la construcción de aplicaciones web interactivas ricas utilizando técnicas como el scripting DOM, DHTML y AJAX. YUI está disponible bajo una licencia BSD y es gratuito para todos los usos. {32}

### **2.3.7 Gestión de clientes**

La administración basada en la relación con los clientes, CRM, es un modelo de gestión de toda la organización, basada en la orientación al cliente, el concepto más cercano es Marketing relacional. A su vez, incluye los sistemas informáticos de apoyo a la gestión de las relaciones con los clientes, a la venta y al marketing. Una parte fundamental de este concepto es, precisamente, la de recopilar la mayor cantidad de información posible sobre los clientes, para poder dar valor a la oferta.

La empresa debe trabajar para conocer las necesidades de los mismos y así poder adelantar una oferta y mejorar la calidad en la atención. Cuando hablamos de mejorar la oferta nos referimos a poder brindarles soluciones a los clientes que se adecuen perfectamente a sus necesidades. {33}

### **2.3.7.1 Funcionalidades**

**Control de Acceso.** Los niveles de control de acceso permiten que la información sea compartida o restringida en base al perfil, o rol.

**Proyectos.** Inicie proyectos a través de las tareas orientadas a los clientes, incluyendo las campañas, oportunidades, cuentas y procesos.

**Calendario:** Como su propio nombre indica es un calendario sobre el que poder fijar los distintos eventos.

**Actividades:** Nos permite tanto buscar como abrir distintos tareas, a saber: reuniones, componer emails, planificación de llamadas, etc.

**Contactos:** Son las personas o grupos de referencia de cada cuenta.

**Cuentas:** Lo forman los datos de los clientes tanto potenciales como efectivos

**Toma de Contacto:** Se suele utilizar cuando las referencias provienen de colaboradores u otras empresas que por el motivo que sea nos interesa registrar.

**Casos e Incidencias:** tienen funcionalidades similares, en algunas ocasiones se utilizan casos para las incidencias con los clientes, y el menú de incidencias para los reportes de incidencias internos.

**Documentos:** Archivo documental donde colgar los archivos corporativos por ejemplo a modo de plantillas, de tal manera que cuando cualquier persona de la organización desee realizar una oferta, acuda a este repositorio de información para seleccionar la plantilla adecuada.

**Reportes.** Convierten los datos en información viable. {34}

## 2.3.8 Seguridades

### 2.3.8.1 Seguridad en la Base de Datos

Es la capacidad del Sistema para proteger Datos, Servicios y Recursos de usuarios no autorizados.

El fin de la seguridad es garantizar la protección o estar libre de todo peligro y/o daño, y que en cierta manera es infalible. {35}

- ✓ **Confidencialidad:** nos dice que los objetos de un sistema han de ser accedidos únicamente por elementos autorizados a ello, y que esos elementos autorizados no van a convertir esa información en disponible para otras entidades.
- ✓ **Integridad:** significa que los objetos sólo pueden ser modificados por elementos autorizados, y de una manera controlada.
- ✓ **Disponibilidad:** indica que los objetos del sistema tienen que permanecer accesibles a elementos autorizados; es el contrario de la negación de servicio.

### 2.3.8.2 Seguridad a Nivel de Web

La seguridad es un elemento de primer nivel que entra en juego desde la concepción inicial de un sistema y participa desde un principio en las decisiones de diseño.

Los requisitos de seguridad deben considerarse explícitamente durante todo el proceso de desarrollo, lo que da lugar a la inclusión de fases o actividades dedicadas a la seguridad.

#### 2.3.8.2.1 Seguridad en las Tecnologías de Información

Entre las medidas administrativas tenemos los siguientes mecanismos de protección:

- ✓ Autenticación

- ✓ Control de acceso
- ✓ Cifrado de datos
- ✓ Funciones de resumen
- ✓ Firma digital
- ✓ Registro de auditoria. {36}

### **2.3.9 Besixplus CIA LTDA**

Besixplus es una empresa joven y emprendedora que nace con el fin de brindar servicios informáticos especializados y de calidad a todo tipo de empresas con soporte profesional en lo que concierne a código abierto y propietario de software.

El recurso humano de BESIXPLUS CIA. LTDA. Posee experiencia en el manejo de plataformas Unix y Windows, Programación de aplicaciones Web Dinámicas y Estáticas, Desarrollo de Software, Consultoría Informática, Cableado Estructurado, entre otros.

#### **Visión**

BESIXPLUS CIA. LTDA. Es una empresa dedicada al desarrollo de soluciones empresariales orientadas al planteamiento de nuevos paradigmas basados en la utilización de nuevas tecnologías, estándares y técnicas que apoyen a la toma de decisiones y lógica de negocios de las empresas que conlleve a la Evolución Informática en las mismas.

#### **Misión**

BESIXPLUS CIA. LTDA. Busca convertirse en una empresa líder en el mercado de la informática con ideas innovadoras, a través de la capacidad y entrega del talento humano que la integran para nuestros clientes. {37}

### 2.3.10 Administración Contable

La contabilidad es el instrumento básico para controlar la administración y gestión empresarial. La administración contable y, por ende, la contabilidad es uno de los instrumentos principales para realizar la administración y gestión de la empresa. Gracias a la contabilidad se clasifican y ordenan las actividades económicas de la empresa, cuantificándose en unidades dinerarias.

La importancia de un sistema contable radica en constituir un medio para así obtener información económica fiable que permita adoptar decisiones. La información que facilita tomar decisiones debe ser recopilada, elaborada e interpretada. {38}

#### 2.3.10.1 Documentos Comerciales

Los documentos comerciales son todos los comprobantes extendidos por escrito en los que se deja constancia de las operaciones que se realizan en la actividad mercantil, de acuerdo con las disposiciones de la ley. Estos son de vital importancia para mantener un apropiado control de todas las acciones que se realizan en una compañía o empresas.

Estos documentos permiten controlar las operaciones practicadas por la empresa o el comerciante y la comprobación de los asientos de contabilidad. {39}

##### 2.3.10.1.1 Tipos de Documentos

**Orden de compra:** Es el comprobante que emite el comprador para pedir mercaderías al proveedor indicando condiciones de pago y forma de entrega. También es llamado nota de pedido.

**Remito:** Es el documento que confecciona el vendedor y que acompaña la entrega de las mercaderías vendidas.

**Factura:** Es el documento que el proveedor entrega al comprador comunicándole el importe de las mercaderías vendidas a crédito o al contado. Es el comprobante de compra.

**Ticket:** Cumple la misma función que la factura al contado y se utiliza en operaciones de escasos importes.

**Nota de débito:** Es el comprobante que envía el proveedor al comprador, comunicándole que aumento su deuda por motivos que en él se especifica.

**Nota de crédito:** Es el comprobante escrito que el proveedor entrega al comprador como constancia de haber recibido dinero, cheque, pagare u otros valores. {40}

## **2.4 Hipótesis**

El desarrollo e integración del módulo de Gestión de Clientes y Seguridades bajo la arquitectura SOA, mejorará los procesos del Sistema Administrativo contable de la empresa BESIXPLUS CIA LTDA.

## **2.5 Señalamiento de variables de la hipótesis**

### **2.5.1 Variable Independiente**

Gestión de Clientes y Seguridades bajo arquitectura SOA.

### **2.5.2 Variable Dependiente**

Procesos del Sistema Administrativo Contable.



## **CAPÍTULO III**

### **MARCO METODOLÓGICO**

#### **3.1 Enfoque**

La investigación es eminentemente cualicuantitativa debido a que el técnico conoce, analiza y toma las decisiones más óptimas para la solución del problema que se produce dentro de un contexto, de igual manera la población nos proporciona con la información necesaria con respecto al problema logrando así tomar las mejores decisiones entre las dos partes.

#### **3.2 Modalidad básica de la investigación**

##### **3.2.1 Investigación de campo**

Esta investigación de campo permite es estudio sistemático de los hechos en el lugar en que se produce los acontecimientos. En esta modalidad el investigador toma contacto en forma directa con la realidad, para tener información de acuerdo con los objetivos del proyecto.

##### **3.2.2 Investigación documental - bibliográfica**

Esta modalidad permite conocer, comparar, ampliar, profundizar y deducir diferentes enfoques, teorías, conceptualizaciones y criterios de diversos autores sobre una cuestión determinada basándose en documentos (fuentes primarias) o en libros, revistas, periódicos y otras publicaciones (fuentes secundarias) que se recomienda para estudios sociales, geográficos, históricos, literarios, entre otros.

### **3.2.3 Proyecto factible**

Se desarrollará una propuesta en base a la construcción de un módulo de Gestión de Clientes y Seguridades para el Sistema Administrativo Contable que es un modelo práctico que permitirá solucionar los problemas detectados en la empresa BESIXPLUS CIA LTDA previo al diagnóstico realizado con antelación y sustentado en el marco teórico.

### **3.3 Nivel o tipo de investigación**

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

### **3.4 Población y muestra**

#### **3.4.1 Población**

La población que será utilizada para la investigación está conformada por siete personas, entre las cuales seis son los socios de la empresa BESIXPLUS CIA LTDA y 1 persona es la encargada de llevar la contabilidad de la empresa.

#### **3.4.2 Muestra**

Como la población es pequeña toda pasa a constituir la muestra.

### **3.5 Operacionalización de Variables**

### 3.5.1 Variable Independiente

Conceptualización	Dimensión	Indicadores	Ítems	Téc. – Inst.
<b>Gestión de Clientes y Seguridades bajo la arquitectura SOA.</b> Conjunto de datos que interactúan entre sí con el fin de registrar y apoyar todas las actividades que realiza la empresa para con los clientes.	<p>Gestión de Clientes</p>         <p>Seguridades</p>	<p>Existencia de un Sistema para el Manejo de Clientes.</p>         <p>Características</p>	<p>¿Disponen un sistema para el manejo de la información sobre los clientes?</p> <p>SI <input type="checkbox"/></p> <p>NO <input type="checkbox"/></p> <p>¿Cree usted que la creación de un módulo sobre gestión de clientes facilitará el manejo de la información?</p> <p>SI <input type="checkbox"/></p> <p>NO <input type="checkbox"/></p> <p>¿El módulo de gestión de clientes facilitará el registro de todas las actividades que se realizan con los clientes?</p> <p>SI <input type="checkbox"/></p> <p>NO <input type="checkbox"/></p> <p>¿El Módulo sobre Gestión de Clientes a desarrollarse deberá ser?</p> <ul style="list-style-type: none"> <li>- Flexible <input type="checkbox"/></li> <li>- Seguro <input type="checkbox"/></li> <li>- Confiable <input type="checkbox"/></li> <li>- Otro <input type="checkbox"/></li> </ul> <p>¿Las seguridades deben ser a nivel de datos y a nivel de aplicación?</p> <p>SI <input type="checkbox"/></p> <p>NO <input type="checkbox"/></p> <p>¿Qué tipo de restricciones cree usted que se deben implementar en el Módulo?</p>	<p>Entrevista y Encuesta</p>

**Tabla 3.1.** Operacionalización de la variable independiente

	Motor de Base de Datos	Sistema Gestor de Base de datos	<p>¿Conoce usted las diferencias entre Software Libre y Software Propietario?</p> <p>SI <input type="checkbox"/></p> <p>NO <input type="checkbox"/></p> <p>¿Qué tipo de Software y que Gestor de Base de Datos sería el más óptimo?</p>
	Programación	Lenguajes de Programación	<p>¿El motor de base de datos utilizado actualmente pertenece a Software Libre?</p> <p>SI <input type="checkbox"/></p> <p>NO <input type="checkbox"/></p> <p>Bajo qué lenguaje de programación cree usted necesario el desarrollo del Módulo de Gestión de Clientes?</p>
		Herramienta de Programación	<p>¿Qué herramienta cree usted necesaria utilizar para la programación?</p>
		Arquitectura SOA	<p>¿Conoce usted la funcionalidad de la arquitectura SOA?</p> <p>SI <input type="checkbox"/></p> <p>NO <input type="checkbox"/></p>
		Transmisión de Datos	<p>¿Es necesario que se pueda acceder a la información desde cualquier sitio, según las necesidades?</p> <p>SI <input type="checkbox"/> NO <input type="checkbox"/></p>

Continuación Tabla 3.1. Operacionalización de la variable

### 3.5.2 Variable Dependiente

Conceptualización	Dimensión	Indicadores	Ítems	Téc. – Inst.
<p><b>Procesos del Sistema Administrativo Contable.-</b> En la administración contable la contabilidad es uno de los instrumentos principales para realizar la administración y gestión de la empresa.</p>	Administración contable	<p>Administración</p> <p>Contabilidad</p>	<p>¿Para determinar el estado financiero de la empresa, cual es el proceso que se sigue?</p> <hr/> <p>¿Actualmente cómo se lo realiza la contabilidad?</p> <p>Automatizada <input type="checkbox"/></p> <p>Manualmente <input type="checkbox"/></p> <p>¿Actualmente los procesos que se realizan para llevar la contabilidad son seguros?</p> <p>SI <input type="checkbox"/></p> <p>NO <input type="checkbox"/></p>	Entrevista y Encuesta

**Tabla 3.2.** Operacionalización de la variable dependiente

### **3.6 Recolección de la información**

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

La recolección de Información se realizará mediante entrevistas realizadas a siete miembros de la institución que son: Gerente General de la empresa, Presidente de la Empresa, contadora encargada de llevar la contabilidad y 4 socios fundadores de la empresa.

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

El plan para el procesamiento y análisis de la información es el siguiente:

- a) Revisión de la encuesta
- b) Ensayo
- c) Aplicación
- d) Tabulación
- e) Estudio estadístico

Los datos recolectados a través de la entrevista realizada al personal de la empresa, nos permitirá recopilar información para aportar en la propuesta en el desarrollo del módulo de Gestión de Clientes y Seguridades para el Sistema Administrativo Contable de la empresa BESIXPLUS CIA LTDA.

## CAPÍTULO IV

### ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

#### 4.1 Análisis e interpretación de los resultados.

Para la realización del análisis de resultados e interpretación se aplicaron entrevistas, las mismas que fueron realizadas a siete miembros de la empresa Gerente General, President, contadora encargada de llevar la contabilidad y 4 socios fundadores de la empresa.

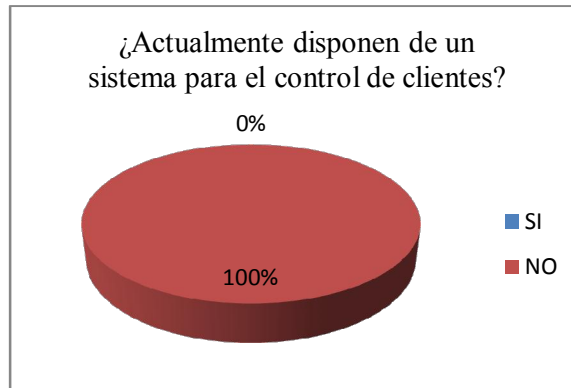
##### 4.1.1 Análisis de los resultados de las entrevistas

#### 1. ¿Actualmente disponen de un sistema para el control de clientes?

**Objetivo:** Especificar si disponen o no de un sistema para el manejo de información sobre clientes.

RESPUESTA	CANTIDAD	PORCENTAJE
SI	0	0%
NO	7	100%
TOTAL	7	100%

**Tabla 4.1** Cuadro porcentual pregunta 1



**Figura 4.1** Gráfico pregunta 1

**Interpretación:** El 100% responde que en la empresa no disponen de un sistema para poder llevar el control de la información de los clientes.

**Análisis:** La empresa no dispone de un sistema para llevar toda la información que concierne al cliente, provocando pérdida de información.

2. **¿Qué tipo de información se maneja en la empresa con respecto a los clientes?**

**Objetivo:** Conocer que información se administra sobre los clientes que posee la empresa Besixplus Cia. Ltda.

**Respuesta:** La información necesaria de los clientes es tener un número que identifique al mismo, el número de documento de identidad, nombres en caso de ser una persona natural o el nombre de la empresa si fuera una persona jurídica, adicionalmente a eso la dirección y una especificación para determinar el tipo de cliente.

**Análisis:** La información que se maneja con respecto a los clientes es primordial almacenarla, tratando de seleccionar la información más específica que se utilice dentro del sistema contable

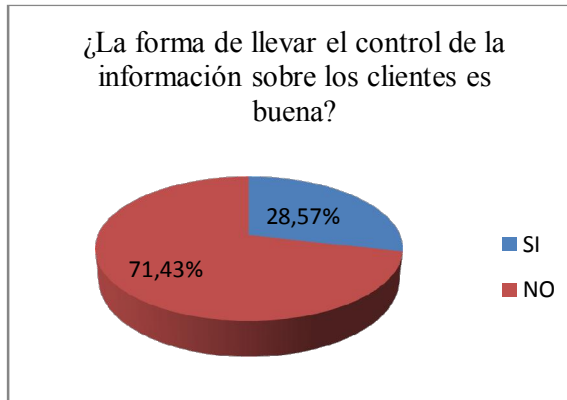
3. **¿La forma de llevar el control de la información sobre los clientes es buena?**

**Objetivo:** Determinar si la forma de llevar la información sobre los clientes es la adecuada.



RESPUESTA	CANTIDAD	PORCENTAJE
SI	2	28.57%
NO	5	71.43%
TOTAL	7	100%

**Tabla 4.3** Cuadro porcentual pregunta 3.



**Figura4.3** Gráfico pregunta 3

**Interpretación:**

El 28.57% de los entrevistados respondieron que es buena la forma de llevar la información, mientras que el 71,43% respondieron que es pésima la manera de llevar la información sobre los clientes.

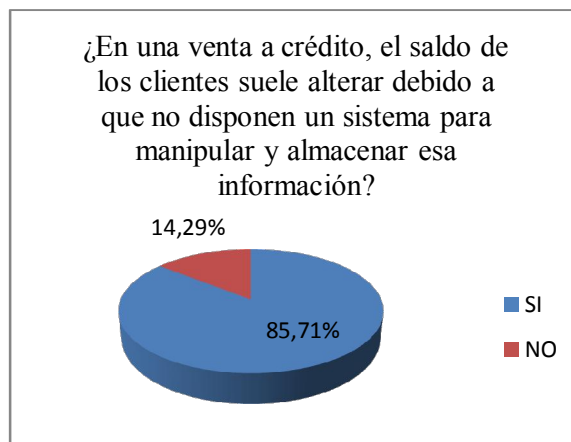
**Análisis:** La empresa lleva la información manualmente provocando grandes inconvenientes al momento de realizar algún proceso con dicha información. Por lo que es preferible la implementación del módulo de Clientes para así poder llevar la información de una manera adecuada y estable.

4. **¿En una venta a crédito, el saldo de los clientes suele alterar debido a que no disponen un sistema para manipular y almacenar esa información?**

**Objetivo:** Conocer si la información de las ventas a crédito suelen ser alteradas debido al mal manejo de la información.

RESPUESTA	CANTIDAD	PORCENTAJE
SI	6	85.71%
NO	1	14.29%
TOTAL	7	100%

**Tabla 4.4** Cuadro porcentual pregunta 4



**Figura 4.4** Gráfico pregunta 4

**Interpretación:** El 85.71% de los entrevistados responden que en muchas de las veces los saldos de los clientes son inconsistentes debido a que esta información no esta almacenada en una base de datos, y el 14.29% respondieron que la información no suele ser alterada al momento de realizar una transacción.

**Análisis:** La información sobre los saldos de las ventas a crédito no es almacenada en una base de datos, provocando así que la información sea errónea, Por lo cual es recomendable almacenar la información en una base de datos.

**5. ¿Los resultados finales son consistentes cuando se realizan procesos manuales?**

**Objetivo:** Determinar si la forma manual de llevar la información provoca inconsistencias al momento de realizar algún proceso.

RESPUESTA	CANTIDAD	PORCENTAJE
SI	1	14.29%
NO	6	85.71%
TOTAL	7	100%

**Tabla 4.5** Cuadro porcentual pregunta 5



**Figura 4.5** Gráfico pregunta 5

**Interpretación:** El 14.29% de los entrevistados respondieron que los resultados finales de los procesos manuales son exactos, mientras que el 85.71% respondieron que la manera manual de llevar la información si provoca inconsistencias al momento de realizar algún tipo de procesos.

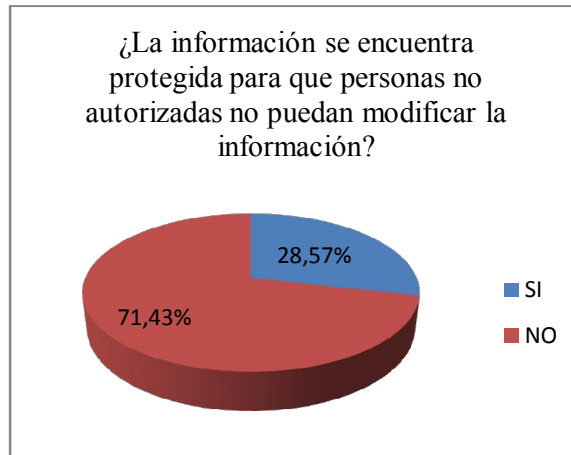
**Análisis:** La forma manual de llevar la información provoca grandes inconsistencias en el manejo de la misma al momento de realizar algún tipo de procesos con los clientes, provocando que los clientes se sientan insatisfechos por las compras realizadas

**6. ¿La información se encuentra protegida para que personas no autorizadas no puedan modificar la información?**

**Objetivo:** Conocer si la información se encuentra protegida, y libre de cualquier plagio o adulteración de la misma.

RESPUESTA	CANTIDAD	PORCENTAJE
SI	2	28.57%
NO	5	71.43%
TOTAL	7	100%

**Tabla 4.6** Cuadro porcentual pregunta 6



**Figura 4.6** Gráfico pregunta 6

**Interpretación:** el 28.57 % de los entrevistados respondió que la información si se encuentra protegida, pero el 71.43% respondió que esa información se encuentra desprotegida, dejando así que cualquier persona pueda manipular esa información.

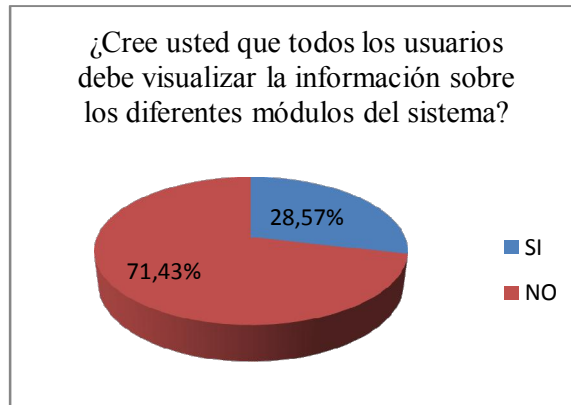
**Análisis:** Debido a la forma de llevar la información en muchos de los caso la información es adulterada por el hecho de que la información no esta protegida por los niveles de seguridad necesaria.

**7. ¿Cree usted que todos los usuarios deben visualizar la información sobre los diferentes módulos del sistema?**

**Objetivo:** Determinar que tipos de usuario son las encargas de manipular cada módulo del sistema administrativo contable.

RESPUESTA	CANTIDAD	PORCENTAJE
SI	2	28.57%
NO	5	71.43%
TOTAL	7	100%

**Tabla 4.7** Cuadro porcentual pregunta 7



**Figura 4.7** Gráfico pregunta 7

**Interpretación:** El 28.57% de los entrevistados respondió que la información solo podrá ser manipulada por personas autorizadas a cada módulo, pero el 71.43% respondió que cualquier usuario puede manipular esa información.

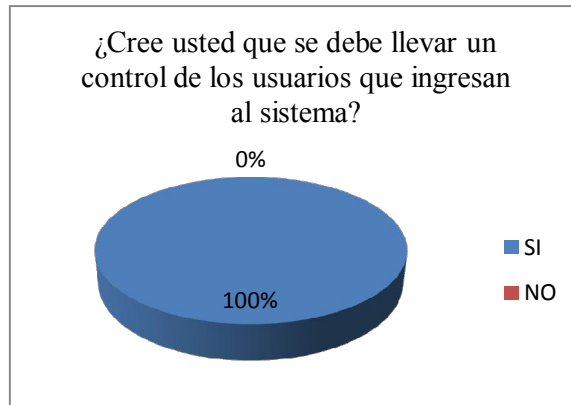
**Análisis:** La información debe ser controlada por módulos, para lo cuál se debe implementar roles y privilegios para cada uno de los módulos que intervienen en el Sistema Administrativo Contable.

**8. ¿Cree usted que se debe llevar un control de los usuarios que ingresan al sistema?**

**Objetivo:** Establecer el control del sistema llevando un registro sobre los usuarios que manipulan el sistema.

RESPUESTA	CANTIDAD	PORCENTAJE
SI	7	100%
NO	0	0%
TOTAL	7	100%

**Tabla 4.8** Cuadro porcentual pregunta 8



**Figura 4.8** Gráfico pregunta 8

**Interpretación:** El 100% de los entrevistados responden que si es necesario llevar un control de todos los usuarios que ingresan y manipulan el sistema.

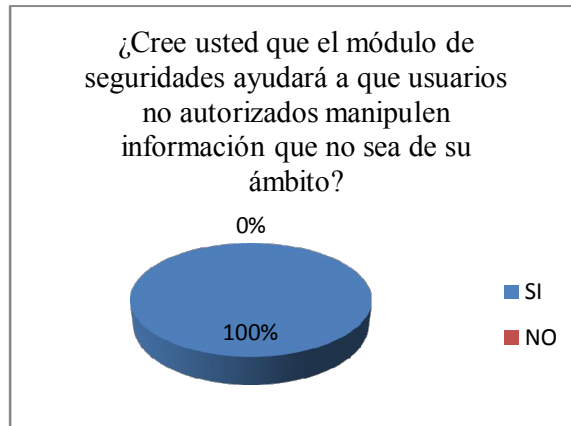
**Análisis:** Se debe establecer el control sobre los usuarios que ingresan y manipulan el sistema, considerando la fecha y hora en la que ha ingresado, dirección IP de la máquina de la cual accedió al sistema, entro otros.

9. **¿Cree usted que el módulo de seguridades ayudará a que usuarios no autorizados manipulen información que no sea de su ámbito?**

**Objetivo:** Determinar si el módulo de seguridades evitará que usuarios no autorizados manipulen información que no es de su ámbito.

RESPUESTA	CANTIDAD	PORCENTAJE
SI	7	100%
NO	0	0%
TOTAL	7	100%

**Tabla 4.9** Cuadro porcentual pregunta 9



**Figura 4.9** Gráfico pregunta 9

**Interpretación:** El 100% de entrevistados responde que el módulo de seguridades evitará que los usuarios no autorizados manipulen la información de otros módulos

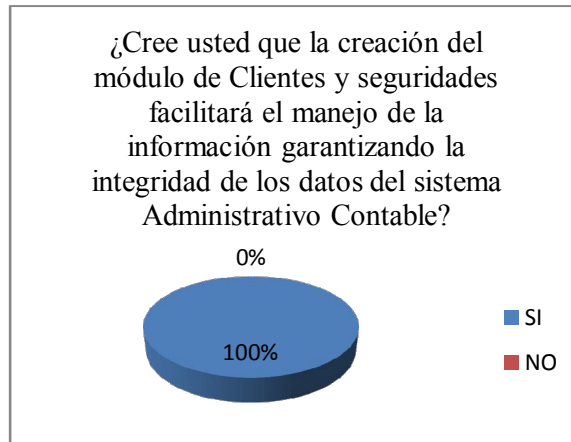
**Análisis:** El módulo de Seguridades dentro del sistema administrativo contable garantizará la estabilidad de la información, ya que se implementaron niveles de seguridad para cada uno de los módulos.

**10. ¿Cree usted que la creación del módulo de Clientes y seguridades facilitará el manejo de la información garantizando la integridad de los datos del Sistema Administrativo Contable?**

**Objetivo:** Conocer si el módulo de gestión de clientes y seguridades facilitará el manejo de la información garantizando la integridad de los datos del Sistema Administrativo Contable.

RESPUESTA	CANTIDAD	PORCENTAJE
SI	7	100%
NO	0	0%
TOTAL	7	100%

**Tabla 4.10** Cuadro porcentual pregunta 10



**Figura 4.10** Gráfico pregunta 10

**Interpretación:** El 100% de los entrevistados respondieron que el módulo de gestión de clientes y seguridades facilita el manejo de la información y que la misma esta protegida de acuerdo a los niveles de seguridad de cada uno de los módulos.

**Análisis:** Según los datos recolectados se puede decir que la creación del módulo de Clientes y Seguridades facilitará el manejo de la información llevando un control sobre los usuarios que manipulan el sistema según las políticas de seguridad de acceso al mismo.

#### 4.2 Verificación de la Hipótesis

Luego de haber tabulado las entrevistas se procede a la comprobación de la hipótesis, mediante el método estadístico chi- cuadrado:

$$= \frac{(\quad - \quad)}{\quad}$$

En donde:

= Chi-cuadrado

$\Sigma$  = Sumatoria

O = Frecuencia Observada

E = Frecuencia esperada o técnica



#### 4.2.1 Combinación de frecuencias

Nº	Pregunta	Si	No	Total
1	¿Actualmente disponen de un sistema para el control de clientes?	0	7	7
3	¿La forma de llevar el control de la información sobre los clientes es buena?	2	5	7
5	¿La forma manual de llevar la información sobre los clientes provoca inconsistencias al momento de realizar algún proceso?	1	6	7
6	¿La información se encuentra protegida para que personas no autorizadas no puedan modificar la información?	2	5	7
7	¿Cree usted que todos los usuarios deben visualizar la información sobre los diferentes módulos del sistema?	2	5	7
<b>TOTALES</b>		7	28	35

**Tabla 4.2.1.** Combinación de frecuencias

#### 4.2.2 Frecuencias esperadas

Nº	Pregunta	Si	No	Total
1	¿Actualmente disponen de un sistema para el control de clientes?	1.4	5.6	7
3	¿La forma de llevar el control de la información sobre los clientes es buena?	1.4	45.6	7
5	¿La forma manual de llevar la información sobre los clientes provoca inconsistencias al momento de realizar algún proceso?	1.4	5.6	7
6	¿La información se encuentra protegida para que personas no autorizadas no puedan modificar la información?	1.4	5.6	7
7	¿Cree usted que todos los usuarios deben visualizar la información sobre los diferentes módulos del sistema?	1.4	5.6	7
<b>TOTALES</b>		7	28	35

**Tabla 4.2.2.** Frecuencias esperadas

### 4.2.3 Modelo Lógico

**Ho**= El desarrollo e integración del módulo de Gestión de Clientes y Seguridades bajo la arquitectura SOA, no mejorará los procesos del Sistema Administrativo contable de la empresa BESIXPLUS CIA LTDA.

**Hi**= El desarrollo e integración del módulo de Gestión de Clientes y Seguridades bajo la arquitectura SOA, mejorará los procesos del Sistema Administrativo contable de la empresa BESIXPLUS CIA LTDA.

### 4.2.4 Nivel de significancia y reglas de decisión.

#### 4.2.4.1 Grado de libertad.

$$GL = (c-1)(f-1)$$

$$GL = (5-1)(2-1)$$

$$GL = 4 * 1$$

$$GL = 4$$

#### 4.2.4.2 Grado de significancia

Nivel de significación (P): Denominado nivel de confianza, se refiere a la probabilidad de que los resultados observados se deban al azar. Este valor es fijado por el investigador, usualmente es el 5% o 10%.

Lo que indica que si se toma  $P=0.05$ , se está significando que solo en un 5% de las veces en que se realice la medición, el resultado obtenido podría deberse al azar. De lo contrario sería decir que existe un nivel de confianza del 95% que el resultado es real y no debido a la casualidad.

Nivel de confiabilidad = 95%

El grado de significancia será 0.05

#### Valores críticos de chi-cuadrado

La tabla Chi-cuadrado es usada para realizar pruebas de independencia, que nos permite determinar si existe una relación entre dos variables categóricas. La tabla tiene dos entradas:

- **Alfa ( $\alpha$ ):** este valor hace referencia al nivel de confianza que deseamos que tengan los cálculos de la prueba; es decir, si queremos tener un nivel de confianza del 95%, el valor de alfa debe ser del 0.05, lo cual corresponde al complemento porcentual de la confianza.
- **Grados de Libertad (k):** Es un estimador del número de categorías independientes en la prueba de independencia o experimento estadístico. Se encuentran mediante la fórmula  $n-r$ , donde  $n$ =número de sujetos y  $r$  es el número de grupos estadísticamente dependientes.

Distribución de $X^2$			
GL	Probabilidad		
	0.05	0.01	0.001
1	3.84	6.64	10.83
2	5.99	9.21	13.82
3	7.82	11.34	16.27
4	9.49	13.28	18.47
5	11.07	15.09	20.52
6	12.59	16.81	22.46
7	14.07	18.48	24.32
8	15.51	20.09	26.12
9	16.92	21.67	27.88
10	18.31	23.21	29.59

**Tabla 4.2.3.** Distribución de chi-cuadrado

$$X^2_{(c-1)(f-1)} = 9,49$$

#### 4.2.5 Cálculo del Chi-cuadrado

En donde:

O = Frecuencia observada

E = Frecuencia esperada

O-E = Frecuencia observada – frecuencias esperada

$(O-E)^2$  = Resultado de las frecuencias observadas y esperadas al cuadrado.

$(O-E)^2/E$  = Resultado de las frecuencias observadas y esperadas al cuadrado dividido para las frecuencias esperadas.

O	E	O-E	$(O-E)^2$	$(O-E)^2/E$
0	1.4	-1.4	1.96	1.4
7	5.6	1.4	1.96	0.35
2	1.4	0.6	0.36	0,25714285
5	5.6	-0,6	0,36	0,06428571
1	1.4	-0.4	0.16	0.11428571
6	5.6	0,4	0.16	0.02857142
2	1.4	0.6	0.36	0.25714285
5	5.6	-0,6	0,36	0.06428571
2	1.4	0.6	0.36	0.25714285
5	5.6	-0,6	0,36	0.06428571
<b>TOTALES</b>				<b>2,85714281</b>

**Tabla 4.2.3.1.** Cálculo de chi - cuadrado

**Criterio de decisión:**

$$X^2 < X_{\alpha}^2(c-1)(f-1) \rightarrow \text{Acepta } H_i.$$

En donde:

$$X^2 = 2.86$$

$$X_{\alpha}^2(c-1)(f-1) = 9,49$$

**Valores de decisión:**

$$2.86 < 9,49$$

Debido a que  $X^2$  es menor a  $X_{\alpha}^2(c-1)(f-1)$  se rechaza  $H_0$  y se acepta  $H_i$ . Por lo tanto la realización de un módulo de gestión de Clientes y seguridades para el Sistema Administrativo Contable de la empresa BESIXPLUS CIA LTDA es factible.

## CAPÍTULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 Conclusiones.

- La información sobre cada uno de los clientes se encuentra almacenada en archivos físicos y algunos en Excel, provocando duplicidad de la información.
- La forma manual de realizar los procesos de cada una de las transacciones que intervienen en la gestión de Clientes provoca que la información final sea inconsistente.
- La falta de un módulo que gestione toda la información de clientes ha provocado los resultados no sean los esperados ya que la información no se encuentra almacenada en una base de datos consistente.
- Al momento de realizar una transacción con los clientes, los datos no suelen ser los esperados ya que cualquier persona puede utilizar esa información; esto se da por que la información no se encuentra protegida adecuadamente con las restricciones suficientes para el acceso a la información.
- La información recolectada en las entrevistas debe ser coherente para que al momento de desarrollo del módulo no exista ningún tipo de problema.

## 5.2 Recomendaciones.

- Se recomienda almacenar la información de los clientes en una base de datos consistente, para así evitar datos duplicados.
- Se recomienda automatizar los procesos que se presentan al momento de realizar una transacción para que la información final sea consistente.
- El módulo de Gestión de Clientes apoyará para la toma de decisiones así como también facilitará el manejo de la información, llevando así un control adecuado sobre cada una de las transacciones realizadas.
- Se recomienda la implementación de un módulo de seguridades en donde se lleve un control sobre cada uno de los usuarios que acceden al sistema, además en este módulo se podrá establecer las restricciones necesarias para que usuarios no autorizados puedan manipular esa información.
- El módulo además deberá dar la opción de visualizar, e imprimir reportes según sean las necesidades de la empresa.
- Se recomienda capacitar a todo el personal que vaya a utilizar el sistema, para que no exista ningún tipo de inconveniente al momento que el sistema se encuentre en producción.

## **CAPÍTULO VI**

### **PROPUESTA**

#### **6.1 Tema**

“MÓDULO DE GESTIÓN DE CLIENTES Y SEGURIDADES BAJO ARQUITECTURA SOA PARA EL SISTEMA ADMINISTRATIVO CONTABLE DE LA EMPRESA BESIXPLUS CIA LTDA”.

#### **6.2 Datos informativos**

**Institución:** Besixplus CIA LTDA

**Ciudad:** Ambato

**Dirección:** Seymour 03 y Tortuga

**Investigador:** Luis Damián Laura Paucar.

**Tutor:** Ing. Galo López

#### **6.3 Antecedentes**

La empresa BESIXPLUS CIA LTDA fue creada el 28 de abril de 2006 como una empresa de consultoría y asesoría informática entre ellos desarrollo de software empresarial, además también distribuidor de mercadería de equipos de computo pero durante todo ese período de tiempo no se ha desarrollado un sistema a medida que gestione todas la información sobre las transacciones de los clientes.

En la empresa se evidencia el proceso manual para llevar toda la información, lo cual dificulta la forma de administrar ya que en muchos de los casos los resultados finales no son los esperados. Este problema se da debido a que la empresa no

cuenta con un Sistema que Gestione todo el tipo de información relacionada con el Cliente.

Por todos los inconvenientes que se presentan como la pérdida y alteración de información surge la necesidad y la oportunidad de desarrollar un módulo que gestione todo este tipo de información acerca de los clientes, optando a su vez el desarrollo de un módulo adicional sobre Seguridades para poder controlar el acceso a la información.

Las herramientas con las que se trabajan actualmente son hojas de cálculo en Excel, en Word y hojas volantes lo cual provoca pérdida de tiempo y malos resultados al momento de realizar un reporte.

Según la tendencia de la empresa se ha decidido el desarrollo de este módulo bajo herramientas de software libre, así:

- Desarrollo de los servicios web (Servidor): Fue desarrollado bajo la herramienta Eclipse Indigo 3.7 bajo el lenguaje de programación JAVA.
- Servidor de Aplicaciones JBOSS 5.1.0 GA
- Ext JS framework para Javascript. (Cliente).
- Base de Datos: PostgreSQL 9
- JasperServer 3.7 para los reportes
- Entre otros.

#### **6.4 Justificación**

El desarrollo de la presente propuesta se lo realizará debido al interés de buscar posibles soluciones al problema que se presenta en la empresa al momento de registrar algún tipo de proceso con la información de los Clientes lo cual mejorará en la consistencia de los datos al momento de ser utilizados.

El encargado de vender la mercadería lleva el control de la información manualmente ocasionando que los datos sean erróneos, por tal motivo es mejor desarrollar un módulo que gestione toda la información referente a Clientes el cual permita administrar de mejor manera la información, facilitando y brindando un mejor confort para el usuario.

Debido a que la información esta al alcance de cualquier usuario, es mejor implementar un módulo adicional sobre las Seguridades que va tener el módulo de



Clientes así como para los diferentes módulos que incluirán en el Sistema Administrativo Contable. Esto garantizará que personas no autorizadas accedan a información que no es de su ámbito.

La presente propuesta es factible y viable ya que en la empresa se puede obtener la información necesaria para el desarrollo del módulo y a su vez todo el personal de la empresa esta dispuesta en instruirnos con normas de desarrollo.

## **6.5 Objetivos**

### **6.5.1 Objetivo General**

Desarrollar el Módulo de Gestión de Clientes y Seguridades bajo la arquitectura SOA para el sistema administrativo contable de la empresa BESIXPLUS CIA LTDA.

### **6.5.2 Objetivos Específicos**

- Analizar los requerimientos necesarios para poder desarrollar el módulo de Gestión de Clientes y Seguridades.
- Diseñar adecuadamente la estructura de la base de datos para garantizar la información.
- Establecer los niveles de seguridad necesarios para el acceso a la información.
- Realizar las pruebas necesarias para garantizar el óptimo funcionamiento del módulo.
- Realizar y proporcionar los manuales de instalación, usuario y administración.

## **6.6 Análisis de Factibilidad**

### **6.6.1 Factibilidad operativa**

El módulo fue desarrollado a la medida razón por la cual interactúa directamente con el usuario, ya que cuenta con una interfaz amigable, ubicando cada una de los ítems principales en un menú agradable y entendible.

Además el módulo de Seguridades cuenta con un formulario en donde se puede controlar todas las sesiones que están siendo utilizadas en el sistema principal.

El único usuario que tiene todos los privilegios en el sistema es un usuario administrador, el mismo que establecerá a los demás usuarios las restricciones necesarias para así evitar que usuarios no autorizados accedan a información no permitida. Cabe recalcar que el Sistema Administrativo Contable posee roles para cada uno de los módulos, es decir según los privilegios de cada uno de los roles se podrá acceder a dicho módulo, realizando operaciones como: consultas, inserciones, modificaciones, eliminaciones y generación de reportes.

El único que puede crear nuevos usuarios es el usuario administrador, mientras que los demás usuarios solo podrán acceder al módulo asignado y además podrá modificar datos acerca de su cuenta como: alias, contraseña, descripción etc.

### **6.6.2 Factibilidad económica**

Este proyecto es factible ya que según la tendencia de la empresa se ha decidido el desarrollo de este módulo bajo herramientas de software libre, además la empresa nos proporciona los recursos necesarios para el desarrollo del módulo.

El módulo de Gestión de Clientes ayudara a llevar de una mejor manera la información, facilitando y reduciendo tiempos al momento de realizar una transacción. Además el módulo de seguridades ayudará a que usuarios no autorizados no puedan alterar la información.

### **6.6.3 Factibilidad técnica**

Para la creación del módulo se cuenta con las siguientes herramientas:

#### **Software:**

- **Sistema Operativo Linux Fedora 15:** Sistema operativo estable de código abierto para así poder implementar nuestro sistema sin la necesidad adquirir licencias que son sumamente costosas y que no están al alcance de la empresa en cuestión.

- **Motor de Base de Datos PostgreSQL 9:** Sistema de gestión de base de datos relacional orientado a objetos y además es software libre. Se puede utilizar el lenguaje PL/PgSQL que es muy similar a PL/SQL de oracle, además posee la facilidad de utilizar lenguajes como C, C++, Java PL/Java Web, PL/Perl entre otros.
- **Herramienta de programación Eclipse Indigo 3.7:** Es un entorno de desarrollo de código abierto, la cual se utilizo para la creación de Servicios Web (Servidor) bajo el lenguaje de programación JAVA. Además a esto funciona bajo las plataformas Linux como en Windows.
- **Ext Js 3.3.1:** Es un framework para Javascript (Cliente) de código abierto específicamente para el desarrollo de aplicaciones web interactivas. Con este framework podemos utilizar componentes con una interfaz muy agradable para el usuario y además estos poseen la facilidad de comunicarse con el servidor usando AJAX.
- **Servidor de Aplicaciones JBOSS 5.1.0 GA:** Servidor de aplicaciones de código abierto para la publicación de nuestro sistema, además este servidor proporciona la facilidad de crear un dominio de seguridad para nuestra aplicación.
- **Servidor de versiones (subversion-1.6.17-1):** Sistema de control de versiones de software libre. Permite ser usado por personas que se encuentran en diferentes computadoras, la cual realiza el trabajo de versionar códigos de archivos que se encuentran en el repositorio de nuestro servidor, evitando así la pérdida o duplicidad de código cuando se trata de desarrollar en grupo.
- **JasperServer 3.7 (Servidor de reportes):** Es un servidor de reportes que se ejecuta dentro de un servidor de aplicaciones Java, en nuestro caso JBOSS, utilizando una pequeña base de datos relacional, adicional a este servidor de reportes se utilizo irreport para el diseño de reportes de cada uno de los módulos:

## Hardware:

La empresa BESIXPLUS CIA LTDA cuenta con equipos de cómputo de última tecnología facilitando el desarrollo del módulo de gestión de Clientes y Seguridades para el sistema administrativo Contable.

Servidor	Cliente
Procesador Core2 Duo de 2.00 Ghz o superior 2GB RAM o superiores 120 GB en disco o superior Trajeta de red	Procesador Pentium IV o superiores 256MB RAM o superiores Tarjeta de red

Con todas las herramientas y los equipos de cómputo necesarios para el desarrollo de este módulo; este procede a ser factible.

## 6.7 Fundamentación

### Servidor de aplicaciones JBOSS

JBoss es un servidor de aplicaciones J2EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo para el que esté disponible Java.

### Plataforma JAVA EE (Enterprise Edition)

Es una plataforma de programación, parte de la Plataforma Java para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

### Jasper Server

Es una aplicación web desarrollada en Java que permite generar reportes y acceder a ellos a través del explorador.

**IRreport.** Es una herramienta visual que sirve para generar ficheros XML (plantillas de informe) que se puedan utilizar con la herramienta de generación de informes JasperReport escrito en Java.

### **Características de IRreport**

La lista siguiente describe algunas de las características importantes de IRreport:

- 100% escrito en Java y además OpenSource y gratuito.
- Maneja el 98% de las etiquetas de JasperReport.
- Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de los textfields, cartas, subreports (subreportes).
- Soporta internacionalización nativamente.
- Browser de la estructura del documento.
- Soporta JDBC.
- Soporta JavaBeans como orígenes de datos (éstos deben implementar la interface JRDataSource).
- Incluye Wizard's (asistentes) para crear automáticamente informes.
- Tiene asistentes para generar los subreportes.
- Tiene asistentes para las plantillas.
- Facilidad de instalación.

### **PostgresSQL**

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales.

PostgreSQL permite la alta concurrencia lo quiere decir que permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

## **6.8 Metodología**

Una metodología de desarrollo de software tiene como objetivo presentar un conjunto de técnicas tradicionales y modernas de modelado de sistemas que permitan desarrollar software de calidad.

En nuestro caso se utilizó el paradigma del ciclo de vida clásico ya que en la actualidad sigue siendo el más utilizado debido a que cumple con todas las fases de desarrollo para obtener un buen resultado.

El método de ciclo de vida clásico para el desarrollo de sistemas debe cumplir con las fases más importantes, entre ellas esta: el análisis, diseño, codificación o implementación, pruebas, y mantenimiento.

Para aplicar el ciclo de vida clásico se realizó la fase de análisis con la obtención de requisitos iniciales, hasta la realización de casos de uso utilizando el lenguaje Unificado de Modelo (UML), el cual es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema, además sirve para especificar o para describir métodos o procesos que intervienen en el módulo que se va a desarrollar.

## **6.9 Modelo operativo**

### **6.9.1 Análisis del Sistema.**

#### **6.9.1.1 Análisis del sistema Actual**

Según los requerimientos recolectados se puede mencionar que no disponen de un control adecuado de la información acerca de los clientes provocando serios inconvenientes al momento de realizar una transacción.

Razones por la cual la empresa ha propuesto que se desarrolle un sistema adecuado para poder administrar toda la información Contable de la empresa y para ello es necesaria la creación de un módulo que gestione la información de los clientes así como también un módulo específico para las seguridades del sistema en general, para así poder evitar serios problemas que viene afectando a la empresa desde hace mucho tiempo.

### 6.9.1.2 Requerimientos del Sistema

Uno de los puntos de los requerimientos es registrar los datos más específicos de los clientes para así no tener información obsoleta, además se debe interactuar con la información del módulo de Ventas para poder registrar las ventas a crédito si fuera el caso, en donde se debe generar automáticamente el número de cuotas según las condiciones del cliente.

Unas ves que el cliente realiza un pago por una cuota el saldo se debe actualizar automáticamente, y una vez finalizado todas las cuotas se debe cambiar de estado a finalizado. Además a esto se podrá generar reportes sobre cada uno de los clientes que poseen una compra a crédito.

Otro de los puntos más importantes de los requerimientos es también las seguridades que va tener el sistema en general, ya que se debe llevar un control sobre cada uno de los usuarios que ingresan al sistema, tomando en cuenta la información de la fecha y hora en la que accedieron, la dirección IP de la que se conectaron y posteriormente la fecha en la que el usuario cerro su sesión.

Con respecto a las seguridades también se debe controlar el acceso al sistema por módulos, ya que no todos los usuarios van a poder manipular la información, el único usuario capaz de acceder a toda la información es el administrador el cual deberá establecer los accesos y las operaciones necesarios para cada usuario.

El módulo a desarrollar debe contar con una interfaz amigable y q sea capaz de interactuar con el usuario, además el módulo será capaz de generar reportes según sea la necesidad de usuario.

Este módulo se lo realizara bajo herramientas de Software libre ya que la empresa no cuenta con los recursos necesarios para adquirir Software comercial, razón por la cual se ha establecido herramientas de software libre tales como: Base de Datos PostgreSQL, Servidor de Aplicaciones JBOSS, Herramienta de programación Eclipse bajo la plataforma JAVA EE, Ext Js framework para Javascript, JasperReport, irreport, entre otros.

### 6.9.1.3 Diagramas UML

#### 6.9.1.3.1 Diagramas de Casos de Uso

A continuación se relatan los casos de uso más relevantes

#### Caso de Uso 1: Gestión de Clientes y Seguridades

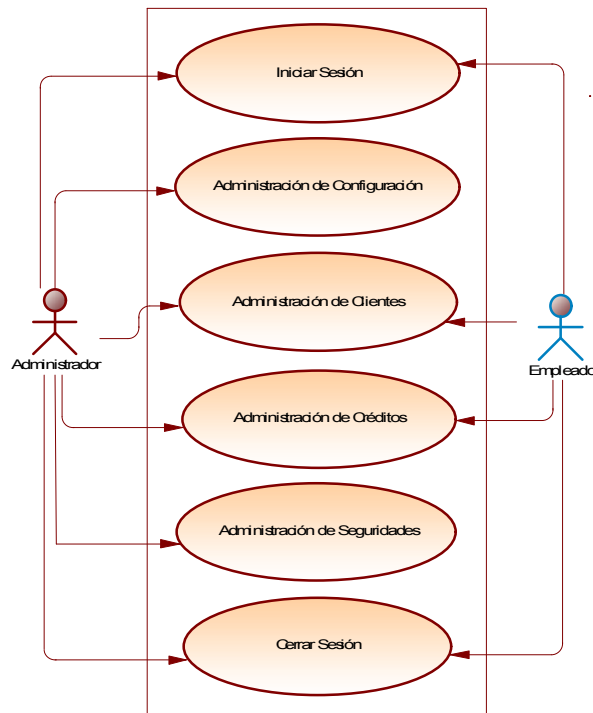


Figura 6.1 Diagrama de caso de uso de Gestión de Clientes y Seguridades

Especificaciones de caso de uso:

#### Caso de Uso: Iniciar Sesión

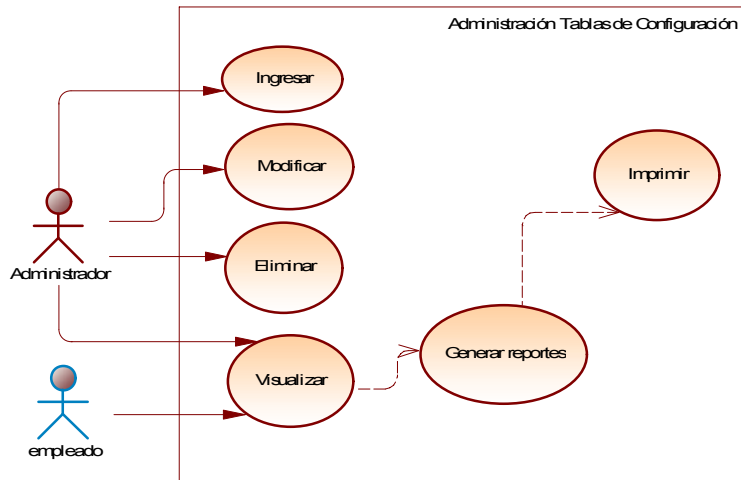
Caso de Uso: Iniciar de Sesión
<b>Actores:</b> Administrador, empleado
<b>Descripción:</b> Permite ingresar al sistema y visualizar el menú según el usuario conectado, tomando en cuenta el perfil asignado a dicho usuario.
<b>Precondiciones:</b> Para poder acceder al sistema se deben registrar los usuarios que van a manipular el sistema, para lo cual el usuario administrador es el único autorizado para poder realizar el registro y la asignación de perfiles.



<p><b>Flujo Normal:</b></p> <ol style="list-style-type: none"> <li>1. El administrador o el empleado acceden al formulario de validación de usuario</li> <li>2. El administrador o el empleado ingresan su alias de usuario y su contraseña.</li> <li>3. El administrador o el empleado pulsán el botón Conectar para que los datos sean validados.</li> </ol>
<p><b>Flujo Alternativo:</b></p> <ol style="list-style-type: none"> <li>3.1. El módulo de Seguridades se encarga de validar los datos ingresados por el usuario. <ol style="list-style-type: none"> <li>3.1.1. Si los datos son correctos se procede al ingreso al sistema.</li> <li>3.1.2. Si los datos son incorrectos se presenta un mensaje de error.</li> </ol> </li> </ol>
<p><b>Pos condiciones:</b></p> <p>El usuario ingresa al sistema en donde se le mostrara todo lo que puede manipular de acuerdo al perfil que tenga dicho usuario.</p>

**Tabla 6.1** Caso de uso de Inicio de Sesión.

**Caso de uso: Administración de tablas de Configuración**



**Figura 6.2** Diagrama de caso de uso Administración de Configuración

Caso de Uso: Ingresar	
<b>Actores:</b>	Administrador
<b>Descripción:</b>	Permite ingresar datos en las diferentes tablas de configuración necesarias para manipular el sistema, entre ellas están: Configuración, Provincia, Ciudad, Cantón, Parroquia, Tipos de Documento, Agencia, Departamento, Empleados, Tipo de Contacto, entre otros.
<b>Precondiciones:</b>	El usuario administrador es quien va poder ingresar este tipo de información, para lo cual debe estar logueado en el sistema.

<p><b>Flujo Normal:</b></p> <ol style="list-style-type: none"> <li>1. El administrador accede a los formularios de las tablas de configuración mencionadas anteriormente.</li> <li>2. El administrador ingresa la información en cada uno de los formularios mencionados.</li> <li>3. El administrador pulsa el botón Guardar para almacenar la información en la Base de Datos.</li> </ol>
<p><b>Flujo Alternativo:</b></p> <ol style="list-style-type: none"> <li>3.1. El sistema verifica la información del formulario. <ol style="list-style-type: none"> <li>3.1.1. Si los datos son correctos se procede al almacenar en la base de datos y posteriormente se muestra un mensaje indicando que los datos se han almacenado correctamente.</li> <li>3.1.2. Si los datos son incorrectos se presenta un mensaje de error.</li> <li>3.1.3. Si ocurrió una excepción en la Base de Datos, se presenta un mensaje de error indicando la excepción.</li> </ol> </li> </ol>
<p><b>Pos condiciones:</b></p> <p>Los datos se reflejan en los diferentes listados del menú configuraciones.</p>

**Tabla 6.2** Caso de uso Ingreso de Tablas de Configuración.

<b>Caso de Uso: Modificar</b>
<p><b>Actores:</b></p> <p>Administrador</p>
<p><b>Descripción:</b></p> <p>Permite modificar los datos en las diferentes tablas de configuración necesarias para manipular el sistema, entre ellas están: Configuración, Provincia, Ciudad, Cantón, Parroquia, Tipos de Documento, Agencia, Departamento, Empleados, Tipo de Contacto, entre otros.</p>
<p><b>Precondiciones:</b></p> <p>El usuario administrador es quien va poder modificar este tipo de información, para lo cual debe estar logueado en el sistema.</p>
<p><b>Flujo Normal:</b></p> <ol style="list-style-type: none"> <li>1. El administrador accede al menú Configuraciones y accede al formulario Listado de la información que desea modificar.</li> <li>2. El usuario administrador establece los criterios de búsqueda para determinar el registro a modificar.</li> <li>3. El usuario administrador pulsa editar registro y posterior procede a realizar los cambios necesarios.</li> <li>4. El usuario administrador pulsa el botón Guardar para almacenar la información en la Base de Datos.</li> </ol>
<p><b>Flujo Alternativo:</b></p> <ol style="list-style-type: none"> <li>4.1. El sistema verifica la información del formulario. <ol style="list-style-type: none"> <li>4.1.1. Si los datos son correctos se procede al almacenar en la base de datos y posteriormente se muestra un mensaje indicando que los datos se han almacenado correctamente.</li> <li>4.1.2. Si los datos son incorrectos se presenta un mensaje de error.</li> <li>4.1.3. Si ocurrió una excepción en la Base de Datos, se presenta un mensaje</li> </ol> </li> </ol>

de error indicando la excepción.
<b>Pos condiciones:</b> Los datos se reflejan en los diferentes listados del menú configuraciones.

**Tabla 6.3** Caso de uso modificación de registros de las tablas de Configuración

Caso de Uso: Eliminar
<b>Actores:</b> Administrador
<b>Descripción:</b> Permite eliminar registros de las diferentes tablas de configuración necesarias para manipular el sistema, entre ellas están: Configuración, Provincia, Ciudad, Cantón, Parroquia, Tipos de Documento, Agencia, Departamento, Empleados, Tipo de Contacto, entre otros.
<b>Precondiciones:</b> El usuario administrador es quien va poder eliminar este tipo de información, para lo cual debe estar logueado en el sistema.
<b>Flujo Normal:</b> <ol style="list-style-type: none"> <li>1. El administrador accede al menú Configuraciones y accede al formulario Listado de la información que desea eliminar.</li> <li>2. El usuario administrador establece los criterios de búsqueda para determinar el registro a eliminar.</li> <li>3. El usuario administrador pulsa Eliminar registro.</li> </ol>
<b>Flujo Alternativo:</b> <ol style="list-style-type: none"> <li>3.1. El sistema verifica en la base de datos si existe otros datos relacionados con ese registro. <ol style="list-style-type: none"> <li>3.1.1. Si no existe datos relacionados se procede al eliminar el registro lógicamente y posteriormente se muestra un mensaje indicando que los datos se han eliminado correctamente.</li> <li>3.1.2. En caso de que exista datos relacionados se presenta un mensaje de error.</li> <li>3.1.3. Si ocurrió una excepción en la Base de Datos, se presenta un mensaje de error indicando la excepción.</li> </ol> </li> </ol>
<b>Pos condiciones:</b> Los datos eliminados se reflejan en los diferentes listados del menú configuraciones ya que esos datos ya no estarán constando.

**Tabla 6.4** Caso de uso Eliminación de datos de configuración

Caso de Uso: Visualizar
<b>Actores:</b> Administrador
<b>Descripción:</b> Permite visualizar la información sobre las diferentes tablas de configuración necesarias para manipular el sistema, entre ellas están: Configuración, Provincia, Ciudad, Cantón, Parroquia, Tipos de Documento, Agencia, Departamento, Empleados, Tipo de Contacto, entre otros.
<b>Precondiciones:</b>

El usuario administrador es quien van ha poder visualizar este tipo de información, para lo cual debe estar logueado en el sistema.
<b>Flujo Normal:</b> <ol style="list-style-type: none"> <li>1. El usuario administrador accede al menú Configuraciones y accede a los formularios de listados que desea visualizar.</li> <li>2. En caso de que exista más de una página de datos, el usuario procede a cambiar de página.</li> </ol>
<b>Flujo Alternativo:</b> <ol style="list-style-type: none"> <li>2.1. Si el usuario necesita buscar un registro determinado, establece los parámetros de búsqueda. <ol style="list-style-type: none"> <li>2.1.1. Si existe datos de acuerdo a los parámetros de busca se visualizan los registros encontrados.</li> <li>2.1.2. Si no existe datos de acuerdo a los parámetros de búsqueda no se visualizará ningún registro.</li> </ol> </li> </ol>
<b>Pos condiciones:</b> Los datos se reflejan en los diferentes listados del menú configuraciones.

**Tabla 6.5** Caso de uso Visualización de datos de las tablas de Configuración

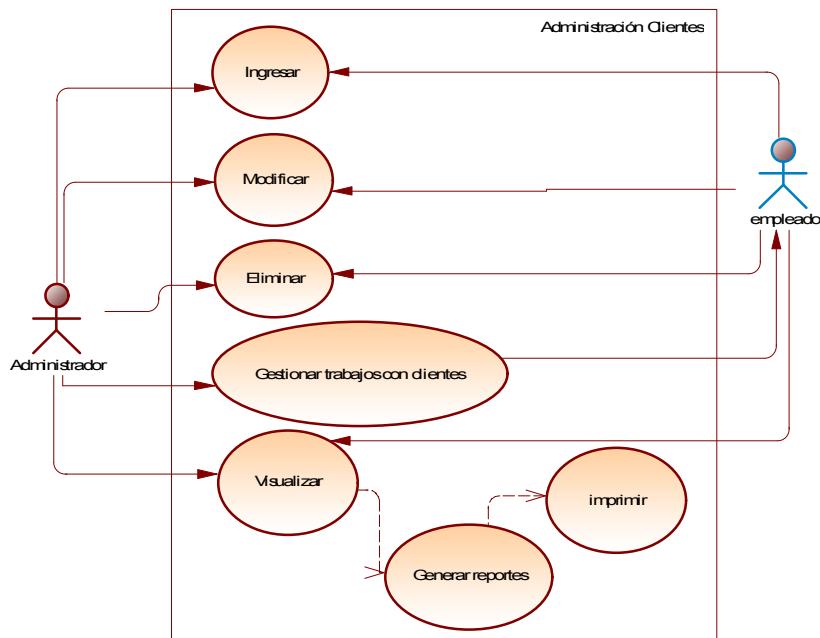
Caso de Uso: Generar Reportes
<b>Actores:</b> Administrador, empleado
<b>Descripción:</b> Permite generar reportes sobre la información de tablas de configuración necesarias para manipular el sistema, entre ellas están: Configuración, Provincia, Ciudad, Cantón, Parroquia, Tipos de Documento, Agencia, Departamento, Empleados, Tipo de Contacto, entre otros.
<b>Precondiciones:</b> El usuario administrador y el empleado son capaces de generar un reporte, para lo cual deben estar logueados en el sistema.
<b>Flujo Normal:</b> <ol style="list-style-type: none"> <li>1. El usuario administrador o el empleado accede al menú Configuraciones y posterior accede al formulario del cual se va a generar el reporte.</li> <li>2. En caso de que el usuario necesite generar reportes específicos, se debe establecer los parámetros de consulta.</li> </ol>
<b>Flujo Alternativo:</b> <ol style="list-style-type: none"> <li>2.1. Si el usuario necesita generar un reporte específico, se ingresan los parámetros de búsqueda y se realiza la consulta. <ol style="list-style-type: none"> <li>2.1.1. Si existe datos de acuerdo a los parámetros de busca se visualizan los registros encontrados en el reporte.</li> <li>2.1.2. Si no existe datos de acuerdo a los parámetros de búsqueda no se visualizará ningún registro en el reporte.</li> </ol> </li> <li>2.2. El usuario tiene la opción de elegir el formato del reporte: pdf, html, excel, entre otros</li> </ol>
<b>Pos condiciones:</b> Los datos se reflejan en cada uno de los reportes realizados.

**Tabla 6.6** Caso de uso Generar Reportes

<b>Caso de Uso: Imprimir</b>	
<b>Actores:</b>	Administrador, empleado
<b>Descripción:</b>	Permite imprimir reportes sobre la información de tablas de configuración necesarias para manipular el sistema, entre ellas están: Configuración, Provincia, Ciudad, Cantón, Parroquia, Tipos de Documento, Agencia, Departamento, Empleados, Tipo de Contacto, entre otros.
<b>Precondiciones:</b>	El usuario administrador y el empleado son capaces de imprimir un reporte, para lo cual deben estar logeados en el sistema.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario administrador o el empleado accede al menú Configuraciones y posterior accede al formulario del cual se va a generar el reporte.</li> <li>2. Una vez generado el reporte el usuario puede realizar una impresión del mismo.</li> </ol>
<b>Flujo Alternativo:</b>	
<b>Pos condiciones:</b>	Los datos se reflejan en cada una de las impresiones realizadas.

**Tabla 6.7** Caso de uso Imprimir Reporte.

**Caso de uso: Administración de Clientes**



**Figura 6.3** Diagrama de caso de uso Administración de Clientes

<b>Caso de Uso: Ingresar</b>	
<b>Actores:</b>	Administrador, empleado
<b>Descripción:</b>	Permite ingresar nuevos clientes al sistema Administrativo Contable.
<b>Precondiciones:</b>	El usuario administrador o el empleado es quien va poder ingresar nuevos clientes al sistema, para lo cual deben estar logueados en el sistema.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El administrador o el empleado accede al formularios de Clientes</li> <li>2. El administrador o el empleado ingresa la información del Cliente.</li> <li>3. El administrador o el empleado pulsa el botón Guardar para almacenar la información en la Base de Datos.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>3.1. El sistema verifica la información del formulario.               <ol style="list-style-type: none"> <li>3.1.1. Si los datos son correctos se procede al almacenar en la base de datos y posteriormente se muestra un mensaje indicando que los datos se han almacenado correctamente.</li> <li>3.1.2. Si los datos son incorrectos se presenta un mensaje de error.</li> <li>3.1.3. Si ocurrió una excepción en la Base de Datos, se presenta un mensaje de error indicando la excepción.</li> </ol> </li> </ol>
<b>Pos condiciones:</b>	Los datos se reflejan en los listados de clientes.

**Tabla 6.8** Caso de uso Ingresar Cliente

<b>Caso de Uso: Modificar</b>	
<b>Actores:</b>	Administrador, empleado
<b>Descripción:</b>	Permite modificar los datos de los clientes.
<b>Precondiciones:</b>	El usuario administrador o el empleado son quienes pueden modificar este tipo de información, para lo cual deben estar logueados en el sistema.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El administrador o el empleado accede al formulario del Listado de Clientes.</li> <li>2. El usuario administrador o el empleado establece los criterios de búsqueda para determinar el registro a modificar.</li> <li>3. El usuario administrador o el empleado pulsa editar registro y posterior procede a realizar los cambios necesarios.</li> <li>4. El usuario administrador o el empleado pulsa el botón Guardar para almacenar la información en la Base de Datos.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>4.1. El sistema verifica la información del formulario.               <ol style="list-style-type: none"> <li>4.1.1. Si los datos son correctos se procede al almacenar en la base de datos y posteriormente se muestra un mensaje indicando que los datos se han</li> </ol> </li> </ol>

<p>almacenado correctamente.</p> <p>4.1.2. Si los datos son incorrectos se presenta un mensaje de error.</p> <p>4.1.3. Si ocurrió una excepción en la Base de Datos, se presenta un mensaje de error indicando la excepción.</p>
<p><b>Pos condiciones:</b> Los datos se reflejan en el listado de clientes.</p>

**Tabla 6.9** Caso de uso Modificar Cliente

Caso de Uso: Eliminar
<p><b>Actores:</b> Administrador, empleado</p>
<p><b>Descripción:</b> Permite eliminar clientes mal ingresados.</p>
<p><b>Precondiciones:</b> El usuario administrador o el empleado son quienes pueden eliminar este tipo de información, para lo cual deben estar logueados en el sistema.</p>
<p><b>Flujo Normal:</b></p> <ol style="list-style-type: none"> <li>1. El administrador o el empleado accede al Listado Clientes.</li> <li>2. El usuario administrador establece los criterios de búsqueda para determinar el registro a eliminar.</li> <li>3. El usuario administrador o el empleado pulsa eliminar registro</li> </ol>
<p><b>Flujo Alternativo:</b></p> <ol style="list-style-type: none"> <li>3.1. El sistema verifica en la base de datos si existe otros datos relacionados con ese registro. <ol style="list-style-type: none"> <li>3.1.1. Si no existe datos relacionados se procede al eliminar el registro lógicamente y posteriormente se muestra un mensaje indicando que los datos se han eliminado correctamente.</li> <li>3.1.2. En caso de que exista datos relacionados se presenta un mensaje de error.</li> <li>3.1.3. Si ocurrió una excepción en la Base de Datos, se presenta un mensaje de error indicando la excepción.</li> </ol> </li> </ol>
<p><b>Pos condiciones:</b> Los datos eliminados se reflejan en el formulario de Clientes.</p>

**Tabla 6.10** Caso de uso eliminar Cliente

Caso de Uso: Visualizar
<p><b>Actores:</b> Administrador, empleado</p>
<p><b>Descripción:</b> Permite visualizar la información sobre los clientes.</p>
<p><b>Precondiciones:</b> El usuario administrador y el empleado son quienes va poder visualizar este tipo de información, para lo cual debe estar logueados en el sistema.</p>

<p><b>Flujo Normal:</b></p> <ol style="list-style-type: none"> <li>1. El usuario administrador o el empleado accede al formulario de listado de Clientes.</li> <li>2. En caso de que exista más de una página de datos, el usuario procede a cambiar de página.</li> </ol>
<p><b>Flujo Alternativo:</b></p> <ol style="list-style-type: none"> <li>2.1. Si el usuario necesita buscar un registro determinado, establece los parámetros de búsqueda. <ol style="list-style-type: none"> <li>2.1.1. Si existe datos de acuerdo a los parámetros de busca se visualizan los registros encontrados.</li> <li>2.1.2. Si no existe datos de acuerdo a los parámetros de búsqueda no se visualizará ningún registro.</li> </ol> </li> </ol>
<p><b>Pos condiciones:</b></p> <p>Los datos se reflejan en el listado de Clientes.</p>

**Tabla 6.11** Caso de uso Visualizar Datos de los Clientes

Caso de Uso: Generar Reportes
<p><b>Actores:</b></p> <p>Administrador, empleado</p>
<p><b>Descripción:</b></p> <p>Permite generar reportes sobre los clientes.</p>
<p><b>Precondiciones:</b></p> <p>El usuario administrador y el empleado son capaces de generar un reporte, para lo cual deben estar logueados en el sistema.</p>
<p><b>Flujo Normal:</b></p> <ol style="list-style-type: none"> <li>1. El usuario administrador o el empleado accede al formulario de Clientes.</li> <li>2. En caso de que el usuario necesite generar reportes específicos, se debe establecer los parámetros de consulta.</li> </ol>
<p><b>Flujo Alternativo:</b></p> <ol style="list-style-type: none"> <li>2.1. Si el usuario necesita generar un reporte específico, se ingresan los parámetros de búsqueda y se realiza la consulta. <ol style="list-style-type: none"> <li>2.1.1. Si existe datos de acuerdo a los parámetros de busca se visualizan los registros encontrados en el reporte.</li> <li>2.1.2. Si no existe datos de acuerdo a los parámetros de búsqueda no se visualizará ningún registro en el reporte.</li> </ol> </li> <li>2.2. El usuario tiene la opción de elegir el formato del reporte: pdf, html, excel, entre otros</li> </ol>
<p><b>Pos condiciones:</b></p> <p>Los datos se reflejan en cada uno de los reportes realizados.</p>

**Tabla 6.12** Caso de uso Generar Reportes sobre los Clientes



<b>Caso de Uso: Imprimir</b>	
<b>Actores:</b>	Administrador, empleado
<b>Descripción:</b>	Permite imprimir reportes sobre los reportes efectuados.
<b>Precondiciones:</b>	El usuario administrador y el empleado son capaces de imprimir un reporte, para lo cual deben estar logueados en el sistema.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario administrador o el empleado acceden al formulario de Clientes.</li> <li>2. Una vez generado el reporte el usuario puede realizar una impresión del mismo.</li> </ol>
<b>Flujo Alternativo:</b>	
<b>Pos condiciones:</b>	Los datos se reflejan en cada una de las impresiones realizadas.

**Tabla 6.13** Caso de uso imprimir reporte de Clientes

<b>Caso de Uso: Gestionar trabajos con clientes</b>	
<b>Actores:</b>	Administrador, empleado
<b>Descripción:</b>	Permite gestionar trabajos con clientes, según se presenten las oportunidades, estas pueden ser: soporte técnico, mantenimientos, proyectos, etc. Los cuales serán asignados a los empleados.
<b>Precondiciones:</b>	El usuario administrador debe estar logueado en el sistema.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El administrador abre el formulario de las actividades ya sean de soporte técnico, mantenimiento o proyectos.</li> <li>2. El administrador ingresa la información.</li> <li>3. El administrador pulsa el botón Guardar para almacenar la información en la Base de Datos.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>3.1. El sistema verifica la información del formulario. <ol style="list-style-type: none"> <li>3.1.1. Si los datos son correctos se procede al almacenar en la base de datos y posteriormente se muestra un mensaje indicando que los datos se han almacenado correctamente.</li> <li>3.1.2. Si los datos son incorrectos se presenta un mensaje de error.</li> <li>3.1.3. Si ocurrió una excepción en la Base de Datos, se presenta un mensaje de error indicando la excepción.</li> </ol> </li> </ol>
<b>Pos condiciones:</b>	Los datos se reflejan en los diferentes listados de actividades.

**Tabla 6.14** Caso de uso Gestionar trabajos con Clientes

## Caso de uso: Administración de Créditos

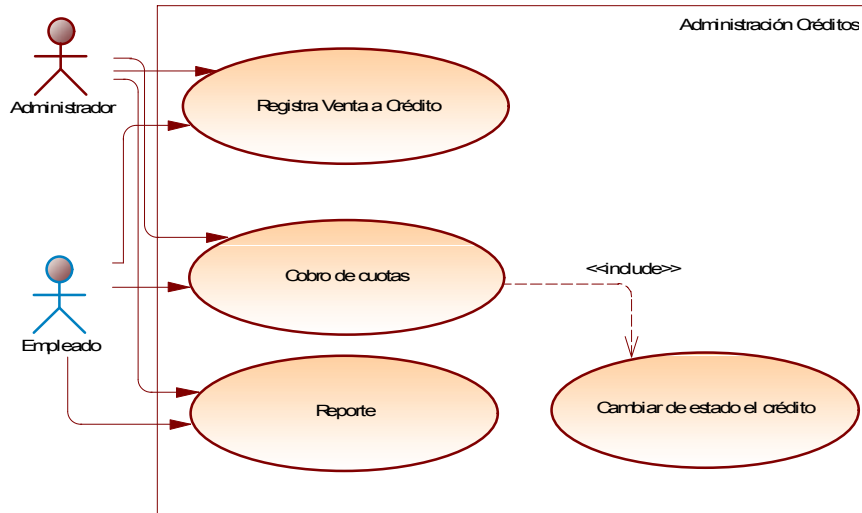


Figura 6.4 Diagrama de caso de uso administración de Créditos

Caso de Uso: Registrar venta a crédito	
<b>Actores:</b>	Administrador, empleado
<b>Descripción:</b>	Permite registrar una venta a crédito. Este proceso se lo realiza en el módulo de Ventas.
<b>Precondiciones:</b>	El cliente debe establecer el número de cuotas
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El administrador o el empleado accede al módulo de Ventas</li> <li>2. El administrador o el empleado ingresan al formulario de ventas.</li> <li>3. El administrador o el empleado ingresan la venta.</li> <li>4. El administrador o el empleado pulsa el botón Guardar para almacenar la información en la Base de Datos.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>4.1. El sistema verifica la información del formulario.                             <ol style="list-style-type: none"> <li>4.1.1. Si los datos son correctos se procede al almacenar en la base de datos y posteriormente se muestra un mensaje indicando que los datos se han almacenado correctamente.</li> <li>4.1.2. Si los datos son incorrectos se presenta un mensaje de error.</li> <li>4.1.3. Si ocurrió una excepción en la Base de Datos, se presenta un mensaje de error indicando la excepción.</li> </ol> </li> </ol>
<b>Pos condiciones:</b>	Los datos se reflejan en los listados de Ventas.

Tabla 6.15 Caso de uso Registrar venta a crédito

<b>Caso de Uso: Cobro de cuotas</b>	
<b>Actores:</b>	Administrador, empleado
<b>Descripción:</b>	Permite realizar el cobro de una determinada cuota de una venta a crédito
<b>Precondiciones:</b>	El usuario administrador o el empleado son quienes pueden realizar el cobro de dicha cuota para ello deben estar logueados en el sistema.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El administrador o el empleado accede al formulario de créditos pendientes.</li> <li>2. El usuario administrador o el empleado establece los criterios de búsqueda para determinar el cliente que va a cancelar.</li> <li>3. El usuario administrador o el empleado ingresa los valores que el cliente va a cancelar.</li> <li>4. El usuario administrador o el empleado pulsa el botón Aceptar para registrar el valor de la cuota en la Base de Datos.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>4.1. El sistema verifica la información del formulario.               <ol style="list-style-type: none"> <li>4.1.1. Si los datos son correctos se procede al almacenar en la base de datos y posteriormente se muestra un mensaje indicando que los datos se han almacenado correctamente.</li> <li>4.1.2. Si los datos son incorrectos se presenta un mensaje de error.</li> <li>4.1.3. Si ocurrió una excepción en la Base de Datos, se presenta un mensaje de error indicando la excepción.</li> </ol> </li> </ol>
<b>Pos condiciones:</b>	<p>Los datos se reflejan en los listados de créditos.</p> <p>El saldo del crédito pendiente disminuye.</p>

**Tabla 6.16** Caso de uso Cobro de Cuotas

<b>Caso de Uso: Cambiar de estado al crédito</b>	
<b>Actores:</b>	Administrador, empleado
<b>Descripción:</b>	Permite cambiar el estado del crédito a finalizado una vez que el cliente haya cancelado todas las cuotas.
<b>Precondiciones:</b>	Para que se pueda cambiar de estado del crédito a finalizado, primero se deben haber cancelado todas las cuotas que tiene el cliente.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El administrador o el empleado acceden al formulario de crédito.</li> <li>2. El administrador o el empleado realiza el cobro de la última cuota del crédito que tienen el cliente.</li> <li>3. El usuario administrador o el empleado pulsa el botón Guardar para almacenar la información en la Base de Datos.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>3.1. El sistema verifica la información del formulario.</li> </ol>

<p>3.1.1. Si los datos son correctos se procede al almacenar en la base de datos y posteriormente se muestra un mensaje indicando que los datos se han almacenado correctamente.</p> <p>3.1.2. Si los datos son incorrectos se presenta un mensaje de error.</p> <p>3.1.3. Si ocurrió una excepción en la Base de Datos, se presenta un mensaje de error indicando la excepción.</p> <p>3.2. El sistema verifica que sea la última cuota que se cobra.</p> <p>3.2.1 Si resulta ser la última cuota, se procede al cambio de estado del crédito</p> <p>3.2.2 Si no es la última cuota, no se realiza ningún cambio de estado del crédito.</p>
<p><b>Pos condiciones:</b> El crédito pasa a estado finalizado después que se hayan cobrado todas las cuotas, esto se refleja en los créditos finalizados.</p>

**Tabla 6.17** Caso de uso Cambiar de estado al crédito

Caso de Uso: Generar Reportes
<p><b>Actores:</b> Administrador, empleado</p>
<p><b>Descripción:</b> Permite generar reportes sobre los clientes.</p>
<p><b>Precondiciones:</b> El usuario administrador y el empleado son capaces de generar un reporte, para lo cual deben estar logueados en el sistema.</p>
<p><b>Flujo Normal:</b></p> <ol style="list-style-type: none"> <li>1. El usuario administrador o el empleado accede al formulario de Créditos de Clientes.</li> <li>2. En caso de que el usuario necesite generar reportes específicos, se debe establecer los parámetros de consulta.</li> </ol>
<p><b>Flujo Alternativo:</b></p> <ol style="list-style-type: none"> <li>2.1. Si el usuario necesita generar un reporte específico, se ingresan los parámetros de búsqueda y se realiza la consulta. <ol style="list-style-type: none"> <li>2.1.1. Si existe datos de acuerdo a los parámetros de busca se visualizan los registros encontrados en el reporte.</li> <li>2.1.2. Si no existe datos de acuerdo a los parámetros de búsqueda no se visualizará ningún registro en el reporte.</li> </ol> </li> <li>2.2. El usuario tiene la opción de elegir el formato del reporte: pdf, html, excel, entre otros</li> </ol>
<p><b>Pos condiciones:</b> Los datos se reflejan en cada uno de los reportes realizados.</p>

**Tabla 6.18** Caso de uso Generar Reportes de créditos

## Caso de Uso: Administración de Seguridades

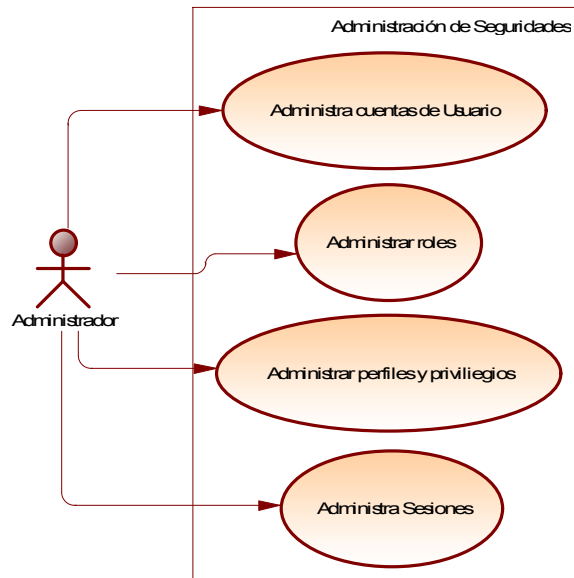


Figura 6.5 Diagrama de caso de uso Administración de Seguridades

Caso de Uso: Administrar cuentas de usuario	
<b>Actores:</b>	Administrador
<b>Descripción:</b>	Permite administrar las diferentes cuentas de usuarios.
<b>Precondiciones:</b>	El usuario administrador es el único que podrá administrar esta información, para lo cual debe estar logueados en el sistema.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario administrador accede al menú de configuraciones.</li> <li>2. El usuario administrador accede al formulario de Usuarios.</li> <li>3. El usuario administrador ingresa la información.</li> <li>4. El usuario administrador presiona el botón guardar para enviar los datos a la Base de dato.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>4.1. El sistema verifica la información del formulario.               <ol style="list-style-type: none"> <li>4.1.1. Si los datos son correctos se procede al almacenar en la base de datos y posteriormente se muestra un mensaje indicando que los datos se han almacenado correctamente.</li> <li>4.1.2. Si los datos son incorrectos se presenta un mensaje de error.</li> <li>4.1.3. Si ocurrió una excepción en la Base de Datos, se presenta un mensaje de error indicando la excepción.</li> </ol> </li> </ol>
<b>Pos condiciones:</b>	Los datos se reflejan en el listado de usuarios.

Tabla 6.19 Caso de uso Administrar Cuentas de usuario

<b>Caso de Uso: Administrar roles</b>	
<b>Actores:</b>	Administrador
<b>Descripción:</b>	Permite administrar las diferentes roles que va tener el sistema.
<b>Precondiciones:</b>	El usuario administrador es el único que podrá administrar esta información, para lo cual debe estar logueados en el sistema.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El usuario administrador accede al menú de configuraciones.</li> <li>2. El usuario administrador accede al formulario de Roles.</li> <li>3. El usuario administrador ingresa la información.</li> <li>4. El usuario administrador presiona el botón guardar para enviar los datos a la Base de dato.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>4.1. El sistema verifica la información del formulario.               <ol style="list-style-type: none"> <li>4.1.1. Si los datos son correctos se procede al almacenar en la base de datos y posteriormente se muestra un mensaje indicando que los datos se han almacenado correctamente.</li> <li>4.1.2. Si los datos son incorrectos se presenta un mensaje de error.</li> <li>4.1.3. Si ocurrió una excepción en la Base de Datos, se presenta un mensaje de error indicando la excepción.</li> </ol> </li> </ol>
<b>Pos condiciones:</b>	<p>Los datos se reflejan en el listado de roles.</p> <p>Los roles son establecido de acuerdo a las necesidades de cada módulo.</p>

**Tabla 6.20** Caso de uso Administrar roles

<b>Caso de Uso: Administrar perfiles y privilegios</b>	
<b>Actores:</b>	Administrador
<b>Descripción:</b>	Permite administrar los diferentes perfiles de usuario, para que dicho usuario ya pueda ingresar al sistema.
<b>Precondiciones:</b>	<p>El usuario administrador es el único que podrá administrar esta información, para lo cual deben estar logueados en el sistema.</p> <p>Debe estar creado por lo menos un Usuario.</p> <p>Debe estar creado por lo menos un Rol.</p>

<p><b>Flujo Normal:</b></p> <ol style="list-style-type: none"> <li>1. El usuario administrador accede al menú de configuraciones.</li> <li>2. El usuario administrador accede al formulario de perfiles.</li> <li>3. El usuario administrador ingresa la información.</li> <li>4. El usuario administrador selecciona el los roles que va tener el usuario.</li> <li>5. El usuario administrador presiona el botón guardar para enviar los datos a la Base de dato.</li> </ol>
<p><b>Flujo Alternativo:</b></p> <ol style="list-style-type: none"> <li>5.1. El sistema verifica la información del formulario. <ol style="list-style-type: none"> <li>5.1.1. Si los datos son correctos se procede al almacenar en la base de datos y posteriormente se muestra un mensaje indicando que los datos se han almacenado correctamente.</li> <li>5.1.2. Si los datos son incorrectos se presenta un mensaje de error.</li> <li>5.1.3. Si ocurrió una excepción en la Base de Datos, se presenta un mensaje de error indicando la excepción.</li> </ol> </li> </ol>
<p><b>Pos condiciones:</b></p> <p>Los datos se reflejan en el listado de perfiles. El usuario asignado a un perfil ya puede acceder al sistema.</p>

**Tabla 6.21** Caso de uso Administrar perfiles y privilegios

Caso de Uso: Administrar sesiones
<p><b>Actores:</b></p> <p>Administrador</p>
<p><b>Descripción:</b></p> <p>Permite administrar las sesiones abiertas. (Control de usuarios conectados en el sistema).</p>
<p><b>Precondiciones:</b></p> <p>El usuario administrador es el único que podrá administrar esta información, para lo cual deben estar logueados en el sistema. Deben estar creados los perfiles de usuario, para que los usuarios puedan ingresar al sistema.</p>
<p><b>Flujo Normal:</b></p> <ol style="list-style-type: none"> <li>1. El usuario administrador accede al menú de configuraciones.</li> <li>2. El usuario administrador accede al formulario de sesiones.</li> <li>3. El usuario administrador Visualiza las sesiones abiertas.</li> <li>4. El usuario manipula las sesiones q están abiertas.</li> </ol>
<p><b>Flujo Alternativo:</b></p> <ol style="list-style-type: none"> <li>4.1. El usuario verifica que no existan sesiones abiertas por equivocación. <ol style="list-style-type: none"> <li>4.1.1. Si existen sesiones abiertas por equivocación el administrador cierra la sesión.</li> </ol> </li> </ol>
<p><b>Pos condiciones:</b></p> <p>Los datos se reflejan en el listado Sesiones.</p>

**Tabla 6.22** Caso de uso Administrar sesiones

### 6.9.1.3.2 Diagramas de Clases

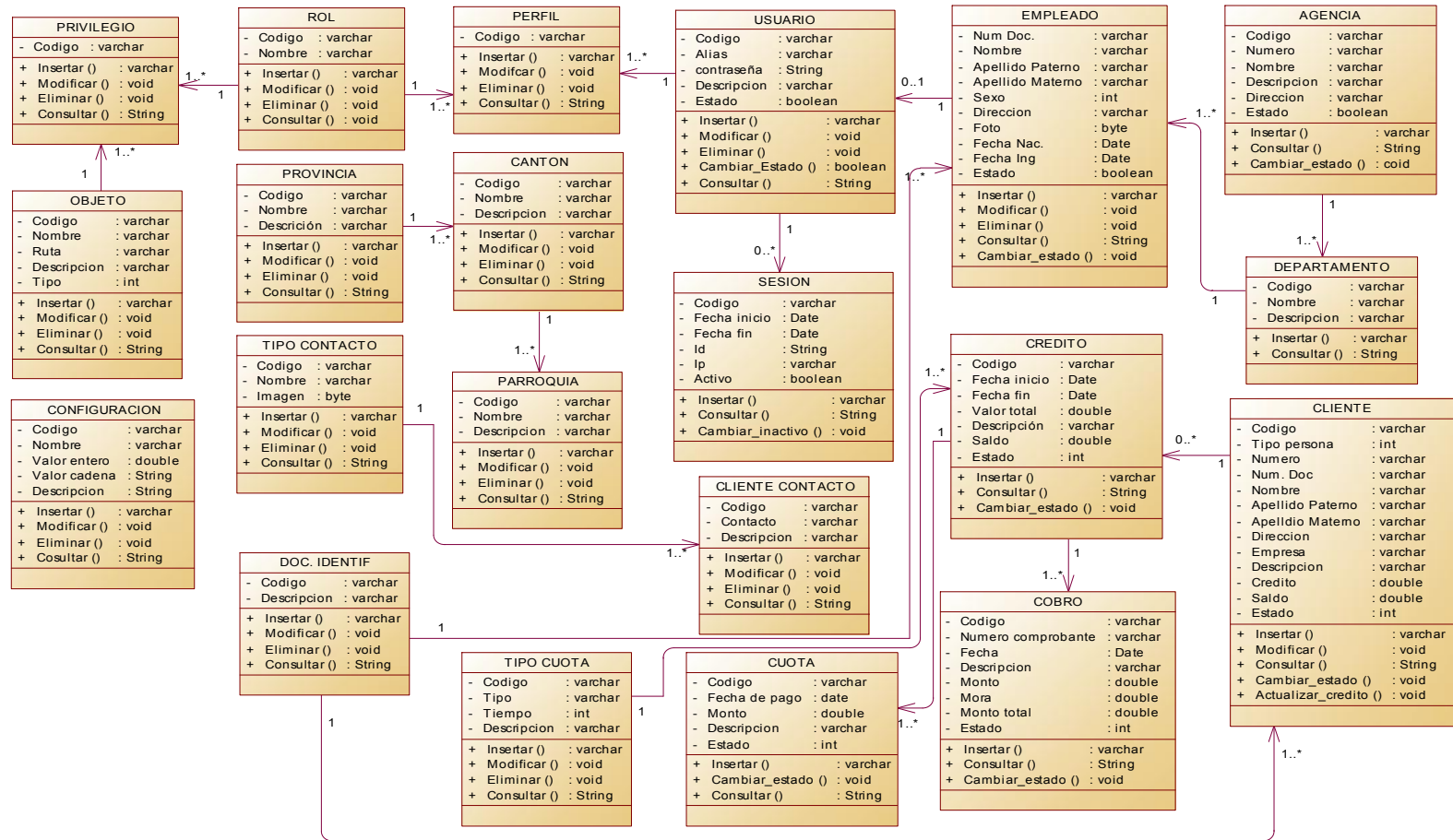


Figura 6.6 Diagrama de Clases



### 6.9.1.3.3 Diagramas de Secuencia

**Nota:** En los siguientes diagramas en donde este asignado como actor Usuario cabe recalcar que tanto el usuario administrador como el empleado puede manipular la información.

#### 6.9.1.3.3.1 Diagrama de Secuencia: Inicio de Sesión

##### Inicio de Sesión

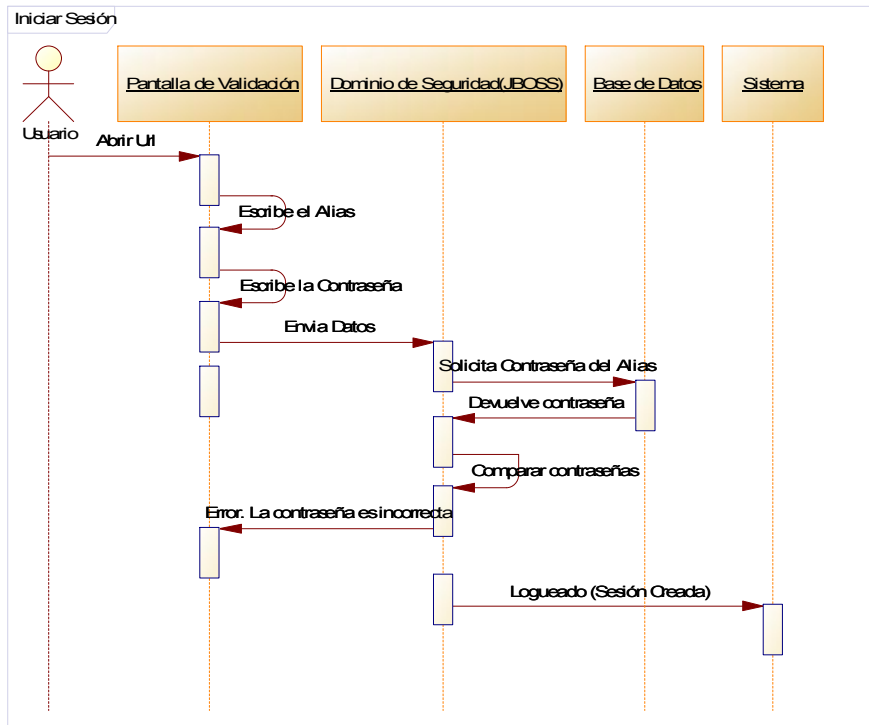


Figura 6.7 Diagrama de secuencia iniciar sesión

### 6.9.1.3.3.2 Diagrama de secuencia Administración de tablas de Configuración

#### Ingreso de Datos

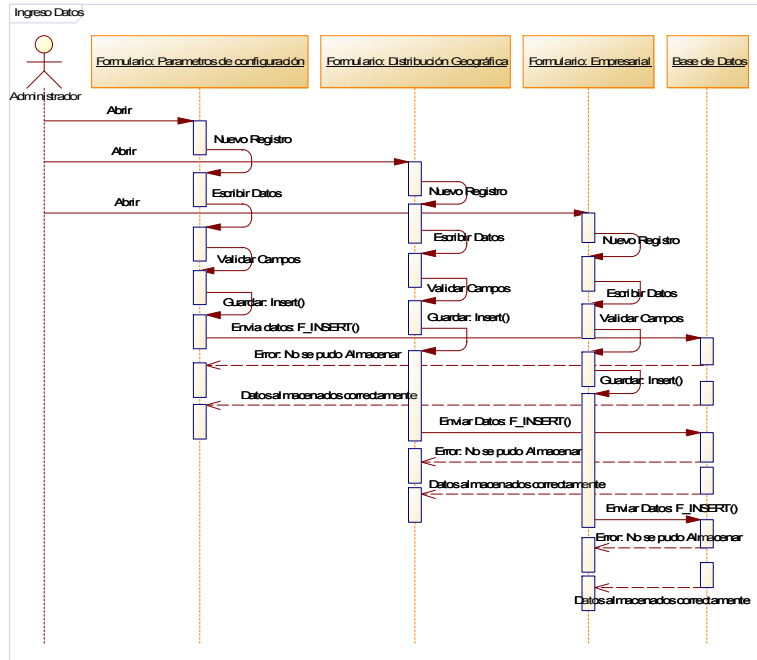


Figura 6.8 Diagrama de secuencia Ingreso de Datos de configuración

#### Modificación de Datos

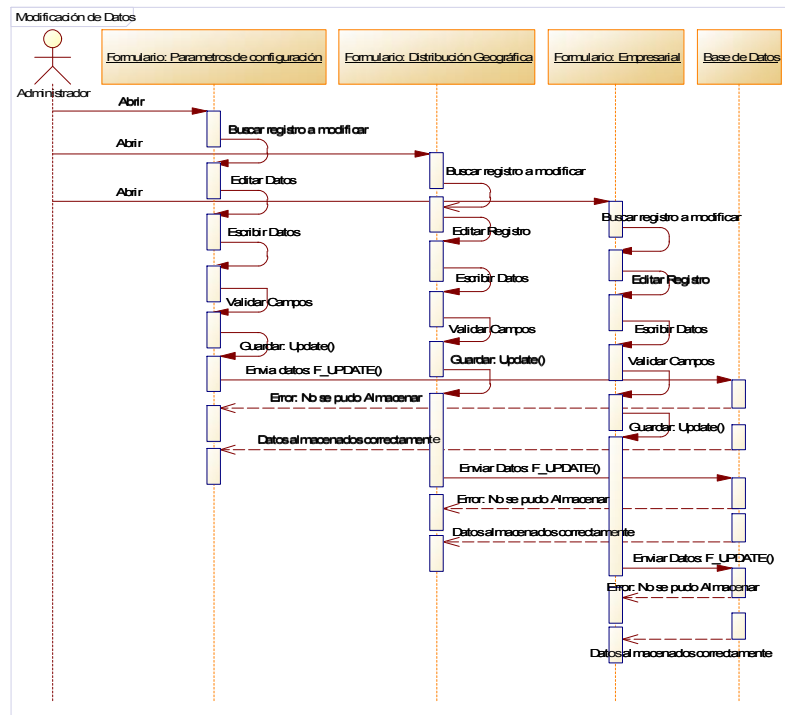


Figura 6.9 Diagrama de secuencia modificación de datos de configuración

## Eliminar Datos

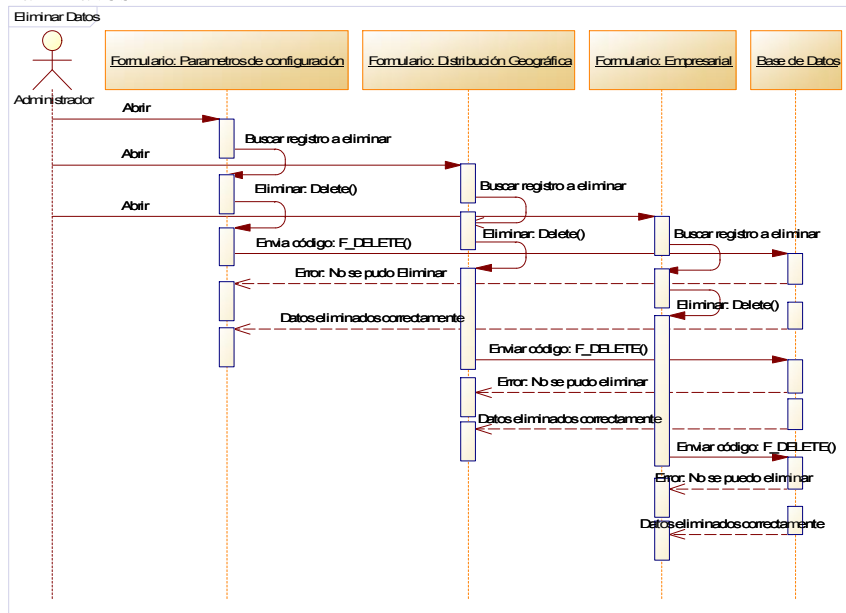


Figura 6.10 Diagrama de secuencia eliminar de datos de configuración

## Visualizar Datos

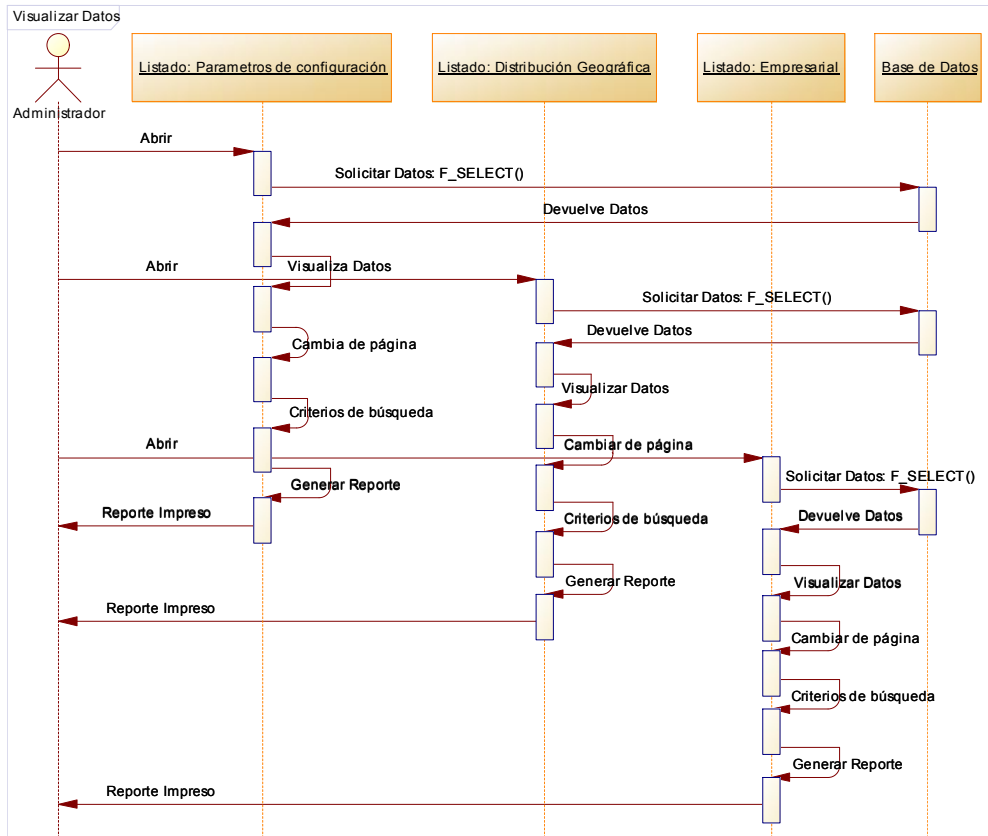


Figura 6.11 Diagrama de secuencia Visualización de Datos de configuración

### 6.9.1.3.3 Diagramas de secuencia: Administración de Clientes

#### Ingresar Datos

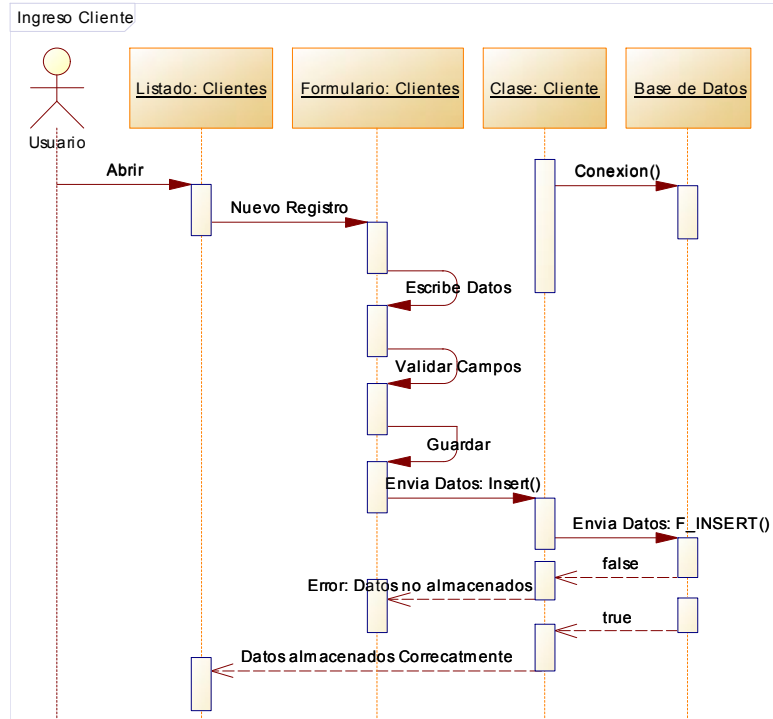


Figura 6.12 Diagrama de secuencia Ingreso de Datos de Clientes

#### Modificación de Datos:

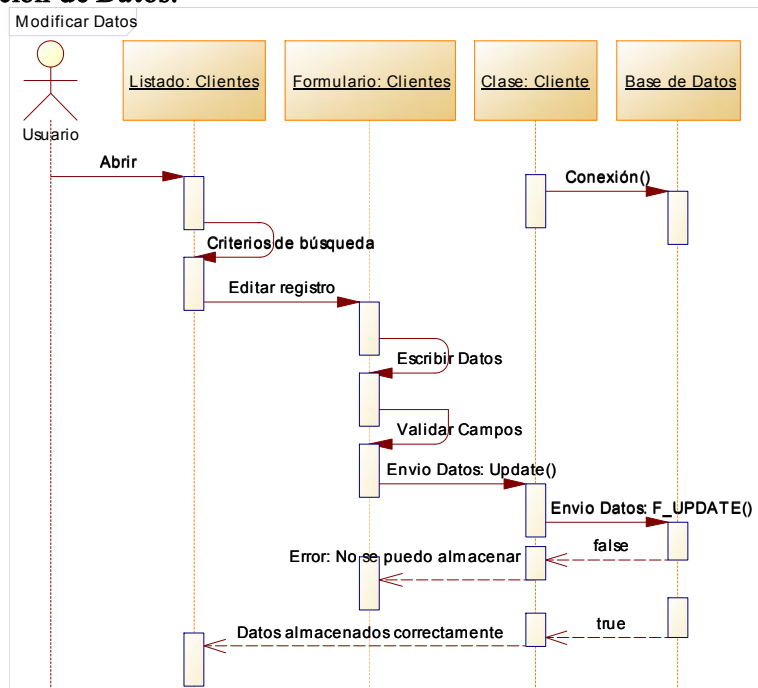
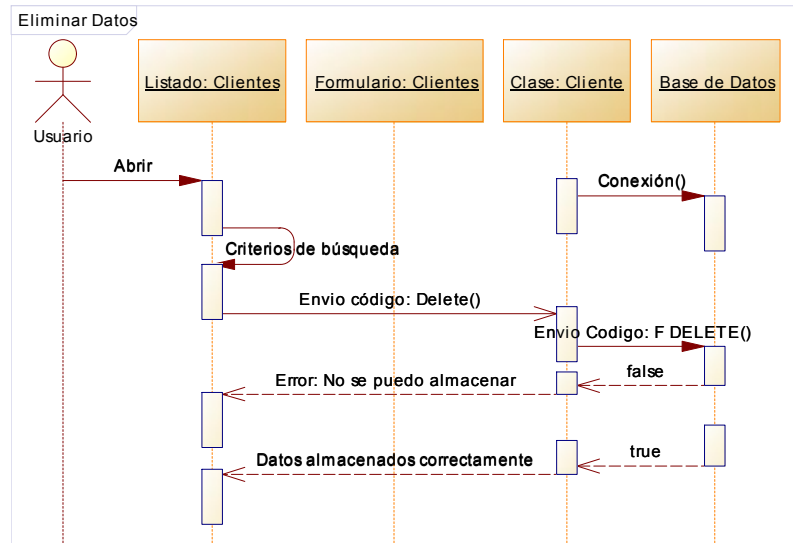


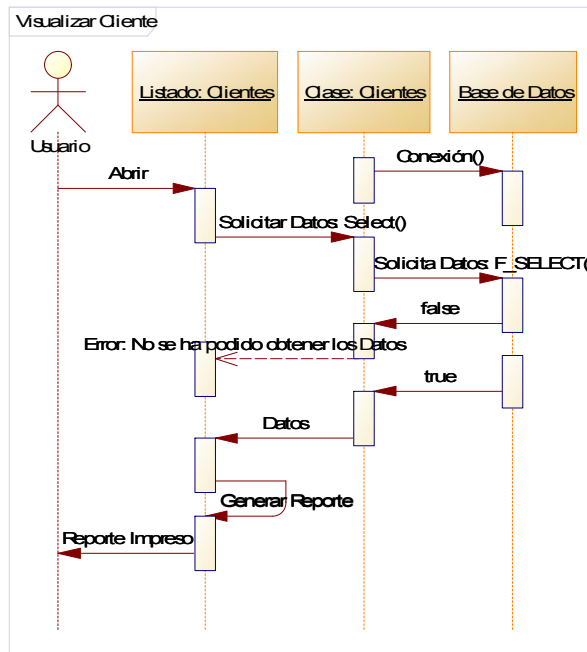
Figura 6.13 Diagrama de secuencia Modificación de Datos de Clientes

**Eliminar Datos:**



**Figura 6.14** Diagrama de secuencia Eliminar Clientes

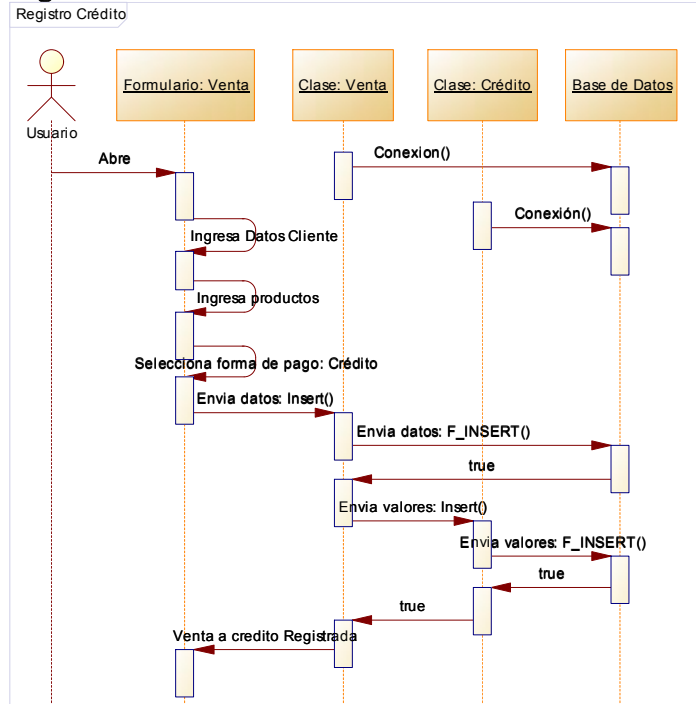
**Visualizar Datos:**



**Figura 6.15** Diagrama de secuencia Visualizar Datos de Clientes

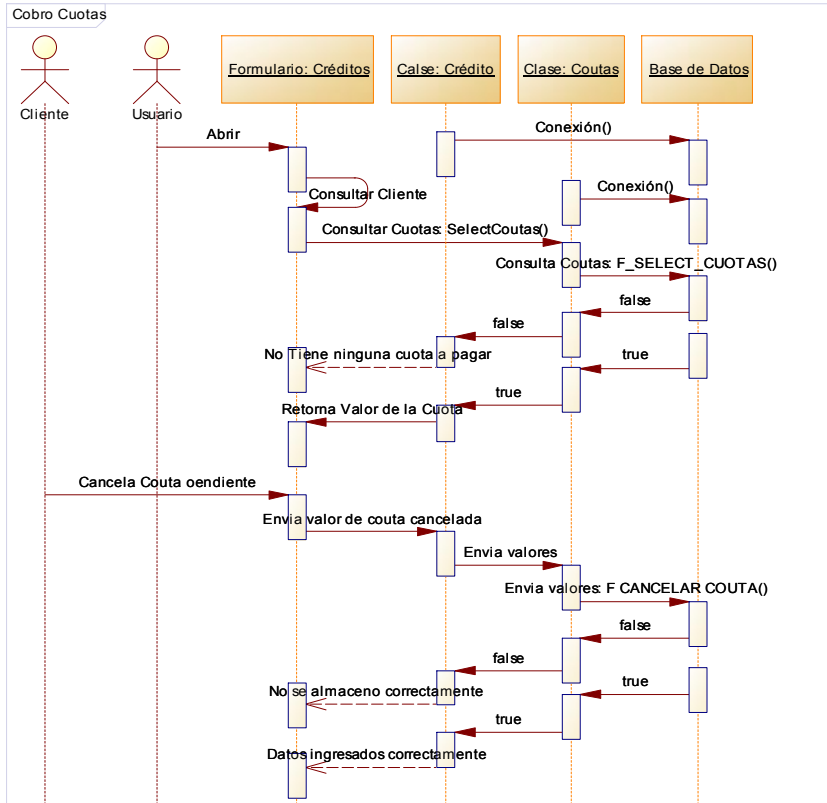
### 6.9.1.3.3.4 Diagramas de secuencia: Administración de Créditos

#### Registrar Venta a Crédito:



**Figura 6.16**  
Diagrama de secuencia registrar venta a crédito

#### Cobro de Cuotas:



**Figura 6.17** Diagrama de secuencia Cobro de Cuotas de un crédito

### 6.9.1.3.3.5 Diagramas de secuencia: Administración de Seguridades.

#### Administrar de Cuentas de Usuario:

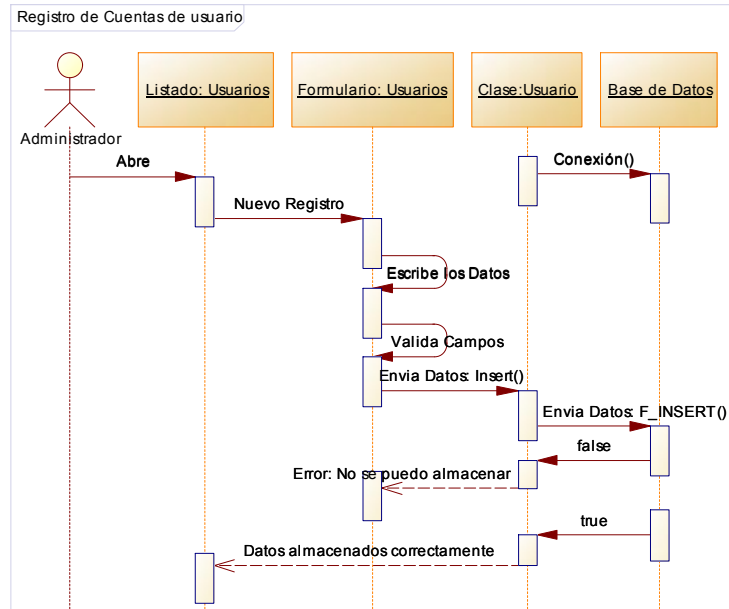


Figura 6.18 Diagrama de secuencia Administración de Cuentas de Usuario

#### Administrar Roles de Usuario:

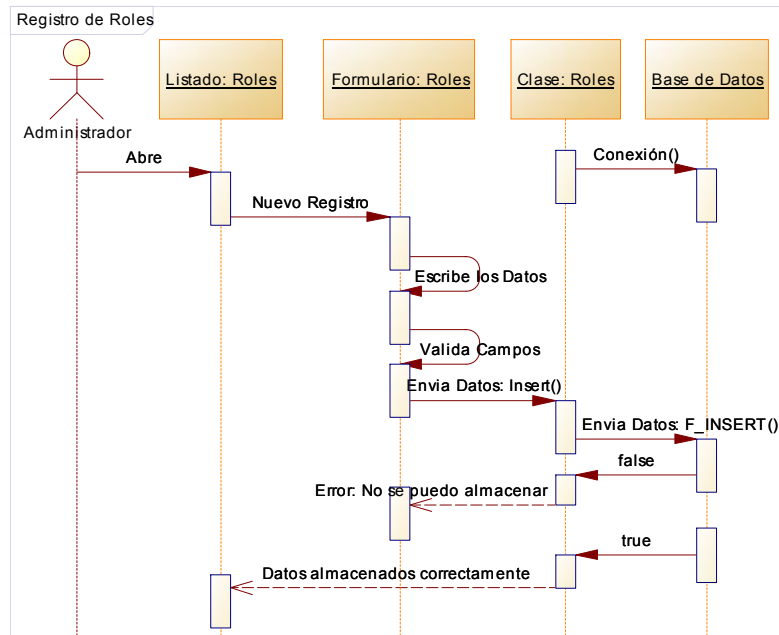


Figura 6.19 Diagrama de secuencia Administración de Roles de Usuario

## Administración de Perfiles de Usuario:

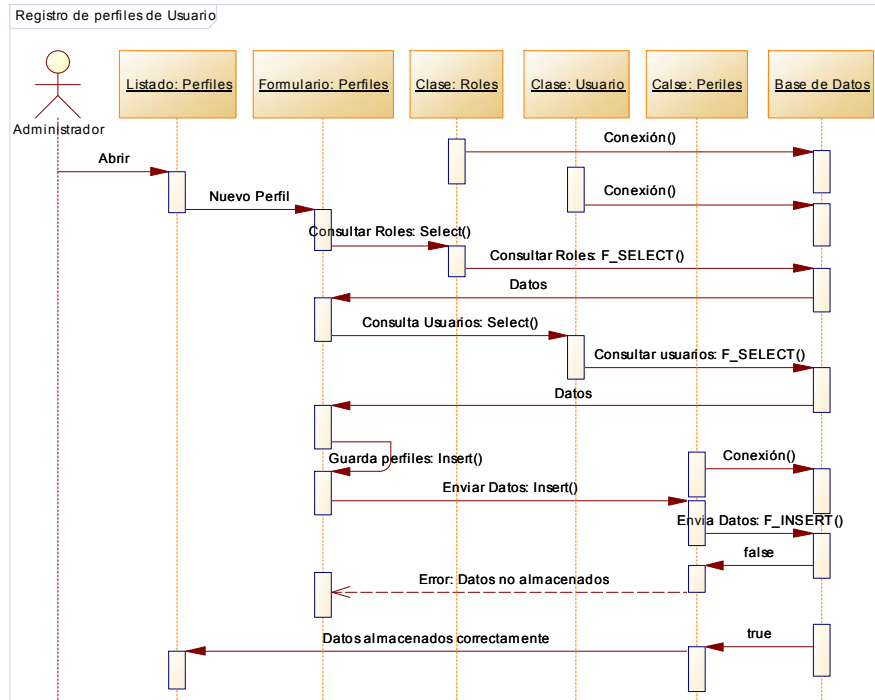


Figura 6.20 Diagrama de secuencia Administración de perfiles de usuario

## Administración de Sesiones de Usuario:

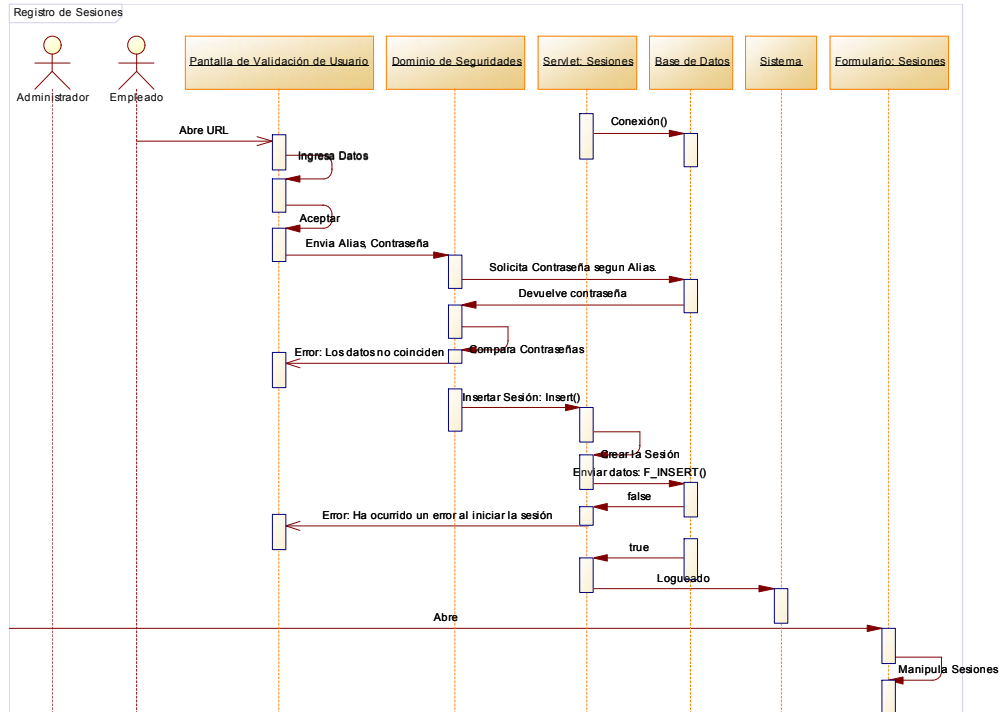


Figura 6.21 Diagrama de secuencia Administración de Sesiones de usuario



### 6.9.1.3.4 Diagrama de Actividades

#### 6.9.1.3.4.1 Diagrama de Actividad: Administración Tablas de Configuración

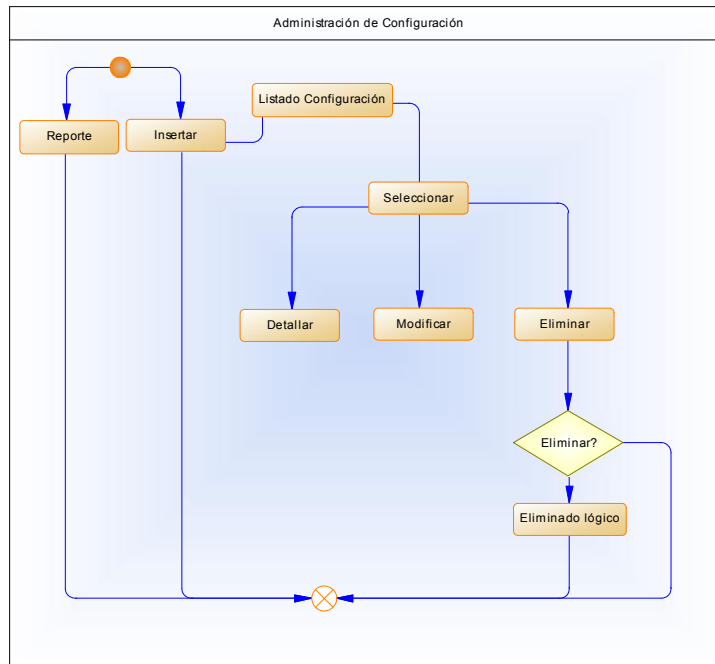


Figura 6.22 Diagrama de actividad Administración de tablas de configuración

#### 6.9.1.3.4.2 Diagrama de Actividad: Administración de Créditos

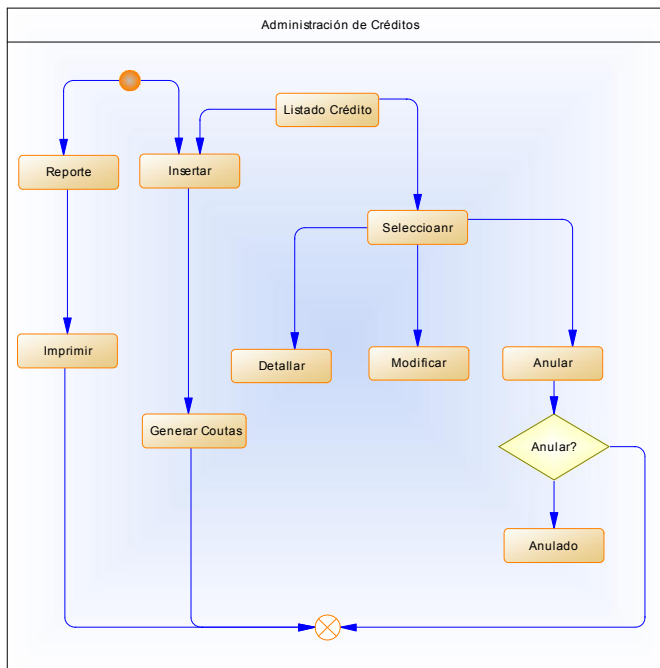


Figura 6.23 Diagrama de actividad Administración de Créditos

### 6.9.1.3.4.3 Diagrama de actividad: Control de Sesiones de Usuario

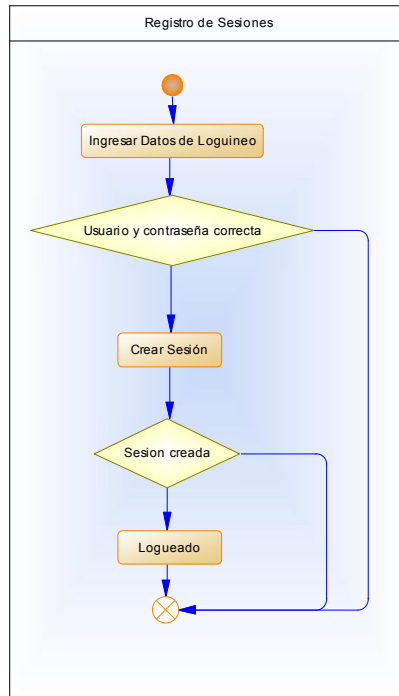


Figura 6.24 Diagrama de actividad Registros de Sesiones de usuario

### 6.9.1.3.5 Diagrama de Despliegues

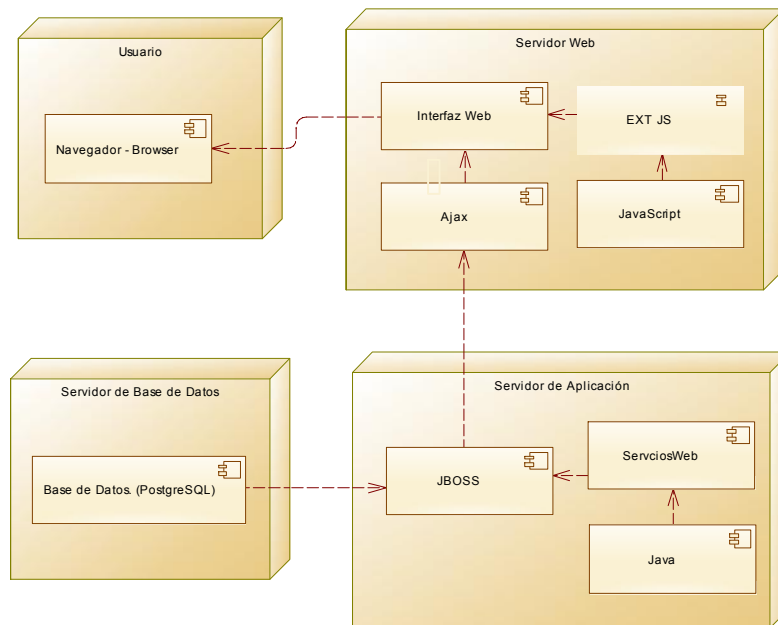


Figura 6.25 Diagrama de despliegue del Módulo de Gestión de Clientes y Seguridades

## 6.9.2 Diseño del Sistema

### 6.9.2.1 Diseño de Base de Datos.

El módulo de Gestión de Clientes y Seguridades forma parte del Sistema Administrativo Contable de la empresa Besixplus CIA LTDA, y según el análisis de la información obtenida se presenta el siguiente diseño de Base de Datos.

**Nota:** Cabe recalcar que el modelado de la base de datos se lo refleja con los siguientes colores para distinguir la funcionalidad.

	Gestión de de Seguridades
	Gestión de Clientes
	Gestión de Tablas de Configuración
	Módulo de Contabilidad

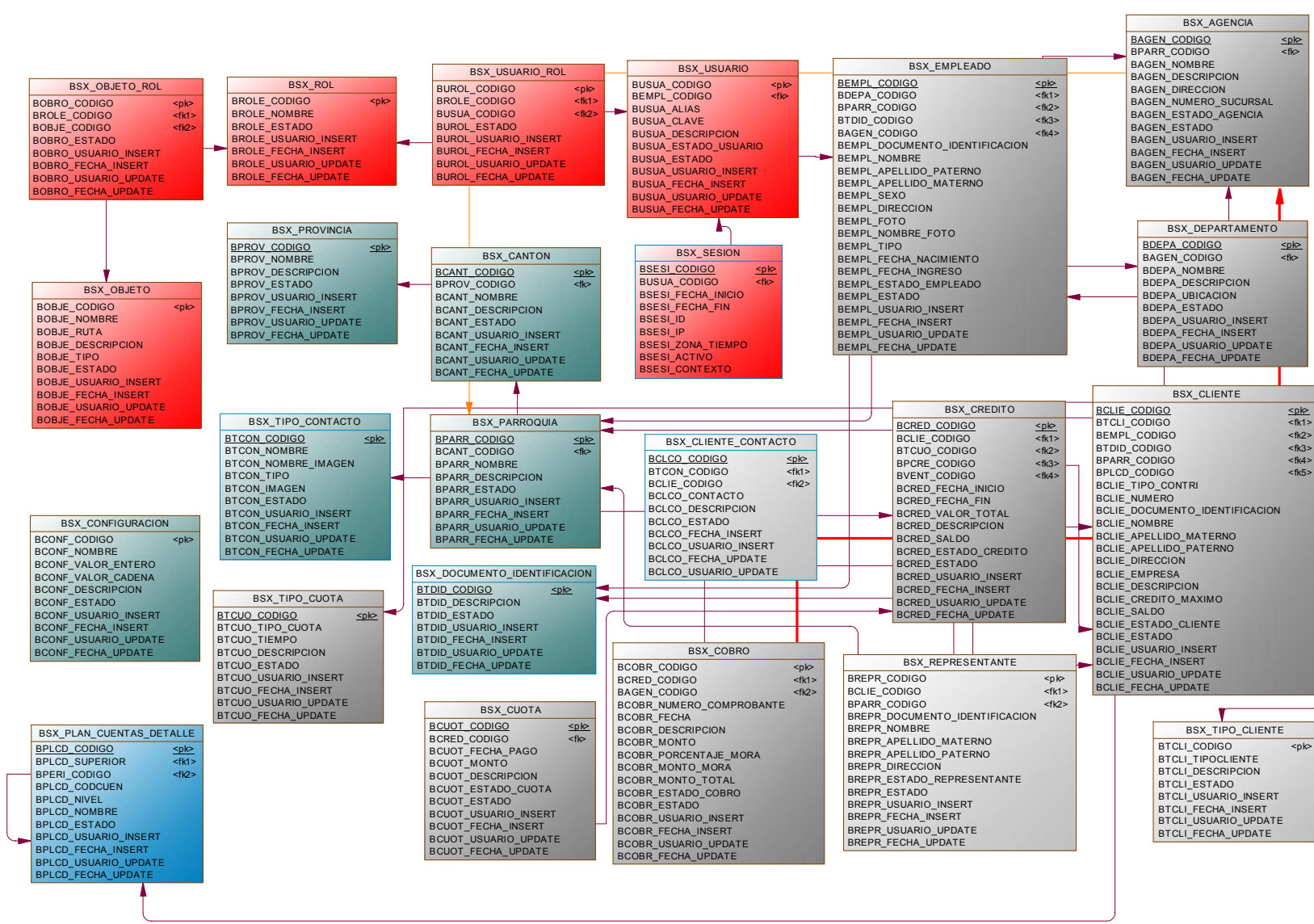


Figura 6.26 Diseño de la base de datos del Módulo de Clientes y seguridades

## 6.9.2.2 Diccionario de Datos

**TABLA: configuración**

<b>Clave Primaria:</b>	bconf_codigo	
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
bconf_codigo	varchar(20)	Campo que almacena la clave primaria
bconf_nombre	varchar(200)	Nombre del parámetro de configuración
bconf_valor_entero	numeric	Valor entero del parámetro de configuración
bconf_valor_cadena	varchar(100)	Valor cadena del parámetro de configuración
bconf_descripcion	varchar(200)	Descripción sobre el parámetro de configuración
bconf_estado	bool	Estado del registro: True: Activo False: Inactivo
bconf_usuario_insert	varchar(50)	Usuario que inserta
bconf_fecha_insert	timestamp with time	Fecha que inserta
bconf_usuario_update	varchar(50)	Usuario que actualiza
bconf_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.23** Descripción de la tabla Configuración

**TABLA: provincia**

<b>Clave Primaria:</b>	bprov_codigo	
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
bprov_codigo	varchar(20)	Campo que almacena la clave primaria
bprov_nombre	varchar(50)	Nombre de la provincia.
bprov_descripcion	text	Descripción detallada referente a la provincia
bprov_estado	bool	Estado del registro: True: Activo False: Inactivo
bprov_usuario_insert	varchar(50)	Usuario que inserta
bprov_fecha_insert	timestamp with time	Fecha que inserta
bprov_usuario_update	varchar(50)	Usuario que actualiza
bprov_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.24** Descripción de la tabla Provincia

<b>TABLA: cantón</b>		
<b>Clave Primaria:</b>	bcant_codigo	
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
bcant_codigo	varchar(20)	Campo que almacena la clave primaria
bprov_codigo	varchar(20)	Campo que almacena el código de la provincia a la que pertenece
bcant_nombre	varchar(50)	Nombre del cantón.
bcant_descripcion	text	Descripción detallada referente al cantón
bcant_estado	bool	Estado del registro: True: Activo False: Inactivo
bcant_usuario_insert	varchar(50)	Usuario que inserta
bcant_fecha_insert	timestamp with time	Fecha que inserta
bcant_usuario_update	varchar(50)	Usuario que actualiza
bcant_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.25** Descripción de la tabla cantón

<b>TABLA: parroquia</b>		
<b>Clave Primaria:</b>	bparr_codigo	
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
bparr_codigo	varchar(20)	Campo que almacena la clave primaria
bcant_codigo	varchar(20)	Campo que almacena el código del cantón al que pertenece
bparr_nombre	varchar(50)	Nombre de la parroquia
bparr_descripcion	text	Descripción detallada referente a la parroquia
bparr_estado	bool	Estado del registro: True: Activo False: Inactivo
bparr_usuario_insert	varchar(50)	Usuario que inserta
bparr_fecha_insert	timestamp with time	Fecha que inserta
bparr_usuario_update	varchar(50)	Usuario que actualiza
bparr_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.26** Descripción de la tabla Parroquia

**TABLA: agencia**

<b>Clave Primaria:</b>		
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
bagen_codigo	varchar(20)	Campo que almacena la clave primaria
bparr_codigo	varchar(20)	Código de la parroquia en la que esta ubicada la agencia
bagen_nombre	varchar(50)	Nombre de la parroquia
bagen_descripcion	text	Descripción detallada referente a la parroquia
bagen_direccion	varchar(50)	Dirección en donde esta ubicada la agencia
bagen_numero_sucursal	int2	Numero de agencia
bagen_estado_agencia	bool	Estado de la Agencia True: Activa False: Fuera de servicio
bagen_estado	bool	Estado del registro: True: Activo False: Inactivo
bagen_usuario_insert	varchar(50)	Usuario que inserta
bagen_fecha_insert	timestamp with time	Fecha que inserta
bagen_usuario_update	varchar(50)	Usuario que actualiza
bagen_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.27** Descripción de la tabla Agencia**TABLA: departamento**

<b>Clave Primaria:</b>		
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
bdepa_codigo	varchar(20)	Campo que almacena la clave primaria
bagen_codigo	varchar(20)	Código de la agencia a la que pertenece el departamento
bdepa_nombre	varchar(30)	Nombre del departamento
bdepa_descripcion	varchar(200)	Descripción detallada referente al departamento
bdepa_ubicacion	varchar(20)	Ubicación en donde esta ubicado el departamento dentro de la agencia.
bdepa_estado	bool	Estado del registro: True: Activo False: Inactivo
bdepa_usuario_insert	varchar(50)	Usuario que inserta
bdepa_fecha_insert	timestamp with time	Fecha que inserta

	time	
bdepa_usuario_update	varchar(50)	Usuario que actualiza
bdepa_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.28** Descripción de la tabla departamento

<b>TABLA: empleado</b>		
<b>Clave Primaria:</b>	bempl_codigo	
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
bempl_codigo	varchar(20)	Campo que almacena la clave primaria
bdepa_codigo	varchar(20)	Código del departamento al que pertenece
bparr_codigo	varchar(20)	Código de la parroquia
btdid_codigo	varchar(20)	Código del tipo de documento de identificación
bagen_codigo	varchar(20)	Código de la agencia a la que pertenece
bempl_documento_identificacion	varchar(13)	Numero de documento de identificación
bempl_nombre	varchar(30)	Nombre del empleado
bempl_apellido_paterno	varchar(50)	Apellido paterno del empleado
bempl_apellido_materno	varcahr(50)	Apellido materno del empleado
bempl_sexo	int2	Sexo del empleado 0 – masculino 1 – femenino
bempl_direccion	varchar(100)	Dirección específica del empleado
bempl_foto	bytea	Fotografía del empleado
bempl_nombre_foto	varchar(100)	Nombre de la fotografía
bempl_tipo	varchar(10)	Extensión del archivo de la fotografía
bempl_fecha_nacimiento	date	Fecha de nacimiento del empleado
bempl_fecha_ingreso	date	Fecha en la que el empleado ingreso a la empresa
bempl_estado_empleado	bool	Estado del empleado True: Activo False: Fuera de servicio
bempl_estado	bool	Estado del registro: True: Activo False: Inactivo
bempl_usuario_insert	varchar(50)	Usuario que inserta



bempl_fecha_insert	timestamp with time	Fecha que inserta
bempl_usuario_update	varchar(50)	Usuario que actualiza
bempl_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.29** Descripción de la tabla empleado

<b>TABLA: cliente</b>		
<b>Clave Primaria:</b>	<b>bclie_codigo</b>	
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
bclie_codigo	varchar(20)	Campo que almacena la clave primaria
btcli_codigo	varchar(20)	Código del tipo de cliente.
bempl_codigo	varchar(20)	Código del empleado que esta a cargo de tal empleado.
btdid_codigo	varchar(20)	Código del tipo de documento de identificación
bparr_codigo	varchar(20)	Código de la parroquia a la que pertenece el cliente.
bplcd_codigo	varchar(20)	Código del la cuenta Cliente que se refleja en el módulo de contabilidad
bclie_tipo_contri	int2	Especifica el tipo de contribuyente. 0 – Personas Naturales 1 – Personas Jurídicas (Empresas)
bclie_numero	varchar(20)	Número de cliente
bclie_documento_identificacion	varchar(13)	Numero de documento de identificación
bclie_nombre	varchar(30)	Nombre del cliente
bclie_apellido_paterno	varchar(50)	Apellido paterno del cliente
bclie_apellido_materno	varcahr(50)	Apellido materno del cliente
bclie_direccion	varchar(100)	Dirección específica del cliente
bclie_empresa	varchar(50)	Nombre de la empresa
bclie_descripcion	text	Descripción del cliente
bclie_credito_maximo	numeric	Crédito máximo que se le puede conceder para una venta a crédito.

bclie_saldo	date	Saldo del cliente.
bclie_estado_cliente	bool	Estado del cliente True: Activo False: Fuera de servicio
bclie_estado	bool	Estado del registro: True: Activo False: Inactivo
bclie_usuario_insert	varchar(50)	Usuario que inserta
bclie_fecha_insert	timestamp with time	Fecha que inserta
bclie_usuario_update	varchar(50)	Usuario que actualiza
bclie_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.30** Descripción de la tabla Cliente

**TABLA: tipo\_cuota**

<b>Clave Primaria:</b>	btcuo_codigo	
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
btcuo_codigo	varchar(20)	Campo que almacena la clave primaria
btcuo_tipo_cuota	varchar(50)	Nombre del tipo de cuota.
btcuo_tiempo	numeric	Número de días
btcuo_descripcion	text	Descripción detallada referente al tipo de cuota
btcuo_estado	bool	Estado del registro: True: Activo False: Inactivo
btcuo_usuario_insert	varchar(50)	Usuario que inserta
btcuo_fecha_insert	timestamp with time	Fecha que inserta
btcuo_usuario_update	varchar(50)	Usuario que actualiza
btcuo_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.31** Descripción de la tabla Tipo Cuota

**TABLA: credito**

<b>Clave Primaria:</b>	bcred_codigo	
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
bcred_codigo	varchar(20)	Campo que almacena la clave primaria
bclie_codigo	varchar(20)	Código del cliente que tiene el crédito.
btcuo_codigo	varchar(20)	Código del tipo de cuota
bpre_codigo	varchar(20)	Código del plazo de crédito
bvent_codigo	varchar(20)	Código de la venta a

		crédito.
bcred_fecha_inicio	date	Fecha de inicio del crédito
bcred_fecha_fin	date	Fecha de finalización del crédito
bcred_valor_total	numeric	Valor total del crédito
bcred_descripcion	text	Descripción sobre el crédito.
bcred_saldo	numeric	Saldo pendiente del crédito
bcred_estado_credito	int2	Estado del crédito: 0 – Pendiente 1 – Finalizado 2 – Anulado
bcred_estado	bool	Estado del registro: True: Activo False: Inactivo
bcred_usuario_insert	varchar(50)	Usuario que inserta
bcred_fecha_insert	timestamp with time	Fecha que inserta
bcred_usuario_update	varchar(50)	Usuario que actualiza
bcred_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.32** Descripción de la tabla Crédito

**TABLA: cuota**

<b>Clave Primaria:</b>	bcuot_codigo	
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
bcuot_codigo	varchar(20)	Campo que almacena la clave primaria
bcred_codigo	varchar(20)	Código del crédito
bcuot_fecha_pago	date	Fecha en la que debe cancelar
bcuot_monto	numeric	Monto del valor a cancelar
bcuot_descripcion	text	Descripción sobre la cuota
bcuot_estado_cuota	int2	Estado de la cuota: 0 – Anulada 1 – Pendiente 2 – Finalizada
bcuot_estado	bool	Estado del registro: True: Activo False: Inactivo
bcuot_usuario_insert	varchar(50)	Usuario que inserta
bcuot_fecha_insert	timestamp with time	Fecha que inserta
bcuot_usuario_update	varchar(50)	Usuario que actualiza
bcuot_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.33** Descripción de la tabla Cuota

<b>TABLA: cobro</b>		
<b>Clave Primaria:</b>	bcobr_codigo	
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
bcobr_codigo	varchar(20)	Campo que almacena la clave primaria
bcred_codigo	varchar(20)	Código del crédito
bagen_ccodigo	varchar(20)	Código de la agencia en la que esta cobrando
bcobr_numero_comprobante	varchar(20)	Número de comprobante
bcobr_fecha	date	Fecha en la que se cobro
bcobr_descripcion	Text	Descripción del cobro
bcobr_monto	numeric	Monto a cancelar cancelado
bcobr_porcentaje_mora	numeric	Porcentaje de Mora
bcobr_monto_mora	numeric	Monto por mora
bcobr_monto_total	numeric	Monto total a cancelar
bcobr_estado_cobro	int2	Estado del cobro: 0 – Emitido 1 – Pendiente 2 – Anulado
bcobr_estado	bool	Estado del registro: True: Activo False: Inactivo
bcobr_usuario_insert	varchar(50)	Usuario que inserta
bcobr_fecha_insert	timestamp with time	Fecha que inserta
bcobr_usuario_update	varchar(50)	Usuario que actualiza
bcobr_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.34** Descripción de la tabla Cobro

<b>TABLA: usuario</b>		
<b>Clave Primaria:</b>	busua_codigo	
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
busua_codigo	varchar(20)	Campo que almacena la clave primaria
bempl_codigo	varchar(20)	Código del empleado
busua_alias	varchar(20)	Alias del usuario del sistema.
busua_clave	text	Clave del usuario
busua_descripcion	varchar(200)	Descripción del usuario
busua_estado_usuario	bool	Estado del usuario: True – Activo False – No esta disponible de acceder al sistema
busua_estado	bool	Estado del registro: True: Activo

		False: Inactivo
busua_usuario_insert	varchar(50)	Usuario que inserta
busua_fecha_insert	timestamp with time	Fecha que inserta
busua_usuario_update	varchar(50)	Usuario que actualiza
busua_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.35** Descripción de la tabla Usuario

<b>TABLA: rol</b>		
<b>Clave Primaria:</b>	brole_codigo	
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
brole_codigo	varchar(20)	Campo que almacena la clave primaria
brole_nombre	varchar(100)	Nombre del rol.
brole_estado	bool	Estado del registro: True: Activo False: Inactivo
brole_usuario_insert	varchar(50)	Usuario que inserta
brole_fecha_insert	timestamp with time	Fecha que inserta
brole_usuario_update	varchar(50)	Usuario que actualiza
brole_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.36** Descripción de la tabla Rol

<b>TABLA: perfil</b>		
<b>Clave Primaria:</b>	burol_codigo	
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
burol_codigo	varchar(20)	Campo que almacena la clave primaria
brole_codigo	varchar(20)	Código del rol
busua_codigo	varchar(20)	Código del usuario
burol_estado	bool	Estado del registro: True: Activo False: Inactivo
burol_usuario_insert	varchar(50)	Usuario que inserta
burol_fecha_insert	timestamp with time	Fecha que inserta
burol_usuario_update	varchar(50)	Usuario que actualiza
burol_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.37** Descripción de la tabla perfil

<b>TABLA: sesion</b>		
<b>Clave Primaria:</b>	bsesi_codigo	
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
bsesi_codigo	varchar(20)	Campo que almacena la clave primaria
busua_codigo	varchar(20)	Código del usuario
bsesi_fecha_inicio	date	Fecha que se inicia la sesión
bsesi_fecha_fin	date	Fecha que termina la sesión
bsesi_id	text	Id de la sesión
bsesi_ip	text	Dirección IP de la que el usuario se ha conectado
bsesi_zona_tiempo	varchar(3)	Zona de la que se conecta
bsesi_activo	bool	Estado de la sesión: True – Activa False - Cerrada
bsesi_contexto	varchar(50)	Contexto de la aplicación a la que se conecto

**Tabla 6.38** Descripción de la tabla Sesión

<b>TABLA: objeto</b>		
<b>Clave Primaria:</b>	bobje_codigo	
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
bobje_codigo	varchar(20)	Campo que almacena la clave primaria
bobje_nombre	varchar(100)	Nombre del objeto
bobje_ruta	text	Establece la ruta del objeto
bobje_descripcion	text	Descripción del objeto
bobje_tipo	int2	Tipo de objeto: 0 – Interface visual 1 – Servicios Web
bobje_estado	bool	Estado del registro: True: Activo False: Inactivo
bobje_usuario_insert	varchar(50)	Usuario que inserta
bobje_fecha_insert	timestamp with time	Fecha que inserta
bobje_usuario_update	varchar(50)	Usuario que actualiza
bobje_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.39** Descripción de la tabla Objeto

TABLA: objeto-rol		
Clave Primaria:	bobro_codigo	
CAMPO	TIPO	DESCRIPCION
bobro_codigo	varchar(20)	Campo que almacena la clave primaria
brole_codigo	varchar(20)	Código del rol
bobje_codigo	varchar(20)	Código del objeto
bobro_estado	bool	Estado del registro: True: Activo False: Inactivo
bobro_usuario_insert	varchar(50)	Usuario que inserta
bobro_fecha_insert	timestamp with time	Fecha que inserta
bobro_usuario_update	varchar(50)	Usuario que actualiza
bobro_fecha_update	timestamp with time	Fecha que actualiza

**Tabla 6.40** Descripción de la tabla Objeto rol.

### 6.9.2.3 Script de la Base de Datos

```

COMMENT ON TABLE SCH_BSX.BSX_CONFIGURACION is
'TABLA: BSX_CONFIGURACION
ALMACENA DATOS SOBRE LA CONFIGURACION QUE PUEDE TENER EL SISTEMA,
ASI COMO PARAMETROS';
Create Table SCH_BSX.BSX_CONFIGURACION (
  BCONF_CODIGO    VARCHAR(20)    not null,
  BCONF_NOMBRE    VARCHAR(200)   null,
  BCONF_VALOR_ENTERO NUMERIC      null,
  BCONF_VALOR_CADENA VARCHAR(100) null,
  BCONF_DESCRIPCION VARCHAR(200) null,
  BCONF_ESTADO    BOOL           null,
  BCONF_USUARIO_INSERT VARCHAR(50) null,
  BCONF_FECHA_INSERT TIMESTAMP WITH TIME ZONE null,
  BCONF_USUARIO_UPDATE VARCHAR(50) null,
  BCONF_FECHA_UPDATE  TIMESTAMP WITH TIME ZONE null,
  constraint PK_BSX_CONFIGURACION primary key (BCONF_CODIGO)
);

COMMENT ON TABLE SCH_BSX.BSX_PROVINCIA IS
'TABLA: BSX_PROVINCIA
ALMACENA DATOS SOBRE LAS DIFERENTES PROVINCIAS QUE TIENE EL PAIS';
CREATE TABLE SCH_BSX.BSX_PROVINCIA (
  BPROV_CODIGO    VARCHAR(20)    not null,
  BPROV_NOMBRE    VARCHAR(50)    null,
  BPROV_DESCRIPCION TEXT         null,
  BPROV_ESTADO    BOOL           null,
  BPROV_USUARIO_INSERT VARCHAR(50) null,
  BPROV_FECHA_INSERT TIMESTAMP WITH TIME ZONE null,
  BPROV_USUARIO_UPDATE VARCHAR(50) null,
  BPROV_FECHA_UPDATE  TIMESTAMP WITH TIME ZONE null,
  CONSTRAINT PK_BSX_PROVINCIA primary key (BPROV_CODIGO)
);

```

COMMENT ON TABLE SCH\_BSX.BSX\_CANTON IS  
'TABLA: BSX\_CANTON  
ALMACENA DATOS SOBRE LAS DIFERENTES CANTONES QUE TIENE UNA  
PROVINCIA';

```
CREATE TABLE SCH_BSX.BSX_CANTON (  
  BCANT_CODIGO   VARCHAR(20)   not null,  
  BPROV_CODIGO  VARCHAR(20)   null,  
  BCANT_NOMBRE  VARCHAR(50)   null,  
  BCANT_DESCRIPCION TEXT       null,  
  BCANT_ESTADO  BOOL          null,  
  BCANT_USUARIO_INSERT VARCHAR(50) null,  
  BCANT_FECHA_INSERT  TIMESTAMP WITH TIME ZONE null,  
  BCANT_USUARIO_UPDATE VARCHAR(50) null,  
  BCANT_FECHA_UPDATE  TIMESTAMP WITH TIME ZONE null,  
  CONSTRAINT PK_BSX_CANTON primary key (BCANT_CODIGO)  
);
```

COMMENT ON TABLE SCH\_BSX.BSX\_PARROQUIA IS  
'TABLA: BSX\_PARROQUIA  
ALMACENA DATOS SOBRE LAS DIFERENTES PARROQUIAS QUE TIENE UN  
CANTON';

```
CREATE TABLE SCH_BSX.BSX_PARROQUIA (  
  BPARR_CODIGO   VARCHAR(20)   not null,  
  BCANT_CODIGO  VARCHAR(20)   null,  
  BPARR_NOMBRE  VARCHAR(50)   null,  
  BPARR_DESCRIPCION TEXT       null,  
  BPARR_ESTADO  BOOL          null,  
  BPARR_USUARIO_INSERT VARCHAR(50) null,  
  BPARR_FECHA_INSERT  TIMESTAMP WITH TIME ZONE null,  
  BPARR_USUARIO_UPDATE VARCHAR(50) null,  
  BPARR_FECHA_UPDATE  TIMESTAMP WITH TIME ZONE null,  
  CONSTRAINT PK_BSX_PARROQUIA primary key (BPARR_CODIGO)  
);
```

COMMENT ON TABLE SCH\_BSX.BSX\_AGENCIA IS  
'TABLA: BSX\_AGENCIA  
PERMITE ALMACENAR LOS DATOS GENERALES DE LA AGENCIA';

```
CREATE TABLE SCH_BSX.BSX_AGENCIA (  
  BAGEN_CODIGO   VARCHAR(20)   not null,  
  BPARR_CODIGO  VARCHAR(20)   null,  
  BAGEN_NOMBRE  VARCHAR(50)   null,  
  BAGEN_DESCRIPCION VARCHAR(200) null,  
  BAGEN_DIRECCION VARCHAR(50)   null,  
  BAGEN_NUMERO_SUCURSAL INT2     null,  
  BAGEN_ESTADO_AGENCIA BOOL      null,  
  BAGEN_ESTADO  BOOL          null,  
  BAGEN_USUARIO_INSERT VARCHAR(50) null,  
  BAGEN_FECHA_INSERT  TIMESTAMP WITH TIME ZONE null,  
  BAGEN_USUARIO_UPDATE VARCHAR(50) null,  
  BAGEN_FECHA_UPDATE  TIMESTAMP WITH TIME ZONE null,  
  CONSTRAINT PK_BSX_AGENCIA primary key (BAGEN_CODIGO)  
);
```

COMMENT ON TABLE SCH\_BSX.BSX\_DEPARTAMENTO IS  
'TABLA: BSX\_DEPARTAMENTO  
PERMITE ALMACENAR LOS DATOS GENERALES DEL DEPARTAMENTO QUE  
PERTENECEN A DICHA AGENCIA';



```

CREATE TABLE SCH_BSX.BSX_DEPARTAMENTO (
  BDEPA_CODIGO VARCHAR(20) not null,
  BAGEN_CODIGO VARCHAR(20) null,
  BDEPA_NOMBRE VARCHAR(30) null,
  BDEPA_DESCRIPCION VARCHAR(200) null,
  BDEPA_UBICACION VARCHAR(20) null,
  BDEPA_ESTADO BOOL null,
  BDEPA_USUARIO_INSERT VARCHAR(50) null,
  BDEPA_FECHA_INSERT TIMESTAMP WITH TIME ZONE null,
  BDEPA_USUARIO_UPDATE VARCHAR(50) null,
  BDEPA_FECHA_UPDATE TIMESTAMP WITH TIME ZONE null,
  CONSTRAINT PK_BSX_DEPARTAMENTO primary key (BDEPA_CODIGO)
);

```

COMMENT ON TABLE SCH\_BSX.BSX\_EMPLEADO IS  
'TABLA: BSX\_EMPLEADO  
PERMITE ALMACENAR LOS DATOS GENERALES DE LOS EMPLEADOS';

```

CREATE TABLE SCH_BSX.BSX_EMPLEADO (
  BEMPL_CODIGO VARCHAR(20) not null,
  BDEPA_CODIGO VARCHAR(20) null,
  BPARR_CODIGO VARCHAR(20) null,
  BTDID_CODIGO VARCHAR(20) null,
  BAGEN_CODIGO VARCHAR(20) null,
  BEMPL_DOCUMENTO_IDENTIFICACION VARCHAR(13) null,
  BEMPL_NOMBRE VARCHAR(50) null,
  BEMPL_APELLIDO_PATERNO VARCHAR(50) null,
  BEMPL_APELLIDO_MATERNO VARCHAR(50) null,
  BEMPL_SEXO INT2 null,
  BEMPL_DIRECCION VARCHAR(100) null,
  BEMPL_FOTO BYTEA null,
  BEMPL_NOMBRE_FOTO VARCHAR(100) null,
  BEMPL_TIPO VARCHAR(10) null,
  BEMPL_FECHA_NACIMIENTO DATE null,
  BEMPL_FECHA_INGRESO DATE null,
  BEMPL_ESTADO_EMPLEADO BOOL null,
  BEMPL_ESTADO BOOL null,
  BEMPL_USUARIO_INSERT VARCHAR(50) null,
  BEMPL_FECHA_INSERT TIMESTAMP WITH TIME ZONE null,
  BEMPL_USUARIO_UPDATE VARCHAR(50) null,
  BEMPL_FECHA_UPDATE TIMESTAMP WITH TIME ZONE null,
  CONSTRAINT PK_BSX_EMPLEADO primary key (BEMPL_CODIGO)
);

```

COMMENT ON TABLE SCH\_BSX.BSX\_CLIENTE IS  
'TABLA: BSX\_CLIENTE  
ALMACENA LOS DATOS PRINCIPALES DE LA EMPRESA CLIENTE O LOS DATOS DE  
CLIENTES NATURALES ';

```

CREATE TABLE SCH_BSX.BSX_CLIENTE (
  BCLIE_CODIGO VARCHAR(20) not null,
  BTCLI_CODIGO VARCHAR(20) null,
  BEMPL_CODIGO VARCHAR(20) null,
  BTDID_CODIGO VARCHAR(20) null,
  BPARR_CODIGO VARCHAR(20) null,
  BPLCD_CODIGO VARCHAR(20) null,
  BCLIE_TIPO_CONTRI INT2 null,
  BCLIE_NUMERO VARCHAR(20) null,
  BCLIE_DOCUMENTO_IDENTIFICACION VARCHAR(13) null,
  BCLIE_NOMBRE VARCHAR(50) null,

```

```

BCLIE_APELLIDO_MATERNO VARCHAR(50) null,
BCLIE_APELLIDO_PATERNO VARCHAR(50) null,
BCLIE_DIRECCION VARCHAR(100) null,
BCLIE_EMPRESA VARCHAR(50) null,
BCLIE_DESCRIPCION TEXT null,
BCLIE_CREDITO_MAXIMO NUMERIC null,
BCLIE_SALDO NUMERIC null,
BCLIE_ESTADO_CLIENTE BOOL null,
BCLIE_ESTADO BOOL null,
BCLIE_USUARIO_INSERT VARCHAR(50) null,
BCLIE_FECHA_INSERT TIMESTAMP WITH TIME ZONE null,
BCLIE_USUARIO_UPDATE VARCHAR(50) null,
BCLIE_FECHA_UPDATE TIMESTAMP WITH TIME ZONE null,
CONSTRAINT PK_BSX_CLIENTE primary key (BCLIE_CODIGO)
);

```

```

COMMENT ON TABLE SCH_BSX.BSX_TIPO_CUOTA is
'TABLA: BSX_TIPO_CUOTA
ALAMCENA LOS TIPOS DE CUOTAS QUE PUEDE TENER UN CREDITO COMO PUEDEN
SER SEMANALES, QUINCENALES, MENSUALES, ETC';
CREATE TABLE SCH_BSX.BSX_TIPO_CUOTA (
BTCUO_CODIGO VARCHAR(20) not null,
BTCUO_TIPO_CUOTA VARCHAR(50) null,
BTCUO_TIEMPO NUMERIC null,
BTCUO_DESCRIPCION TEXT null,
BTCUO_ESTADO BOOL null,
BTCUO_USUARIO_INSERT VARCHAR(50) null,
BTCUO_FECHA_INSERT TIMESTAMP WITH TIME ZONE null,
BTCUO_USUARIO_UPDATE VARCHAR(50) null,
BTCUO_FECHA_UPDATE TIMESTAMP WITH TIME ZONE null,
CONSTRAINT PK_BSX_TIPO_CUOTA primary key (BTCUO_CODIGO)
);

```

```

COMMENT ON TABLE SCH_BSX.BSX_CREDITO IS
'TABLA: BSX_CREDITO
ALMACENA DATOS SOBRE CREDITOS GENERADOS POR UNA VENTA A CREDITO
PERTENECIENTE A LOS CLIENTES';
CREATE TABLE SCH_BSX.BSX_CREDITO (
BCRED_CODIGO VARCHAR(20) not null,
BCLIE_CODIGO VARCHAR(20) null,
BTCUO_CODIGO VARCHAR(20) null,
BPCRE_CODIGO VARCHAR(20) null,
BVENT_CODIGO VARCHAR(20) null,
BCRED_FECHA_INICIO DATE null,
BCRED_FECHA_FIN DATE null,
BCRED_VALOR_TOTAL NUMERIC null,
BCRED_DESCRIPCION TEXT null,
BCRED_SALDO NUMERIC null,
BCRED_ESTADO_CREDITO INT2 null,
BCRED_ESTADO BOOL null,
BCRED_USUARIO_INSERT VARCHAR(50) null,
BCRED_FECHA_INSERT TIMESTAMP WITH TIME ZONE null,
BCRED_USUARIO_UPDATE VARCHAR(50) null,
BCRED_FECHA_UPDATE TIMESTAMP WITH TIME ZONE null,
CONSTRAINT PK_BSX_CREDITO primary key (BCRED_CODIGO)
);

```

```

COMMENT ON TABLE SCH_BSX.BSX_CUOTA is

```

TABLA: BSX\_CUOTA  
 ALMACENA DATOS SOBRE LAS CUOTAS QUE TIENE LA VENTA A CREDITO QUE REALIZA EL CLIENTES';

```
CREATE TABLE SCH_BSX.BSX_CUOTA (
  BCUOT_CODIGO VARCHAR(20) not null,
  BCRED_CODIGO VARCHAR(20) null,
  BCUOT_FECHA_PAGO DATE null,
  BCUOT_MONTO NUMERIC null,
  BCUOT_DESCRIPCION TEXT null,
  BCUOT_ESTADO_CUOTA INT2 null,
  BCUOT_ESTADO BOOL null,
  BCUOT_USUARIO_INSERT VARCHAR(50) null,
  BCUOT_FECHA_INSERT TIMESTAMP WITH TIME ZONE null,
  BCUOT_USUARIO_UPDATE VARCHAR(50) null,
  BCUOT_FECHA_UPDATE TIMESTAMP WITH TIME ZONE null,
  CONSTRAINT PK_BSX_CUOTA primary key (BCUOT_CODIGO)
);
```

COMMENT ON TABLE SCH\_BSX.BSX\_COBRO IS

TABLA: BSX\_COBRO  
 ALMACENA DATOS SOBRE LOS PAGOS QUE REALIZA EL CLIENTE';

```
CREATE TABLE SCH_BSX.BSX_COBRO (
  BCOBR_CODIGO VARCHAR(20) not null,
  BCRED_CODIGO VARCHAR(20) null,
  BAGEN_CODIGO VARCHAR(20) null,
  BCOBR_NUMERO_COMPROBANTE VARCHAR(20) null,
  BCOBR_FECHA TIMESTAMP WITH TIME ZONE null,
  BCOBR_DESCRIPCION TEXT null,
  BCOBR_MONTO NUMERIC null,
  BCOBR_PORCENTAJE_MORA NUMERIC null,
  BCOBR_MONTO_MORA NUMERIC null,
  BCOBR_MONTO_TOTAL NUMERIC null,
  BCOBR_ESTADO_COBRO INT2 null,
  BCOBR_ESTADO BOOL null,
  BCOBR_USUARIO_INSERT VARCHAR(50) null,
  BCOBR_FECHA_INSERT TIMESTAMP WITH TIME ZONE null,
  BCOBR_USUARIO_UPDATE VARCHAR(50) null,
  BCOBR_FECHA_UPDATE TIMESTAMP WITH TIME ZONE null,
  CONSTRAINT PK_BSX_COBRO primary key (BCOBR_CODIGO)
);
```

COMMENT ON TABLE SCH\_BSX.BSX\_USUARIO IS

TABLA: BSX\_USUARIO  
 ALMACENA DATOS PRINCIPALES SOBRE LOS USUARIOS QUE INGRESAN AL SISTEMA';

```
CREATE TABLE SCH_BSX.BSX_USUARIO (
  BUSUA_CODIGO VARCHAR(20) not null,
  BEMPL_CODIGO VARCHAR(20) null,
  BUSUA_ALIAS VARCHAR(20) null,
  BUSUA_CLAVE TEXT null,
  BUSUA_DESCRIPCION VARCHAR(200) null,
  BUSUA_ESTADO_USUARIO BOOL null,
  BUSUA_ESTADO BOOL null,
  BUSUA_USUARIO_INSERT VARCHAR(50) null,
  BUSUA_FECHA_INSERT TIMESTAMP WITH TIME ZONE null,
  BUSUA_USUARIO_UPDATE VARCHAR(50) null,
  BUSUA_FECHA_UPDATE TIMESTAMP WITH TIME ZONE null,
  CONSTRAINT PK_BSX_USUARIO primary key (BUSUA_CODIGO)
);
```

);

COMMENT ON TABLE SCH\_BSX.BSX\_ROL IS

TABLA: ROL

ALMACENA INFORMACION SOBRE LOS ROLES QUE TIENE EL SISTEMA';

```
CREATE TABLE SCH_BSX.BSX_ROL (  
  BROLE_CODIGO    VARCHAR(20)    not null,  
  BROLE_NOMBRE    VARCHAR(100)    null,  
  BROLE_ESTADO    BOOL            null,  
  BROLE_USUARIO_INSERT VARCHAR(50)    null,  
  BROLE_FECHA_INSERT  TIMESTAMP WITH TIME ZONE null,  
  BROLE_USUARIO_UPDATE VARCHAR(50)    null,  
  BROLE_FECHA_UPDATE  TIMESTAMP WITH TIME ZONE null,  
  CONSTRAINT PK_BSX_ROL primary key (BROLE_CODIGO)
```

);

COMMENT ON TABLE SCH\_BSX.BSX\_USUARIO\_ROL IS

TABLA: BSX\_USUARIO\_ROL PERFILES

DESCRIPCION: ALMACENA DATOS SOBRE LOS PERFILES DE USUARIO SEGÚN EL ROL DE USUARIO';

```
CREATE TABLE SCH_BSX.BSX_USUARIO_ROL (  
  BUROL_CODIGO    VARCHAR(20)    not null,  
  BROLE_CODIGO    VARCHAR(20)    null,  
  BUSUA_CODIGO    VARCHAR(20)    null,  
  BUROL_ESTADO    BOOL            null,  
  BUROL_USUARIO_INSERT VARCHAR(50)    null,  
  BUROL_FECHA_INSERT  TIMESTAMP WITH TIME ZONE null,  
  BUROL_USUARIO_UPDATE VARCHAR(50)    null,  
  BUROL_FECHA_UPDATE  TIMESTAMP WITH TIME ZONE null,  
  CONSTRAINT PK_BSX_USUARIO_ROL primary key (BUROL_CODIGO)
```

);

COMMENT ON TABLE SCH\_BSX.BSX\_SESION IS

TABLA: SESION

ALMACENA INFORMACION DE LAS CONEXIONES DEL USUARIO DEL SISTEMA';

```
CREATE TABLE SCH_BSX.BSX_SESION (  
  BSESI_CODIGO    VARCHAR(20)    not null,  
  BUSUA_CODIGO    VARCHAR(20)    null,  
  BSESI_FECHA_INICIO  TIMESTAMP WITH TIME ZONE null,  
  BSESI_FECHA_FIN    TIMESTAMP WITH TIME ZONE null,  
  BSESI_ID         TEXT            null,  
  BSESI_IP         TEXT            null,  
  BSESI_ZONA_TIEMPO  VARCHAR(3)    null,  
  BSESI_ACTIVIVO    BOOL            null,  
  BSESI_CONTEXTO    VARCHAR(50)    null,  
  CONSTRAINT PK_BSX_SESION primary key (BSESI_CODIGO)
```

);

COMMENT ON TABLE SCH\_BSX.BSX\_OBJETO IS

TABLA: OBJETO

DESCRIPCION: ALMACENA INFORMACION DE LOS OBJETOS QUE CONFORMAN LA APLICACIÓN ESTOS PUEDEN SER LOS BOTONES DE ACCION DE LOS FORMULARIOS, METODOS DE LOS SERVICIOS WEB';

```
CREATE TABLE SCH_BSX.BSX_OBJETO (  
  BOBJE_CODIGO    VARCHAR(20)    not null,  
  BOBJE_NOMBRE    VARCHAR(100)    null,  
  BOBJE_RUTA      TEXT            null,  
  BOBJE_DESCRIPCION TEXT            null,
```

```

BOBJE_TIPO      INT2      null,
BUSUA_ESTADO    BOOL      null,
BOBJE_USUARIO_INSERT VARCHAR(50) null,
BOBJE_FECHA_INSERT  TIMESTAMP WITH TIME ZONE null,
BOBJE_USUARIO_UPDATE VARCHAR(50) null,
BOBJE_FECHA_UPDATE  TIMESTAMP WITH TIME ZONE null,
CONSTRAINT PK_BSX_OBJETO primary key (BOBJE_CODIGO)
);

```

```

COMMENT ON TABLE SCH_BSX.BSX_OBJETO_ROL IS
'TABLA: BSX_OBJETO_ROL PRIVILEGIOS
ALMACENA LOS PRIVILEGIOS QUE TIENEN CADA UNO DE LOS USUARIOS SEGÚN
EL PERFIL DE USUARIO';

```

```

CREATE TABLE SCH_BSX.BSX_OBJETO_ROL (
  BOBRO_CODIGO    VARCHAR(20)    not null,
  BROLE_CODIGO    VARCHAR(20)    null,
  BOBJE_CODIGO    VARCHAR(20)    null,
  BUSUA_ESTADO    BOOL          null,
  BOBRO_USUARIO_INSERT VARCHAR(50) null,
  BOBRO_FECHA_INSERT  TIMESTAMP WITH TIME ZONE null,
  BOBRO_USUARIO_UPDATE VARCHAR(50) null,
  BOBRO_FECHA_UPDATE  TIMESTAMP WITH TIME ZONE null,
  CONSTRAINT PK_BSX_OBJETO_ROL primary key (BOBRO_CODIGO)
);

```

#### 6.9.2.4 Diseño de la Interfaz

Según los requerimientos obtenidos se ha diseñado una interfaz fácil y entendible para los usuarios. El sistema y por ende el Módulo de Clientes y seguridades esta diseñado para la Web.

##### 6.9.2.4.1 Diseño de Entradas y Salidas

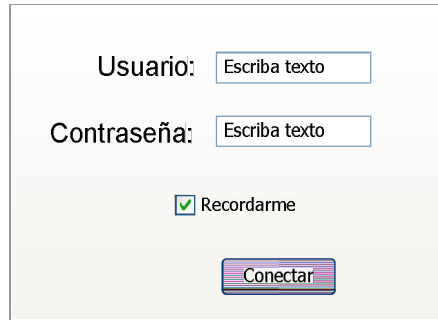
**FORMATO DEL MÓDULO DE CLIENTES Y SEGURIDADES CON LAS DIFERENTES SECCIONES DE MENÚ PRINCIPAL DENTRO DE UN SITIO WEB.**

Para el diseño del módulo de Clientes y seguridades se utilizaron prototipos previos a su desarrollo, para así tener una idea de cómo se va a visualizar los formularios en dicho módulo.

Las pantallas para la aplicación tienen una estructura similar a los formularios de aplicaciones de escritorio ya que están construidas en la web bajo la librería Ext JS las cuales tienen el siguiente formato:

**a) Pantalla de Validación de Usuario:**

En la siguiente pantalla el usuario ingresa el alias y su contraseña para poder acceder al Sistema. Adicionalmente posee la opción para poder almacenar los datos en una cookies.



Prototipo de la pantalla de validación de usuario. El formulario contiene:

- Etiqueta "Usuario:" seguida de un campo de texto con el placeholder "Escriba texto".
- Etiqueta "Contraseña:" seguida de un campo de texto con el placeholder "Escriba texto".
- Una casilla de verificación marcada con un checkmark y el texto "Recordarme".
- Un botón "Conectar" con un efecto de sombra y un borde.

**Figura 6.27** Prototipo de la pantalla de validación de usuario

**Ejemplo:**



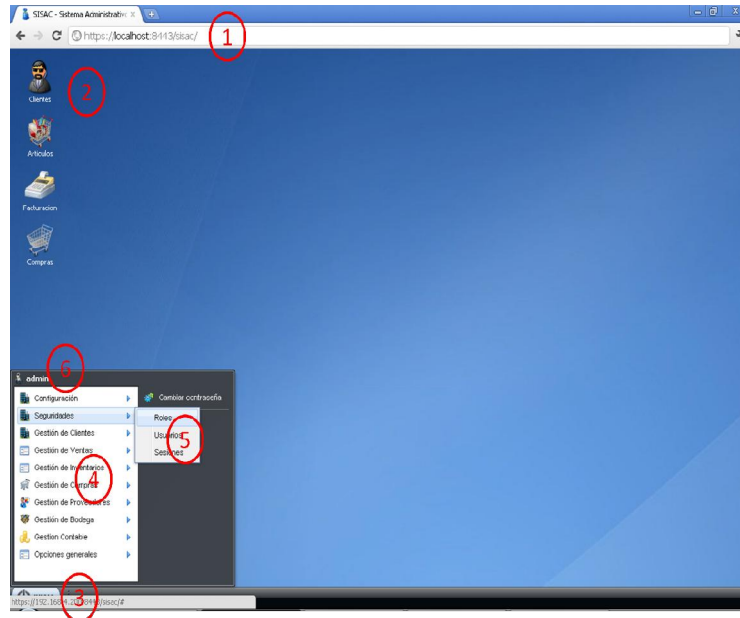
Ejemplo de la pantalla de validación de usuario. El formulario contiene:

- Etiqueta "Validación de usuarios" en la parte superior.
- Etiqueta "Usuario:" seguida de un campo de texto con el valor "admin".
- Etiqueta "Contraseña:" seguida de un campo de texto con caracteres ocultos por puntos ".....".
- Una casilla de verificación marcada con un checkmark y el texto "Recordarme".
- Un botón "Conectar" en la parte inferior derecha.

**Figura 6.28** Pantalla de validación de usuario

**a) Entorno de la Aplicación**

La interfaz de la pantalla principal del Sistema tiene cierta similitud con la interfaz del Sistema Operativo Windows XP por su menú, ventanas y su forma de interactuar con el usuario.



**Figura 6.29** Pantalla principal del sistema

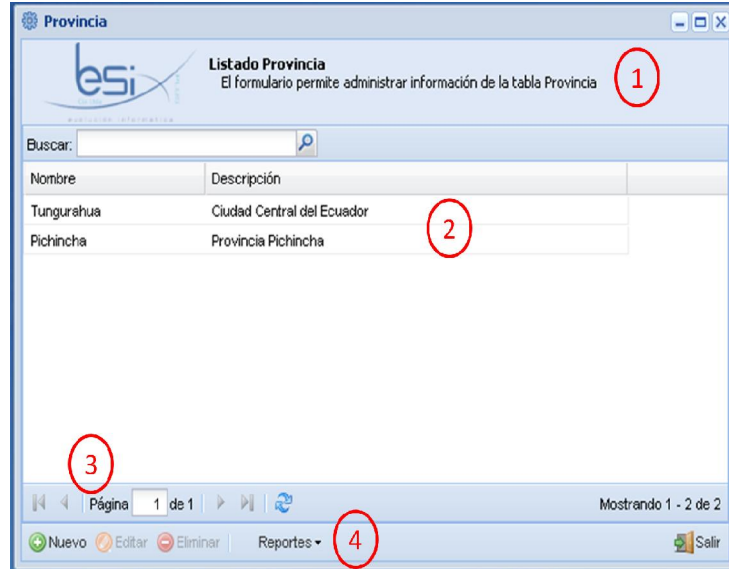
- 1.- URL de la aplicación.
- 2.- Accesos directos a los módulos principales del Sistema.
- 3.- Menú Inicio y barra de tareas.
- 4.- Items principales del Sistema.
- 5.- Sub Items.
- 6.- Usuario que ha iniciado sesión.

#### b) Listados: Provincias

Los diferentes listados utilizan un formato similar presentando los campos más relevantes. Constando con un componente de búsqueda la cual filtrará la información por cada uno de los campos presentados en el listado.

**Figura 6.30** Prototipo de la pantalla del listado de Provincias

## Ejemplo:



**Figura 6.31** Pantalla del Listado de Provincias

Si el usuario en el listado figura 6.29 selecciono Nuevo, se presenta un nuevo formulario para el ingreso de la información. Entre las partes principales del formulario de listados tenemos:

1. Encabezado: En el cual se encuentra el logo de la empresa y una descripción sobre dicho formulario.
2. Datos que están almacenados en la base de Datos
3. Paginación: Este componente se habilita cuando existen mas de 30 datos.
4. Opciones del formulario.
  - a. Nuevo: Se abre el formulario de Ingreso de un nuevo registro.
  - b. Editar: Formulario con los datos del registro a editar.
  - c. Eliminar: Elimina lógicamente el registro seleccionado.
  - d. Reporte: En caso de que un formulario posea la opción de reporte.
  - e. Salir: Permite salir del listado.

### c) Formulario de Agencias

En este formulario se ingresarán las diferentes agencias que posee la empresa.



Logo

Datos Agencia

Nombre:

Parroquia:

Dirección:

Descripción:

Estado:  Activa

Contactos

Tipo	Contacto	Descripción

**Figura 6.32** Prototipo de la pantalla de ingreso de las Agencias

Ejemplo:

bsi Agencia  
El formulario permite administrar la información de la tabla Agencia

Nombre:

Parroquia:

Dirección:

Descripción:

Estado agencia:

Contactos

Tipo co...	Contacto	Descripción

**Figura 6.33** Pantalla de ingreso de las Agencias

En donde el formulario consta de las siguientes partes:

1. Logo y Descripción del formulario
2. Campos a ingresar.
3. Opciones:
  - a. Guardar: Guarda el registro
  - b. Cancelar: Cancela y los controles vuelven a su valor inicial.
  - c. Cerrar: Cierra el formulario.

#### d) Formulario Empleados

En este formulario se ingresara a todos los empleados que tenga la agencia.



Logo

**Datos Personales**

Tipo Doc: Cédula | Escriba texto

Nombre: Escriba texto

Ap. Paterno: Escriba texto | Ap. Materno: Escriba texto

Fech. Nac: Escriba texto | Sexo: Escriba texto

Parroquia: Escriba texto | Buscar

Dirección: Escriba texto

**Fotografía**

Foto: img

**Ubicación**

Departamento: Escriba texto | Buscar | Fech. Ingre: Escriba texto | Estado:

**Contactos**

Nuevo | Editar | Eliminar

Tipo	Contacto	Descripción

Guardar | Cancelar | Cerrar

Figura 6.34 Prototipo de la pantalla de ingreso de empleados

Ejemplo:



Empleado

bsi

Empleado  
El formulario permite administrar la información de la tabla Empleado

**Datos personales**

Tipo Documento: Seleccione un Tipo de |

Nombre: |

Apellido paterno: | Materno: |

Fecha nacimiento: 28/11/2011 | Sexo: Sexo

Parroquia: |

Dirección: |

**Fotografía**

Foto: |

**Ubicación**

Departamento: | Fecha ingreso: 28/11/2011 | Estado Emp:

**Contactos**

Nuevo | Editar | Eliminar

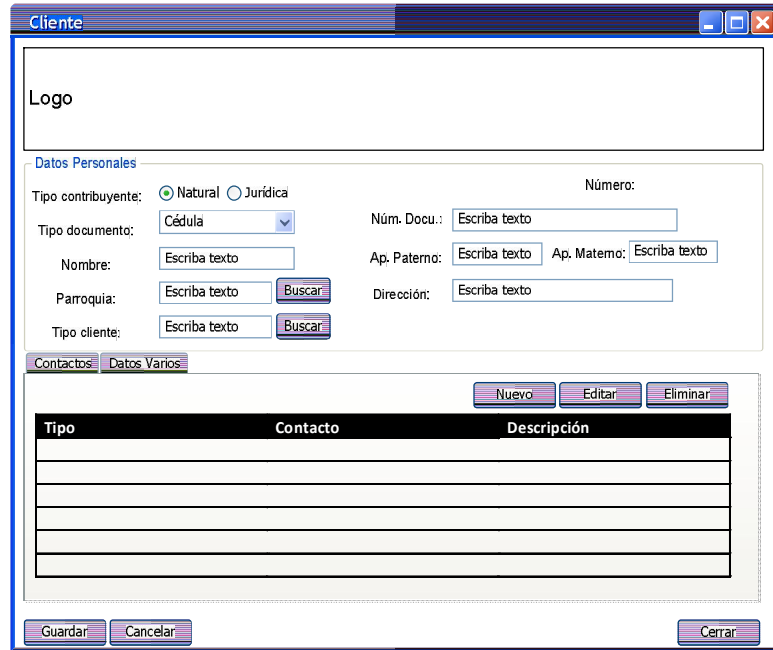
Tipo contacto	Contacto	Descripción

Guardar | Cancelar | Cerrar

Figura 6.35 Pantalla de ingreso de empleados.

### e) Formulario de Clientes

El siguiente formulario permite ingresar los datos respectivos del cliente con sus respectivos contactos que pueden ser: teléfono, celular, email, fax, etc.



Logo

**Datos Personales**

Tipo contribuyente:  Natural  Jurídica

Tipo documento: Cédula (dropdown)      Núm. Docu.: [Escriba texto]

Nombre: [Escriba texto]      Ap. Paterno: [Escriba texto]      Ap. Materno: [Escriba texto]

Parroquia: [Escriba texto] [Buscar]      Dirección: [Escriba texto]

Tipo cliente: [Escriba texto] [Buscar]

**Contactos**    Datos Varios

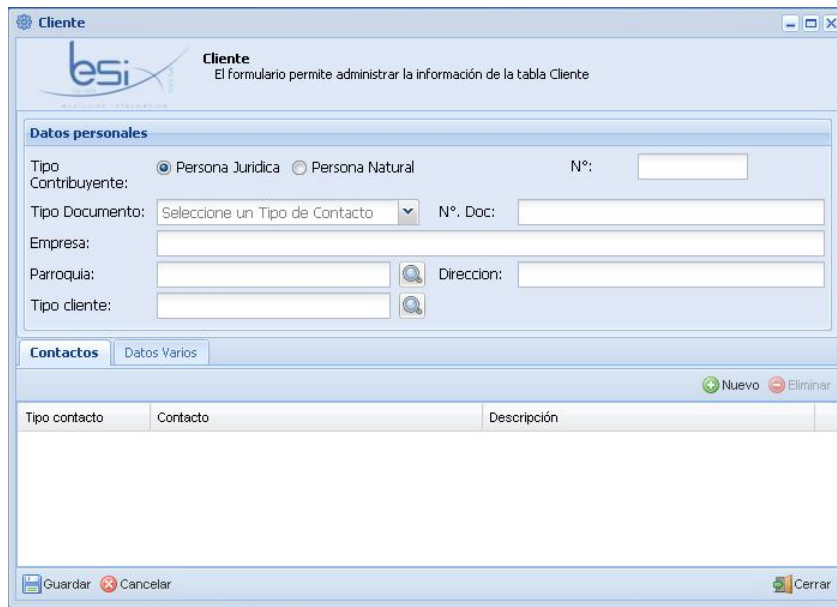
[Nuevo] [Editar] [Eliminar]

Tipo	Contacto	Descripción

[Guardar] [Cancelar] [Cerrar]

Figura 6.36 Prototipo de la pantalla ingreso de clientes

### Ejemplo:



Cliente

El formulario permite administrar la información de la tabla Cliente

**Datos personales**

Tipo Contribuyente:  Persona Jurídica  Persona Natural      N°: [ ]

Tipo Documento: Seleccione un Tipo de Contacto (dropdown)      N°. Doc: [ ]

Empresa: [ ]

Parroquia: [ ] [Buscar]      Dirección: [ ]

Tipo cliente: [ ] [Buscar]

**Contactos**    Datos Varios

[Nuevo] [Eliminar]

Tipo contacto	Contacto	Descripción

[Guardar] [Cancelar] [Cerrar]

Figura 6.37 Pantalla de ingreso de Clientes

En el sistema principal se podrá visualizar una interfaz similar a la plataforma Windows XP, la cual esta desarrollada bajo arquitectura SOA, lenguaje de programación JAVA para la capa del servidor, gestor de base de datos PostgreSQL, un servidor de aplicaciones JBOSS y la librería Ext JS para la capa del cliente.

En el menú principal podemos encontrar tres secciones importantes para lo que concierne al módulo de Gestión de Clientes y Seguridades:

- Configuración
- Seguridades
- Clientes

**Configuración:** Los diferentes formularios que están dentro de configuración permiten administrar información para es utilizada en todo el sistema, en la cual el único usuario para poder manipular esta información es el administrador. Entre las opciones principales de Configuración tenemos:

1. Distribución Geográfica
  - a. Provincia: Almacena todas las provincias.
  - b. Cantón: Almacena todos los cantones.
  - c. Parroquia: Almacena todos las parroquias.
2. Estructura empresarial
  - a. Agencia: Agencias de la empresa.
  - b. Departamento: Departamentos que tienen las agencias
  - c. Empleados: Empleados de las agencias
3. Clientes
  - a. Descuentos: Se encuentran todos los descuentos posibles que pueden aplicar a los clientes. Esto depende del tipo de cliente.
  - b. Tipo Cliente: Se encuentran los diferentes tipos de clientes. Esto se lo define de acuerdo a las circunstancias del administrador.
4. Tipo Contacto: Están definidos los diferentes tipos de contactos: entre los principales tenemos:
  - a. Teléfono Convencional.
  - b. Celular.
  - c. E-mail

- d. Fax, etc.
- 5. Documento de Identificación: Entre los diferentes tipos de documento tenemos:
  - a. Cédula.
  - b. Pasaporte.
  - c. Partida de Nacimiento, etc.

**Seguridades:** Dentro de las seguridades tenemos.

- 1. Usuarios: Permite administrar todas las cuentas de usuarios que están definidos en el sistema.
- 2. Roles: Administra los roles usuario.
- 3. Perfiles: Perfiles de usuario para dar permisos a los diferente módulos.
- 4. Objetos: Administra los objetos que se manipulan en el sistema, estos son:
  - a. Formularios.
  - b. Servicios Web, etc.
- 5. Sesiones: Permite administrar todas las sesiones abiertas del sistema, esto manipulara el administrador, en caso de que una sesión no autorizada este en uso el administrador podrá darle de baja a dicha sesión.

**Clientes:** Aquí se encuentran los formularios en donde se podrá manipular toda la información con respecto a clientes, entre los principales tenemos:

- 1. Clientes: Administra todos los clientes de la empresa en general.
- 2. Créditos por cobrar: Permite administrar todos los créditos que están vigentes por cobrar.
- 3. Créditos finalizados: Aquí se encuentran todos los créditos que ya han sido culminados.
- 4. Soportes a clientes: Administra todos los trabajos que se tiene con un cliente, tales como soporte técnico, cableados, entre otros.
- 5. Proyectos a clientes: Permite administrar los diferentes proyectos que tiene a cargo la empresa.

### 6.9.3 Implementación

El módulo de gestión de Clientes y seguridades se desarrollo bajo la arquitectura de SOA con el gestor de base de datos PostgreSQL, Java como lenguaje de programación en la parte del Servidor, y en el nivel de clientes se utilizó extjs, javaScript y ajax.

Adicionalmente se utilizo un servidor de aplicaciones JBOSS en el cual se ha definido un dominio de seguridades para nuestra aplicación.

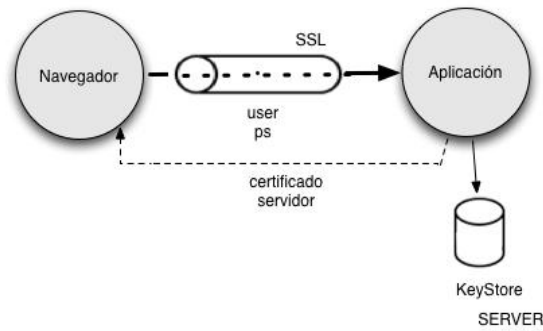
A continuación se presenta la creación del dominio de seguridades en el servidor de aplicaciones para así poder proporcionar confidencialidad en la comunicación.

Dentro de los archivos de configuración tenemos **login-config.xml** en el cual vamos a definir nuestro dominio de seguridades.

```
<!--DOMINIO DE SEGURIDAD PARA EL SAD PARA LA EMPRESA BESIXPLUS CIA LTDA -->

<application-policy name="JBossSISACWS">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.DatabaseServerLoginModule" flag="required">
      <module-option name="dsJndiName">java:jdbc/sisac</module-option>
      <module-option name="principalsQuery">SELECT BUSUA_CLAVE FROM
      SCH_BSX.BSX_USUARIO WHERE BUSUA_ESTADO AND
      BUSUA_ESTADO_USUARIO AND BUSUA_ALIAS = ?</module-option>
      <module-option name="rolesQuery">SELECT R.BROLE_NOMBRE, 'Roles' FROM
      SCH_BSX.BSX_ROL R INNER JOIN SCH_BSX.BSX_USUARIO_ROL UR ON
      UR.BROLE_CODIGO = R.BROLE_CODIGO AND UR.BUROL_ESTADO INNER
      JOIN SCH_BSX.BSX_USUARIO U ON U.BUSUA_CODIGO = UR.BUSUA_CODIGO
      WHERE U.BUSUA_ESTADO AND U.BUSUA_ESTADO_USUARIO AND
      U.BUSUA_ALIAS = ?</module-option>
      <module-option name="hashAlgorithm">MD5</module-option>
      <module-option name="hashEncoding">base64</module-option>
    </login-module>
  </authentication>
</application-policy>
```

Para que el servidor de aplicaciones JBoss sea capaz de gestionar este sistema de seguridad debe manejar almacenes de claves Java y certificados digitales.



**Figura 6.38** Almacén de claves y entradas asimétricas para el servidor de seguridades

**Creación del Almacén de claves:** Para crear un almacén de claves estándar JKS (Java KeyStore) y un par de claves asimétricas (pública/privada). Para ello utilizaremos la herramienta administrativa keytool:

```
[root@linux-emily emily]# keytool -genkey -alias sisac -keyalg RSA -keystore
./sisackey.keystore
Escriba la contraseña del almacén de claves:
Volver a escribir la contraseña nueva:
¿Cuáles son su nombre y su apellido?
[Unknown]: Damián Laura
¿Cuál es el nombre de su unidad de organización?
[Unknown]: UTA
¿Cuál es el nombre de su organización?
[Unknown]: UTA
¿Cuál es el nombre de su ciudad o localidad?
[Unknown]: Ambato
¿Cuál es el nombre de su estado o provincia?
[Unknown]: Tungurahua
¿Cuál es el código de país de dos letras de la unidad?
[Unknown]: ec
¿Es correcto CN=Damián Laura, OU=UTA, O=UTA, L=Ambato, ST=Tungurahua,
C=ec?
[no]: si
Escriba la contraseña clave para <sisac>
(INTRO si es la misma contraseña que la del almacén de claves):
```

En donde **sisac** va ser nuestra entrada de tipo RSA(Algoritmo de cifrado de clave pública).

También es necesario definir el certificado para poder identificar a nuestro sitio web como seguro, para lo cual ejecutamos:

```
[root@linux-emily emily]# keytool -certreq -keystore ./sisackey.keystore -alias sisac -file
sisacCsr.csr
```

A continuación vamos a definir el conector seguro para http, para lo cual creamos el conector en el siguiente archivo de configuración

``${JBOSS_HOME}/deploy/jbossweb.sar/server.xml`:`

```
<!-- SSL/TLS Connector configuration using the admin devl guide keystore-->
<Connector protocol="HTTP/1.1" SSLEnabled="true"
    port="8443" address="`${jboss.bind.address}`"
    scheme="https" secure="true" clientAuth="false"
    keystoreFile="`${jboss.server.home.dir}/conf/sisackey.keystore"
    keystorePass="sisac2011" sslProtocol = "TLS" />
```

En `keystoreFile` colocamos nuestro almacén de datos antes creado y en `keystorePass` colocamos la clave de nuestro almacén de datos

Hay que tener en cuenta que los archivos creado para nuestro dominio de seguridades hay q colocarlos dentro del directorio ``${JBOSS_HOME}/deploy/conf`.`

La configuración para nuestro dominio de seguridades garantizará la confidencialidad de nuestra información.



**Figura 6.39** Confidencialidad en la comunicación



Ahora solo nos queda consumir el dominio de seguridades desde nuestra aplicación, para lo cual en el archivo de configuración de nuestra aplicación llamamos a nuestro dominio ya creado. Los archivos de configuración son:

### **jboss-web.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss-web>
    <context-root>/sisac</context-root>
    <security-domain>java:jaas/JBossSISACWS</security-domain>
</jboss-web>
```

### **web.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.5"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
<display-name>SISAC_UI</display-name>
<listener>
<display-name>SessionListener</display-name>
<listener-class>com.besixplus.sisac.listeners.SessionListener</listener-class>
</listener>
<servlet>
<display-name>SessionManager</display-name>
<servlet-name>SessionManager</servlet-name>
<servlet-class>com.besixplus.sisac.servlets.SessionManager</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>SessionManager</servlet-name>
<url-pattern>/SessionManager</url-pattern>
</servlet-mapping>
<session-config>
<session-timeout>10</session-timeout>
</session-config>
<welcome-file-list>
<welcome-file>index.html</welcome-file>
<welcome-file>index.htm</welcome-file>
<welcome-file>index.jsp</welcome-file>
<welcome-file>default.html</welcome-file>
<welcome-file>default.htm</welcome-file>
<welcome-file>default.jsp</welcome-file>
</welcome-file-list>
<error-page>
<error-code>403</error-code>
<location>/login/error.html</location>
</error-page>
<security-constraint>
<display-name>SISAC_UI</display-name>
<web-resource-collection>
<web-resource-name>SAC</web-resource-name>
<url-pattern>/*</url-pattern>
<http-method>POST</http-method>
<http-method>GET</http-method>
```

```

</web-resource-collection>
<auth-constraint>
  <role-name>administrador</role-name>
  <role-name>usuario</role-name>
</auth-constraint>
<user-data-constraint>
  <transport-guarantee>CONFIDENTIAL</transport-guarantee>
</user-data-constraint>
</security-constraint>
<security-constraint>
  <display-name>Session</display-name>
  <web-resource-collection>
    <web-resource-name>SessionManager</web-resource-name>
    <url-pattern>/SessionManager</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
</security-constraint>
<login-config>
  <auth-method>FORM</auth-method>
  <form-login-config>
    <form-login-page>/login/login.html</form-login-page>
    <form-error-page>/login/bad-login.html</form-error-page>
  </form-login-config>
</login-config>
<security-role>
  <role-name>administrador</role-name>
</security-role>
<security-role>
  <role-name>usuario</role-name>
</security-role>
</web-app>

```

Ahora para poder interactuar con la base de datos y visualizar la información desde el sistema se presenta la codificación siguiente, tanto para la capa del cliente, datos, y servidor. La Cual servida como referencia para los demás ya desarrollados.

Proceso de guardado de datos de la tabla BSX\_EMPLEADO

## 1. Capa de datos

### **Función para generar clave primaria de la tabla.**

```

/**
 * FUNCION QUE GENERA CLAVES PARA LAS TABLAS DEL ESQUEMA.
 * @param INTABLENAME NOMBRE DE LA TABLA QUE REQUIERE LA CLAVE.
 * @param INCOLNAME COLUMNA DE LA TABLA QUE REQUIERE LA CLVAE.
 * @param INABR ABREVIATURA DEL NOMBRE DE LA TABLA SOLICITANTE.
 * @return NUEVA CLAVE GENERADA.
 */
CREATE OR REPLACE FUNCTION sch_bsx.F_KEYGEN(
IN INTABLENAME VARCHAR,

```

```

IN INCOLNAME VARCHAR,
IN INABR VARCHAR
)RETURNS VARCHAR AS
$$
DECLARE
TMP_NUMREGISTROS NUMERIC;
TMP_EXISTS NUMERIC;
OUT_KEY VARCHAR;
BEGIN
EXECUTE 'SELECT COUNT('||INCOLNAME||')+1 FROM sch_bsx.'||INTABLENAME
INTO TMP_NUMREGISTROS;
OUT_KEY := INABR||TMP_NUMREGISTROS;
EXECUTE 'SELECT COUNT('||INCOLNAME||') FROM sch_bsx.'||INTABLENAME||'
WHERE '||INCOLNAME||' = $1' INTO TMP_EXISTS USING OUT_KEY;
IF (TMP_EXISTS > 0) THEN
FOR I IN 1..TMP_NUMREGISTROS LOOP
EXECUTE 'SELECT COUNT('||INCOLNAME||') FROM
sch_bsx.'||INTABLENAME||' WHERE '||INCOLNAME||' = $1' INTO TMP_EXISTS USING
INABR||I;
IF (TMP_EXISTS = 0) THEN
OUT_KEY := INABR||I;
EXIT;
END IF;
END LOOP;
END IF;
RETURN OUT_KEY;
END
$$
LANGUAGE 'plpgsql' VOLATILE CALLED ON NULL INPUT;

```

### **Función para insertar los Datos**

```

/**
* FUNCION SCH_BSX.F_BSX_EMPLEADO_INSERT
* INGRESA UN REGISTRO EN LA TABLA SCH_BSX.BSX_EMPLEADO
* @param IN_BEMPL_CODIGO CAMPO QUE ALMACENA EL CODIGO DEL EMPLEADO
* @param IN_BDEPA_CODIGO CAMPO QUE ALMACENA EL CODIGO DEL
DEPARTAMENTO AL QUE PERTENECE EL EMPLEADO
* @param IN_BPARR_CODIGO IDENTIFICATIVO UNICO DEL REGISTRO PARROQUIA
* @param IN_BTDDID_CODIGO IDENTIFICATIVO UNICO DE REGISTRO DE
DOCUMENTO DE IDENTIFICACION
* @param IN_BEMPL_DOCUMENTO_IDENTIFICACION CEDULA O RUC DE LA
PERSONA
* @param IN_BEMPL_NOMBRE NOMBRE DE LA PERSONA
* @param IN_BEMPL_APELLIDO_MATERNO APELLIDO DE LA PERSONA
* @param IN_BEMPL_APELLIDO_PATERNO APELLIDO DE LA PERSONA
* @param IN_BEMPL_DIRECCION DIRECCION ESPECIFICA DE LA PERSONA
* @param IN_BEMPL_FOTO CAMPO QUE ALMACENA LA FOTO O ALGUN LOGOTIPO
DEL EMPLEADO
* @param IN_BEMPL_NOMBRE_FOTO ALMACENA EL NOMBRE DEL ADJUNTO DE LA
IMAGEN
* @param IN_BEMPL_TIPO ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL
EMPLEADO
* @param IN_BEMPL_FECHA_NACIMIENTO CAMPO QUE ALMACENA LA FECHA DE
NACIMIENTO DEL EMPLEADO
* @param IN_BEMPL_FECHA_INGRESO CAMPO QUE ALMACENA LA FECHA EN LA
QUE EL EMPLEADO INGRESO A LA EMPRESA

```

```

* @param IN_BEMPL_ESTADO_EMPLEADO CAMPO QUE ALMACENA EL ESTADO DEL
EMPLEADO
0 PASIVO
1 ACTIVO
2 ANULADO
* @param IN_BEMPL_ESTADO CAMPO QUE ALMACENA EL ESTADO DEL REGISTRO.
0 CUANDO EL REGISTRO ESTA ACTIVO
1 CUANDO EL REGISTRO ESTA INACTIVO, BORRADO LOGICO
* @param IN_BEMPL_USUARIO_INSERT CAMPO QUE ALMACENA EL USUARIO QUE
INSERTO UN NUEVO REGISTRO EN LA TABLA
* @param IN_BEMPL_USUARIO_UPDATE CAMPO QUE ALMACENA EL USUARIO QUE
MODIFICO O ACTUALIZO UN REGISTRO DE LA TABLA
* @return VARCHAR
*/
CREATE OR REPLACE FUNCTION SCH_BSX.F_BSX_EMPLEADO_INSERT(
IN IN_BEMPL_CODIGO SCH_BSX.BSX_EMPLEADO.BEMPL_CODIGO%TYPE,
IN IN_BDEPA_CODIGO SCH_BSX.BSX_EMPLEADO.BDEPA_CODIGO%TYPE,
IN IN_BPARR_CODIGO SCH_BSX.BSX_EMPLEADO.BPARR_CODIGO%TYPE,
IN IN_BTDDID_CODIGO SCH_BSX.BSX_EMPLEADO.BTDDID_CODIGO%TYPE,
IN IN_BEMPL_DOCUMENTO_IDENTIFICACION
SCH_BSX.BSX_EMPLEADO.BEMPL_DOCUMENTO_IDENTIFICACION%TYPE,
IN IN_BEMPL_NOMBRE SCH_BSX.BSX_EMPLEADO.BEMPL_NOMBRE%TYPE,
IN IN_BEMPL_APELLIDO_MATERNO
SCH_BSX.BSX_EMPLEADO.BEMPL_APELLIDO_MATERNO%TYPE,
IN IN_BEMPL_APELLIDO_PATERNO
SCH_BSX.BSX_EMPLEADO.BEMPL_APELLIDO_PATERNO%TYPE,
IN IN_BEMPL_SEXO SCH_BSX.BSX_EMPLEADO.BEMPL_SEXO%TYPE,
IN IN_BEMPL_DIRECCION SCH_BSX.BSX_EMPLEADO.BEMPL_DIRECCION%TYPE,
IN IN_BEMPL_FOTO SCH_BSX.BSX_EMPLEADO.BEMPL_FOTO%TYPE,
IN IN_BEMPL_NOMBRE_FOTO
SCH_BSX.BSX_EMPLEADO.BEMPL_NOMBRE_FOTO%TYPE,
IN IN_BEMPL_TIPO SCH_BSX.BSX_EMPLEADO.BEMPL_TIPO%TYPE,
IN IN_BEMPL_FECHA_NACIMIENTO
SCH_BSX.BSX_EMPLEADO.BEMPL_FECHA_NACIMIENTO%TYPE,
IN IN_BEMPL_FECHA_INGRESO
SCH_BSX.BSX_EMPLEADO.BEMPL_FECHA_INGRESO%TYPE,
IN IN_BEMPL_ESTADO_EMPLEADO
SCH_BSX.BSX_EMPLEADO.BEMPL_ESTADO_EMPLEADO%TYPE,
IN IN_BEMPL_ESTADO SCH_BSX.BSX_EMPLEADO.BEMPL_ESTADO%TYPE,
IN IN_BEMPL_USUARIO_INSERT
SCH_BSX.BSX_EMPLEADO.BEMPL_USUARIO_INSERT%TYPE,
IN IN_BADTM_CODIGO SCH_BSX.BSX_ADJUNTO_TMP.BADTM_CODIGO%TYPE
)RETURNS VARCHAR AS
$$
DECLARE
    OUT_KEY VARCHAR;
BEGIN
    IF (UPPER(IN_BEMPL_CODIGO) = 'KEYGEN') THEN
        SELECT
SCH_BSX.F_KEYGEN('BSX_EMPLEADO','BEMPL_CODIGO','BEMPL') INTO OUT_KEY;
    ELSE
        OUT_KEY := IN_BEMPL_CODIGO;
    END IF;

    INSERT INTO SCH_BSX.BSX_EMPLEADO(
        BEMPL_CODIGO,
        BDEPA_CODIGO,
        BPARR_CODIGO,

```

```

        BTDID_CODIGO,
        BEMPL_DOCUMENTO_IDENTIFICACION,
        BEMPL_NOMBRE,
        BEMPL_APELLIDO_MATERNO,
        BEMPL_APELLIDO_PATERNO,
        BEMPL_SEXO,
        BEMPL_DIRECCION,
        BEMPL_FOTO,
        BEMPL_NOMBRE_FOTO,
        BEMPL_TIPO,
        BEMPL_FECHA_NACIMIENTO,
        BEMPL_FECHA_INGRESO,
        BEMPL_ESTADO_EMPLEADO,
        BEMPL_ESTADO,
        BEMPL_USUARIO_INSERT,
        BEMPL_FECHA_INSERT,
        BEMPL_USUARIO_UPDATE,
        BEMPL_FECHA_UPDATE
    ) VALUES (
        OUT_KEY,
        IN_BDEPA_CODIGO,
        IN_BPARR_CODIGO,
        IN_BTDID_CODIGO,
        IN_BEMPL_DOCUMENTO_IDENTIFICACION,
        IN_BEMPL_NOMBRE,
        IN_BEMPL_APELLIDO_MATERNO,
        IN_BEMPL_APELLIDO_PATERNO,
        IN_BEMPL_SEXO,
        IN_BEMPL_DIRECCION,
        IN_BEMPL_FOTO,
        IN_BEMPL_NOMBRE_FOTO,
        IN_BEMPL_TIPO,
        IN_BEMPL_FECHA_NACIMIENTO,
        IN_BEMPL_FECHA_INGRESO,
        IN_BEMPL_ESTADO_EMPLEADO,
        IN_BEMPL_ESTADO,
        IN_BEMPL_USUARIO_INSERT,
        CURRENT_TIMESTAMP,
        IN_BEMPL_USUARIO_INSERT,
        CURRENT_TIMESTAMP
    );
    RETURN OUT_KEY;
END
$$
LANGUAGE 'plpgsql' VOLATILE CALLED ON NULL INPUT;

```

## 2. Capa de Servidor.

### Clase Java – Objetos

```

package com.besixplus.sisac.objects;
import java.io.Serializable;
/**
 * CLASE Bsx_empleado
 * TABLA: SCH_BSX.BSX_EMPLEADO
 * PERMITE ALMACENAR LOS DATOS GENERALES DE LOS EMPLEADOS
 * ABREVIATURA: BEMPL

```

```

*/
public class Bsx_empleado implements Serializable{
    private static final long serialVersionUID = 1486721151;
    /**
     * CAMPO QUE ALMACENA EL CODIGO DEL EMPLEADO
     */
    private String myBempl_codigo;
    /**
     * CAMPO QUE ALMACENA EL CODIGO DEL DEPARTAMENTO AL QUE
    PERTENECE EL EMPLEADO
     */
    private String myBdepa_codigo;
    /**
     * IDENTIFICATIVO UNICO DEL REGISTRO PARROQUIA
     */
    private String myBparr_codigo;
    /**
     * IDENTIFICATIVO UNICO DE REGISTRO DE DOCUMENTO DE
    IDENTIFICACION
     */
    private String myBtdid_codigo;
    /**
     * CEDULA O RUC DE LA PERSONA
     */
    private String myBempl_documento_identificacion;
    /**
     * NOMBRE DE LA PERSONA
     */
    private String myBempl_nombre;
    /**
     * APELLIDO DE LA PERSONA
     */
    private String myBempl_apellido_materno;
    /**
     * APELLIDO DE LA PERSONA
     */
    private String myBempl_apellido_paterno;
    /**
     * SEXO DE LA PERSONA
     */
    private int myBempl_sexo;
    /**
     * DIRECCION ESPECIFICA DE LA PERSONA
     */
    private String myBempl_direccion;
    /**
     * CAMPO QUE ALMACENA LA FOTO O ALGUN LOGOTIPO DEL EMPLEADO
     */
    private byte[] myBempl_foto;
    /**
     * ALMACENA EL NOMBRE DEL ADJUNTO DE LA IMAGEN
     */
    private String myBempl_nombre_foto;
    /**
     * ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL EMPLEADO
     */
    private String myBempl_tipo;
    /**

```

```

* ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL EMPLEADO
*/
private String myBagen_nombre;
/**
* ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL EMPLEADO
*/
private String myBagen_codigo;
/**
* CAMPO QUE ALMACENA LA FECHA DE NACIMIENTO DEL EMPLEADO
*/
private java.util.Date myBempl_fecha_nacimiento;
/**
* CAMPO QUE ALMACENA LA FECHA EN LA QUE EL EMPLEADO INGRESO A
LA EMPRESA
*/
private java.util.Date myBempl_fecha_ingreso;
/**
* CAMPO QUE ALMACENA EL ESTADO DEL EMPLEADO
0 PASIVO
1 ACTIVO
2 ANULADO
*/
private boolean myBempl_estado_empleado;
/**
* CAMPO QUE ALMACENA EL ESTADO DEL REGISTRO.
0 CUANDO EL REGISTRO ESTA ACTIVO
1 CUANDO EL REGISTRO ESTA INACTIVO, BORRADO LOGICO
*/
private boolean myBempl_estado;
/**
* CAMPO QUE ALMACENA EL USUARIO QUE INSERTO UN NUEVO
REGISTRO EN LA TABLA
*/
private String myBempl_usuario_insert;
/**
* CAMPO QUE ALMACENA EL USUARIO QUE MODIFICO O ACTUALIZO UN
REGISTRO DE LA TABLA
*/
private String myBempl_usuario_update;
/**
* CONSTRUCTOR DE LA CLASE Bsx_empleado
*/
public Bsx_empleado(){
/**
* CONSTRUCTOR DE LA CLASE Bsx_empleado
* @param inBempl_codigo CAMPO QUE ALMACENA EL CODIGO DEL
EMPLEADO
* @param inBdepa_codigo CAMPO QUE ALMACENA EL CODIGO DEL
DEPARTAMENTO AL QUE PERTENECE EL EMPLEADO
* @param inBparr_codigo IDENTIFICATIVO UNICO DEL REGISTRO PARROQUIA
* @param inBtdid_codigo IDENTIFICATIVO UNICO DE REGISTRO DE
DOCUMENTO DE IDENTIFICACION
* @param inBempl_documento_identificacion CEDULA O RUC DE LA PERSONA
* @param inBempl_nombre NOMBRE DE LA PERSONA
* @param inBempl_apellido_materno APELLIDO DE LA PERSONA
* @param inBempl_apellido_paterno APELLIDO DE LA PERSONA
* @param inBempl_direccion DIRECCION ESPECIFICA DE LA PERSONA

```

```

    * @param inBempl_foto CAMPO QUE ALMACENA LA FOTO O ALGUN
    LOGOTIPO DEL EMPLEADO
    * @param inBempl_nombre_foto ALMACENA EL NOMBRE DEL ADJUNTO DE LA
    IMAGEN
    * @param inBempl_tipo ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN
    DEL EMPLEADO
    * @param inBempl_fecha_nacimiento CAMPO QUE ALMACENA LA FECHA DE
    NACIMIENTO DEL EMPLEADO
    * @param inBempl_fecha_ingreso CAMPO QUE ALMACENA LA FECHA EN LA
    QUE EL EMPLEADO INGRESO A LA EMPRESA
    * @param inBempl_estado_empleado CAMPO QUE ALMACENA EL ESTADO DEL
    EMPLEADO
    0 PASIVO
    1 ACTIVO
    2 ANULADO
    * @param inBempl_estado CAMPO QUE ALMACENA EL ESTADO DEL
    REGISTRO.
    0 CUANDO EL REGISTRO ESTA ACTIVO
    1 CUANDO EL REGISTRO ESTA INACTIVO, BORRADO LOGICO
    * @param inBempl_usuario_insert CAMPO QUE ALMACENA EL USUARIO QUE
    INSERTO UN NUEVO REGISTRO EN LA TABLA
    * @param inBempl_usuario_update CAMPO QUE ALMACENA EL USUARIO QUE
    MODIFICO O ACTUALIZO UN REGISTRO DE LA TABLA
    */
    public Bsx_empleado(
        String inBempl_codigo,
        String inBdepa_codigo,
        String inBparr_codigo,
        String inBtdid_codigo,
        String inBempl_documento_identificacion,
        String inBempl_nombre,
        String inBempl_apellido_materno,
        String inBempl_apellido_paterno,
        int inBempl_sexo,
        String inBempl_direccion,
        byte[] inBempl_foto,
        String inBempl_nombre_foto,
        String inBempl_tipo,
        String inBagen_codigo,
        String inBagen_nombre,
        java.util.Date inBempl_fecha_nacimiento,
        java.util.Date inBempl_fecha_ingreso,
        boolean inBempl_estado_empleado,
        boolean inBempl_estado,
        String inBempl_usuario_insert,
        String inBempl_usuario_update
    ){
        this.setBEMPL_CODIGO(inBempl_codigo);
        this.setBDEPA_CODIGO(inBdepa_codigo);
        this.setBPARR_CODIGO(inBparr_codigo);
        this.setBTDID_CODIGO(inBtdid_codigo);

        this.setBEMPL_DOCUMENTO_IDENTIFICACION(inBempl_documento_identificacio
n);

        this.setBEMPL_NOMBRE(inBempl_nombre);
        this.setBEMPL_APELLIDO_MATERNO(inBempl_apellido_materno);
        this.setBEMPL_APELLIDO_PATERNO(inBempl_apellido_paterno);
        this.setBEMPL_SEXO(inBempl_sexo);

```



```

        this.setBEMPL_DIRECCION(inBempl_direccion);
        this.setBEMPL_FOTO(inBempl_foto);
        this.setBEMPL_NOMBRE_FOTO(inBempl_nombre_foto);
        this.setBEMPL_TIPO(inBempl_tipo);
        this.setBAGEN_CODIGO(inBagen_codigo);
        this.setBAGEN_NOMBRE(inBagen_nombre);
        this.setBEMPL_FECHA_NACIMIENTO(inBempl_fecha_nacimiento);
        this.setBEMPL_FECHA_INGRESO(inBempl_fecha_ingreso);
        this.setBEMPL_ESTADO_EMPLEADO(inBempl_estado_empleado);
        this.setBEMPL_ESTADO(inBempl_estado);
        this.setBEMPL_USUARIO_INSERT(inBempl_usuario_insert);
        this.setBEMPL_USUARIO_UPDATE(inBempl_usuario_update);
    }

    /**
     * ESTABLECE CAMPO QUE ALMACENA EL CODIGO DEL EMPLEADO
     * @param inbempl_codigo CAMPO QUE ALMACENA EL CODIGO DEL
EMPLEADO
     */
    public void setBEMPL_CODIGO(String inBempl_codigo){
        this.myBempl_codigo = inBempl_codigo;
    }
    /**
     * OBTIENE CAMPO QUE ALMACENA EL CODIGO DEL EMPLEADO
     * @return String CAMPO QUE ALMACENA EL CODIGO DEL EMPLEADO
     */
    public String getBEMPL_CODIGO(){
        return this.myBempl_codigo;
    }

    /**
     * ESTABLECE CAMPO QUE ALMACENA EL CODIGO DEL DEPARTAMENTO
AL QUE PERTENECE EL EMPLEADO
     * @param inbdepa_codigo CAMPO QUE ALMACENA EL CODIGO DEL
DEPARTAMENTO AL QUE PERTENECE EL EMPLEADO
     */
    public void setBDEPA_CODIGO(String inBdepa_codigo){
        this.myBdepa_codigo = inBdepa_codigo;
    }
    /**
     * OBTIENE CAMPO QUE ALMACENA EL CODIGO DEL DEPARTAMENTO AL
QUE PERTENECE EL EMPLEADO
     * @return String CAMPO QUE ALMACENA EL CODIGO DEL DEPARTAMENTO
AL QUE PERTENECE EL EMPLEADO
     */
    public String getBDEPA_CODIGO(){
        return this.myBdepa_codigo;
    }

    /**
     * ESTABLECE IDENTIFICATIVO UNICO DEL REGISTRO PARROQUIA
     * @param inbparr_codigo IDENTIFICATIVO UNICO DEL REGISTRO PARROQUIA
     */
    public void setBPARR_CODIGO(String inBparr_codigo){
        this.myBparr_codigo = inBparr_codigo;
    }
    /**
     * OBTIENE IDENTIFICATIVO UNICO DEL REGISTRO PARROQUIA

```

```

* @return String IDENTIFICATIVO UNICO DEL REGISTRO PARROQUIA
*/
public String getBPARR_CODIGO(){
    return this.myBparr_codigo;
}

/**
* ESTABLECE IDENTIFICATIVO UNICO DE REGISTRO DE DOCUMENTO DE
IDENTIFICACION
* @param inbtdid_codigo IDENTIFICATIVO UNICO DE REGISTRO DE
DOCUMENTO DE IDENTIFICACION
*/
public void setBTDID_CODIGO(String inBtdid_codigo){
    this.myBtdid_codigo = inBtdid_codigo;
}

/**
* OBTIENE IDENTIFICATIVO UNICO DE REGISTRO DE DOCUMENTO DE
IDENTIFICACION
* @return String IDENTIFICATIVO UNICO DE REGISTRO DE DOCUMENTO DE
IDENTIFICACION
*/
public String getBTDID_CODIGO(){
    return this.myBtdid_codigo;
}

/**
* ESTABLECE CEDULA O RUC DE LA PERSONA
* @param inbempl_documento_identificacion CEDULA O RUC DE LA PERSONA
*/
public void setBEMPL_DOCUMENTO_IDENTIFICACION(String
inBempl_documento_identificacion){
    this.myBempl_documento_identificacion = inBempl_documento_identificacion;
}

/**
* OBTIENE CEDULA O RUC DE LA PERSONA
* @return String CEDULA O RUC DE LA PERSONA
*/
public String getBEMPL_DOCUMENTO_IDENTIFICACION(){
    return this.myBempl_documento_identificacion;
}

/**
* ESTABLECE NOMBRE DE LA PERSONA
* @param inbempl_nombre NOMBRE DE LA PERSONA
*/
public void setBEMPL_NOMBRE(String inBempl_nombre){
    this.myBempl_nombre = inBempl_nombre;
}

/**
* OBTIENE NOMBRE DE LA PERSONA
* @return String NOMBRE DE LA PERSONA
*/
public String getBEMPL_NOMBRE(){
    return this.myBempl_nombre;
}

/**
* ESTABLECE APELLIDO DE LA PERSONA

```

```

* @param inBempl_apellido_materno APELLIDO DE LA PERSONA
*/
public void setBEMPL_APELLIDO_MATERNO(String inBempl_apellido_materno){
    this.myBempl_apellido_materno = inBempl_apellido_materno;
}
/**
* OBTIENE APELLIDO DE LA PERSONA
* @return String APELLIDO DE LA PERSONA
*/
public String getBEMPL_APELLIDO_MATERNO(){
    return this.myBempl_apellido_materno;
}

/**
* ESTABLECE APELLIDO DE LA PERSONA
* @param inBempl_apellido_paterno APELLIDO DE LA PERSONA
*/
public void setBEMPL_APELLIDO_PATERNO(String inBempl_apellido_paterno){
    this.myBempl_apellido_paterno = inBempl_apellido_paterno;
}
/**
* OBTIENE APELLIDO DE LA PERSONA
* @return String APELLIDO DE LA PERSONA
*/
public String getBEMPL_APELLIDO_PATERNO(){
    return this.myBempl_apellido_paterno;
}

/**
* ESTABLECE SEXO DE LA PERSONA
* @param inBempl_apellido_paterno SEXO DE LA PERSONA
*/
public void setBEMPL_SEXO(int inBempl_sexo){
    this.myBempl_sexo = inBempl_sexo;
}
/**
* OBTIENE APELLIDO DE LA PERSONA
* @return String APELLIDO DE LA PERSONA
*/
public int getBEMPL_SEXO(){
    return this.myBempl_sexo;
}

/**
* ESTABLECE DIRECCION ESPECIFICA DE LA PERSONA
* @param inBempl_direccion DIRECCION ESPECIFICA DE LA PERSONA
*/
public void setBEMPL_DIRECCION(String inBempl_direccion){
    this.myBempl_direccion = inBempl_direccion;
}
/**
* OBTIENE DIRECCION ESPECIFICA DE LA PERSONA
* @return String DIRECCION ESPECIFICA DE LA PERSONA
*/
public String getBEMPL_DIRECCION(){
    return this.myBempl_direccion;
}
/**

```

```

    * ESTABLECE CAMPO QUE ALMACENA LA FOTO O ALGUN LOGOTIPO DEL
    EMPLEADO
    * @param inbempl_foto CAMPO QUE ALMACENA LA FOTO O ALGUN
    LOGOTIPO DEL EMPLEADO
    */
    public void setBEMPL_FOTO(byte[] inBempl_foto){
        this.myBempl_foto = inBempl_foto;
    }
    /**
    * OBTIENE CAMPO QUE ALMACENA LA FOTO O ALGUN LOGOTIPO DEL
    EMPLEADO
    * @return byte[] CAMPO QUE ALMACENA LA FOTO O ALGUN LOGOTIPO DEL
    EMPLEADO
    */
    public byte[] getBEMPL_FOTO(){
        return this.myBempl_foto;
    }
    /**
    * ESTABLECE ALMACENA EL NOMBRE DEL ADJUNTO DE LA IMAGEN
    * @param inbempl_nombre_foto ALMACENA EL NOMBRE DEL ADJUNTO DE LA
    IMAGEN
    */
    public void setBEMPL_NOMBRE_FOTO(String inBempl_nombre_foto){
        this.myBempl_nombre_foto = inBempl_nombre_foto;
    }
    /**
    * OBTIENE ALMACENA EL NOMBRE DEL ADJUNTO DE LA IMAGEN
    * @return String ALMACENA EL NOMBRE DEL ADJUNTO DE LA IMAGEN
    */
    public String getBEMPL_NOMBRE_FOTO(){
        return this.myBempl_nombre_foto;
    }
    /**
    * ESTABLECE ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL
    EMPLEADO
    * @param inbempl_tipo ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL
    EMPLEADO
    */
    public void setBEMPL_TIPO(String inBempl_tipo){
        this.myBempl_tipo = inBempl_tipo;
    }
    /**
    * OBTIENE ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL
    EMPLEADO
    * @return String ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL
    EMPLEADO
    */
    public String getBEMPL_TIPO(){
        return this.myBempl_tipo;
    }
    /**
    * ESTABLECE ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL
    EMPLEADO
    * @param inbempl_tipo ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL
    EMPLEADO
    */

```

```

public void setBAGEN_CODIGO(String inBagen_codigo){
    this.myBagen_codigo = inBagen_codigo;
}
/**
 * OBTIENE ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL
EMPLEADO
 * @return String ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL
EMPLEADO
 */
public String getBAGEN_CODIGO(){
    return this.myBagen_codigo;
}
/**
 * ESTABLECE ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL
EMPLEADO
 * @param inbempl_tipo ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL
EMPLEADO
 */
public void setBAGEN_NOMBRE(String inBagen_nombre){
    this.myBagen_nombre = inBagen_nombre;
}
/**
 * OBTIENE ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL
EMPLEADO
 * @return String ALMACENA EL TIPO DE ADJUNTO DE LA IMAGEN DEL
EMPLEADO
 */
public String getBAGEN_NOMBRE(){
    return this.myBagen_nombre;
}
}
/**
 * ESTABLECE CAMPO QUE ALMACENA LA FECHA DE NACIMIENTO DEL
EMPLEADO
 * @param inbempl_fecha_nacimiento CAMPO QUE ALMACENA LA FECHA DE
NACIMIENTO DEL EMPLEADO
 */
public void setBEMPL_FECHA_NACIMIENTO(java.util.Date
inBempl_fecha_nacimiento){
    this.myBempl_fecha_nacimiento = inBempl_fecha_nacimiento;
}
/**
 * OBTIENE CAMPO QUE ALMACENA LA FECHA DE NACIMIENTO DEL
EMPLEADO
 * @return java.util.Date CAMPO QUE ALMACENA LA FECHA DE NACIMIENTO
DEL EMPLEADO
 */
public java.util.Date getBEMPL_FECHA_NACIMIENTO(){
    return this.myBempl_fecha_nacimiento;
}
}
/**
 * ESTABLECE CAMPO QUE ALMACENA LA FECHA EN LA QUE EL
EMPLEADO INGRESO A LA EMPRESA
 * @param inbempl_fecha_ingreso CAMPO QUE ALMACENA LA FECHA EN LA
QUE EL EMPLEADO INGRESO A LA EMPRESA
 */
public void setBEMPL_FECHA_INGRESO(java.util.Date inBempl_fecha_ingreso){

```

```

        this.myBempl_fecha_ingreso = inBempl_fecha_ingreso;
    }
    /**
     * OBTIENE CAMPO QUE ALMACENA LA FECHA EN LA QUE EL EMPLEADO
    INGRESO A LA EMPRESA
     * @return java.util.Date CAMPO QUE ALMACENA LA FECHA EN LA QUE EL
    EMPLEADO INGRESO A LA EMPRESA
     */
    public java.util.Date getBEMPL_FECHA_INGRESO(){
        return this.myBempl_fecha_ingreso;
    }

    /**
     * ESTABLECE CAMPO QUE ALMACENA EL ESTADO DEL EMPLEADO
    0 PASIVO
    1 ACTIVO
    2 ANULADO
     * @param inbempl_estado_empleado CAMPO QUE ALMACENA EL ESTADO DEL
    EMPLEADO
    0 PASIVO
    1 ACTIVO
    2 ANULADO
     */
    public void setBEMPL_ESTADO_EMPLEADO(boolean inBempl_estado_empleado){
        this.myBempl_estado_empleado = inBempl_estado_empleado;
    }
    /**
     * OBTIENE CAMPO QUE ALMACENA EL ESTADO DEL EMPLEADO
    0 PASIVO
    1 ACTIVO
    2 ANULADO
     * @return boolean CAMPO QUE ALMACENA EL ESTADO DEL EMPLEADO
    0 PASIVO
    1 ACTIVO
    2 ANULADO
     */
    public boolean getBEMPL_ESTADO_EMPLEADO(){
        return this.myBempl_estado_empleado;
    }

    /**
     * ESTABLECE CAMPO QUE ALMACENA EL ESTADO DEL REGISTRO.
    0 CUANDO EL REGISTRO ESTA ACTIVO
    1 CUANDO EL REGISTRO ESTA INACTIVO, BORRADO LOGICO
     * @param inbempl_estado CAMPO QUE ALMACENA EL ESTADO DEL
    REGISTRO.
    0 CUANDO EL REGISTRO ESTA ACTIVO
    1 CUANDO EL REGISTRO ESTA INACTIVO, BORRADO LOGICO
     */
    public void setBEMPL_ESTADO(boolean inBempl_estado){
        this.myBempl_estado = inBempl_estado;
    }
    /**
     * OBTIENE CAMPO QUE ALMACENA EL ESTADO DEL REGISTRO.
    0 CUANDO EL REGISTRO ESTA ACTIVO
    1 CUANDO EL REGISTRO ESTA INACTIVO, BORRADO LOGICO
     * @return boolean CAMPO QUE ALMACENA EL ESTADO DEL REGISTRO.
    0 CUANDO EL REGISTRO ESTA ACTIVO

```

```

1 CUANDO EL REGISTRO ESTA INACTIVO, BORRADO LOGICO
    */
    public boolean getBEMPL_ESTADO(){
        return this.myBempl_estado;
    }

    /**
    * ESTABLECE CAMPO QUE ALMACENA EL USUARIO QUE INSERTO UN
    NUEVO REGISTRO EN LA TABLA
    * @param inbempl_usuario_insert CAMPO QUE ALMACENA EL USUARIO QUE
    INSERTO UN NUEVO REGISTRO EN LA TABLA
    */
    public void setBEMPL_USUARIO_INSERT(String inBempl_usuario_insert){
        this.myBempl_usuario_insert = inBempl_usuario_insert;
    }
    /**
    * OBTIENE CAMPO QUE ALMACENA EL USUARIO QUE INSERTO UN NUEVO
    REGISTRO EN LA TABLA
    * @return String CAMPO QUE ALMACENA EL USUARIO QUE INSERTO UN
    NUEVO REGISTRO EN LA TABLA
    */
    public String getBEMPL_USUARIO_INSERT(){
        return this.myBempl_usuario_insert;
    }

    /**
    * ESTABLECE CAMPO QUE ALMACENA EL USUARIO QUE MODIFICO O
    ACTUALIZO UN REGISTRO DE LA TABLA
    * @param inbempl_usuario_update CAMPO QUE ALMACENA EL USUARIO QUE
    MODIFICO O ACTUALIZO UN REGISTRO DE LA TABLA
    */
    public void setBEMPL_USUARIO_UPDATE(String inBempl_usuario_update){
        this.myBempl_usuario_update = inBempl_usuario_update;
    }
    /**
    * OBTIENE CAMPO QUE ALMACENA EL USUARIO QUE MODIFICO O
    ACTUALIZO UN REGISTRO DE LA TABLA
    * @return String CAMPO QUE ALMACENA EL USUARIO QUE MODIFICO O
    ACTUALIZO UN REGISTRO DE LA TABLA
    */
    public String getBEMPL_USUARIO_UPDATE(){
        return this.myBempl_usuario_update;
    }
}

```

### Clase Java – db

```

package com.besixplus.sisac.db;
import java.util.ArrayList;
import java.util.HashMap;
import java.io.Serializable;
import java.sql.CallableStatement;
import java.sql.ResultSet;
import java.sql.Savepoint;
import java.sql.SQLException;
import java.sql.Types;
/**

```

```

* CLASE Bsx_empleado
* ESTA CLASE CONTIENE LOS METODOS NECESARIOS PARA EL MANEJO DE LA
INFORMACION CONTENIDA EN LA TABLA Bsx_empleado.
*/
public class Bsx_empleado implements Serializable{
    private static final long serialVersionUID= 836520166;
    /**
     * VARIABLE QUE ALMACENA UNA INSTANCIA DE LA CLASE
com.besixplus.sisac.objects.Bsx_empleado.
     */
    private com.besixplus.sisac.objects.Bsx_empleado myBsx_empleado = null;
    /**
     * CONSTRUCTOR DE LA CLASE Bsx_empleado.
     */
    public Bsx_empleado(){
    /**
     * CONSTRUCTOR DE LA CLASE Bsx_empleado.
     * @param inBsx_empleado OBJETO QUE CONTIENE TODA LA INFORMACION
NECESARIA PARA INSERTAR EN LA TABLA Bsx_empleado.
     */
    public Bsx_empleado(
        com.besixplus.sisac.objects.Bsx_empleado inBsx_empleado
    ){
        this.setBsx_empleado(inBsx_empleado);
    }
    /**
     * ESTABLECE LA INSTANCIA DEL OBJETO Bsx_empleado.
     * @param inBsx_empleado OBJETO Bsx_empleado.
     * @return void
     */
    public void setBsx_empleado(com.besixplus.sisac.objects.Bsx_empleado
inBsx_empleado){
        this.myBsx_empleado = inBsx_empleado;
    }
    /**
     * OBTIENE LA INSTANCIA DEL OBJETO Bsx_empleado.
     * @return Bsx_empleado OBJETO Bsx_empleado.
     */
    public com.besixplus.sisac.objects.Bsx_empleado getBsx_empleado(){
        return this.myBsx_empleado;
    }
    /**
     * INSERTA UN REGISTRO EN LA TABLA Bsx_empleado.
     * @param inConnection CONEXION A LA BASE DE DATOS.
     * @return bool <code>true</code> SI SE INGRESO EL REGISTRO, CASO
CONTRARIO <code>false</code>.
     */
    public String insert(
        java.sql.Connection inConnection,
        String inBadtm_codigo
    ){
        String outResult = "true";
        try{
            CallableStatement stmInsert = inConnection.prepareCall("{ ? = call
sch_bsx.F_BSX_EMPLEADO_INSERT(?, ?, ?, ?, ?, ?, ?, ?, ?::smallint, ?, ?, ?, ?,
?::date, ?::date, ?, ?, ?) }");
            stmInsert.registerOutParameter(1, java.sql.Types. VARCHAR);
            stmInsert.setString(2, this.getBsx_empleado().getBEMPL_CODIGO());
            stmInsert.setString(3, this.getBsx_empleado().getBDEPA_CODIGO());

```



```

stmInsert.setString(4, this.getBsx_empleado().getBPARR_CODIGO());
stmInsert.setString(5, this.getBsx_empleado().getBTDID_CODIGO());
stmInsert.setString(6,
this.getBsx_empleado().getBEMPL_DOCUMENTO_IDENTIFICACION());
stmInsert.setString(7, this.getBsx_empleado().getBEMPL_NOMBRE());
stmInsert.setString(8,
this.getBsx_empleado().getBEMPL_APELLIDO_MATERNO());
stmInsert.setString(9,
this.getBsx_empleado().getBEMPL_APELLIDO_PATERNO());
stmInsert.setInt(10, this.getBsx_empleado().getBEMPL_SEXO());
stmInsert.setString(11, this.getBsx_empleado().getBEMPL_DIRECCION());
stmInsert.setBytes(12, this.getBsx_empleado().getBEMPL_FOTO());
stmInsert.setString(13,
this.getBsx_empleado().getBEMPL_NOMBRE_FOTO());
stmInsert.setString(14, this.getBsx_empleado().getBEMPL_TIPO());
stmInsert.setDate(15, new
java.sql.Date(this.getBsx_empleado().getBEMPL_FECHA_NACIMIENTO().ge
tTime()));
stmInsert.setDate(16, new
java.sql.Date(this.getBsx_empleado().getBEMPL_FECHA_INGRESO().getTim
e()));
stmInsert.setBoolean(17,
this.getBsx_empleado().getBEMPL_ESTADO_EMPLEADO());
stmInsert.setBoolean(18, this.getBsx_empleado().getBEMPL_ESTADO());
stmInsert.setString(19,
this.getBsx_empleado().getBEMPL_USUARIO_INSERT());
stmInsert.setString(20, inBadtm_codigo!=null?inBadtm_codigo:null);
stmInsert.execute();
this.getBsx_empleado().setBEMPL_CODIGO(stmInsert.getString(1));
stmInsert.close();
} catch (SQLException e) {
com.besixplus.sisac.db.SqlErrorHandler.errorHandler(e);
outResult = e.getMessage();
}
return outResult;
}
}

```

## Clase Java – Servlet

```

package com.besixplus.sisac.servlets;
import java.io.IOException;
import java.io.Serializable;
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.Timestamp;
import java.util.Iterator;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.commons.fileupload.FileItem;
import org.apache.commons.fileupload.FileUploadException;
import org.apache.commons.fileupload.disk.DiskFileItemFactory;
import org.apache.commons.fileupload.servlet.ServletFileUpload;
import org.json.JSONException;

```

```

import org.json.JSONObject;
import com.besixplus.sisac.objects.ServerResponse;
import com.sun.tools.ws.wsdl.document.jaxws.Exception;

public class Bsx_empleado extends HttpServlet implements Serializable{
    private static final long serialVersionUID = 89432014;
    public Bsx_empleado() {
        super();
    }
    protected void service(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        boolean tmpIsMultipart = ServletFileUpload.isMultipartContent(request);
        String res = "true";
        String tmpRequest = null;
        String inJSONContactos = null;
        String inBadtm_codigo = null;
        String inJSONEliminarContactos = null;
        ServerResponse appResponse= new ServerResponse();
        if(tmpIsMultipart){
            com.besixplus.sisac.objects.Bsx_empleado obj = new
com.besixplus.sisac.objects.Bsx_empleado();
            DiskFileItemFactory factory = new DiskFileItemFactory();
            ServletFileUpload upload = new ServletFileUpload(factory);
            upload.setSizeMax(10485760);
            List items;
            try {
                items = upload.parseRequest(request);
                Iterator iter = items.iterator();
                while (iter.hasNext()) {
                    FileItem item = (FileItem) iter.next();
                    if (item.isFormField()) {
                        if(item.getFieldName().equals("request"))
                            tmpRequest = item.getString();
                        if(item.getFieldName().equals("bempl_codigo") )
                            obj.setBEMPL_CODIGO(item.getString());
                        if(item.getFieldName().equals("bdepa_codigo") &&
item.getString().trim().length() > 0)
                            obj.setBDEPA_CODIGO(item.getString());
                        if(item.getFieldName().equals("bparr_codigo") &&
item.getString().trim().length() > 0)
                            obj.setBPARR_CODIGO(item.getString());
                        if(item.getFieldName().equals("btid_codigo") &&
item.getString().trim().length() > 0)
                            obj.setBTDID_CODIGO(item.getString());
                        if(item.getFieldName().equals("bempl_documento_identificacion") )
                            obj.setBEMPL_DOCUMENTO_IDENTIFICACION(item.getString());
                        if(item.getFieldName().equals("bempl_nombre") )
                            obj.setBEMPL_NOMBRE(item.getString());
                        if(item.getFieldName().equals("bempl_apellido_materno") )
                            obj.setBEMPL_APELLIDO_MATERNO(item.getString());
                        if(item.getFieldName().equals("bempl_apellido_paterno") )
                            obj.setBEMPL_APELLIDO_PATERNO(item.getString());
                        if(item.getFieldName().equals("bempl_sexo") )
                            obj.setBEMPL_SEXO(Integer.valueOf(item.getString()));
                        if(item.getFieldName().equals("bempl_direccion") )
                            obj.setBEMPL_DIRECCION(item.getString());
                        if(item.getFieldName().equals("bempl_nombre_foto") )
                            obj.setBEMPL_NOMBRE_FOTO(item.getString());
                    }
                }
            }
        }
    }
}

```

```

        if(item.getFieldName().equals("bempl_tipo") )
            obj.setBEMPL_TIPO(item.getString());
        if(item.getFieldName().equals("bempl_fecha_nacimiento") )
obj.setBEMPL_FECHA_NACIMIENTO(java.sql.Date.valueOf(item.getString()));
        if(item.getFieldName().equals("bempl_fecha_ingreso") )
            obj.setBEMPL_FECHA_INGRESO(java.sql.Date.valueOf(item.getString()));
        if(item.getFieldName().equals("bempl_estado_employed") )
obj.setBEMPL_ESTADO_EMPLEADO(Boolean.valueOf(item.getString()));
        if(item.getFieldName().equals("inBadtm_codigo") )
            inBadtm_codigo = item.getString();
        if(item.getFieldName().equals("inJSONContactos") )
            inJSONContactos = item.getString();
        if(item.getFieldName().equals("inJSONEliminarContactos") )
            inJSONEliminarContactos = item.getString();
    } else {
        if(item.getFieldName().equals("bempl_foto"))
            obj.setBEMPL_FOTO(item.get());
    }
}
obj.setBEMPL_ESTADO(true);
Connection con = com.besixplus.sisac.db.ManagerConnection.getConnection();
con.setAutoCommit(false);
if(tmpRequest.trim().toUpperCase().equals("INSERT")){
obj.setBEMPL_CODIGO("KEYGEN");
obj.setBEMPL_USUARIO_INSERT(request.getUserPrincipal().getName());
res = new com.besixplus.sisac.db.Bsx_empleado(obj).insert(con,inBadtm_codigo);
if(!res.equalsIgnoreCase("false")){
    org.json.JSONArray JSONContactos = new org.json.JSONArray(inJSONContactos);
    for (int i = 0 ; i < JSONContactos.length(); i++){
        org.json.JSONObject JSONContactoObj =
            (org.json.JSONObject)JSONContactos.get(i);
        com.besixplus.sisac.objects.Bsx_empleado_contacto objCont = new
            com.besixplus.sisac.objects.Bsx_empleado_contacto();
        objCont.setBEMCO_CODIGO("KEYGEN");
objCont.setBTCON_CODIGO(JSONContactoObj.get("BTCON_CODIGO").toString());
        objCont.setBEMPL_CODIGO(obj.getBEMPL_CODIGO());
objCont.setBEMCO_CONTACTO(JSONContactoObj.get("BEMCO_CONTACTO").toString());
objCont.setBEMCO_DESCRIPCION(JSONContactoObj.get("BEMCO_DESCRIPCION").toString());
objCont.setBEMCO_ESTADO(true);
objCont.setBEMCO_USUARIO_INSERT(request.getUserPrincipal().getName());
res =new com.besixplus.sisac.db.Bsx_empleado_contacto(objCont).insert(con);
        if (res.equals("false"))
            break;
    }
}
}
if(res.equals("true") ){
    if(!con.isClosed()){
        con.commit();
        con.setAutoCommit(true);
    }
} else {
    if(!con.isClosed())
        con.rollback();
}
if(!con.isClosed())
    con.close();

```

```

}
try {
    con.close();
} catch (SQLException e) {
    com.besixplus.sisac.db.SQLErrorHandler.errorHandler(e);
    response.sendError(HttpServletResponse.SC_INTERNAL_SERVER_ERROR,
        e.getMessage());
    res = "false";
}
} catch (FileUploadException e) {
    e.printStackTrace();
    response.sendError(HttpServletResponse.SC_INTERNAL_SERVER_ERROR,
        e.getMessage());
    res = "false";
} catch (SQLException e) {
// TODO Auto-generated catch block
    e.printStackTrace();
} catch (JSONException e) {
// TODO Auto-generated catch block
    e.printStackTrace();
}
}
appResponse.setSuccess(Boolean.parseBoolean(res));
appResponse.setMsg("Datos Almacenados correctamente");
response.setContentType("text/html");
response.getWriter().println(new JSONObject(appResponse).toString());
} else {
    String tmpCodigo = request.getParameter("codigo");
    String tmpColumna = request.getParameter("columna");
    String tmpOpcion = request.getParameter("request");
    if(tmpCodigo != null){
        com.besixplus.sisac.objects.Bsx_empleado obj = null;
        obj = new com.besixplus.sisac.objects.Bsx_empleado();
        obj.setBEMPL_CODIGO(tmpCodigo);
        Connection con =
            com.besixplus.sisac.db.ManagerConnection.getConnection();
        try{
            con.setAutoCommit(!com.besixplus.sisac.db.ManagerConnection.isDeployed());
            new com.besixplus.sisac.db.Bsx_empleado(obj).select_FOTO(con);
            con.close();
        } catch (SQLException e) {
            com.besixplus.sisac.db.SQLErrorHandler.errorHandler(e);
        }
        ServletOutputStream out = response.getOutputStream();
        if(tmpColumna.toUpperCase().equals("BEMPL_FOTO")){
            try{
                out.write(obj.getBEMPL_FOTO());
            } catch(IOException e){
                response.sendError(HttpServletResponse.SC_INTERNAL_SERVER_ERROR,
                    e.getMessage());
            }
        }
    }
    if(tmpOpcion.equals("ver")){
        if(obj.getBEMPL_NOMBRE_FOTO() != null){
            response.setHeader("Content-Disposition","inline;
                filename="+obj.getBEMPL_NOMBRE_FOTO()+".");
            response.setContentType("application/"+obj.getBEMPL_NOMBRE_FOTO()+"."+obj.getBEMPL_TIPO());
            out.flush();
        }
    }
}

```

```

        out.close();
    }
    }
    out.flush();
    out.close();
}
}
}
}

```

## Clase JAVA – Servicio Web

```

package com.besixplus.sisac.ws;
import java.util.ArrayList;
import java.util.HashMap;
import java.io.Serializable;
import java.sql.Connection;
import java.sql.SQLException;
import javax.annotation.Resource;
import javax.jws.WebMethod;
import javax.jws.WebService;
import javax.xml.namespace.QName;
import javax.xml.soap.SOAPBinding;
import javax.xml.soap.SOAPBinding.Style;
import javax.xml.soap.SOAPEXception;
import javax.xml.soap.SOAPFactory;
import javax.xml.ws.soap.SOAPFaultException;
import javax.jws.WebParam;
import javax.servlet.http.HttpServletRequest;
import javax.xml.ws.WebServiceContext;
import javax.xml.ws.handler.MessageContext;
import com.besixplus.sisac.db.ManagerConnection;
/**
 * CLASE Bsx_empleado
 * CONTIENE LOS METODOS NECESARIOS PARA EL MANEJO DE LA INFORMACION
 * CONTENIDA EN LA TABLA Bsx_empleado POR MEDIO DE SERVICIOS WEB.
 */
@WebService()
@SOAPBinding(style=Style.RPC)
public class Bsx_empleado implements Serializable{

    private static final long serialVersionUID = 1404244281;

    @Resource WebServiceContext wctx;
    /**
     * SELECCIONA UN REGISTRO EN LA TABLA Bsx_empleado EN UNA
     * ESTRUCTURA JSON o XML.
     * @param inBempl_codigo bempl_codigo
     * @return String REGISTRO SELECCIONADO.
     */
    @WebMethod
    public String select(
        @WebParam(name="inBempl_codigo")String inBempl_codigo,
        @WebParam(name="format")String format
    ) throws SOAPEXception {
        com.besixplus.sisac.misc.Formatter tmpFormat = null;
        HttpServletRequest tmpRequest = (HttpServletRequest)
        wctx.getMessageContext().get(MessageContext.SERVLET_REQUEST);

```

```

com.besixplus.sisac.objects.Bsx_empleado tmpObj = new
com.besixplus.sisac.objects.Bsx_empleado();
tmpObj.setBEMPL_CODIGO(inBempl_codigo);
try{
    Connection con = ManagerConnection.getConnection();
    if(!com.besixplus.sisac.db.Bsx_objeto.isGrant(con,
    Thread.currentThread().getStackTrace()[1].getMethodName(),
    Thread.currentThread().getStackTrace()[1].getClassName(),
    tmpRequest.getUserPrincipal().getName(), 1)){
        con.close();
        throw new
        SOAPFaultException(SOAPFactory.newInstance().createFault
        ("Acceso no autorizado.", new
        QName("http://schemas.xmlsoap.org/soap/envelope/", Thread.c
        urrentThread().getStackTrace()[1].getClassName()+
        "+Thread.currentThread().getStackTrace()[1].getMethodName
        ())),);
    }
    con.setAutoCommit(!ManagerConnection.isDeployed());
    tmpObj = new
    com.besixplus.sisac.db.Bsx_empleado(tmpObj).select(con);
    ArrayList<com.besixplus.sisac.objects.Bsx_empleado> tmpArray =
    new ArrayList<com.besixplus.sisac.objects.Bsx_empleado>();
    tmpArray.add(tmpObj);
    tmpFormat = new com.besixplus.sisac.misc.Formatter(format,
    tmpArray);
    con.close();
} catch (SQLException inException){
    com.besixplus.sisac.db.SQLiteErrorHandler.errorHandler(inException);
    throw new
    SOAPFaultException(SOAPFactory.newInstance().createFault(inExcep
    tion.getMessage(), new
    QName("http://schemas.xmlsoap.org/soap/envelope/",
    Thread.currentThread().getStackTrace()[1].getClassName()+
    "+Thread.currentThread().getStackTrace()[1].getMethodName())));
} catch (Exception inException){
    com.besixplus.sisac.db.SQLiteErrorHandler.errorHandler(inException);
    throw new
    SOAPFaultException(SOAPFactory.newInstance().createFault(inExcep
    tion.getMessage(), new
    QName("http://schemas.xmlsoap.org/soap/envelope/",
    Thread.currentThread().getStackTrace()[1].getClassName()+
    "+Thread.currentThread().getStackTrace()[1].getMethodName())));
}
}
if (tmpObj != null)
    return tmpFormat.getData().toString();
return null;
}
}

```

### 3. Capa Cliente.

```

function
FrmBsx_empleado(IN_SENTENCIA_BSX_EMPLEADO,IN_RECORD_BSX_EMPLEADO){
/**
* Boton que permite almacenar la informacion de la ventana winFrmBsx_empleado
*/
var btnGuardarBsx_empleado = new Ext.Button( {

```

```

id:'btnGuardarBsx_empleado',
text:'Guardar',
iconCls:'iconGuardar',
listeners:{
    click:function(){
        if(!pnlBsx_empleado.getForm().isValid()){
            return;
        }
        pnlBsx_empleado.getForm().submit({
            url: URL_WS+"Bsx_empleadoSRV",
            waitMsg: 'Enviando formulario...!',
            success: function(fp, o){
                Ext.Msg.show({
                    title:tituloBsx_empleado,
                    msg: 'La informaci\u00f3n de Empleado ha sido
                    almacenada.',
                    buttons: Ext.Msg.OK,
                    icon: Ext.MessageBox.INFO
                });
                winFrmBsx_empleado.close();
            },
            failure:function(o){
                Ext.Msg.show({
                    title:tituloBsx_empleado,
                    msg: 'La informaci\u00f3n de Empleado no ha podido
                    ser almacenada.'+(r.message?r.message:r),
                    buttons: Ext.Msg.OK,
                    icon: Ext.MessageBox.ERROR
                });
            },
            params: {
                request:inSentenciaBsx_empleado,
                bempl_codigo:txtBempl_codigoEmpl.getValue(),
                bdepa_codigo: tmpCodigoDepartamento,
                bparr_codigo: tmpCodigoParroquia,
                btddid_codigo: cbxTipoDocumentoEmple.getValue(),
                bempl_documento_identificacion:
                txtBempl_documento_identificacion.getValue(),
                bempl_nombre: txtBempl_nombre.getValue(),
                bempl_apellido_materno:
                txtBempl_apellido_materno.getValue(),
                bempl_apellido_paterno:
                txtBempl_apellido_paterno.getValue(),
                bempl_sexo: cbxSexoEmple.getValue(),
                bempl_direccion: txtBempl_direccion.getValue(),
                bempl_nombre_foto: txtBempl_nombre_foto.getValue(),
                bempl_tipo: txtBempl_tipo.getValue(),
                bempl_fecha_nacimiento:
                dtBempl_fecha_nacimiento.getValue().format(TypeDateForm
                at.ISO8601Short),
                bempl_fecha_ingreso:
                dtBempl_fecha_ingreso.getValue().format(TypeDateFormat.I
                SO8601Short),
                bempl_estado_empleado:
                chkBempl_estado_empleado.getValue(),
                inBadtm_codigo:tmpCodigoAdjuntoPersona!=null?tmpCodig
                oAdjuntoPersona:null,

```

```

        inJSONContactos:
        grdBsx_empleado_contacto.getStore().getJsonData(),
        inJSONEliminarContactos:
        isEdit?tmpEliminarContactosEmple.toJSON():''
    }
    });
    });
}

```

#### 6.9.4 Implantación

Para la implementación del módulo de Gestión de Clientes y Seguridades en el Sistema Administrativo Contable de la empresa BESIXPLUS CIA LTDA depende de los módulos de Compras y Proveedores, Inventario y Facturación y el módulo de Configuración.

El sistema esta conformado por 4 módulos y cada uno de ellos han sido desarrollados bajo estándares de programación tanto a nivel de base de datos como de para el diseño de la interfaz.

Para no tener inconvenientes al momento del desarrollo se ha optado por configurar un servidor de control de versiones (Subversion 1.6) dentro la plataforma Linux, la cual permitirá mantener los archivos en una base de datos, en donde el usuario que modifique algún archivo debe confirmar los cambios para que los demás usuario puedan descargarse los cambios y no tener problemas de perdida de código.

SyncroSVN en un cliente bajo la plataforma Linux al igual que Tortoise SVN en Windows tienen la funcionalidad de trabajar con el servidor de versiones ayudando y evitando la pérdida de tiempo en la integración de los módulos.

Para la verificación del funcionamiento del módulo de Clientes y Seguridades se han hecho las pruebas necesarias a nivel de base de datos, servidor de aplicaciones para las seguridades y reportes para la salida de información.

Una vez hechas las pruebas necesarias del sistema se procedió a las capacitaciones correspondientes al módulo de Clientes y seguridades tanto a la persona



autorizada para administrar todo el sistema como a los usuarios finales que utilizaran el sistema.

En la capacitación para el administrador se profundizó el módulo de seguridades ya que el va ser quien administre todas las cuentas de usuario, mientras que a los usuarios encargados de realizar ventas se y al personal de caja solo se dio a conocer el funcionamiento de los créditos pendientes por cobrar.

### 6.9.5 Cronograma de implantación

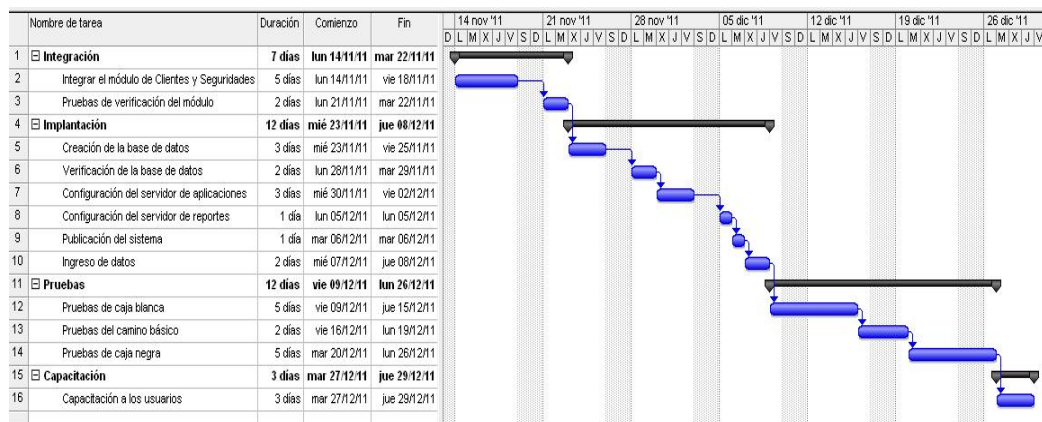


Figura 6.40 Cronograma de implantación.

### 6.9.6 Pruebas

Durante este proceso se debe poner en práctica todas las estrategias posibles para garantizar que el usuario inicial del sistema se encuentre libre de problemas.

#### 6.9.6.1 Pruebas de caja blanca

Mediantes las pruebas de caja blanca podremos examinar la estructura del módulo y las funciones internas de mismo, en donde:

- 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.

Para lograr un correcto funcionamiento del módulo se diseñaron casos de pruebas para evaluar la lógica del módulo y así asegurar que todos los procesos se ajusten a las especificaciones.

Entre las pruebas que se hicieron están las siguientes:

- Control de los campos de cada uno de los formularios de Ingreso, con los tipos de datos y formatos específicos a cada uno de ellos.

**Ejemplo:**

```
if (!pnlBsx_empleado.getForm().isValid()){
    return;
}
```



**Figura 6.41** Validación de campos en el formulario

- Pruebas al momento de almacenar datos de forma transaccional. Verificando que se hayan almacenado la información en las diferentes tablas que intervienen.

**Ejemplo:**

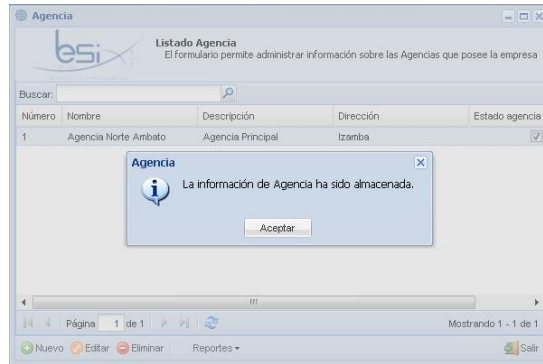
**Código para validar si el valor se almaceno o no en la base de datos**

```
function CallBackBsx_agencia(r){
    winFrmBsx_agencia.getEl().unmask();
    if(r=='true'){
        Ext.Msg.show({
            title: tituloBsx_agencia,
            msg: 'La informaci\u00f3n de Agencia ha sido almacenada.',
            buttons: Ext.Msg.OK,
            icon: Ext.MessageBox.INFO
        });winFrmBsx_agencia.close()
    }
```

```

    }else{
        Ext.Msg.show({
            title:tituloBsx_agencia,
            msg: 'La informaci\u00f3n de Agencia no ha podido ser
            almacenada. '+r.message?r.message:r),
            buttons: Ext.Msg.OK,
            icon: Ext.MessageBox.ERROR
        });
    }
}

```



**Figura 6.42** Verificación de almacenamiento en la base de datos

- Visualización de mensajes de alertas y errores de manejo de datos.

**Ejemplo:**

**Código de advertencia al eliminar un dato**

```

Ext.Msg.show({title:'Aviso', msg:'Est\u00e1 seguro de eliminar el registro seleccionado?',
buttons: Ext.Msg.YESNO,fn: SWRBsx_Agencia_contacto,icon:
Ext.MessageBox.QUESTION});
function SWRBsx_Agencia_contacto(btn){
    if (btn=='yes')
    {
        Ext.Msg.show({
            title:tituloBsx_agencia,
            msg: 'La informaci\u00f3n de contacto ha sido eliminada.',
            buttons: Ext.Msg.OK,
            icon: Ext.MessageBox.INFO
        });
        var seleccionado = grdBsx_agencia_contacto.getSelectionModel().getSelected();
        if(seleccionado != null ){
            grdBsx_agencia_contacto.getStore().remove(seleccionado);
        }
    }
}
}else{
    Ext.Msg.show({
        title:tituloBsx_agencia,
        msg: 'La informaci\u00f3n de contacto no ha podido ser eliminada.',
        buttons: Ext.Msg.OK,
        icon: Ext.MessageBox.ERROR
    });
}
}
}

```



Figura 6.43 Mensajes de advertencia

### 6.9.6.1.1 Pruebas del camino básico

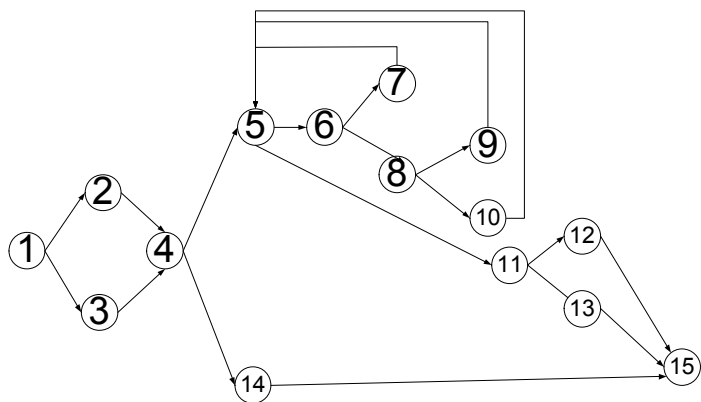
Estas pruebas se realizan para definir una serie de caminos básicos de ejecución, Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

### Grafo de flujo

#### Validación de la cédula/RUC

function validarCedula(inCedula)	
{	
var array = inCedula.split("");	1
var num = array.length;	
if ((num == 13 && array[10] == '0' && array[11] == '0' && array[12] == '1')    num == 10)	
{	
num = 10;	2
} else {	
return false;	
}	3
var total;	
var digito;	
var mult;	
var decena;	
var end;	
if(num == 10)	4
{	
total = 0;	
digito = (array[9]*1);	
for( var i = 0; i<(num-1);i++)	5
{	
mult = 0;	
if((i%2) != 0)	6

<pre>{   total = total + (array[i]*1); }</pre>	7
<pre>else {   mult = array[i]*2;   if( mult &gt; 9)</pre>	8
<pre>{   total = total + (mult - 9); }</pre>	9
<pre>else {   total = total + mult; } }</pre>	10
<pre>decena = total/10; decena = Math.floor(decena); decena = ( decena + 1 ) * 10; end = ( decena - total ) ; if((end == 10 &amp;&amp; digito == 0)   end == digito)</pre>	11
<pre>{   return true; }</pre>	12
<pre>else {   return false; } }</pre>	13
<pre>else {   return false; }</pre>	14
<pre>}</pre>	15



**Figura 6.44** Grafo Validación Cédula/RUC

### Complejidad Ciclomática

$$V(G) = a \text{ (aristas)} - n \text{ (nodos)} + 2$$

$$V(G) = 20 - 15 + 2 = 7$$

### Caminos Básicos

#1: 1, 2, 4, 14, 15

#2: 1, 2, 4, 5, 6, 7, 5, 11, 12, 15

#3: 1, 3, 4, 12, 15

**Para la comprobación de la prueba del camino básico se tomará el camino #2.**

En la variable **array** se va almacenar cada posición de la cadena que ingresa a la función, y en la variable **num** se almacena la longitud de la cadena, posteriormente se procede a verificar si la cadena ingresada es una cédula o un RUC para lo cual si la longitud total es igual a 13 y los tres últimos dígitos son **001** entonces esto quiere decir que es un RUC o si la longitud es 10 se procede a la validación del número de cédula.

Una vez culminado el ciclo for se comprobará si la cédula ingresada es correcta retornando el valor **true** y en caso de que la cédula sea errónea retornará un valor **false**.

En caso de que la longitud de la cadena ingresada no es **10** o **13** inmediatamente retorna **false** dando como error “Cédula o RUC no valido”

#### 6.9.6.2 Pruebas de Caja negra

Esta prueba se lleva a cabo sobre la interfaz del sistema en el módulo de Clientes y seguridades, por lo tanto estas son completamente indiferentes del comportamiento interno y a su estructura.

Con estas pruebas se pudo demostrar que.

- Las funciones del sitio son completamente operativas o factibles.
- El ingreso de usuario se realiza de forma adecuada y sencilla.
- El inicio de sesión y la culminación del mismo muestran la fiabilidad en la seguridad del sitio.
- Los gestores de datos presentan la información de forma correcta, rápida y adecuada.
- La integridad del sistema se mantiene.

Para lograr esto se debió probar en reiteradas veces las entradas y salidas de datos, el inicio y salida de sesión para probar las conexiones a la base de datos y la forma de responder en la visualización de la información.

## **6.10 Conclusiones y recomendaciones**

### **6.10.1 Conclusiones**

- Una vez desarrollado el módulo se puede concluir que es factible implementar dentro de cualquier plataforma, siendo el único requerimiento más indispensable el tener instalado JAVA en nuestro sistema operativo.
- Se puede concluir que mediante los diagramas UML se pudo comprender la manera de interactuar entre el usuario y el módulo de clientes y seguridades.
- El servidor de aplicaciones es muy bueno para la administración de seguridades, además de ser libre cuenta con una gran variedad de beneficios para las aplicaciones Web.
- El framework Ext JS es una librería adecuada para el desarrollo de la interfaz del lado del cliente haciendo que el usuario se sienta a gusto al interactuar con el sistema.
- El desarrollo de este módulo, especialmente el de seguridades reforzó los conocimientos sobre servidores aplicaciones y dominios de seguridad.

## 6.10.2 Recomendaciones

- Se recomienda implementar bajo la plataforma Linux para garantizar niveles de estabilidad e integridad en nuestro sistema. Además es una plataforma de software libre.
- Se recomienda administrar de una buena manera los privilegios de usuario restringiendo el acceso a formularios no autorizados. Para así evitar que usuarios no autorizados manipulen la información.
- A pesar que el servidor de aplicaciones es quien se encarga de la desplegar nuestro sistema y de salvaguardar la información mediante las seguridades implemetadas se recomienda realizar respaldos periódicamente de la base de datos para evitar algún tipo de inconveniente, además a esto se debe realizar mantenimientos preventivos y correctivos.
- Se recomienda al administrador crear usuarios para el sistema únicamente al personal que este capacitado; ademas es importante proveerlos del manual de usuario en caso de alguna duda sobre el sistema.
- Para la manipulación de sistema en general se recomienda utilizar los navegadores Firefox o Google Chrome para poder aprovechar al máximo el desempeño del sistema, ya que en ciertos navegadores suele alteras el diseño de ciertos componentes.

## 6.11 Bibliografía

### 6.11.1 Fuentes Bibliográficas

1. SILBERSCHATZ, Abraham (2006). *“Fundamentos de Base de Datos”* Edicion5, Mc Graw Hill.

### 6.11.2 Linkografía

2. José L. Hevia (2000). *Servidor de Base de datos*. Consultado en (Septiembre del 2010) en: <http://www.idg.es/dealerworld/Servidores-de-bases-de-datos.Grande>



s-proveedores-de-informacion-para-todo-tipo-de-usuarios/seccion-/articulo-106633

3. Damián Pérez Valdés. *¿Que son las Bases de Datos?* Consultado en (Septiembre del 2010) en: <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>
4. Sandoval Jimenez Hector. *SGBD libres*. Consultado en (Septiembre del 2010) en: <http://admin-info.galeon.com/gestor.pdf>
5. Wikipedia org (2010). *PostgreSQL*. Consultado en (Septiembre del 2010) en: <http://es.wikipedia.org/wiki/PostgreSQL>
6. Wikipedia org (2010). *MySQL*. Consultado en (Septiembre del 2010) en: <http://es.wikipedia.org/wiki/MySQL>
7. Wikipedia org (2010). *Firebird*. Consultado en (Septiembre del 2010) en: <http://es.wikipedia.org/wiki/Firebird>
8. Wikipedia org (2010). *Servidor de Aplicaciones*. Consultado en (Septiembre del 2010) en: [http://es.wikipedia.org/wiki/Servidor\\_de\\_aplicaciones](http://es.wikipedia.org/wiki/Servidor_de_aplicaciones)
9. Wikipedia org (2010). *JOnAS*. Consultado en (Septiembre del 2010) en: <http://es.wikipedia.org/wiki/JOnAS>
10. Wikipedia org (2010). *JBOSS*. Consultado en (Septiembre del 2010) en: <http://es.wikipedia.org/wiki/JBoss>
11. Wikipedia org (2010). *GlassFish*. Consultado en (Septiembre del 2010) en: <http://es.wikipedia.org/wiki/GlassFish>
12. OPEN CERTIAC (2009). *Jasper Server*. Consultado en (Septiembre del 2010) en: <http://www.opencertiac.org/jasperserver.html>
13. Sin Autor (s.f). *JasperReport*. Consultado en (Septiembre del 2010) en: <http://www.ultraforos.com/foro/ayudas-tutoriales/34497-jasper-report-e-ireport.html>
14. Wikipedia org (2010). *JAVA EE*. Consultado en (Septiembre del 2010) en: [http://es.wikipedia.org/wiki/Java\\_EE](http://es.wikipedia.org/wiki/Java_EE)
15. Sin Autor (s.f.). *Lenguaje de programación JAVA*. Consultado en (Septiembre del 2010) en: <http://es.scribd.com/doc/81119793/lenguajes-de-prog>
16. Juan Gutiérrez. Universidad de Valencia (2004). *Eclipse*. Consultado en (Septiembre del 2010) en: [http://www.uv.es/~jgutierr/MySQL\\_Java/TutorialEclip](http://www.uv.es/~jgutierr/MySQL_Java/TutorialEclip)

se.pdf

17. Wikipedia org (2010). *NetBeans*. Consultado en (Septiembre del 2010) en: <http://es.wikipedia.org/wiki/NetBeans>
18. Gabriel Gonzales (s.f.). *Servicios Web*. Consultado en (Septiembre del 2010) en: <http://kalistog.wordpress.com/servicios-web/>
19. GXTechnical. *Web Services*. Consultado en (Septiembre del 2010) en: [http://www.gxtechnical.com/gxdlsp/pub/genexus/internet/technicalpapers/web\\_services.htm](http://www.gxtechnical.com/gxdlsp/pub/genexus/internet/technicalpapers/web_services.htm)
20. Wikipedia org (s.f.). *UDDI*. Consultado en (Septiembre del 2010) en: [http://es.wikipedia.org/wiki/Servicio\\_web](http://es.wikipedia.org/wiki/Servicio_web)
21. Antonio Barco (s.f.). *Arquitectura SOA*. Consultado en (Septiembre del 2010) en: <http://arquitecturaorientadaaservicios.blogspot.com/>
22. Universidad Central de Venezuela (s.f.). *Arquitectura SOA*. Consultado en (Septiembre del 2010) en: [http://kuainasi.ciens.ucv.ve/adsi2010-2/business\\_process\\_management/SOA.HTML](http://kuainasi.ciens.ucv.ve/adsi2010-2/business_process_management/SOA.HTML)
23. Sin Autor (s.f.). *Arquitectura orientada a servicios SOA*. Consultado en (Septiembre del 2010) en: <http://temariotic.wikidot.com/arquitectura-soa>
24. Antonio Barcos (s.f.). *Capas que define SOA*. Consultado en (Septiembre del 2010) en: <http://arquitecturaorientadaaservicios.blogspot.com/>
25. Wikipedia org (2010). *Arquitectura orientada a servicios*. Consultada en (Septiembre del 2010) en: [http://es.wikipedia.org/wiki/Arquitectura\\_orientada\\_a\\_servicios](http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios)
26. Damián Pérez Valdés (s.f.). *Los diferentes lenguajes de programación para la web*. Consultado en (Septiembre del 2010) en: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>
27. Miguel Ángel Álvarez (s.f.). *Tecnologías XML*. Consultado en (Septiembre del 2010) en: <http://www.desarrolloweb.com/articulos/449.php>
28. Jorge Iván Meza Martínez. (s.f.). *Prototype*. Consultado en (Septiembre del 2010) en: <http://blog.jorgeivanmeza.com/2008/09/introduccion-a-prototypejs/>
29. Sencha (s.f.). *Librería Ext*. Consultado en (Septiembre del 2010) en: <http://extjs.es/>

30. Wikipedia org (2010). *JQuery*. Consultado en (Septiembre del 2010) en: <http://es.wikipedia.org/wiki/JQuery>
31. Wikipedia org (2010). *Google web Toolkit*. Consultado en (Septiembre del 2010) en: [http://es.wikipedia.org/wiki/Google\\_Web\\_Toolkit](http://es.wikipedia.org/wiki/Google_Web_Toolkit)
32. Yahoo! Inc (s.f.). *YUI*. Traducido Consultado en (Septiembre del 2010) en: <http://developer.yahoo.com/yui/>
33. Wikipedia org (2010). *Customer relationship management CRM*. Consultado en (Septiembre del 2010) en: [http://es.wikipedia.org/wiki/Customer\\_relationship\\_Management](http://es.wikipedia.org/wiki/Customer_relationship_Management)
34. Sin Autor (s.f). *CRM Gestión de Clientes con SugarCRM*. Consultado en (septiembre del 2010) en: <http://www.bairesbusiness.com/pdf/SugarCRM.pdf>
35. Sin Autor (s.f). *Seguridades en la Base de datos*. Consultado en (Septiembre del 2010) en: <http://es.scribd.com/doc/14870070/Seguridad-en-Base-de-Datos>
36. Sin Autor (s.f). *Seguridades en sitios Web*. Consultado en (Septiembre del 2010) en: <http://www.slideshare.net/sanfer08/seguridad-en-sitios-web>
37. Besixplus CIA LTDA (s.f). *Besixplus CIA LTDA*. Consultado en (Septiembre del 2010) en: <http://www.besixplus.com>
38. Emilio María Ramírez Márquez (2010). *Administración Contable*. Consultado en (Septiembre del 2010) en: <http://www.suite101.net/content/la-administracion-contable-en-las-empresas-a21541>
39. Sin Autor (s.f). *Documentos comerciales*. Consultado en (Septiembre del 2010) en: <http://html.rincondelvago.com/documentos-comerciales.html>
40. Sin Autor (s.f). *Tipos de Documentos*. Consultado en (Septiembre del 2010) en: <http://www.monografias.com/trabajos11/ladocont/ladocont.shtml>

## GLOSARIO DE TÉRMINOS.

**SGBD:** Sistema de Gestión de la Base de Datos. Es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

**Licencia BSD:** (*Berkeley Software Distribution*). Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

**OPEN SOURCE:** Código abierto 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 éticas y morales las cuales destacan en el llamado software libre.

**PGDG:** PostgreSQL Global Development Group. Grupo de Desarrollo Global de PostgreSQL.

**MySQL AB:** Es una compañía de software fundada en 1995, creadora del sistema administrador de bases de datos relacionales MySQL, y una de las más grandes empresas de software libre del mundo.

**RDBMS:** Sistema de Gestión de Base de Datos Relacional o SGBDR para base de datos relacionales.

**HP\_UX:** Es la versión de Unix desarrollada y mantenida por Hewlett-Packard desde 1983, ejecutable típicamente sobre procesadores HPPA RISC y en sus últimas versiones sobre Intel Itanium (arquitectura Intel de 64 bits).

**FreeBSD:** Es un sistema operativo libre para computadoras basado en las CPU de arquitectura Intel, incluyendo procesadores Intel 80386, Intel 80486 (versiones SX y DX), y Pentium.

**Mac OS:** Es un sistema operativo creado por Apple para su línea de computadoras Macintosh. Es conocido por haber sido el primer sistema dirigido al gran público en contar con una interfaz gráfica compuesta por la interacción del *mouse* con ventanas, Icono y menús.

**Solaris:** Solaris es un sistema operativo de tipo Unix desarrollado desde 1992 inicialmente por Sun Microsystems y actualmente por Oracle Corporation como sucesor de SunOS. Es un sistema certificado oficialmente como versión de Unix.

**TCP/IP:** Se esconde uno de los protocolos más usados del mundo, debido a que es el más usado por Internet y está muy extendido en el sistema operativo UNIX.

**JAVA:** Es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems.

**J2EE:** (Java 2 Enterprise Edition) define un estándar para el desarrollo de aplicaciones empresariales multicapa diseñado por SUN Microsystems.

**Licencia CDDL:** Es una licencia de código abierto (OSI) y libre, producida por Sun Microsystems, basada en la Mozilla Public License o MPL.

**GNU GPL:** Es una licencia creada por la Free Software Foundation en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

**OLAP:** Procesamiento analítico en línea (*On-Line Analytical Processing*). Es una solución utilizada en el campo de la llamada Inteligencia empresarial (o *Business Intelligence*) cuyo objetivo es agilizar la consulta de grandes cantidades de datos.

**MDX:** La consulta de MDX (Expresiones multidimensionales) básica es la instrucción SELECT: la consulta usada con más frecuencia en MDX.

**JAVA EE:** Java Platform, Enterprise Edition o Java EE (o también conocido como Java 2 Platform, Enterprise Edition o J2EE hasta la versión 1.4), es una plataforma de programación.

**PDF:** Es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems.

**HTML: (HyperText Markup Language)** Lenguaje de Marcación de Hipertexto, es un lenguaje se utiliza comúnmente para establecer la estructura y contenido de un sitio web, tanto de texto, objetos e imágenes.

**XLS:** Hojas de cálculo creada en Microsoft Excel.

**CSV: (*comma-separated values*)** son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas y las filas por saltos de línea.

**XML:** ('lenguaje de marcas extensible'), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

**W3C: (*World Wide Web Consortium*)** Es un consorcio internacional que produce recomendaciones para la World Wide Web.

**CVS:** Concurrent Versioning System, es una aplicación informática que implementa un sistema de control de versiones.

**IDE: (*Integrated Development Environment*)** Entorno de desarrollo integrado es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

**SOAP: (*Simple Object Access Protocol*)** es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

**HTTP:** Hypertext Transfer Protocol o HTTP (Protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web.

**WSDL:** (*Web Services Description Language*) Formato XML que se utiliza para describir servicios Web

**UDDI:** (*Universal Description, Discovery and Integration*) Es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.

**SOA:** La arquitectura orientada a servicios de cliente (Service Oriented Architecture), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.

**XSS:** *Cross-site scripting* es un tipo de inseguridad informática o agujero de seguridad típico de las aplicaciones Web, que permite a una tercera parte inyectar código JavaScript en páginas web vistas por el usuario, evitando medidas de control como la Política del mismo origen.

**AJAX:** Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas.

**CSS:** Hojas de estilo en cascada viene del inglés Cascading Style Sheets, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

**DOM:** Modelo de Objetos del Documento. Es una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML

**JSON:** *JavaScript Object Notation*, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

**GWT:** Google Web Toolkit es un *framework* creado por Google que permite ocultar la complejidad de varios aspectos de la tecnologíaAJAX.

**DHTML:** (*Dynamic HTML*) designa el conjunto de técnicas que permiten crear sitios web interactivos utilizando una combinación de lenguaje HTML estático, un lenguaje interpretado en el lado del cliente (como JavaScript), el lenguaje de hojas de estilo en cascada (CSS) y la jerarquía de objetos de un DOM.

**CRM:** La administración basada en la relación con los clientes.

**HTTPS:** HTTPS viene a ser el Protocolo Seguro de Transferencia de Hipertexto y está obviamente basado en el antes mencionado HTTP pero con la particularidad de utilizar un cifrado basado en la Secure Socket Layers mas conocidas como SSL y así crear un canal de transferencia cifrado con el que obviamente aumenta la seguridad en el tráfico de información en comparación al protocolo HTTP común.

**JDK:** (*Java Development Kit*) Es un software que provee herramientas de desarrollo para la creación de programas en Java.

**JQUERY:** Es una biblioteca de JavaScript, permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

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

**SOFTWARE:** Se conoce como software al equipamiento lógico o soporte lógico de un sistema informático; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.



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

**WEB:** El sistema de documentos (o páginas web) interconectados por enlaces de hipertexto, disponibles en Internet.

# Anexos



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**

Entrevistado: \_\_\_\_\_

Fecha : \_\_\_\_\_

Entrevista enfocada al recurso humano de la empresa BESIXPLUS CIA LTDA.

**Nota:** la información que se recopila es para uso exclusivo del trabajo de investigación a efectuarse en la empresa BESIXPLUS CIA LTDA. por lo tanto se recomienda que la información sea verídica ya que la misma será utilizada para la creación del módulo.

**Cuestionario**

1. ¿Qué tipo de información se maneja en la empresa con respecto a los clientes?  
.....
2. ¿Qué tipo de procesos se realizan con la información de los clientes?  
.....
3. ¿Se realiza algún tipo de transacciones con los clientes? ¿Explique cuáles?  
.....
4. ¿Cuál es el proceso para realizar una transacción con el cliente?  
.....
5. ¿Qué datos son necesarios para realizar este tipo de transacción?  
.....
6. ¿Qué cuentas se generan al momento de realizar una venta?  
.....
7. ¿Qué procesos se realizan luego de haberse emitido la factura?  
.....
8. ¿Cuáles son las formas de pago que el cliente puede realizar?

- .....
- .....
9. ¿Cuál es el proceso de cobro por la venta de quipos o prestación de un servicio?
- .....
- .....
10. ¿Qué documentos se emiten al momento de realizar un cobro?
- .....
- .....
11. ¿Una vez realizado el cobro que cuentas interactúan? ¿De qué forma?
- .....
- .....
12. ¿Qué sucede si un cliente se retrasa en los pago?
- .....
- .....
13. ¿Hasta cuantos meses consecutivos un cliente puede retrasarse en los pagos?
- .....
- .....
14. ¿En caso de que un cliente se retrase en los pagos la tasa de interés sube en que porcentaje?
- .....
- .....
15. ¿En el caso de que un accionista o un empleado de la empresa adquiriera un equipo cual es el proceso de cobro?
- .....
- .....
16. ¿Esas cuentas por cobrar se debitan del sueldo mensual de los empleados?
- .....
- .....
17. ¿Cuándo y cada cuanto tiempo se debe realizar un balance?
- .....
- .....
18. ¿Cuándo y cómo realiza un reporte?
- .....
- .....
19. ¿Según que parámetros se debe realizar un reporte?
- .....
- .....
20. ¿Cada qué tiempo se debe realizar un reporte?
- .....
- .....
21. ¿El sistema utilizado actualmente cuanta con las seguridades suficientes?
- .....
- .....
22. ¿Cada usuario ingresa a sus propios datos o también puede acceder a la información de otros usuarios?
- .....
- .....

23. ¿Q tipo de permisos se le debe dar a cada uno de los usuarios?

.....  
.....

24. ¿Quiénes son los únicos que pueden acceder y manipular todo tipo de información?

.....  
.....

---

Entrevistado

## **Manual de instalación y configuración**

Para la instalación del sistema y por ende el módulo de Gestión de Clientes y Seguridades es necesario que contar con lo siguiente:

### **Antecedentes.**

- Servidor de aplicaciones JBOSS 5.1.0.
- Motor de base de datos PostgreSQL 8.4 en adelante.
- Tener instalado JAVA JDK y JRE 6 o superiores en nuestro servidor (Linux, Windows, entro otros).
- Jasper Server. Servidor de reportes.

### **Instalación**

Hay que tener en cuenta que para la instalación del Módulo y del sistema en general hay que contar con todos los recursos mencionados anteriormente. Este proceso de instalación se lo hace solamente en el servidor, ya que en la parte del cliente el requisito principal es tener un navegador que soporte Javascript y tener instalado o configurado el JDK 6 o superiores.

### **Primer Paso: Base de Datos**

Crear la base de datos en nuestro servidor, para lo cual debemos tener instalado el gestor de base de datos PostgreSQL 8.4 o superiores, además instalaremos PgAdmin 3 para poder manipular nuestra base de datos.

A nuestra base de datos la llamaremos **sisacdb**, en la cual vamos a definir un esquema llamado **sch\_bsx** en la cual vamos a crear nuestras tablas con el script de nuestro modelado.

### **Segundo Paso: Servidor de Aplicaciones**

Una vez creada nuestra base de datos procedemos a configurar nuestro servidor de aplicaciones para lo cual accedemos al directorio `${JBOSS_HOME}/conf/` en

donde vamos a crear nuestro dominio de seguridades en el archivo **login-config.xml**.

```
<!--DOMINIO DE SEGURIDAD PARA EL SAD PARA LA EMPRESA BESIXPLUS CIA LTDA -->

<application-policy name="JBossSISACWS">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.DatabaseServerLoginModule" flag="required">
      <module-option name="dsJndiName">java:jdbc/sisac</module-option>
      <module-option name="principalsQuery">SELECT BUSUA_CLAVE FROM
      SCH_BSX.BSX_USUARIO WHERE BUSUA_ESTADO AND
      BUSUA_ESTADO_USUARIO AND BUSUA_ALIAS = ?</module-option>
      <module-option name="rolesQuery">SELECT R.BROLE_NOMBRE, 'Roles' FROM
      SCH_BSX.BSX_ROL R INNER JOIN SCH_BSX.BSX_USUARIO_ROL UR ON
      UR.BROLE_CODIGO = R.BROLE_CODIGO AND UR.BUROL_ESTADO INNER
      JOIN SCH_BSX.BSX_USUARIO U ON U.BUSUA_CODIGO = UR.BUSUA_CODIGO
      WHERE U.BUSUA_ESTADO AND U.BUSUA_ESTADO_USUARIO AND
      U.BUSUA_ALIAS = ?</module-option>
      <module-option name="hashAlgorithm">MD5</module-option>
      <module-option name="hashEncoding">base64</module-option>
    </login-module>
  </authentication>
</application-policy>
```

Para que el servidor de aplicaciones JBoss sea capaz de gestionar este sistema de seguridad debe manejar almacenes de claves Java y certificados digitales.

**Creación del Almacén de claves:** Para crear un almacén de claves estándar JKS (Java KeyStore) y un par de claves asimétricas (pública/privada). Para ello utilizaremos la herramienta administrativa **keytool**: y para ellos no ubicamos dentro del directorio `/${JBASS_HOME}/deploy/conf/`

```
[root@linux-emily emily]# keytool -genkey -alias sisac -keyalg RSA -keystore
./sisackey.keystore
Escriba la contraseña del almacén de claves:
Volver a escribir la contraseña nueva:
¿Cuáles son su nombre y su apellido?
 [Unknown]: Damián Laura
¿Cuál es el nombre de su unidad de organización?
 [Unknown]: UTA
¿Cuál es el nombre de su organización?
 [Unknown]: UTA
¿Cuál es el nombre de su ciudad o localidad?
 [Unknown]: Ambato
¿Cuál es el nombre de su estado o provincia?
 [Unknown]: Tungurahua
¿Cuál es el código de país de dos letras de la unidad?
 [Unknown]: ec
```

¿Es correcto CN=Damián Laura, OU=UTA, O=UTA, L=Ambato, ST=Tungurahua,  
C=ec?  
[no]: si  
Escriba la contraseña clave para <sisac>  
(INTRO si es la misma contraseña que la del almacén de claves):

En donde **sisac** va ser nuestra entrada de tipo RSA(Algoritmo de cifrado de clave pública).

También es necesario definir el certificado para poder identificar a nuestro sitio web como seguro, para lo cual ejecutamos:

```
[root@linux-emily emily]# keytool -certreq -keystore ./sisackey.keystore -alias sisac -file  
sisacCsr.csr
```

A continuación vamos a definir el conector seguro para **http**, para lo cual creamos el conector en el siguiente archivo de configuración

`${JBOSS_HOME}/deploy/jbossweb.sar/server.xml`:

```
<!-- SSL/TLS Connector configuration using the admin devl guide keystore-->  
<Connector protocol="HTTP/1.1" SSLEnabled="true"  
    port="8443" address="{jboss.bind.address}"  
    scheme="https" secure="true" clientAuth="false"  
    keystoreFile="{jboss.server.home.dir}/conf/sisackey.keystore"  
    keystorePass="sisac2011" sslProtocol = "TLS" />
```

En donde `keystoreFile` va nuestro almacén de datos antes creado y en `keystorePass` colocamos la clave de nuestro almacén de datos

### Tercer Paso: Subir nuestra aplicación al servidor de aplicaciones

Dentro del directorio de nuestro servidor de aplicaciones `${JBOSS_HOME}/deploy/` colocar nuestra aplicación Cliente **SISAC\_UI.war** y la aplicación de servicios web **sisac.war**

Una vez colocado nuestras aplicaciones en el directorio de despliegue de nuestro servidor de aplicaciones creamos un archivo para el pool de conexiones. El archivo de configuración lo llamaremos **postgresql-ds.xml** en donde realizaremos nuestra conexión a la base de datos.



```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>

  <local-tx-datasource>
    <!-- This connection pool will be bound into JNDI with the name
         "java:/MyDB" -->
    <jndi-name>jdbc/sisac</jndi-name>
    <!-- jdbc:postgresql://[servername]:[port]/[database name] -->
    <connection-url>jdbc:postgresql://localhost:5432/sisacdb</connection-url>
    <driver-class>org.postgresql.Driver</driver-class>
    <user-name>postgres</user-name>
    <password>damian</password>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>100</max-pool-size>
    <track-statements>>false</track-statements>
  </local-tx-datasource>
</datasources>
```

#### **Cuarto paso: Ejecución.**

Una vez terminado con las configuraciones ponemos en ejecución nuestra aplicación para lo cual accedemos a una consola de comandos en la cual ingresamos a l directorio del servidor de aplicaciones a la carpeta **bin**. En donde vamos a levantar nuestro servidor.

```
root@linux-emily emily]# cd /jboss-5.1.0.GA/bin
root@linux-emily emily]# ./run.sh -b 0.0.0.0
```

Al levantarse completamente el servidor de aplicaciones abrimos el navegador y colocamos la dirección siguiente: <http://localhost:8080/sisac> en donde el dominio de seguridad redireccionará a <https://localhost:8443/sisac>

## Manual de Usuario

### SISAC(Sistema Administrativo Contable) Módulo de Clientes y Seguridades

#### Introducción

Este manual es una guía para el usuario que manipula el sistema, ya que consta de información necesaria para inducir al usuario el manejo correcto sobre el módulo de Clientes y seguridades.

Este sistema y por ende el módulo de Clientes y Seguridades se lo puede utilizar desde cualquier punto de nuestra intranet y bajo cualquier plataforma, siendo los requisitos tener un navegador que soporte javascript y tener instalado el JAVA JDK y JRE 6 o superiores.

#### Objetivo:

Dar a conocer a los usuarios finales las características y las formas de funcionamiento del módulo de Clientes y Seguridades del Sistema Administrativo Contable (SISAC).

#### A quién va dirigido este manual

Este manual está orientado a los usuarios finales que manipularan el sistema y el módulo de Clientes y Seguridades entre ellos tenemos a los usuarios:

**Administrador.-** Es el usuario que tiene acceso a todos los módulos del sistema entre ellos tenemos el módulo de Seguridades ya que en ella se va a poder llevar el control de las sesiones, usuario, roles, perfiles, entre otras.

Además en los diferentes módulos el usuario Administrador podrá acceder a todas las opciones y podrá realizar las siguientes funciones: consultar, ingresar, modificar, eliminar, y generar reportes.

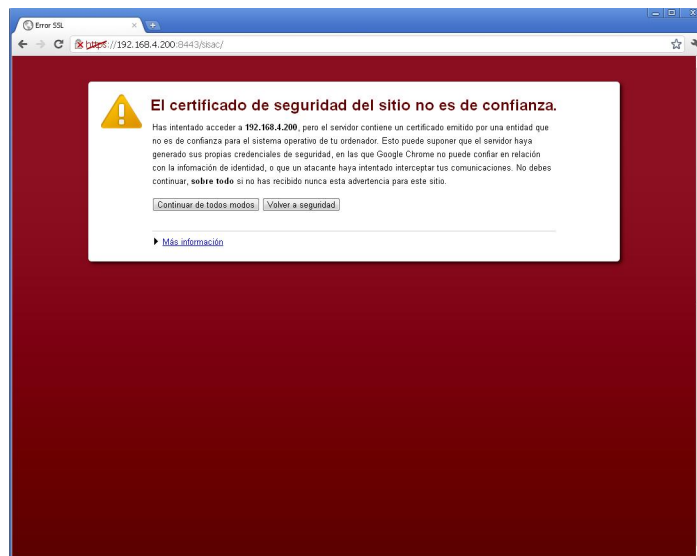
**Usuarios en general.-** Son los usuario que tienen acceso a ciertas partes del sistema. En donde el usuario Administrador es quien otorga los permisos necesarios para su acceso y manipulación de la misma.

Para el módulo de Clientes un usuario con los permisos necesarios de acceso podrá realizar las siguientes funciones: insertar, modificar, eliminar y generar reportes.

### Ingreso al sistema

Para poder ingresar al sistema primero hay que haber instalado y configurado el servidor.

Una vez configurado el servidor ya podemos acceder a nuestro sistema, para lo cual en nuestro navegador accedemos a la dirección: <http://IP-SERVER:8080/sisac> en donde **IP-SERVER** es la dirección IP de nuestro servidor.



Como podemos ver se nos redirecciona a <https://IP-SERVER:8443/sisac> lo que significa que esta bajo el protocolo de seguridades el cual esta dentro de nuestro dominio de seguridades configurado en nuestro servidor de aplicaciones. En donde nuestra información va a viajar de un lugar a otro de una forma segura.

En esta pantalla no pide aceptar el certificado de seguridades de nuestro sitio para lo cual presionamos **Acceder de todos modos**, luego a esto aparecerá nuestra pantalla de validación de usuario.

## Validación de usuario



Validación de usuarios

Usuario:

Contraseña:

Recordarme

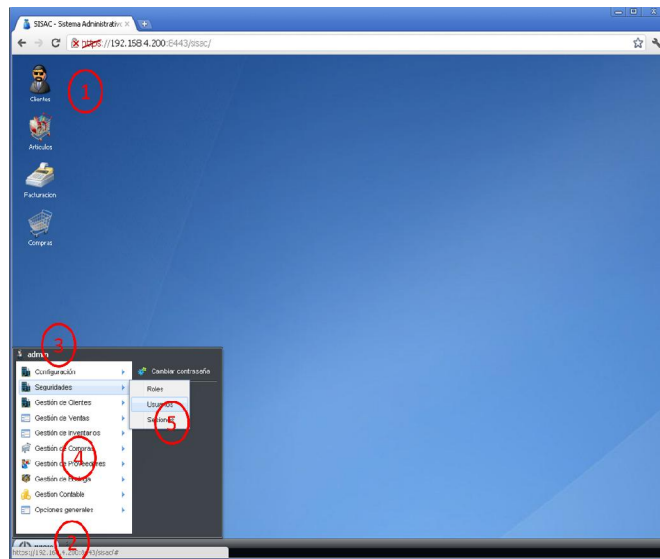
Este formulario de validación de usuario sirve para realizar el proceso de ingreso al sistema para lo cual se debe ingresar el alias de usuario con la respectiva contraseña, adicional a esto tenemos un componente check con la opción de recordar el usuario y la contraseña almacenando en la cookies de nuestro navegador.

Una vez ingresado los datos presionamos el botón Conectar para realizar el proceso de ingreso al sistema. En caso de que los datos no hayan sido ingresados correctamente se nos presentara un mensaje de error.

USUARIO / CONTRASEÑA INVALIDOS

## Pantalla principal del sistema

Si los datos ingresados en la pantalla de validación de usuario fueron correctos se nos presentara la pantalla principal del sistema:



Esta pantalla principal consta de lo siguiente:

1. Escritorio del sistema en donde están los accesos directos hacia los módulos principales del mismo.
2. Menú Inicio y barra de tareas.
3. Alias des usuario conectado.
4. Items principales del sistema
5. Sub Items.

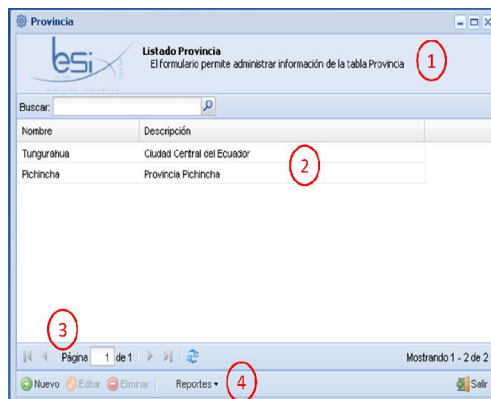
#### Simbolizado de Iconos y Botones.

Icono	Significado
	Botón nuevo registro.
	Botón editar registro.
	Botón eliminar registro.
	Botón Salir del formulario Listado.
	Botón Guardar
	Botón Cancelar
	Botón buscar.
	Ir a la primera página.
	Página anterior
	Página siguiente.
	Ir a la página final

#### Administración de Formularios de Configuración

Permite administrar todas las tablas de configuración para que el sistema funcione de la mejor manera. Entre las más importantes tenemos:

1.- **Provincias:** Almacena todas las provincias.



En este formular de Listado podemos visualizar lo siguiente:

- 1.- Encabezado: Logo de la empresa y una descripción del formulario.
- 2.- Listado de Registros: Muestra todos los registros almacenados en la base de datos.
- 3.- Barra de navegación: facilita para la navegación entre páginas.
- 4.- Barra de opciones: Permite realizar acciones que posee el formulario.

## Ingreso

Para poder ingresar un registro damos click en el botón Nuevo.

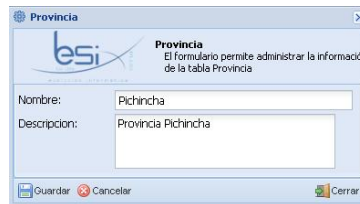


En esta pantalla tenemos de igual manera el encabezado, el área de los campos de ingreso y la barra de botones de acción.

Una vez ingresado los datos correctos procedemos a presionar el botón guardar, en caso de que no haya sucedido ningún error durante el proceso de guardado se nos mostrará el mensaje de Datos almacenados correctamente o caso contrario el error producido al momento de guardar.

## Modificación:

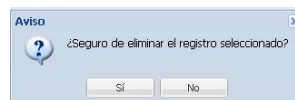
Al dar click en el botón **Editar** aparecerá la pantalla con los datos del registro seleccionado.



Para guardar los datos modificados presionamos el botón **Guardar**, o si no deseamos realizar los cambios presionamos en el botón **Cancelar** y posteriormente presionamos el botón **Cerrar**

## Eliminación:

Al presionar el botón **Eliminar** se nos mostrará un mensaje de advertencia:



## 2.- Empleados: Empleados de la empresa.

Empleado  
El formulario permite administrar la información de la tabla Empleado

**Datos personales**

Tipo Documento: Seleccione un Tipo de Documento  
Nombre:  
Apellido paterno: Materno:  
Fecha nacimiento: 05/12/2011 Sexo: Sexo  
Parroquia:  
Dirección:

**Fotografía**

Foto:

**Ubicación**

Departamento: Fecha ingreso: 05/12/2011 Estado Emp:

**Contactos**

Tipo contacto	Contacto	Descripción
---------------	----------	-------------

Guardar Cancelar Cerrar

En este formulario permite ingresar empleados de la empresa, en donde podemos ingresar los datos personales, fotografía y datos de contacto del empleado. Para poder colocar la fotografía del empleado presionamos el botón buscar foto y seleccionamos nuestra imagen.

'. The 'Contactos' section is empty. At the bottom, there are 'Guardar', 'Cancelar', and 'Cerrar' buttons."/>

Empleado  
El formulario permite administrar la información de la tabla Empleado

**Datos personales**

Tipo Documento: Cédula 1804161949  
Nombre: José Luis  
Apellido paterno: Turja Materno: Peralta  
Fecha nacimiento: 05/12/1987 Sexo: MASCULINO  
Parroquia: Izamba  
Dirección: Junto a la Iglesia de Izamba

**Fotografía**

Foto: C:\yakepath\12

**Ubicación**

Departamento: Ventas Fecha ingreso: 05/12/2010 Estado Emp:

**Contactos**

Tipo contacto	Contacto	Descripción
---------------	----------	-------------

Guardar Cancelar Cerrar

Finalmente nos queda guardar para lo cual presionamos el botón guardar.

## 3.- Clientes.

Cliente  
El formulario permite administrar la información de la tabla Cliente

**Datos personales**

Tipo Contribuyente:  Persona Jurídica  Persona Natural N°:  
Tipo Documento: Seleccione un Tipo de Contacto N° Doc:  
Empresa:  
Parroquia: Dirección:  
Tipo cliente:

**Contactos**

Tipo contacto	Contacto	Descripción
---------------	----------	-------------

Guardar Cancelar Cerrar

Este formulario permite ingresar un nuevo cliente, para lo cual se debe ingresar los datos personales del cliente, contactos, entre otros.

También se puede ingresar el empleado quien estará a cargo de este cliente, el crédito máximo que se le puede al cliente para que pueda hacer una compra a crédito.

El saldo es un valor que se va ir actualizando de acuerdo a las compras a crédito que realiza el cliente.

The screenshot shows a web application window titled "Cliente" with the following content:

- Header:** Logo "bsi" and text "Cliente. El formulario permite administrar la información de la tabla Cliente".
- Datos personales:**
  - Tipo Contribuyente:  Persona Jurídica  Persona Natural. N°: [input field]
  - Tipo Documento: [dropdown menu showing "RUC"]. N° Doc: [input field with value "1804161949001"]
  - Empresa: [input field with value "EEASA"]
  - Parroquia: [input field with value "Izamba"]
  - Dirección: [input field with value "Redondel de las focas"]
  - Tipo cliente: [input field with value "Cliente fijo"]
- Contatos / Datos Varios:**
  - Empleado a cargo: [input field]
  - Estado cliente:
  - Credito máximo: [input field with value "0"]
  - Saldo: [input field with value "0"]
  - Descripción: [text area]
- Footer:** "Guardar" (with floppy disk icon), "Cancelar" (with red X icon), and "Cerrar" (with window icon).