

# Índice General

<b>1.1</b>	<b>Introducción</b>	<b>4</b>
1.1.1	Misión	4
1.1.2	Visión	5
<b>1.2</b>	<b>Antecedentes</b>	<b>5</b>
<b>1.3</b>	<b>Solución Propuesta.</b>	<b>6</b>
<b>1.4</b>	<b>Objetivo General.</b>	<b>8</b>
<b>1.5</b>	<b>Objetivos Específicos.</b>	<b>8</b>
<b>1.6</b>	<b>Alcances</b>	<b>9</b>
<b>1.7</b>	<b>FODA:</b>	<b>10</b>
<b>1.8</b>	<b>Metodología</b>	<b>11</b>
1.8.1	Diagramas UML	11
<b>1.9</b>	<b>Estudio de Factibilidad</b>	<b>12</b>
1.9.1	Estudio de Mercado	12
1.9.1.1	Introducción.	12
1.9.1.2	Mercado Meta:	13
1.9.1.3	Identificación de la competencia	13
<b>1.10</b>	<b>Estudio técnico</b>	<b>14</b>
1.10.1	Hardware.	14
1.10.2	Software.	15
<b>1.11</b>	<b>Recursos</b>	<b>15</b>
1.11.1	Recurso Físico	16
1.11.2	Recurso Humano	16
1.11.3	Recurso Tecnológico	18
1.11.3.1	Hardware	18
1.11.3.2	Software	19
<b>1.12</b>	<b>Presupuesto</b>	<b>19</b>
1.12.1	Costo de Hardware	19
1.12.2	Costo de Software	19
<b>1.13</b>	<b>Estructura del Sistema</b>	<b>19</b>
<b>CAPITULO 2</b>		<b>21</b>
<b>2.</b>	<b>Análisis</b>	<b>22</b>
<b>2.1</b>	<b>Levantamiento de Información</b>	<b>22</b>
2.1.1	Encuesta de sistemas	22
<b>2.2</b>	<b>Análisis de requerimientos</b>	<b>25</b>
<b>2.3</b>	<b>Diagrama de caso de usos</b>	<b>25</b>
2.3.1	Caso de uso 1	25
2.3.1.1	Descripción del caso de uso 1	27
2.3.2	Caso de uso 2	28
2.3.2.1	Descripción del caso de uso 2	31
2.3.3	Caso de uso 3	33

2.3.5 Diagrama de Flujo de Procesos.	37
2.3.5.1 Agente	37
2.3.5.2 Supervisor	38
2.5.3 Administrador	40
<b>CAPITULO 3</b>	<b>41</b>
<b>3. Diseño</b>	<b>41</b>
3.1 Diagrama Entidad-Relación	41
3.2 Diagrama de Clases.	42
3.3 Diagrama de Secuencia	44
3.3.1 Secuencia Agente.	44
3.3.2 Secuencia Supervisor	45
3.3.3 Secuencia Administrador	46
3.4 Diagrama de Flujo de Ventanas	47
3.5 Diseño de la Interfaz	48
3.5.1 Pagina Inicial	48
3.5.2 Pagina Agente	49
3.5.3 Pantallas Realizar Llamada	50
3.5.4 Pantalla Supervisor	51
3.5.5 Pantalla Realizar Consultas	53
3.5.5.1 Consultas Por fecha	53
3.5.5.2 Consulta Por Agente	54
3.5.5.3 Consultas Por Llamadas	54
3.5.5.4 Consultas Por Clientes.	55
3.5.5.5 Consultas Por Status de Llamadas.	56
3.5.6 Pantalla Mostrar Consulta	56
3.5.8 Pantalla Grabaciones	59
	60
3.5.9 Pantalla Definir Evento	60
3.5.10 Pantalla Administrador.jsp	62
<b>CAPITULO 4</b>	<b>65</b>
<b>4. Codificación</b>	<b>65</b>
4.1 Codificación de Procesos Principales	65
4.1.1 Proceso de Llamadas.	65
4.1.2 Proceso de Grabación de Llamadas.	67
4.1.3 Proceso de Generar Reportes	67
4.1.4 Proceso de Estructura Agente-Clientes.	68
<b>Capitulo 5</b>	<b>69</b>
<b>5. Pruebas</b>	<b>69</b>
5.1 Evaluación Operacional	69
5.2 Evaluacion Usuarios	70
<b>CAPITULO 6</b>	<b>71</b>
<b>6. Conclusiones y Recomendaciones</b>	<b>71</b>

# CAPITULO 1

## 1.1 Introducción

En nuestros días, la telefonía en las empresas ya sean grande, medianas o pequeñas es una herramienta muy importante para la comunicación que debe existir entre la empresa con sus respectivos clientes.

Realizar este tipo de procesos implica un gasto elevado ya sea por la compra de equipos, duración de llamadas y adicionalmente la mala administración del uso de este recurso.

Otro factor importante es que el cliente siempre nota la disponibilidad del servicio y la facilidad con que va a ser atendido.

Telefonivrix disminuirá los costos que tenga que realizar la empresa incluyendo una gran administración en el uso de las llamadas y dará una facilidad y disponibilidad a los clientes en la atención manteniéndolo al cliente siempre informado de su situación con la empresa.

### **1.1.1 Misión**

Brindar a los clientes la información necesaria para mantenerlos siempre actualizados de una manera ágil y siempre disponible en cualquier lugar dando al cliente la importancia que se merece a través de nuestro IVR.

### **1.1.2 Visión**

Fomentar el crecimiento de la empresa con nueva tecnología que dará a lugar a que nuevas áreas hagan uso de este medio para tener una gran captación de clientes por la atención permanente y siempre disponible.

Dicha tecnología se llama IVR permitiendo a los clientes una constante actualización de su estado en la empresa otorgándole una facilidad y comodidad al momento de ser informado

## **1.2 Antecedentes**

En la actualidad muchas empresas deben tener constante comunicación con sus clientes, de manera que cada agente realiza la llamada telefónica a cada cliente para mantenerlo informado de lo que la empresa crea necesario para el cliente, entonces cada agente debe realizar normalmente la ubicación de la información del cliente para luego realizar la respectiva llamada telefónica.

Las empresas no cuentan con un reporte preciso de las llamadas que realiza cada agente, dando a lugar a que las empresas no lleven un control del uso correcto del teléfono y tampoco medir la eficiencia con

que los agentes atienden a los clientes ya sea en cantidad de llamadas y en la duración de las mismas.

Existen empresas que no tienen un respaldo en las llamadas que ellos realizan para poder verificar el comportamiento que tuvo el agente con el cliente y viceversa, también sobre los compromisos que realizan los clientes con las empresas ya sea para establecer un día de pago o alguna modificación en sus datos.

En el mercado ya existen aplicativos que solucionan todos los problemas mencionados, pero sus costos son muy elevados.

### **1.3 Solución Propuesta.**

Telefonivrix se ajustará a las necesidades de empresas que manejen información de clientes y deban mantenerlos siempre actualizados ya sea para ofrecer un nuevo artículo, descuentos en una fecha en particular, nuevos servicios, y en caso de que la empresa ofrezca crédito directo para pagar un producto adquirido por el cliente, recordarle la deuda que contrae con la empresa.

El sistema brindará un gran beneficio a las empresas, ya que las personas encargadas de localizar a los clientes lo haga de de una manera muy ordenada y ágil, logrando así tener un mayor control en la información que la empresa debe transmitir a los clientes y también poder realizar un mayor numero de llamadas en el día.

Adicionalmente la empresa impedirá que cada uno de los agentes (personas encargadas de realizar las llamadas al cliente) cometa errores al momento de realizar el proceso normal de llamar el cliente.

Telefonivrix optimizará los procesos de comunicación con los clientes para que las llamadas resulten más económicas, se emitirá reportes de las llamadas realizadas por los agentes y facilitará a los respectivos supervisores grabar las llamadas de cualquier agente ya sea que pertenezca al grupo que tiene asignado y a otros grupos.

Para cubrir el proceso de comunicación con el cliente, el sistema dará la facilidad de que cada agente con solo presionar un botón desde la aplicación, se realice la llamada con el cliente sin necesidad de que tengan que buscar la información del cliente y marcar el respectivo número, pudiendo así controlar el error de marcado permitiendo realizar un mayor número de llamadas y a su vez exista una mayor rapidez en la comunicación con los respectivos clientes.

Además, el sistema nos permitirá obtener un reporte de todas las llamadas realizadas por el agente permitiendo que se realice un análisis para poder medir el desempeño con la que el agente atiende a cada cliente y también la cantidad de llamadas realizadas por el agente.

También el sistema dará la opción a los supervisores de grabar las conversaciones realizadas por los agentes, en donde el supervisor pueda escoger aleatoriamente a cualquier agente que esta teniendo la conversación con el cliente y pueda grabar dicha conversación para poder medir el desempeño de dicho agente.

La empresa obtendrá muchos beneficios con el uso de nuestro producto, ya que mejorará el rendimiento de sus empleados y brindará una mejor atención a los clientes ya sea para que conozcan sus nuevos productos, servicios, promociones o estén al día en sus pagos.

#### **1.4 Objetivo General.**

Nuestro Producto contribuirá para que el proceso de llamadas telefónicas realizadas a nuestros clientes ya sea para realizar cobro o campañas publicitarias se las pueda llevar a cabo de una manera automatizada, económica, eficiente y rápida.

#### **1.5 Objetivos Específicos.**

- Realizar las llamadas telefónicas desde una computadora con un simple click (VoIP).
  
- Llevar un control detallado de las llamadas por agente realizadas a los clientes, y guardarlo en un registro
  
- Administrar el trabajo realizado por los agentes, generando reportes para que puedan ser revisados por los supervisores.

- Creación de usuario para la aplicación con sus respectivas contraseñas para agentes y supervisores.
  
- Establecer una estructura personalizada de clientes a llamar a cada uno de los agentes.
  
- Ofrecer la opción de poder grabar la conversación entre el cliente y mi agente cuando sea requerida y guárdalo en un formato de audio.
  
- Envío de reportes vía correo electrónico.

### **1.6 Alcances**

- El modulo para poder integrar mi aplicación con mi software de PBX, donde se pueda ejecutar con una instrucción la llamada telefónica desde mi aplicación hacia mi PBX.
  
- Se manejaran tres tipos de usuario: el usuario admin. que tiene todo el control para poder crear o eliminar usuario de la aplicación(lógica)  
El usuario tipo supervisor solo podrá realizar consultas y emitir reportes y el usuario tipo agente solo podrá hacer llamadas desde la aplicación.
  
- Una aplicación cliente que será manejada ya sea por el usuario tipo supervisor o el usuario tipo agente donde para ingresar se les pediría una respectiva autenticación, se tiene que presentar en pantallas la información necesaria según el caso por Ej. en el caso del agente tendría que visualizar los clientes a llamar en el caso del supervisor las consultas que desea realizar.

- Un modulo en mi aplicación donde se pueda consultar los registros de llamadas que los genera mi software PBX y puedan ser visualizados ya sean por los supervisores.
- Llevar a cabo un modulo donde se pueda definir una lista de clientes a llamar para cada uno de los agentes, esta lista de clientes se la realiza de manera aleatoria de mi Base datos.
- Una función donde le presente una opción al supervisor para que pueda grabar la conversaron que tiene el agente con el cliente y guardar el archivo en un formato de audio. Dicho archivo de audio también se guardara con la respectiva información del cliente y el agente para poder usarlo como evidencia si lo sea requerido.
- Se realizara una función donde le presente al supervisor una opción en el momento de realizar la consulta, para que dicha consulta pueda ser enviado por correo a otras personas interesadas en la empresa Ej.: gerente financiero.
- Cuando el agente vaya a realizar la llamada, tendrá que visualizar los datos del cliente, y permitirle actualizar cierto campos

## 1.7 FODA:

Vamos a utilizar la herramienta FODA, para analizar los elementos internos o externos de mi Producto.

**Fortalezas:** Se creara registros por cada operación hecha.

**Oportunidad:** Se desarrollara con aplicativos Open source

**Debilidades:** No hay mucha documentación de nuestro software PBX.

**Amenazas:** La competencia en el mercado de aplicaciones Open Source.

## 1.8 Metodología

Para representar gráficamente mi diseño del producto vamos a utilizar el lenguaje unificado modelado (UML), que es un lenguaje que me permite representar en mayor o menor medida todas las fases de realización de mi producto, también me permitirá especificar cuales con las características de mi producto antes de su construcción y dicha información sirve para la documentación de mi producto desarrollado para su futura revisión.

### **1.8.1 Diagramas UML**

Para poder representar gráficamente el diseño de mi producto se utilizaran lo siguientes diagramas UML

- Diagrama de Caso de Uso
- Diagrama de Clase
- Diagrama de Secuencia

## **1.9 Estudio de Factibilidad**

### **1.9.1 Estudio de Mercado**

El estudio de mercado nos ayudara a conocer información necesaria para establecer las diferentes políticas, planes, estrategias mas adecuadas para la construcción de mi producto.

#### **1.9.1.1 Introducción.**

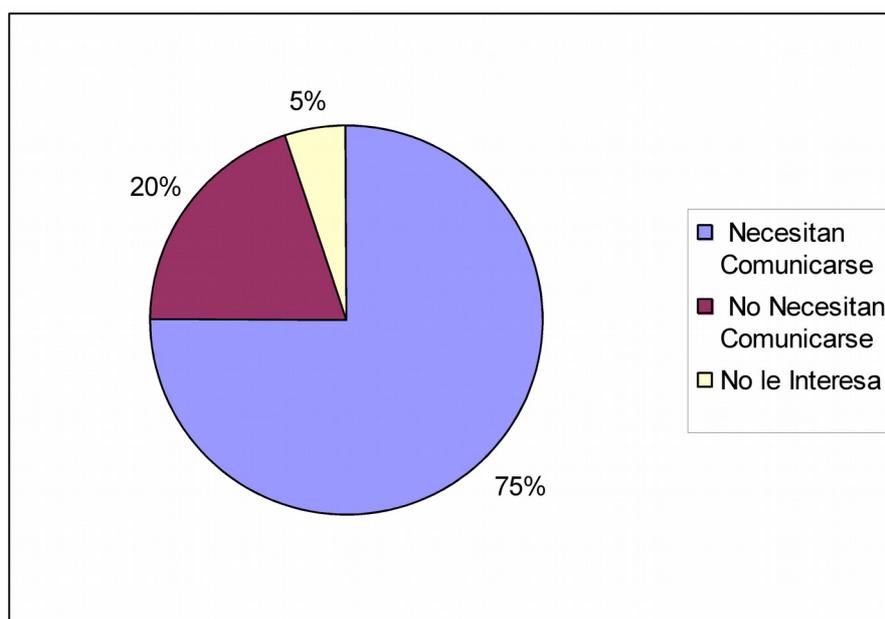
Existe en el mercado, una gran variedad de software PBX que esta completamente gratuitos, la mayoría de dichos software se ejecutan en entorno Linux, y son un poco difíciles su administración es por eso que nuestro producto tiene una interfaz amigable para que no se haga difícil su administración.

En nuestro estudio de mercado, se pudo constatar la falta de comunicación que tienen las empresas con sus clientes, y ese es un factor muy importante para que los clientes se sienta a gusto con ellos. Poca son las empresas que tienen una comunicación telefónica con sus clientes.

#### 1.9.1.2 Mercado Meta:

Mi producto esta orientado a todas las empresas que tienen la necesidad de poder comunicarse con sus clientes, ya sea para realizar promociones, cobros, y un sinnúmero de tareas que se pueden realizar de manera telefónica.

En un muestreo que se realizo se estableció que más del 75% de las empresas de guayaquil tienen una necesidad de comunicación con sus clientes



### **1.9.1.3 Identificación de la competencia**

En el mercado existe algunas soluciones para resuelven el problema que tienen algunas empresas para poder comunicarse con sus clientes, por ejemplo tenemos el ivr Ángel (para mas información de este software visitar [www.angel.com](http://www.angel.com)) que es una aplicación que tiene incluido tanto el ivr tipo inbound como el outbound y ofrece un sinnúmero de tareas adicionales, el punto débil de este software es su precio, es demasiado costoso su implementación.

## **1.10 Estudio técnico**

El objetivo de nuestro objetivo técnico es establecer la tecnología que se necesitara para la construcción de mi producto.

De acuerdo a la tecnología necesaria para la implementación de nuestro producto se estableció dos enfoques: Hardware y Software

### **1.10.1 Hardware.**

En cuanto a hardware, específicamente el servidor donde estará instalado nuestra aplicación junto con nuestra Base de Datos y nuestro software PBX tienen que cubrir los siguientes requerimientos mínimos.

- Pentium III 800Mhz
- 512MB de RAM
- 80GB de espacio en disco
- Tarjeta de red 10/100
- Unidad de CD-ROM
- Monitor

- Teclado
- Mouse
- Unidad de Protección UPS

El hardware donde se va a ejecutar nuestra aplicación cliente tiene que cubrir lo siguientes requerimientos mínimos.

- Pentium III 800Mhz
- 256 MB de RAM
- 8 GB de espacio en disco
- Tarjeta de red 10/100
- Monitor
- Teclado
- Mouse
- Unidad de Protección UPS

Se debe de contar con una red interna donde se pueda interconectar todas las PC que ejecutaran la aplicación cliente con el servidor donde se estará ejecutando la aplicación Server.

#### **1.10.2 Software.**

En cuanto al software se necesitara los siguientes aplicativos para mi servidor

- Sistemas Operativo Linux
- Java 1.5
- Base de Datos
- Software PBX
- Servidor TomCat

Este software es el que se necesitara para poder ejecutar mi aplicación

- Sistema Operativo XP
- Java 1.5
- Softphone

## **1.11 Recursos**

### **1.11.1 Recurso Físico**

Principalmente se usara un espacio físico con las condiciones adecuadas para la estabilidad de nuestros equipos, este lugar debe estar libre de residuos de polvo , libre de humedad y principalmente debe de mantener los estándares básicos de seguridad según normas generales establecidas.

### **1.11.2 Recurso Humano**

Para realizar nuestro IVR distribuido contaremos con 3 personas cada una con ciertas funciones principales, como son estructurar físicamente el sistema, diseñar el software, prueba e implementación, etc.

### **Participantes del proyecto**

El personal encargado del proyecto, considerando las fases de inicio, elaboración e implementación, estará formado por los siguientes puestos de trabajo y personal asociado:

- **Jefe de proyecto:**

Labor de Noe Abraham Pinzon Leones, egresado de la Carrera de Ingeniería en sistemas computacionales en la Facultad de Matemáticas

y Física de la Universidad de Guayaquil. Con una experiencia modesta en metodología de diseño de redes, configuración de dispositivos de comunicaciones y en particular dispositivos CISCO.

- **Administrador de Servidores:**

El perfil establecido es: Ingeniero en Sistemas Computacionales con conocimientos de instalación, configuración y administración de Servidores, RAID, dominio de sistemas operativos tanto Linux como Windows, métodos de contingencia, uno de ellos al menos con experiencia en sistemas afines a la línea del proyecto, labor que llevara a cabo Webster Javier Hernández Salazar.

- **Analista de Sistemas**

Con experiencia en el entorno de desarrollo del proyecto, con el fin de que los prototipos puedan ser lo mas cercanos posibles al producto final. Este trabajo ha sido encomendado a Allan Omar Gallegos Vincés.

### **Interfaces Externas**

Los participantes del proyecto que proporcionaran los requisitos del sistema, entre ellos los encargados de evaluar los artefactos de acuerdo a cada subsistema y según al plan establecido, serán elegidos durante el proceso de muestreo; este proceso nos permitirá a través de una serie de encuestas definir cuales serán las necesidades que deben ser satisfechas por el sistema.

### **Roles y responsabilidades**

<b>Rol</b>	<b>Responsabilidad</b>
Jefe de Proyecto	El jefe de proyecto se encargara de asignar los recursos, gestionar prioridades, coordinar interacciones con los clientes y usuarios, y

	mantener al equipo de proyecto enfocado en los objetivos. Además se encargara de supervisar el establecimiento de la arquitectura del sistema. Gestión de riesgos planificación y control de proyecto.
Administrador de Servidores	Instala el hardware y software necesario para la realización del proyecto, además configurara los servidores, sistemas operativos y establecerá las seguridades debidas para el correcto funcionamiento del sistema.
Analista de Sistemas	Captura, específica y valida los requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elabora el modelo de análisis y diseño. Colabora en la elaboración de las pruebas funcionales y modelo de datos.

Cabe mencionar que el sistema será desarrollado por todos los integrantes del grupo, es decir la programación del sistema será responsabilidad de cada uno de los integrantes del grupo, el jefe de proyecto designara a su debido tiempo, que módulos se trabajaran en conjunto, y que módulos se asignaran a cada uno de los integrantes, optimizando todos los recursos.

### **1.11.3 Recurso Tecnológico**

#### **1.11.3.1 Hardware**

Usaremos un computador simple con las siguientes características:

- Procesador Pentium IV 3.0 GHZ (mínimo de 2.4 GHZ)
- Memoria RAM 1GB (mínimo de 512Mb)
- Disco Duro de 160 GB (mínimo de 80 GB)

### **1.11.3.2 Software**

- MySQL
- Asterisk
- Eclipse
- Java
- Centos 5.1 y Herramientas OpenSource

## **1.12 Presupuesto**

### **1.12.1 Costo de Hardware**

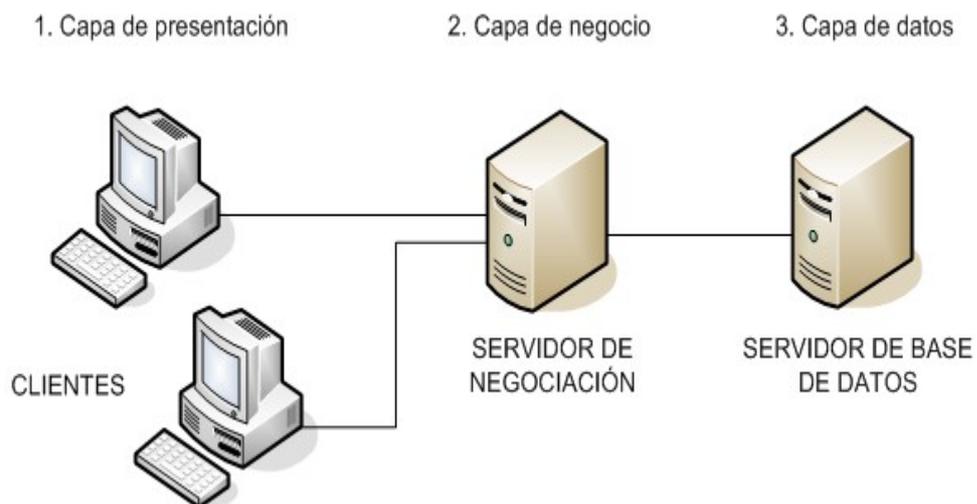
- Procesador Pentium IV 3.0 GHZ
- Memoria RAM 1GB
- Disco Duro de 160GB
- Accesorios
- Costo Total del Equipo                      \$ 600

### **1.12.2 Costo de Software**

Como nos podemos dar cuenta el 100% del software que utilizaremos es open source, así que nos evitamos el costo del software.

### **1.13 Estructura del Sistema**

Para realizar nuestro sistema usaremos programación de tres capas en un nivel.



#### **1.- Capa de presentación:**

La primera capa presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. Esta capa se comunica con la capa de negocio. También es conocida como interfaz gráfica y debe de ser entendible y fácil de usar para el usuario. En nuestro sistema la capa de presentación será desarrollada en jsp con un ambiente amigable y intuitivo, para que el usuario se sienta a gusto con el sistema.

## **2.- Capa de negocio:**

Aquí residen los [programas](#) que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio o lógica del negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de [base de datos](#) para almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

En nuestro sistema usaremos servlet y aplicaciones como eclipse para la programación de los mismos.

## **3.- Capa de datos:**

Aquí residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. En nuestro sistema podríamos usar como gestor de base de datos Postgresql o mysql.

# CAPITULO 2

## 2. Análisis

### 2.1 Levantamiento de Información

Analizando el proceso de comunicación que existe entre las empresas con sus clientes, se pudo detectar que muchas empresas no tiene una comunicación con sus clientes y otras empresas solo se comunican con sus clientes por medio de correos electrónicos, este medio de correo electrónico es un poco económico pero no garantiza que el cliente se entere lo que la empresa esta tratando de indicarle.

Hay empresas que se comunican con sus clientes de manera telefónica, pero no llevan un control detallado de todas las llamadas realizadas y la cartera de clientes a llamar se la realiza de una manera inadecuada.

Cuando el supervisor quiere enterarse de cuantas llamadas se realizaron, o cuantas veces se llamo a un cliente, no podría tener dicha información ya que carece de un sistema que le permita tener registrado todas las llamadas realizados, tampoco puede controlar el trabajo realizado por sus agentes y ni las llamadas realizadas por estas personas.

Se realizó unas pequeñas encuestas para evaluar cómo se está desarrollando la comunicación que se realiza telefónicamente con los clientes de la empresa y el control de estas mismas.

### **2.1.1 Encuesta de sistemas**

**Nombre:** Ing. Freddy Romero

**Cargo:** Jefe de Cobranzas de Camposantos del Ecuador

1) ¿Sabe usted realmente cuánto tiempo ocupan el teléfono sus colaboradores? Comente.

En realidad no.

2) ¿Existe alguna manera de control que usted tiene sobre las llamadas realizadas por sus colaboradores? Si las hay, ¿usted está totalmente satisfecho? Comente.

No específicamente yo, parece que aquí existe un sistema que limita las llamadas a cierto tiempo. Permite hablar máximo 5 minutos y luego suena un timbre y se cuelga.

3) ¿Puede usted dar un reporte de cuánto le está costando a la empresa el uso telefónico?

No, por el momento no.

4) ¿Usted cree que existe alguna forma de que sus empleados optimicen el tiempo de trabajo? Ideas.

Por supuesto, escuche de un software que me permite realizar todo eso, pero el problema es que es un poco costoso, y la empresa por el momento no le da prioridad a esas soluciones.

5) Si usted tuviera la oportunidad de pedir un sistema a su medida que tareas le gustaría que este realice.

- Me gustaría que se puedan generar reportes
- Se pueda tomar una muestra de llamadas
- Se pueda enviar reportes por correos

**Nombre:** Denisse Saltos

**Cargo:** Jefe de Departamento de Ventas

1) ¿Sabe usted realmente cuanto tiempo ocupan el teléfono sus colaboradores? Comente.

No me he puesto a pensar en eso en realidad

2) ¿Existe alguna manera de control que usted tiene sobre las llamadas realizadas por sus colaboradores? Si las hay esta usted totalmente satisfecho? Comente.

Nosotros no tenemos restricciones en llamadas

3) ¿Puede usted dar un reporte de cuanto le esta costando a la empresa el uso telefónico?

No

4) ¿Usted cree que existe alguna forma de que sus empleados optimicen el tiempo de trabajo? Ideas.

Mayor control, comunicación.

5) Si usted tuviera la oportunidad de pedir un sistema a su medida que tareas le gustaría que este realice.

-cuando quiera realizar una promoción, avisar automáticamente a una lista de clientes y que el sistema los seleccione de una base de datos y yo solo determino el porcentaje de descuento.

## 2.2 Análisis de requerimientos

Para poder llevar una comunicación con los clientes de la empresa de una manera mas eficiente se creo Telefonivrix, este tendrá una interfaz amigable tanto para el agente como para el supervisor, cada agente tendrá un supervisor a designado y un supervisor tendrá uno o varios agentes a cargo, el agente se le designara una cartera de clientes a llamar, dicha cartera de clientes serán seleccionados por el tipo de evento que el supervisor a cargo haya seleccionado, esto le permitirá podrá hacer diferentes tipo de evento ya sea para realizar el proceso de cobro como para promocionar un nuevo producto

El supervisor tendrá la facilidad de realizar consultas ya sea para consultar las llamadas realizadas a los clientes, o medir el desempeño de los agentes y otro tipo de consultas. El supervisor podrá decidir si se graban las llamadas realizadas por los agentes para poder evaluar el desempeño que tiene un agente al momento de realizar una llamada.

## 2.3 Diagrama de caso de usos

El diagrama de casos de usos representa gráficamente la interacción que va a tener mis actores (Agente, supervisor y administrado) con mi aplicación Telefonivrix.

### 2.3.1 Caso de uso 1

#### Ingreso del Agente a la Aplicación

##### Primer Nivel

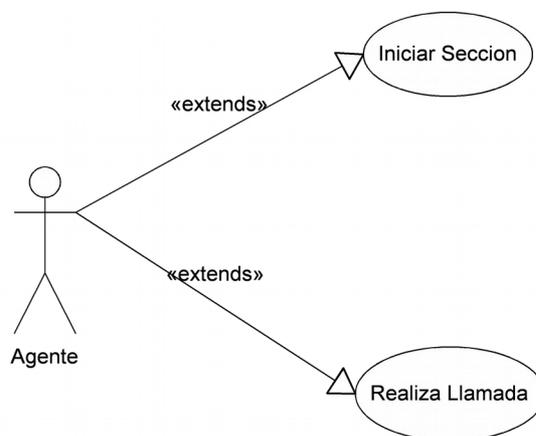
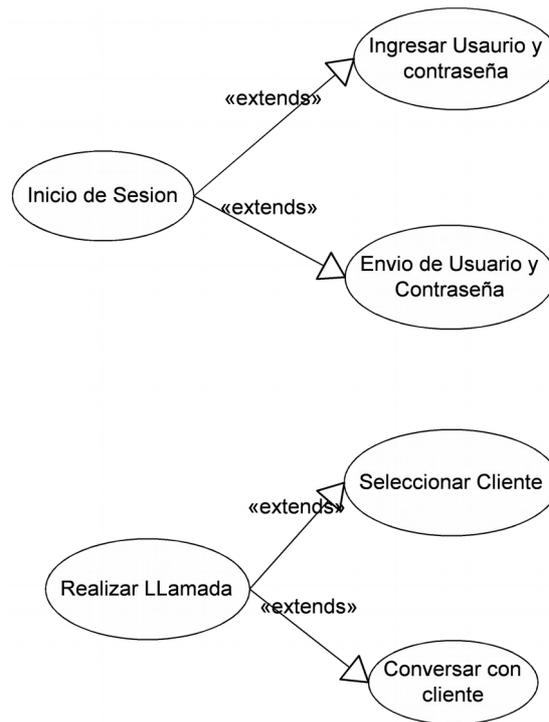


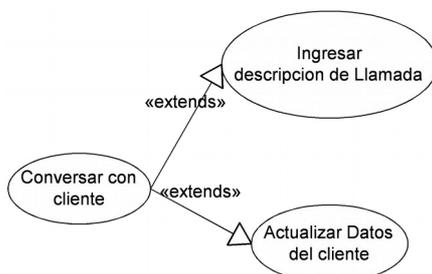
Figura Caso Uso 1 Nivel 1

##### Segundo Nivel



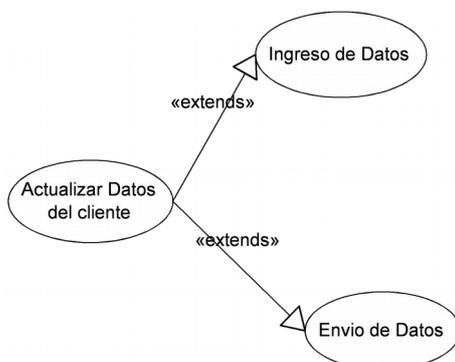
**Figura Caso Uso 1 Nivel 2**

### Tercer Nivel



**Figura Caso Uso 1 Nivel 3**

### Cuarto Nivel



**Figura Caso Uso 1 Nivel 4**

### 2.3.1.1 Descripción del caso de uso 1

<b>Nombre:</b>	Inicio de Sesión
<b>Alias:</b>	
<b>Actores:</b>	Usuario del Sistema, “Agente”
<b>Función:</b>	Ingreso de datos para acceder al aplicativo
	El Sistema valorará y verificará lo siguiente: <ol style="list-style-type: none"> <li>1. Que ingrese datos válidos.</li> <li>2. Que ambos campos usuario/contraseña sean llenos para que valide el sistema y registrar su entrada</li> </ol>
<b>Referencias:</b>	

**Figura a Descripción Caso Uso 1**

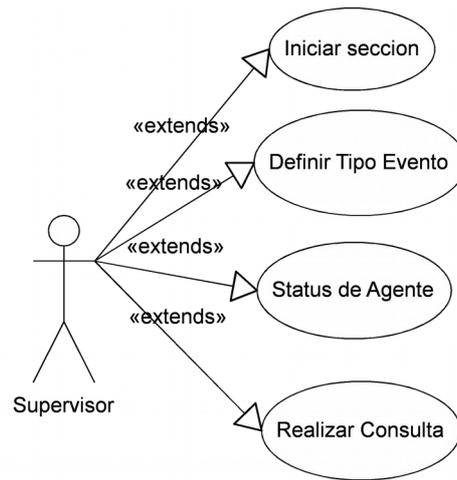
<b>Nombre:</b>	Realizar Llamada
<b>Alias:</b>	
<b>Actores:</b>	Usuario del Sistema, “Agente”
<b>Función:</b>	Comunicarse con los clientes.
	El Sistema valorará y verificará lo siguiente: <ol style="list-style-type: none"> <li>1. Que se comunique con el cliente.</li> <li>2. Originar el respectivo detalle de la llamada.</li> <li>3. Llevar un registro de la llamada realizada</li> </ol>
<b>Referencias:</b>	

**Figura b Descripción Caso Uso 1**

### 2.3.2 Caso de uso 2

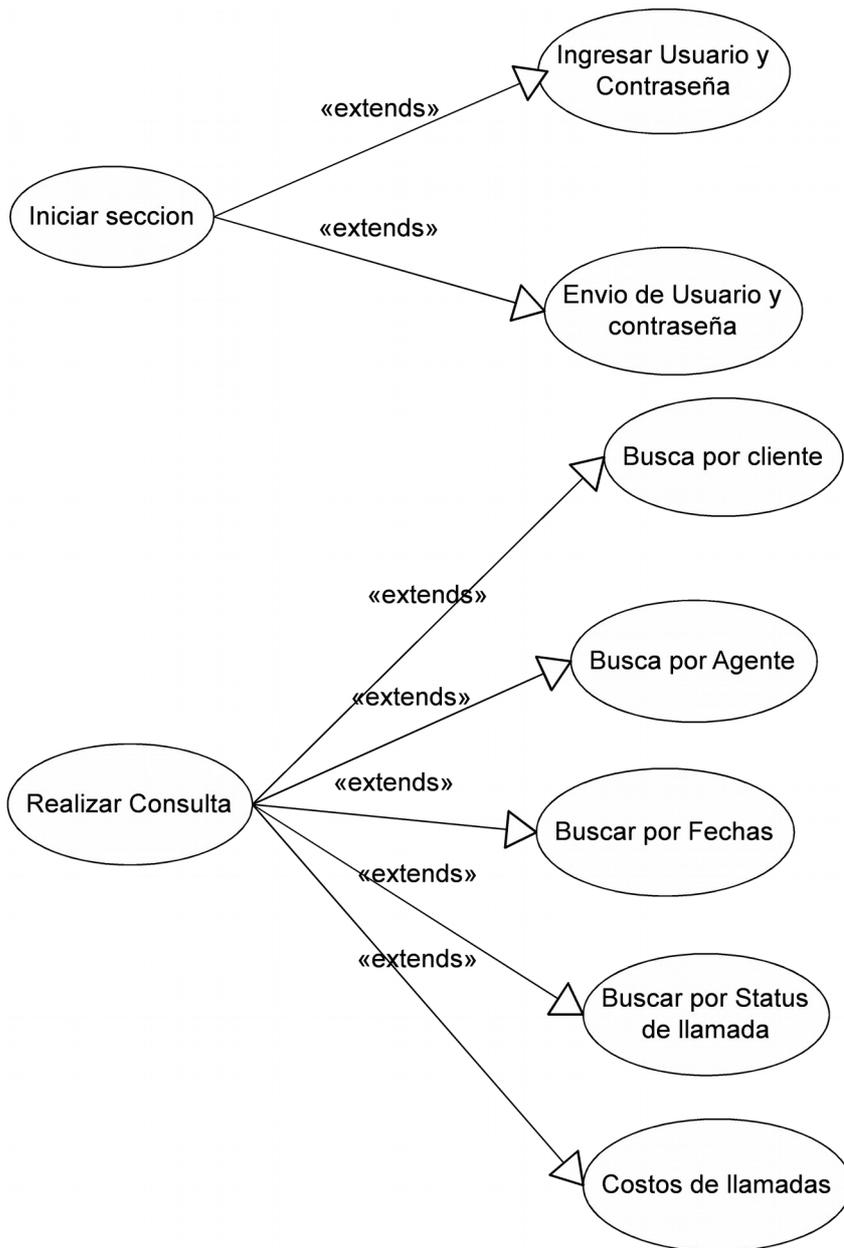
#### Ingreso del Supervisor a la Aplicación

#### Primer Nivel



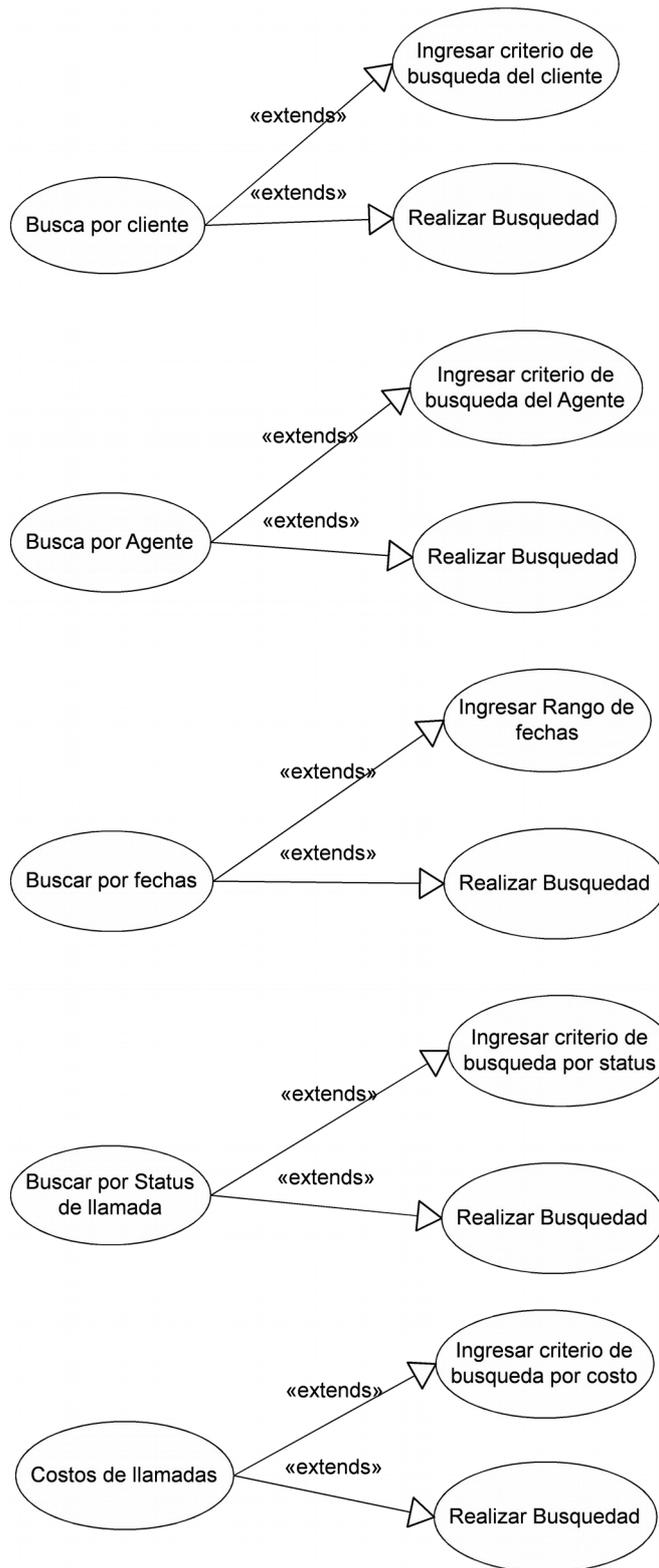
**Figura Caso Uso 2 Nivel 1**

**Segundo Nivel**



**Figura Caso Uso 2 Nivel 2**

**Tercer Nivel**



**Figura Caso Uso 2 Nivel 3**

### 2.3.2.1 Descripción del caso de uso 2

<b>Nombre:</b>	Inicio de Sesión
<b>Alias:</b>	
<b>Actores:</b>	Usuario del Sistema, “Supervisor”
<b>Función:</b>	Ingreso de datos para acceder al aplicativo
	El Sistema valorará y verificará lo siguiente: 1. Que ingrese datos válidos. 2. Que ambos campos usuario/contraseña sean llenos para que valide el sistema y registrar su entrada
<b>Referencias:</b>	

**Figura a Descripción Caso Uso 2**

<b>Nombre:</b>	Definir Tipo Evento
<b>Alias:</b>	
<b>Actores:</b>	Usuario del Sistema, “Supervisor”
<b>Función:</b>	Ingresar Tipo de evento a seguir
	El supervisor puede definir el tipo de evento a realizar por los agentes que pueden ser: Publicidad Cobros
<b>Referencias:</b>	

**Figura b Descripción Caso Uso 2**

<b>Nombre:</b>	Status Agente
<b>Alias:</b>	
<b>Actores:</b>	Usuario del Sistema, “Supervisor”
<b>Función:</b>	Modificar Status de Agente
	El supervisor podrá cambiar el status del agente para definir si el sistema le graba las conversaciones mantenidas por el agente con los clientes

<b>Referencias:</b>	
---------------------	--

**Figura c Descripción Caso Uso 2**

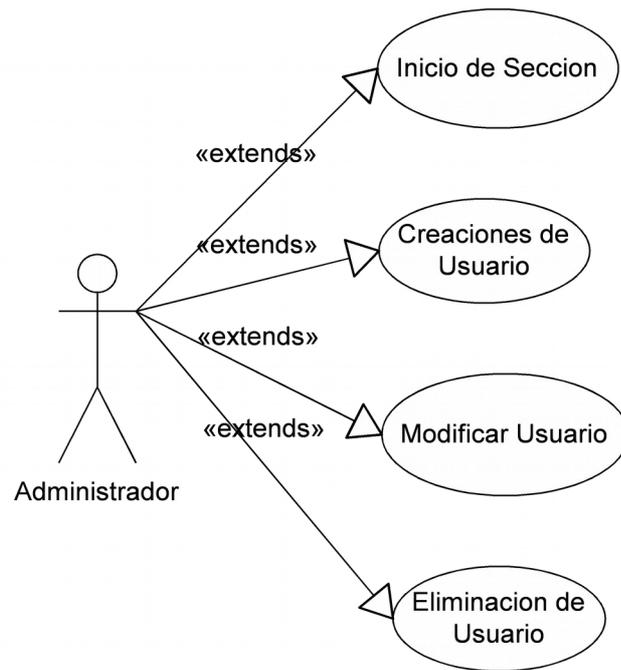
<b>Nombre:</b>	Realizar Consultas
<b>Alias:</b>	
<b>Actores:</b>	Usuario del Sistema, “ <b>Supervisor</b> ”
<b>Función:</b>	Consultas por diferentes criterios de búsqueda.
	<p>El Sistema permitirá lo siguiente:</p> <ol style="list-style-type: none"> <li>1. Que el usuario pueda emitir un detalle sobre las llamadas registradas por el criterio que crea conveniente (agente, fecha, status, cliente, costo)</li> <li>2. Que dichos reportes se puedan enviar por correo electrónico a otras personas de la empresa</li> </ol>
<b>Referencias:</b>	

**Figura d Descripción Caso Uso 2**

### 2.3.3 Caso de uso 3

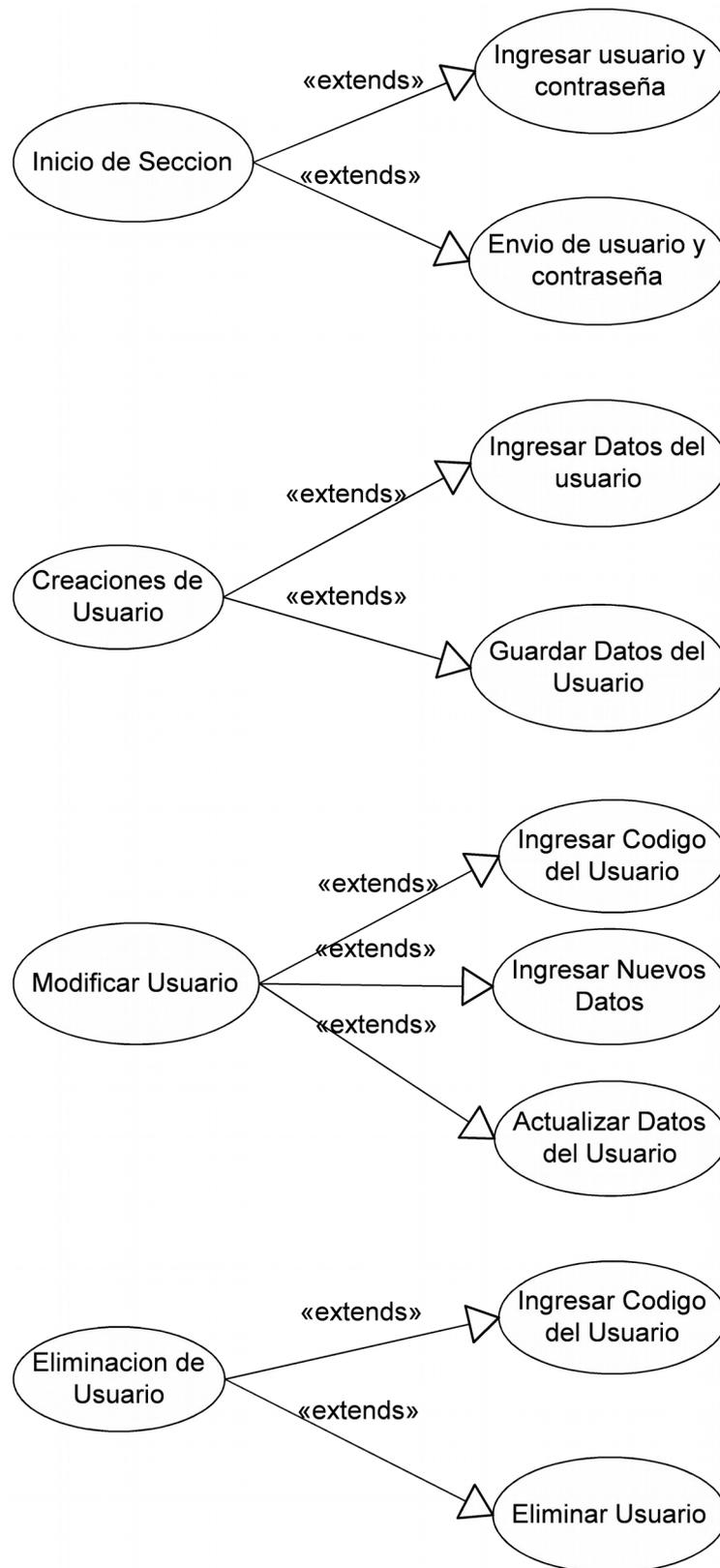
#### Ingreso del Administrador a la Aplicación

##### Primer Nivel



**Figura Caso Uso 3 Nivel 1**

**Segundo Nivel**



**Figura Caso Uso 3 Nivel 2**

### 2.3.3.1 Descripción del caso de uso 3

<b>Nombre:</b>	Inicio de Sesión
<b>Alias:</b>	
<b>Actores:</b>	Usuario del Sistema, “Administrador”
<b>Función:</b>	Ingreso de datos para acceder al aplicativo
	El Sistema valorará y verificará lo siguiente: <ol style="list-style-type: none"> <li>1. Que ingrese datos válidos.</li> <li>2. Que ambos campos usuario/contraseña sean llenos para que valide el sistema y registrar su entrada</li> </ol>
<b>Referencias:</b>	

**Figura a Descripción Caso Uso 3**

<b>Nombre:</b>	Creación de Usuario
<b>Alias:</b>	
<b>Actores:</b>	Usuario del Sistema, “Administrador”
<b>Función:</b>	Crear Usuarios nuevos para la aplicación
	El sistema podrá crea usuarios, ya sean agentes o supervisores. Y designara que cada agente tenga un supervisor que este a cargo de el .
<b>Referencias:</b>	

**Figura b Descripción Caso Uso 3**

<b>Nombre:</b>	Modificar de Usuario
<b>Alias:</b>	
<b>Actores:</b>	Usuario del Sistema, “Administrador”
<b>Función:</b>	Modificar usuario existente en el sistemas
	El sistema podrá modificar usuarios, ya sean agentes o supervisores. Se podrá cambiar datos que se puedan modificar en el transcurso del tiempo
<b>Referencias:</b>	

**Figura c Descripción Caso Uso 3**

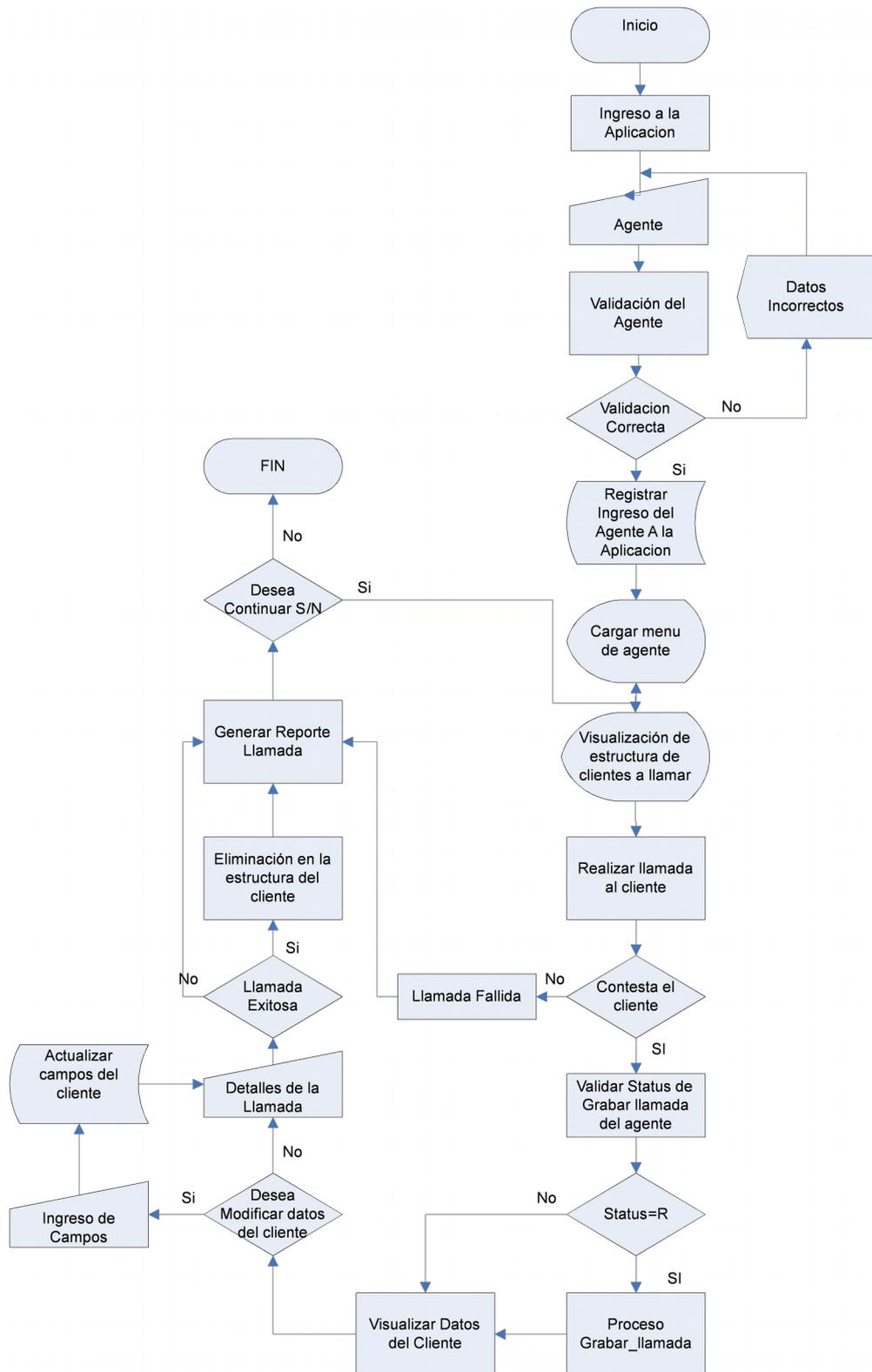
<b>Nombre:</b>	Eliminación de Usuarios
<b>Alias:</b>	
<b>Actores:</b>	Usuario del Sistema, “Administrador”
<b>Función:</b>	Eliminación de Usuarios
	Eliminar usuarios existentes en la aplicación ya sean agentes o supervisores.
<b>Referencias:</b>	

**Figura d Descripción Caso Uso 3**

### 2.3.5 Diagrama de Flujo de Procesos.

Este diagrama me va a representar gráficamente la secuencia de ingreso de los usuarios a la aplicación

#### 2.3.5.1 Agente



**Figura Flujo Proceso Agente**

### 2.3.5.2 Supervisor

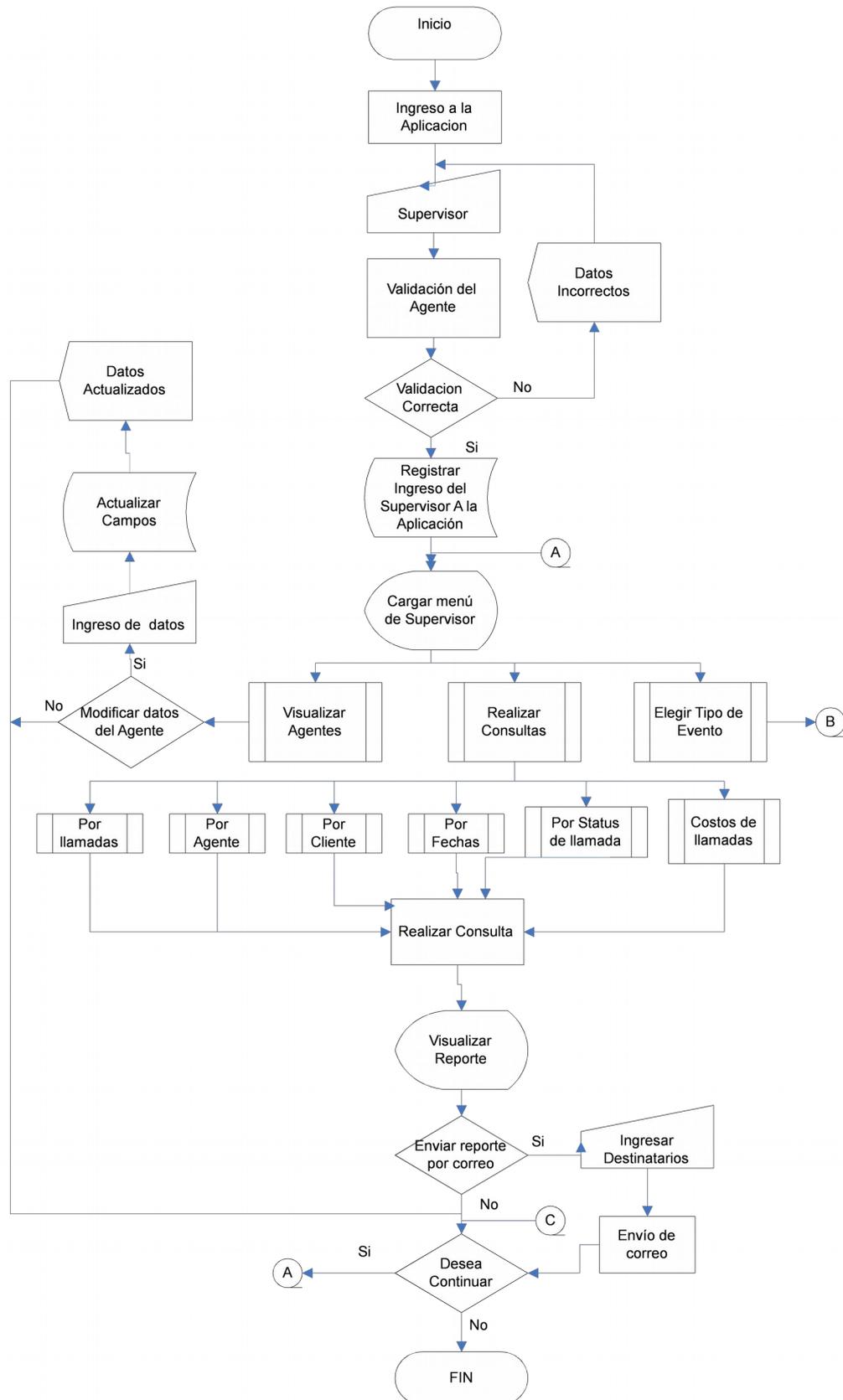
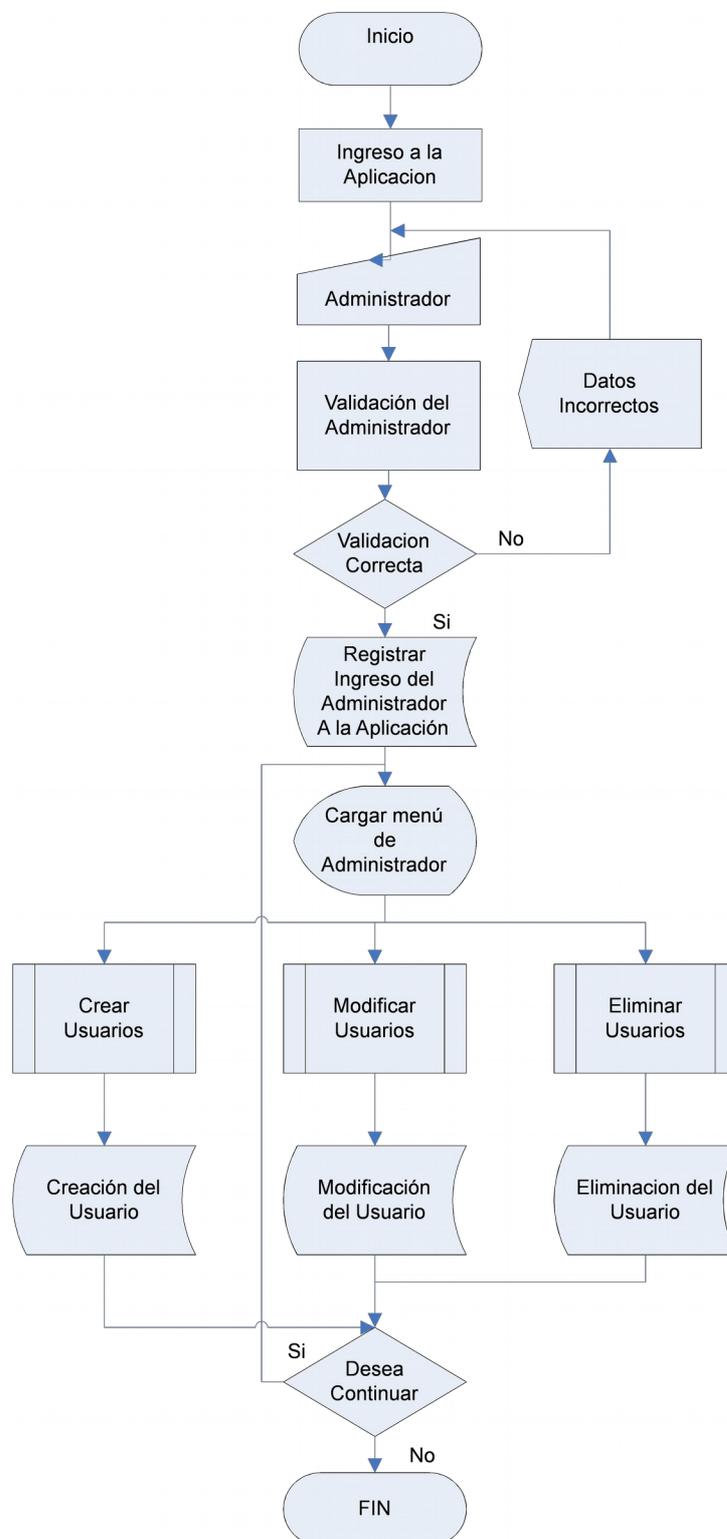


Figura Flujo Proceso Supervisor

### 2.5.3 Administrador



**Figura Flujo Proceso Administrador**

# **CAPITULO 3**

## **3. Diseño**

### **3.1 Diagrama Entidad-Relación**

El diagrama entidad-relación pretende visualizar los objetos que pertenecen a la base de datos como entidades.

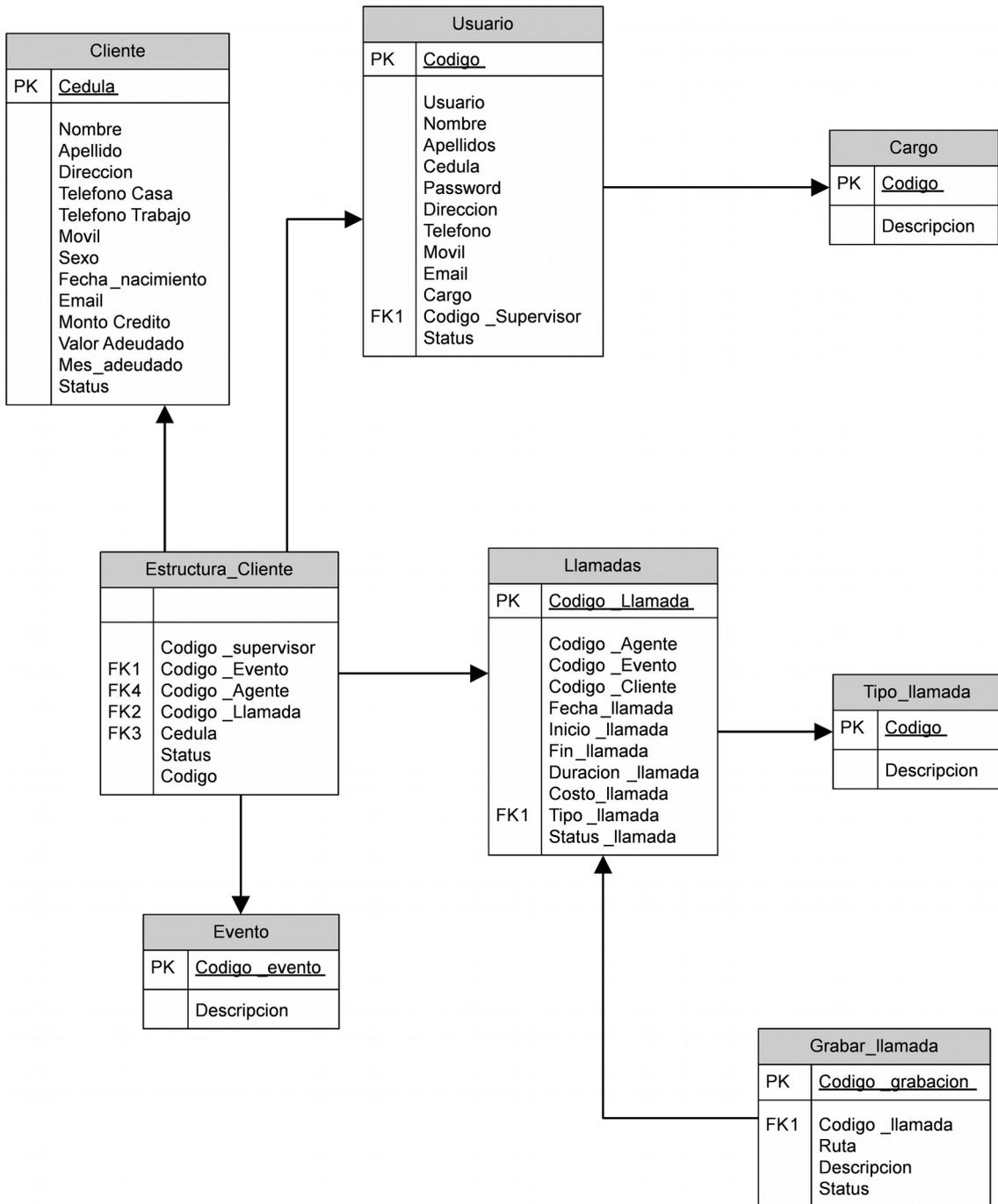


Figura DER

### 3.2 Diagrama de Clases.

Este diagrama de clases va a representar la estructura de mi aplicación mostrando sus principales clases, atributos, operaciones y las relaciones entre ellos.

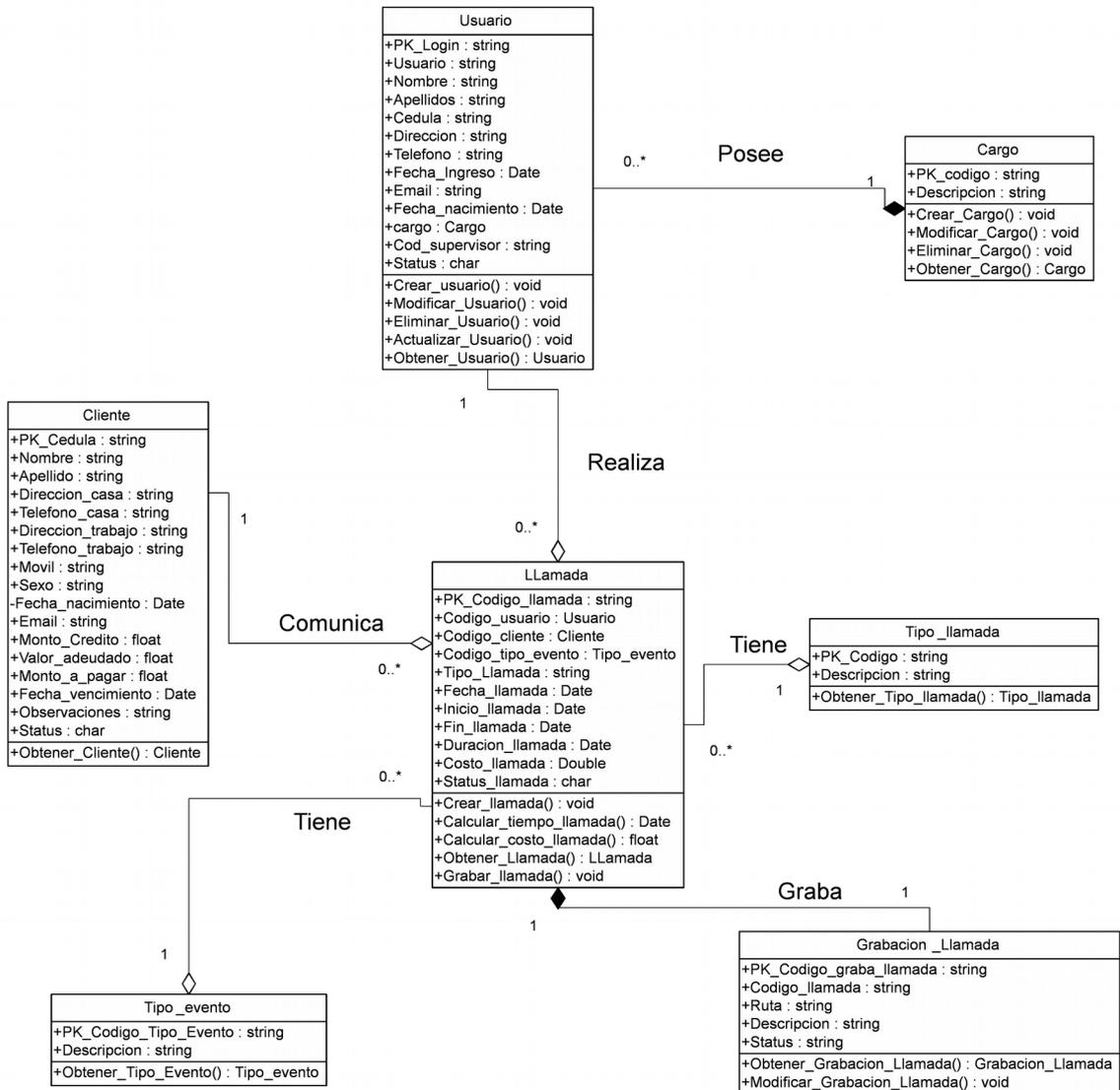
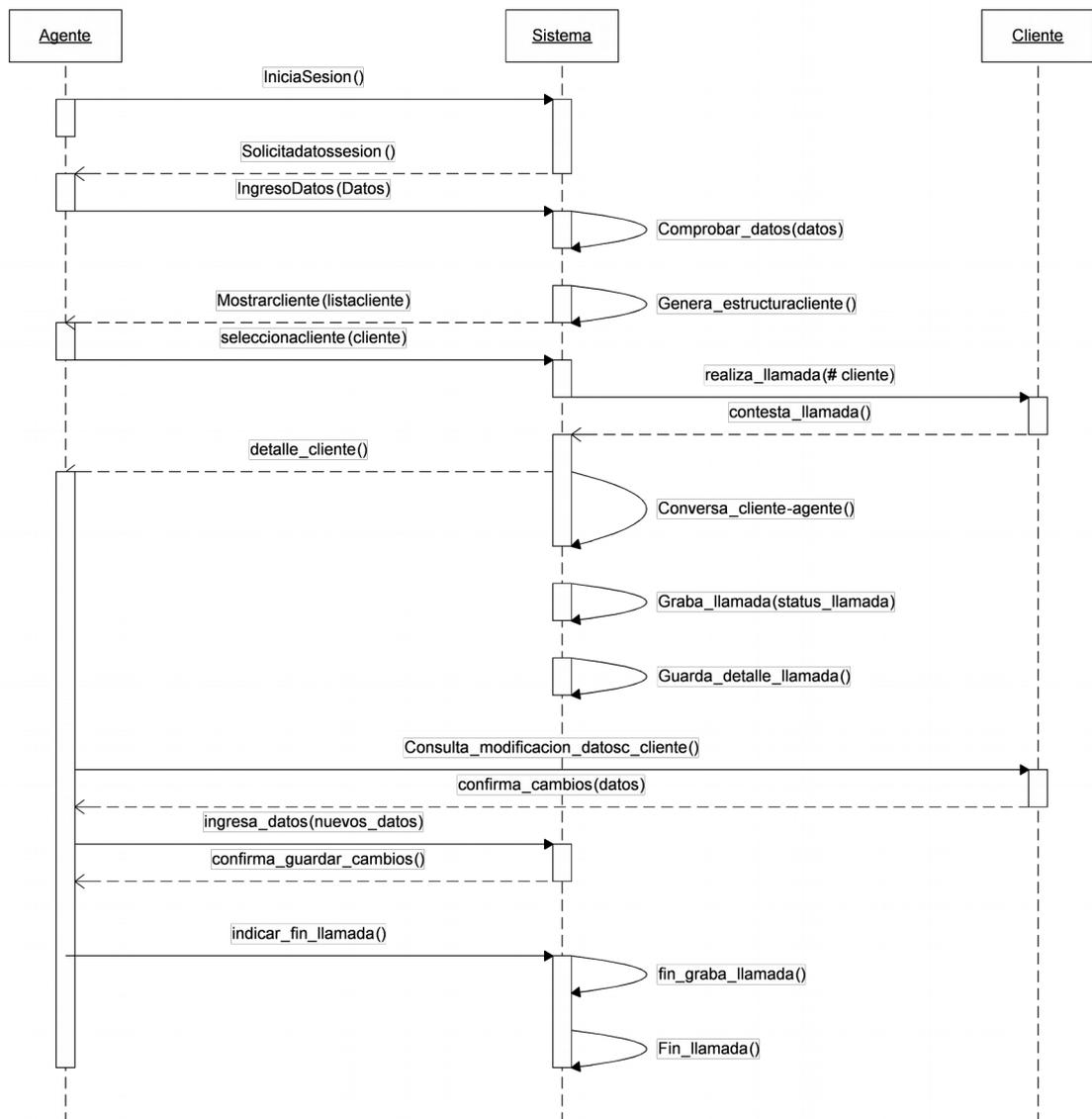


Figura Diagrama de Clases

### **3.3 Diagrama de Secuencia**

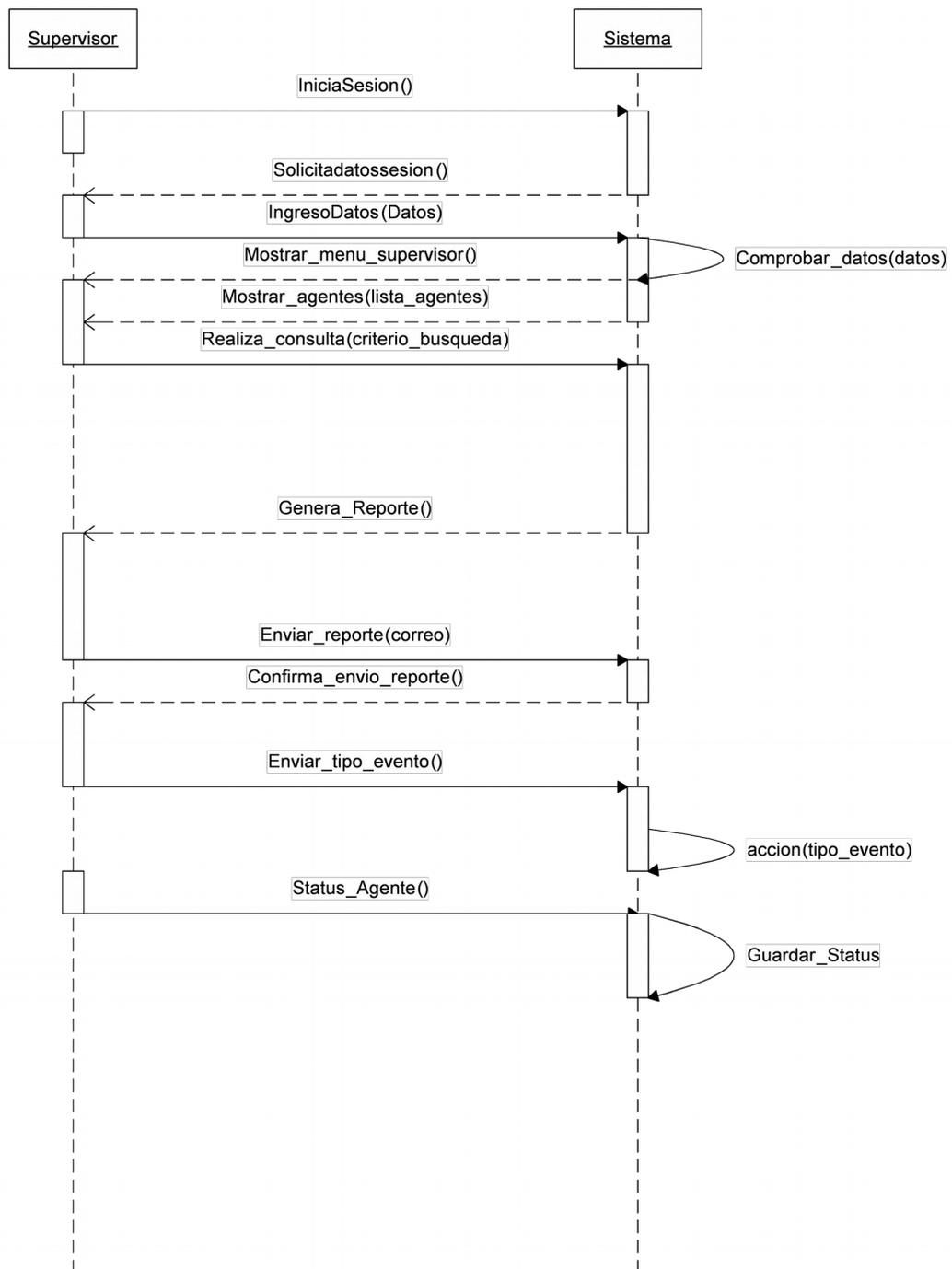
En este diagrama se muestra la interacción de los objetos que componen el sistema de forma temporal.

#### **3.3.1 Secuencia Agente.**



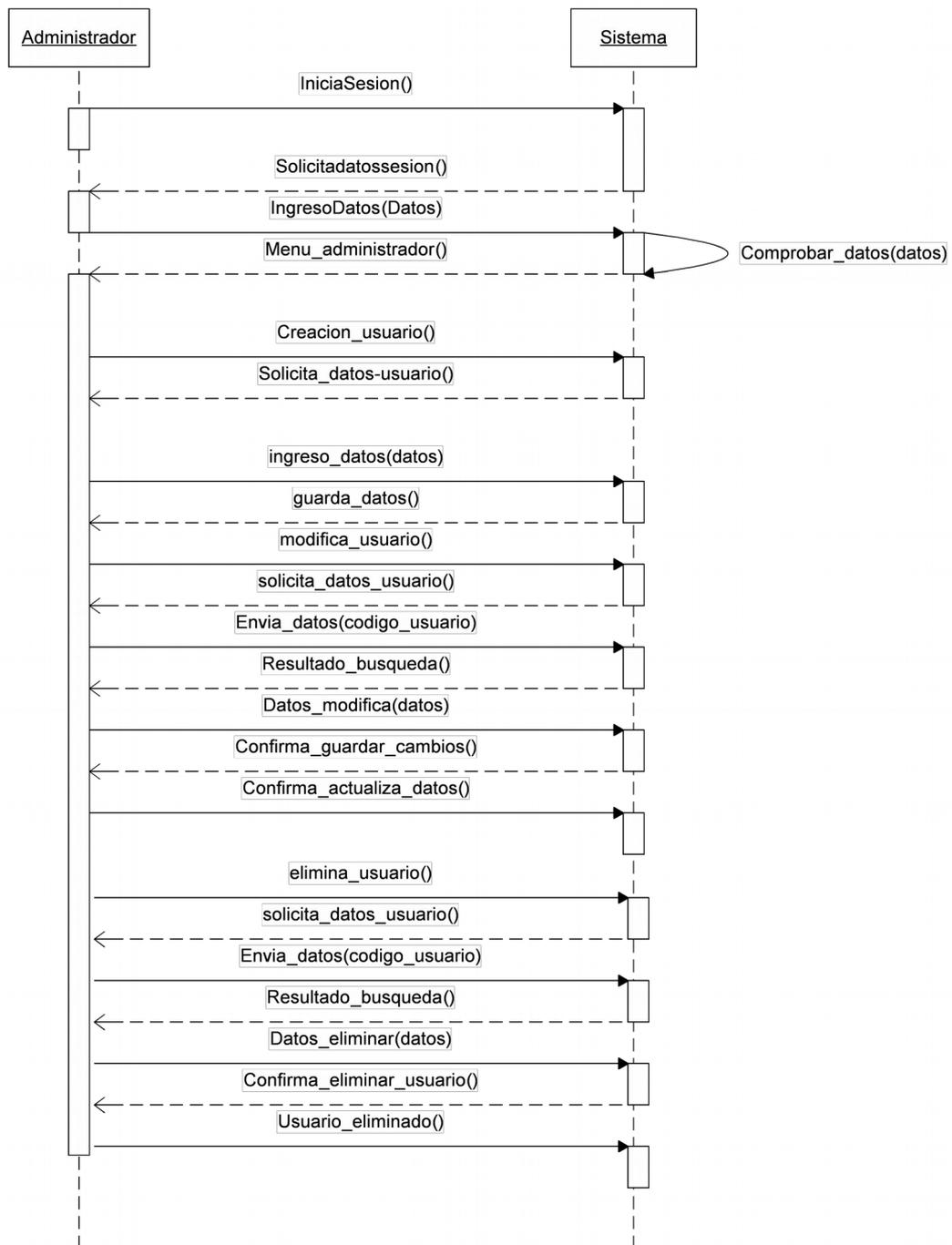
**Figura Secuencia Agente**

### 3.3.2 Secuencia Supervisor



**Figura Secuencia Supervisor**

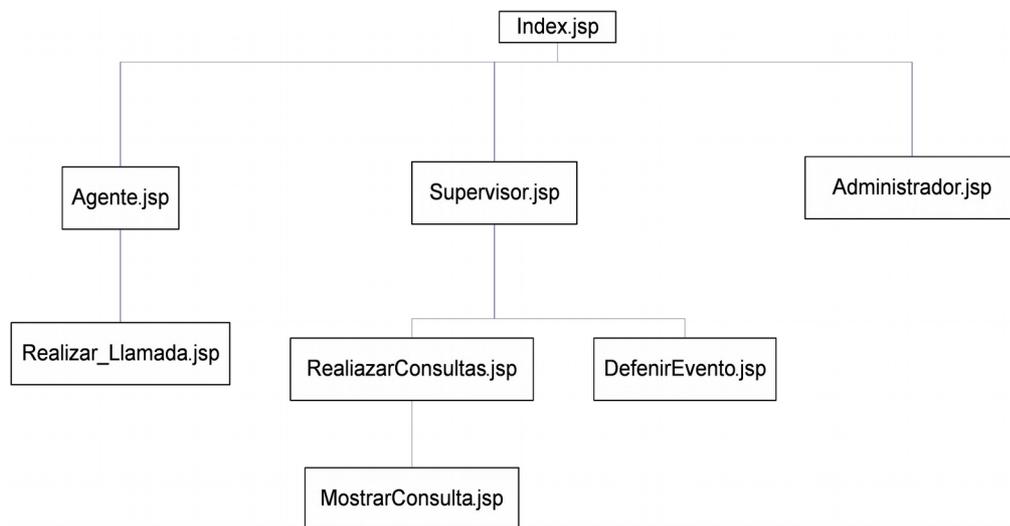
### 3.3.3 Secuencia Administrador



**Figura Secuencia Administrador**

### 3.4 Diagrama de Flujo de Ventanas

Este diagrama nos ayuda a ver la secuencia de las pantallas de mi aplicación.



**Figura Flujo de Ventanas**

### 3.5 Diseño de la Interfaz

El diseño de la interfaz describe como se comunica mi aplicación consigo mismo, y con los usuarios que lo emplean

### 3.5.1 Pagina Inicial

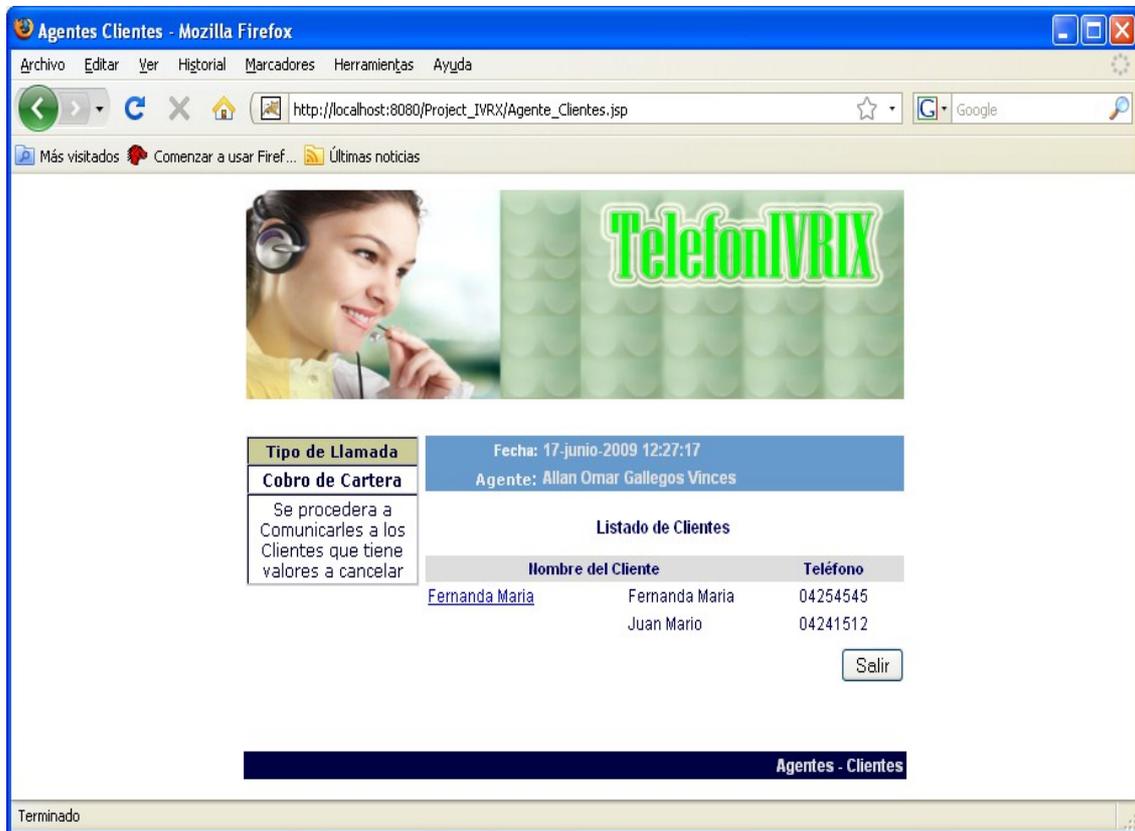
Esta pantalla es la que aparecerá a todos los usuarios de mi aplicación, en donde tendrá que ingresar el usuario y su contraseña para poder continuar.



**Figura Inicio**

### 3.5.2 Pagina Agente

En esta pantalla el agente visualizara todos los clientes que le ha otorgado el sistema para realizar las respectivas llamadas. Aquí el agente observar su nombre de usuario y una lista de clientes a llamar donde le indica el nombre y el numero a que va a llamar, en la parte izquierda le indica el tipo de evento que realizara, que puede ser cobros o publicidad. Cabe recalcar que el agente tendrá que comenzar por el primero cliente que el sistema le indique.



**Figura Agente**

### 3.5.3 Pantallas Realizar Llamada

Aquí el agente podrá tener mas detalle del cliente como el nombre completo, dirección, teléfono y tenga la facilidad de cambiar ciertos campos si se amerita el caso.

En la parte inferior el agente puede calificar si la llamada ah sido exitosa o no ha sido exitosa y tiene un campo para que pueda ingresar cualquier comentario de la llamada realizada.

También el agente tendrá una pequeña ayuda a seguir para el momento que este realizado la llamada, es una pequeña guía de cómo el agente tiene que expresarse el agente en el momento que este dialogando con el cliente.



**Figura Sistema Datos Clientes**

### 3.5.4 Pantalla Supervisor

Aquí el supervisor podrá observar todos los agentes que tiene a cargo y cuales están conectados y lo que se encuentran desconectados, el tiempo que están en línea (Minutos Hablados) de modificar si desea que

el agente se les graben las llamadas que el realiza de ese momento en adelante.

El supervisor tendrá dos opciones a seguir que son la de realizar consultas y las definir el tipo de evento.

Supervisor TelefonIVRX - Mozilla Firefox

http://localhost:8080/Project\_IVRX/Supervisor.jsp

Fecha: 26-junio-2009 10:22:59  
Agente: Benito Darwin Floro Villacres

Listado de Agentes

Agente	Estado	Tiempo Conectado	Minutos Hablados	Graba
Marcela Romina Tutiven Dalmia (agente03)	OFF	0:00 Segundos	2 Minutos	<input checked="" type="checkbox"/>
Eduardo Alvarado (agente04)	OFF	0:00 Segundos	0 Minutos	<input type="checkbox"/>
Cleopatra Chantal Quiñonez Tenorio (agente05)	OFF	0:00 Segundos	0 Minutos	<input type="checkbox"/>

Actualizar

Terminado

**Figura Supervisor**

### 3.5.5 Pantalla Realizar Consultas

Aquí el supervisor podrá realizar diferentes tipos de consultas como por fecha en donde tendrá que ingresar un rango de fechas, por agentes en

donde ingresar el código del agente o el nombre del agente, por llamadas las ultimas llamadas realizadas, por clientes aquí ingresar el numero de cedula del cliente o el nombre o por status de llamadas que pueden ser fallidas, exitosas o las que se grabaron.

### 3.5.5.1 Consultas Por fecha

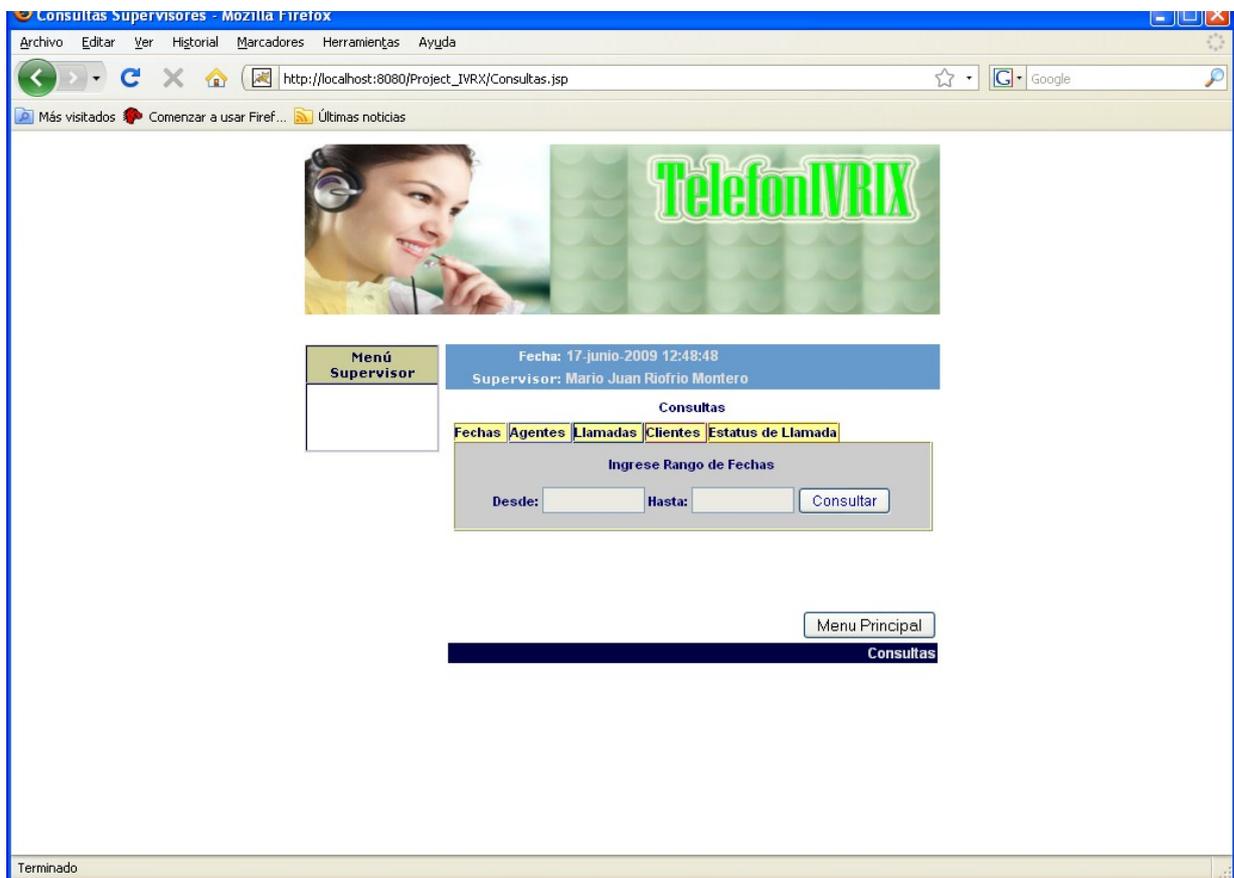


Figura Consulta x Fechas

### 3.5.5.2 Consulta Por Agente

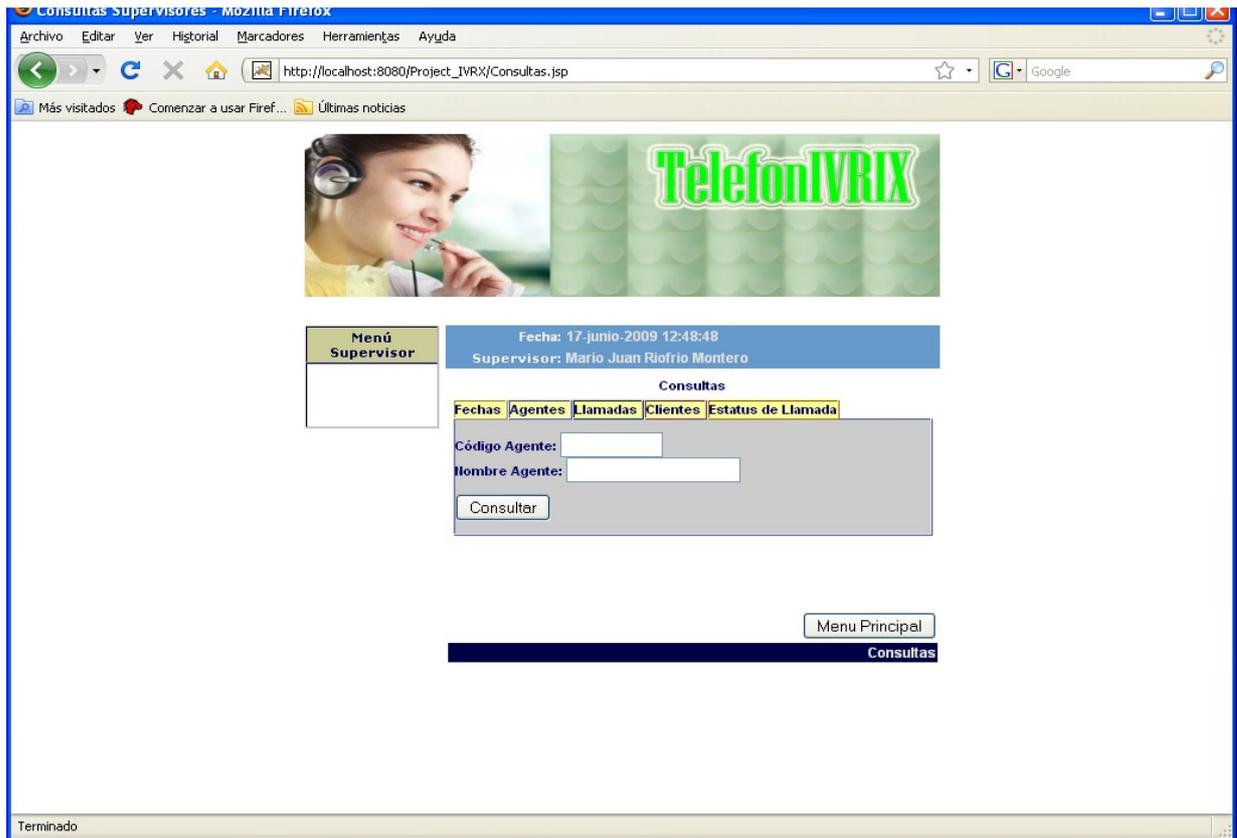


Figura Consulta x Agente

### 3.5.5.3 Consultas Por Llamadas

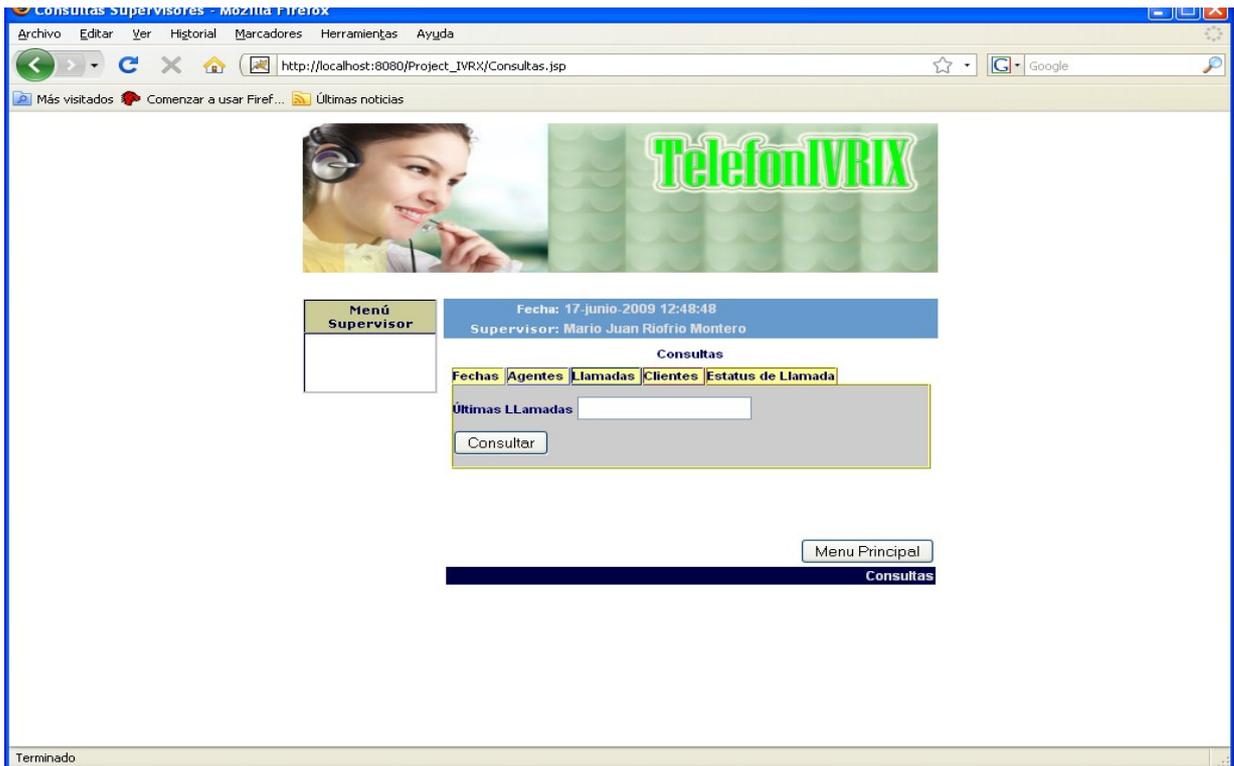


Figura Consulta x Llamadas

### 3.5.5.4 Consultas Por Clientes.



Figura Consulta x Clientes

### 3.5.5 Consultas Por Status de Llamadas.

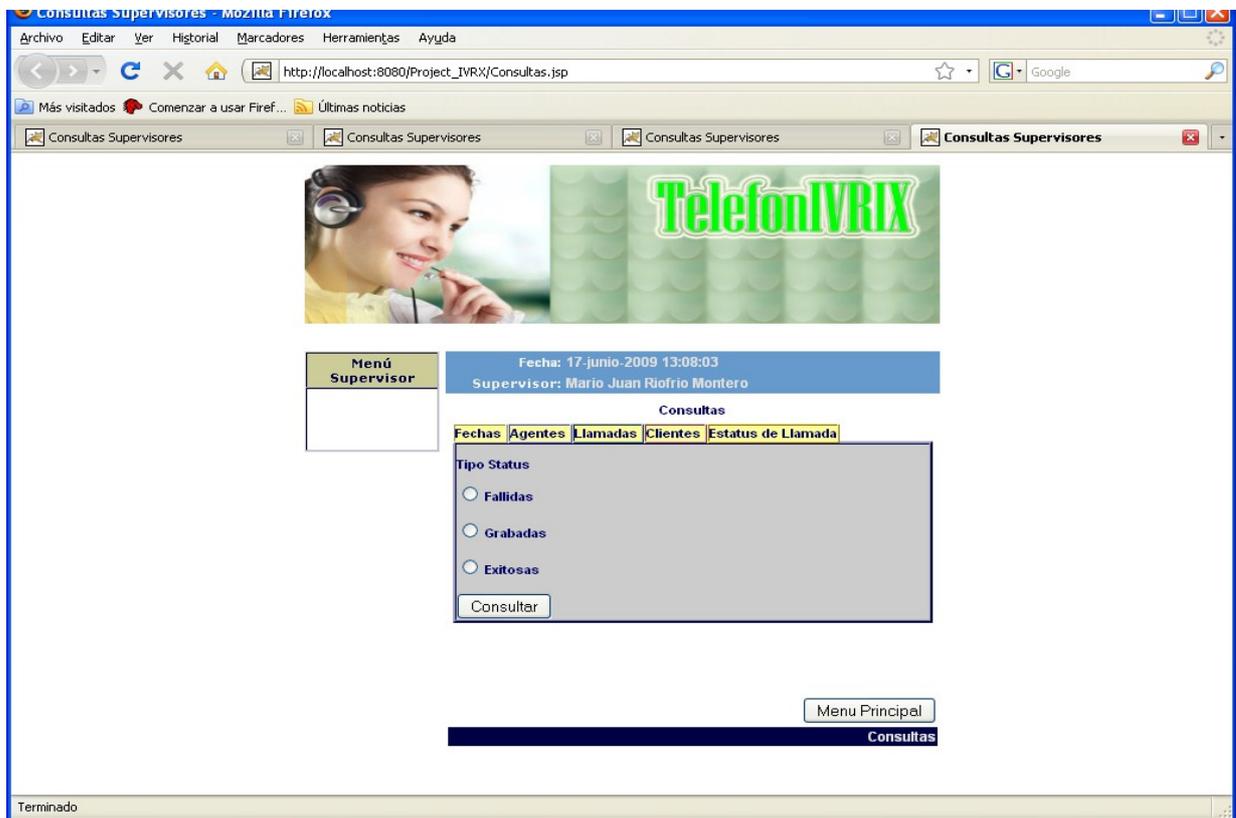
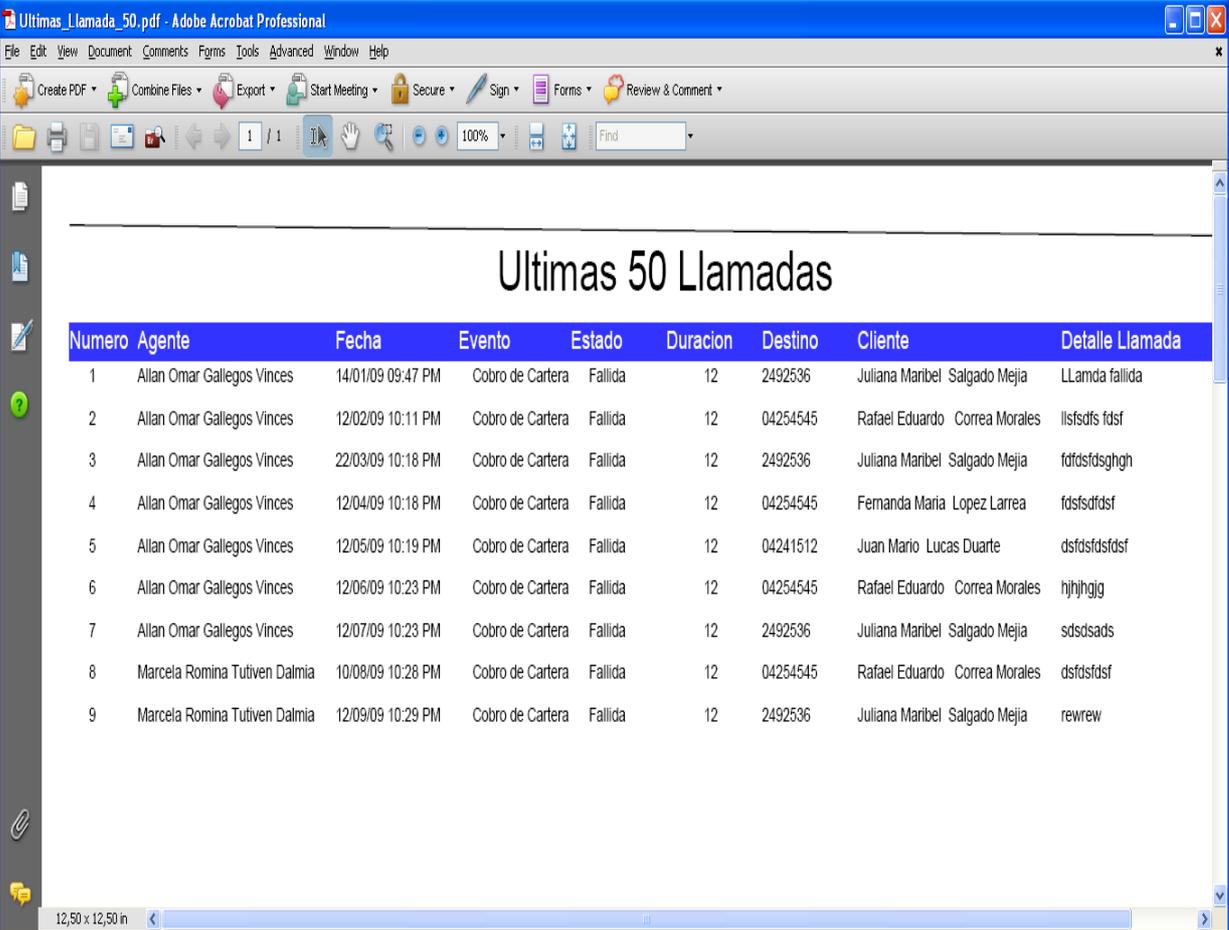


Figura Consulta x Status Llamada

### 3.5.6 Pantalla Mostrar Consulta

En esta pantalla el supervisor visualizar el resultado de la consulta dependiendo del tipo de consulta que haya seleccionado el supervisor en un archivo PDF el cual le dará la opción de que la pueda imprimir o enviarla por email, según el caso lo amerite.



The screenshot shows a PDF document titled "Ultimas\_Llamada\_50.pdf" in Adobe Acrobat Professional. The document content is a report titled "Ultimas 50 Llamadas" which contains a table with the following data:

Numero	Agente	Fecha	Evento	Estado	Duracion	Destino	Cliente	Detalle Llamada
1	Allan Omar Gallegos Vincés	14/01/09 09:47 PM	Cobro de Cartera	Fallida	12	2492536	Juliana Maribel Salgado Mejia	LLamda fallida
2	Allan Omar Gallegos Vincés	12/02/09 10:11 PM	Cobro de Cartera	Fallida	12	04254545	Rafael Eduardo Correa Morales	llsfdsfs fdsf
3	Allan Omar Gallegos Vincés	22/03/09 10:18 PM	Cobro de Cartera	Fallida	12	2492536	Juliana Maribel Salgado Mejia	fdfsfdsfghgh
4	Allan Omar Gallegos Vincés	12/04/09 10:18 PM	Cobro de Cartera	Fallida	12	04254545	Fernanda Maria Lopez Larrea	fdfsfdfdsf
5	Allan Omar Gallegos Vincés	12/05/09 10:19 PM	Cobro de Cartera	Fallida	12	04241512	Juan Mario Lucas Duarte	dsfdfsfdsf
6	Allan Omar Gallegos Vincés	12/06/09 10:23 PM	Cobro de Cartera	Fallida	12	04254545	Rafael Eduardo Correa Morales	hjhjhjgig
7	Allan Omar Gallegos Vincés	12/07/09 10:23 PM	Cobro de Cartera	Fallida	12	2492536	Juliana Maribel Salgado Mejia	sdsdsads
8	Marcela Romina Tutiven Dalmia	10/08/09 10:28 PM	Cobro de Cartera	Fallida	12	04254545	Rafael Eduardo Correa Morales	dsfdfsfdsf
9	Marcela Romina Tutiven Dalmia	12/09/09 10:29 PM	Cobro de Cartera	Fallida	12	2492536	Juliana Maribel Salgado Mejia	rewrew

**Figura Reporte a**

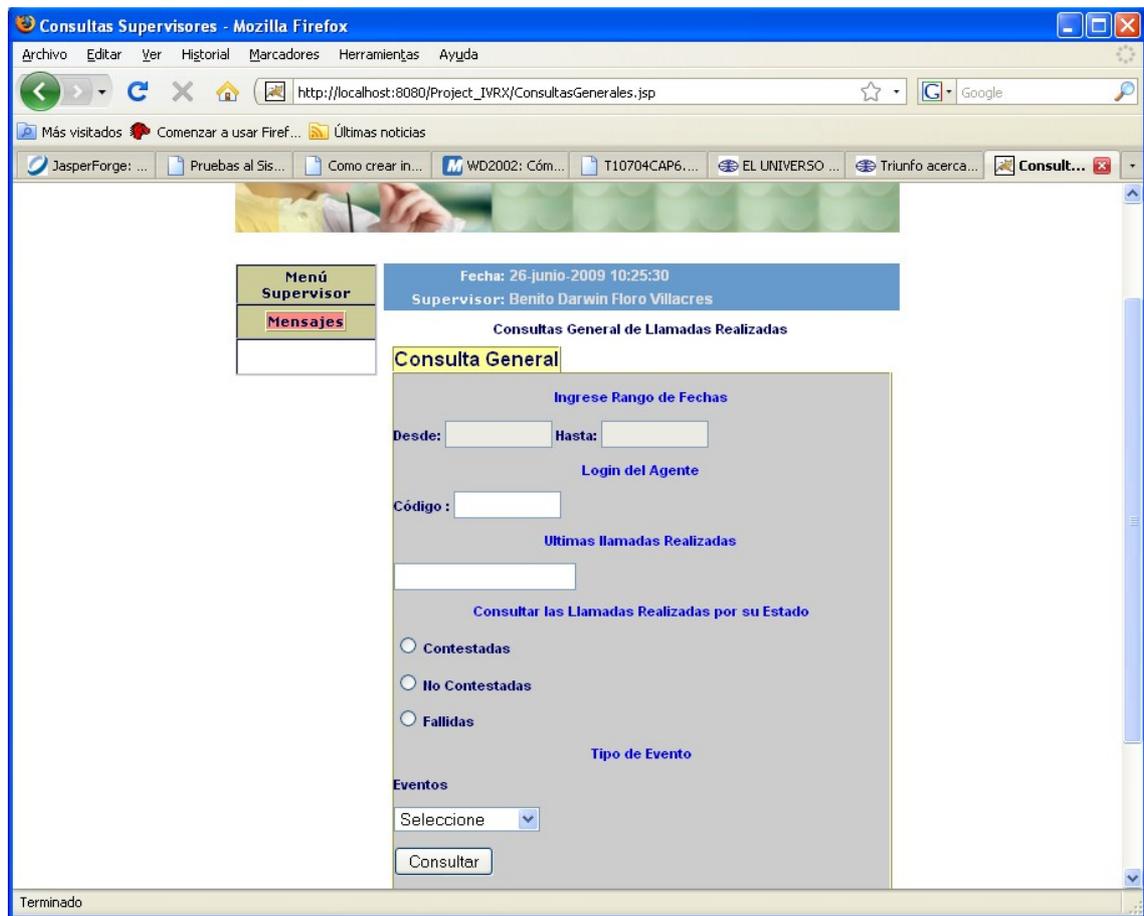
The screenshot shows a PDF document titled 'Clientes-Llamadas' displayed in Adobe Acrobat Professional. The document contains a table with the following data:

Numero	Nombre Cliente	Estado	Fecha	Evento	Duracion	Destino	Agente	Detalle
1	Juliana Maribel Salgado Mejia	Fallida	14/01/09 09:47 PM	Cobro de Cartera	12	2492536	Gallegos Vinces	LLamda fallida
2	Juliana Maribel Salgado Mejia	Fallida	22/03/09 10:18 PM	Cobro de Cartera	12	2492536	Gallegos Vinces	fdfdsfdsggh
3	Juliana Maribel Salgado Mejia	Fallida	12/07/09 10:23 PM	Cobro de Cartera	12	2492536	Gallegos Vinces	sdsdsads
4	Juliana Maribel Salgado Mejia	Fallida	12/09/09 10:29 PM	Cobro de Cartera	12	2492536	Tutiven Dalmia	rewrew

**Figura Reporte b**

### 3.5.7 Pantalla Consulta General

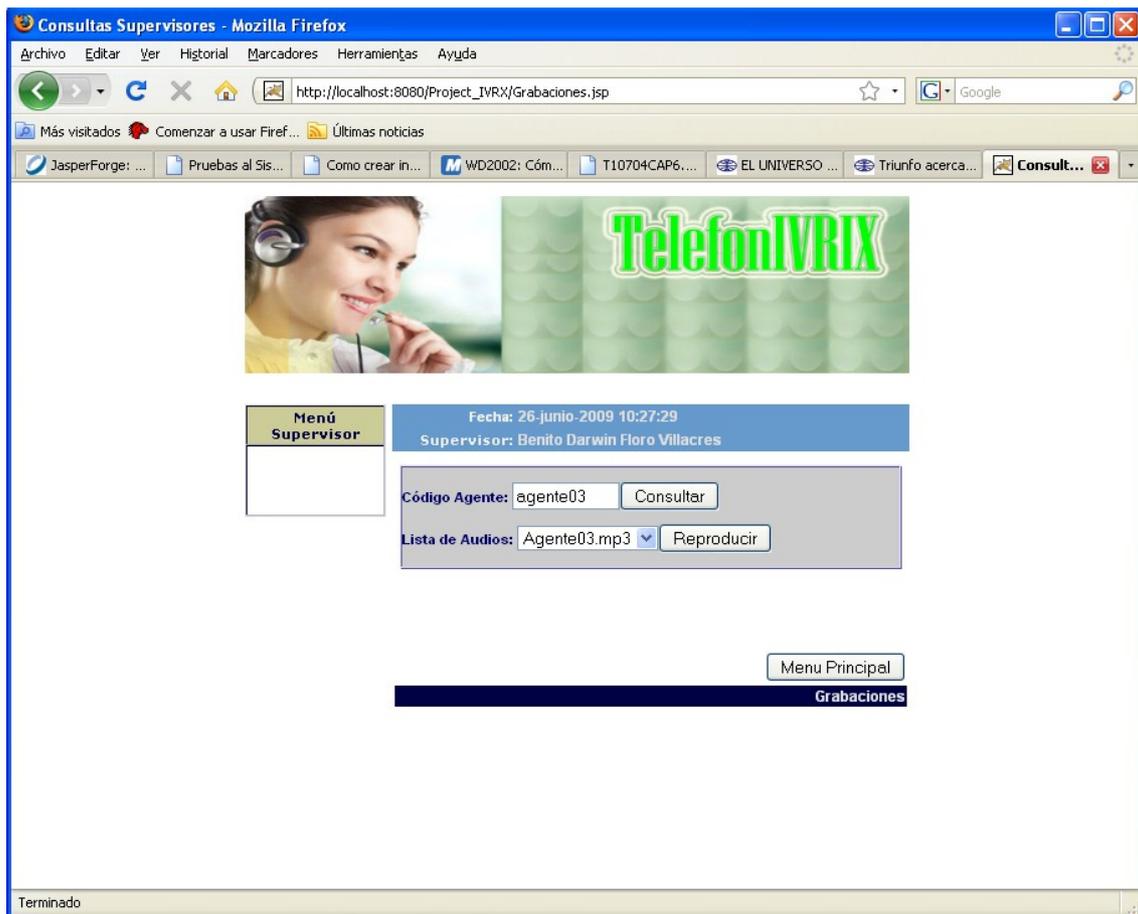
Aquí el supervisor podrá realizar consultas a su manera, manejando los parámetros que el desee acorde a la necesidad para el cual la requiera.



**Figura Consulta General**

### 3.5.8 Pantalla Grabaciones

Muestra todas las grabaciones que ha realizado el supervisor según el agente que haya solicitado buscar.



**Figura Grabaciones**

### 3.5.9 Pantalla Definir Evento

Aquí el supervisor puede elegir que tipo de evento desea que sus agentes realicen, existe dos tipos de eventos que son el de publicidad y el cobro. También tiene un área donde el supervisor debe ingresar la guía que servirá a los agentes en el momento de realizar la llamada.

También se asignarán sectores que serán organizadas por parroquias según el supervisor asigne al agente.

Supervisor-Eventos - Mozilla Firefox

http://localhost:8080/Project\_IVRX/Evento.jsp

Más visitados Comenzar a usar Firef... Últimas noticias

Consultas Supervisores Consultas Supervisores Consultas Supervisores Supervisor-Eventos

**Evento Actual**

**Cobro de Cartera**

**Zona**

Toda la Ciudad

**Detalle**

Promocionar un producto de belleza para las mujeres de hoy

Fecha: 17-junio-2009 13:14:05

Supervisor:: Mario Juan Riofrio Montero

**Eventos**

Tipo de Evento:

Parroquias:

Toda la Ciudad

Ayacucho

Bolívar

Carbo Concepcion

Febres Cordero

Garcia Moreno

Letamendi

9 de Octubre

Olmeda/San Alejo

Francisco Roca

Rocafuerte

Sucre

Tarqui

Urdaneta

Ximena

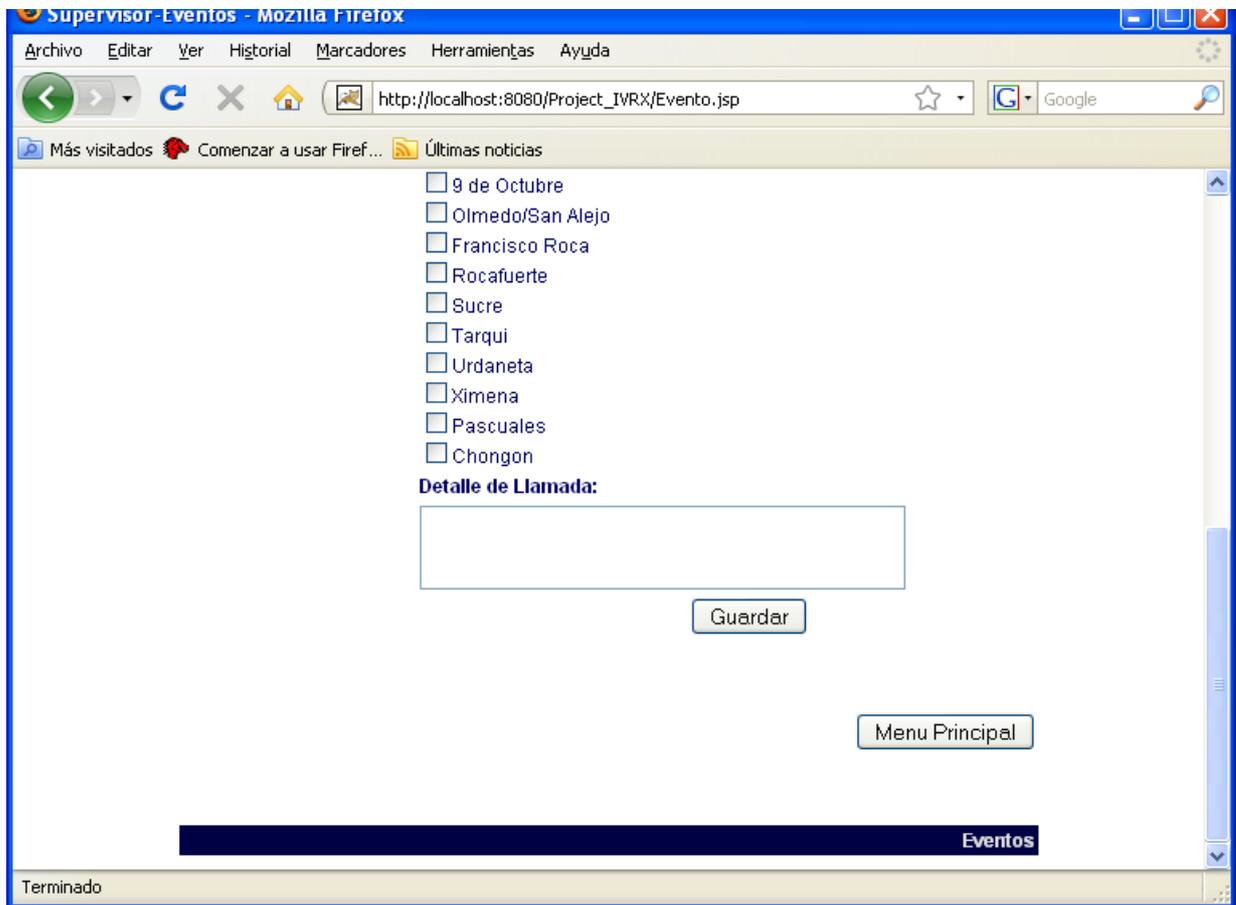
Cobro de Cartera

Cobro de Cartera

Telemercadeo

Terminado

**Figura tipo Evento a**



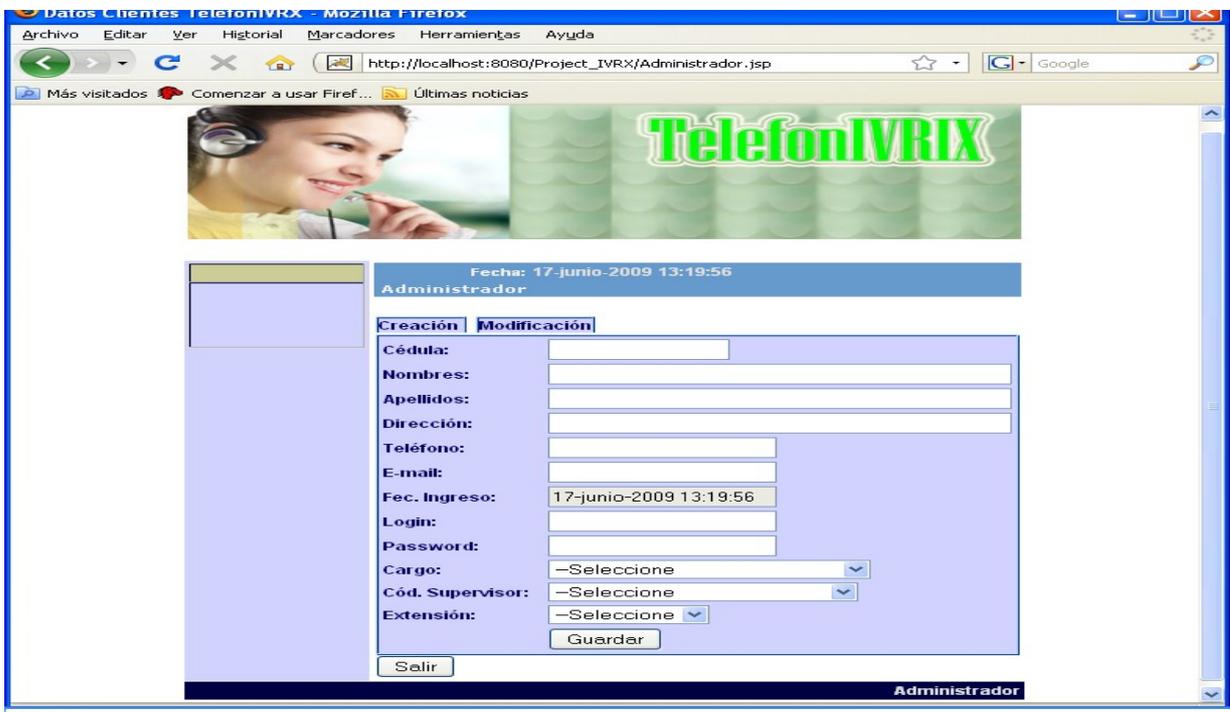
**Figura tipo Evento b**

### 3.5.10 Pantalla Administrador.jsp

Aquí el administrador puede crear y modificar los usuarios de la aplicación ya sea agentes o supervisores.

## Creación

Aquí se ingresan los datos para el usuario a crear, cabe mencionar que en caso de que el usuario sea de tipo Agente se deberá asignar un supervisor y una extensión que es con la cual ellos realizaran las llamadas.



The screenshot shows a web browser window titled 'Datos Clientes TelefonIVRIX - Mozilla Firefox'. The address bar shows 'http://localhost:8080/Project\_IVRX/Administrador.jsp'. The page header includes the 'TelefonIVRIX' logo and a date/time stamp: 'Fecha: 17-junio-2009 13:19:56 Administrador'. The main content area has two tabs: 'Creación' (selected) and 'Modificación'. The 'Creación' form contains the following fields:

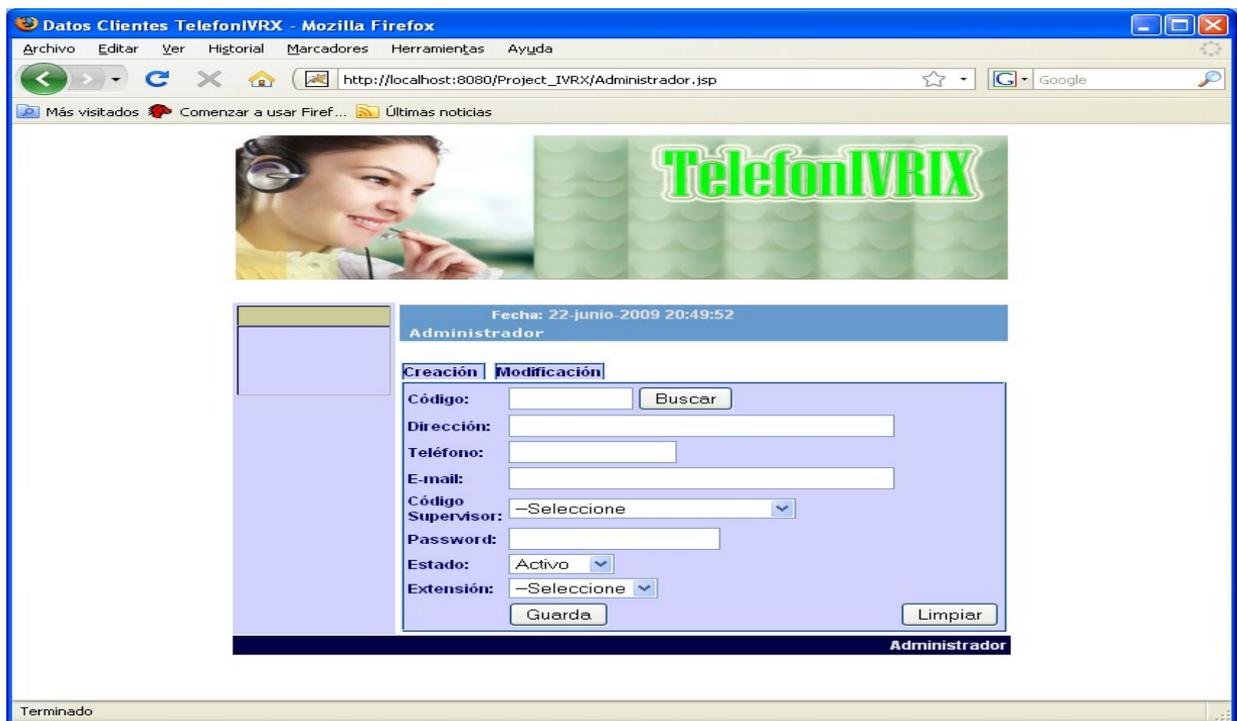
- Cédula:
- Nombres:
- Apellidos:
- Dirección:
- Teléfono:
- E-mail:
- Fec. Ingreso:
- Login:
- Password:
- Cargo:
- Cód. Supervisor:
- Extensión:

Buttons: 'Guardar' and 'Salir'.

**Figura Administrador - Creacion**

## Modificación

En esta pantalla realizaremos las modificaciones que sean requeridas por el supervisor, incluyendo el cambio de la contraseña.



**Figura Administrador - Modificación**

# CAPITULO 4

## 4. Codificación

### 4.1 Codificación de Procesos Principales

#### 4.1.1 Proceso de Llamadas.

Este proceso se realizará cada vez que el agente se comunique con el cliente.

El agente ingresa al sistema y luego se le asignará una estructura de usuarios a los cuales deberá llamar.

El sistema le indicará que cliente llamar, una vez seleccionado el sistema nos enviará a una página donde se mostrará los datos del cliente.

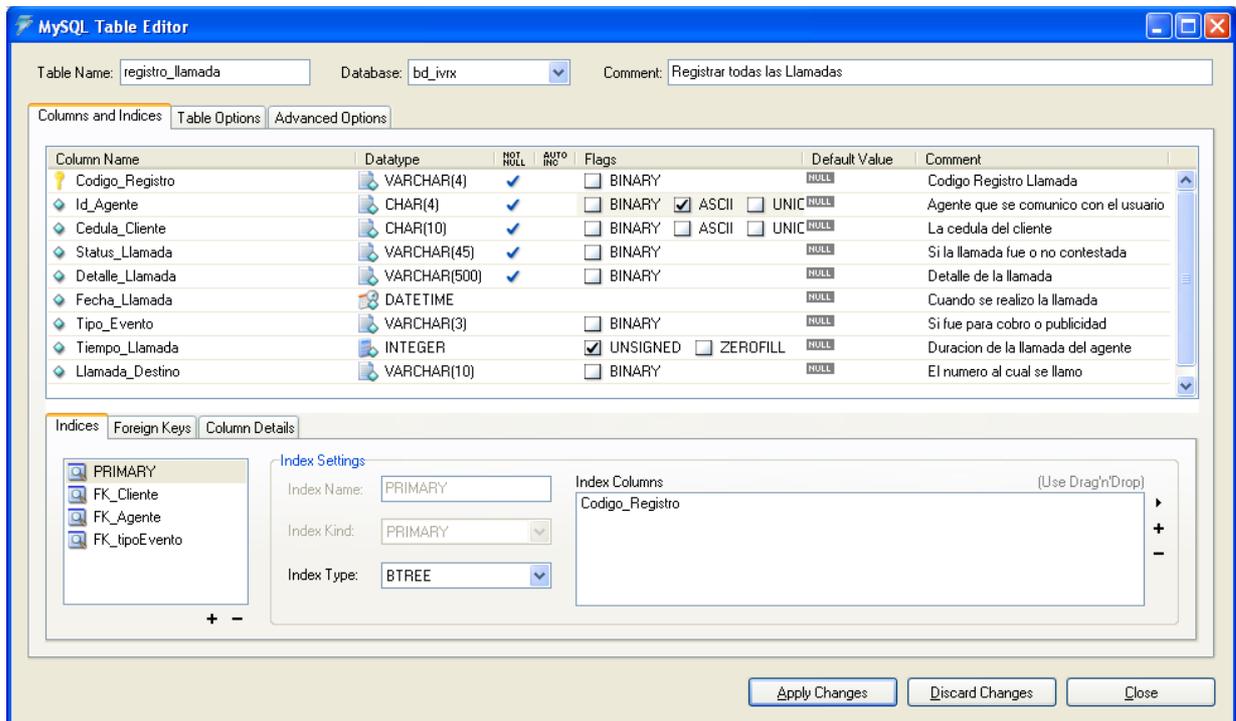
En esta pagina encontraremos el botón *llamar* que es el que establecerá la llamada hacia el cliente, el Asterisk es el encargado de realizar la llamada entre el agente y el cliente.

Cuando el cliente contesta, automáticamente se comienza a registrar el inicio de la llamada. Una vez culminada la conversación se registrará el fin de llamada.

Una vez que damos click en *Actualizar* la duración de la llamada será guardada en la Base de Datos junto con la extensión del agente y el día que se realizó la llamada.

## Registro Llamadas

**Objetivo:** Almacena la duración de la llamada que hizo el agente.



#### **4.1.2 Proceso de Grabación de Llamadas.**

El supervisor es la persona encargada de realizar esta acción, el una vez que ingrese al sistema registrará todos los agentes que estén asignados a el y mostrando el status del agente que será ON si el Agente esta activo en el sistema y OFF en caso de no estarlo.

El supervisor seleccionará con un visto en la columna Grabación donde se realizará la grabación de la llamada del agente seleccionado, para que luego este archivo de audio (.wav) sea almacenado en una ruta del Asterisk.

Luego se crea un crond (tarea programada en Linux) que permitirá realizar un backup de las llamadas grabadas en el directorio donde se encuentra el sistema.

Estos archivos serán almacenados ordenadamente, siendo así que se creara una ruta para cada agente y dependiendo de el se guarde en la ruta correspondiente.

#### **4.1.3 Proceso de Generar Reportes**

Este proceso es manejado por los supervisores que son los encargados de realizar las respectivas consultas de los agentes.

Sus datos son obtenidos de la tabla de Registro Llamadas acorde a los parámetros que sean enviados por el supervisor.

Los archivos que se generen son PDF y permitirá que sen impresos o enviados por correo, según la situación lo amerite.

#### **4.1.4 Proceso de Estructura Agente-Clientes.**

Este proceso permitirá que a los agentes se les asigne aleatoriamente un grupo de clientes deudores, y que son organizados por un job en la Base de Datos.

Si los clientes no contestaron las llamadas al siguiente día es seteado el nivel de llamada de Llamada Fallida a No contestado.

# Capitulo 5

## 5. Pruebas

### 5.1 Evaluación Operacional

Se llevaron a cabo dos tipos de pruebas en el sistema. Las primeras pruebas funcionales se hicieron para estar seguros que el sistema está funcionando como se espera o como fue diseñado.

Estas pruebas se llevaron a cabo creando un grupo de casos de prueba con datos de prueba. Las pruebas incluyeron la validación de campos, reglas de negocio, integración de procesos y la entrada de datos.

Las pruebas fueron realizadas en ambos ambientes presentados de cliente/servidor.

Las segundas pruebas fueron preparadas para asegurarse que el sistema era capaz de manejar el volumen de datos y el tiempo de respuesta al usuario era el esperado.

## **5.2 Evaluación Usuarios**

La facilidad con la que los agentes permiten realizar su trabajo es satisfactoria, previo a una capacitación en la que se le indica el uso de la herramienta.

No se presentaron inconvenientes al momento de que el usuario comenzó a usar el sistema siendo así evaluadas la mayoría de nuestras validaciones.

# CAPITULO 6

## 6. Conclusiones y Recomendaciones

La tecnología IVR de tipo outbound puede ser implementado en cualquier compañía que desee ofrecer servicios o promociones y también dedicarse al Cobro de Cartera siendo así una gran inversión economizando costos gracias a la infraestructura que posee el sistema

El call center en una institución financiera constituye un punto estratégico en el manejo del negocio. El uso del IVR outbound como herramienta para informativo, ayuda a reducir el tráfico telefónico en el call center, y es un mecanismo de informar las 24 horas (llamadas programadas), lo cual mejora el nivel de atención al cliente.

El IVR puede tener diferentes aplicaciones, que dependen de la necesidad del usuario, esta es una gran ventaja del sistema ya que permite realizar IVR y conexión con cualquier motor de bases de

datos a través de los ejecutables diseñados con el objetivo específico de la consulta.

La tarjeta DIGIUM TDM400 es susceptible a la estática por lo que se recomienda un manejo adecuado de la misma.

Para corregir errores en el IVR, primero se debe verificar el correcto desempeño del software de conexión con la base de datos, luego se debe verificar el diagrama de flujo del IVR, finalmente realizar un debug de los datos que está obteniendo el IVR creando archivos de texto que nos muestren el valor de las variables que maneja el IVR,

Tener un servidor de respaldo en caso de que existan fallas en el sistema o por algún motivo se dañe el servidor y poder así tener una contingencia del sistema y no perder la operabilidad del día a día en la empresa.

## **Bibliografía**

[http://www.nettix.com.pe/manuales/instalacion\\_de\\_servidor\\_centos\\_5.2/Page-3.html](http://www.nettix.com.pe/manuales/instalacion_de_servidor_centos_5.2/Page-3.html)

[http://www.contactcentervoip.com/ES/taxonomy\\_menu/9/44](http://www.contactcentervoip.com/ES/taxonomy_menu/9/44)

<http://www.voipforo.com/asterisk/configuracion-sip-conf.php>

<http://www.alcancelibre.org/staticpages/index.php/como-ekiga-asterisk>

[http://www.counterpath.net/assets/files/191/X-Lite3.0\\_UserGuide.pdf](http://www.counterpath.net/assets/files/191/X-Lite3.0_UserGuide.pdf)

<http://asterisk-java.org/development/tutorial.html>

[http://www.contactcentervoip.com/ES/taxonomy\\_menu/9/30](http://www.contactcentervoip.com/ES/taxonomy_menu/9/30)

<http://curso--de-asterisk.blogspot.com/2009/03/dial-plan-features.html>

<http://documentacion.irontec.com/cursoAsteriskVozIP-3-introduccionAsterisk.pdf>

<http://www.voztovoice.org/?q=node/20>

# **MANUAL TECNICO**

# Indice General

<b>Manual Técnico</b> .....	1
Introducción.....	2
<b>Capítulo I</b> .....	3
Introducción.....	3
1.1 Herramientas Utilizadas.....	3
1.1.1 Sistema Operativo Centos.....	4
1.1.2 Asterisk.....	4
1.1.3 Softphone Xlite .....	4
1.2 Instalación Centos 5.2.....	5-20
1.3 Arquitectura Asterisk .....	20-21
1.4 Conceptos Previos.....	21-23
1.5 Instalación Asterisk.....	24-26
1.6 Configuración Asterisk.....	27-37
1.7 Funcionalidad del Asterisk.....	38
1.8 Instalación Softphone.....	39
<b>Capitulo II</b> .....	40
Introducción.....	40
2.1 Herramientas Utilizadas.....	40
2.1.1 My Eclipse.....	41
2.1.2 JDK.....	41
2.1.3 Tomcat.....	41-42
2.2 Comunicación de Asterisk con Java.....	42
2.2.1 Manager API.....	42-44
<b>Capitulo III</b> .....	45
Introducción .....	45
3.1 Herramientas Utilizadas.....	45
3.1.1 My SQL .....	45
3.1.2 Hibernate .....	46
3.2 DER .....	46-47
3.3 Diagrama de Clases .....	47-48
3.4 Diagrama de Secuencias.....	49
3.4.1 Secuencia Agente .....	49
3.4.2 Secuencia Supervisor .....	50
3.4.3 Secuencia Administrador .....	51
3.5 Diagrama Flujo de Ventanas .....	52
3.6 Creación de Objetos en la BD .....	52-61
3.7 Procesos Principales .....	61
3.7.1 Proceso llamadas .....	61-66
3.7.2 Proceso Reportes.....	66-78
3.7.3 Proceso estructura Agente.Clientes .....	79-83
3.7.4Proceso Grabaciones Llamadas .....	84-88
3.8 Jasper Reports .....	89-91

<b>Manual Usuario</b> .....	92
Introducción .....	93
1.1 Ingreso del Sistema .....	93-94
1.2 Lista de Clientes .....	94-95
1.3 Datos del Cliente .....	96-98
1.4 Lista de Agentes .....	99-100
1.5 Realizar Consultas .....	100-105
1.6 Definir Tipo Evento .....	105-108
1.7 Creación y Modificación de usuarios .....	108-110
Bibliografía .....	111

## **Introducción**

En los capítulos que se presentarán a continuación se mostrarán las herramientas utilizadas para permitir que se realicen las respectivas llamadas a los usuarios desde la aplicación JSP.

Para su mejor entendimiento lo hemos dividido en tres capítulos:

**CAPITULO I: ASTERISK**

**CAPITULO II: ASTERISK – JAVA**

**CAPITULO III: JAVA – MY SQL**

# CAPITULO I

## Introducción

En este capítulo mostraremos lo que es el Asterisk, su arquitectura, requerimientos mínimos como instalar y su funcionalidad en el proyecto.

Podemos descargar Asterisk de la siguiente dirección:

**<http://www.asterisk.org/>**

### 1.1 Herramientas Utilizadas

A continuación mencionaremos las herramientas de software necesarias para el desarrollo de este módulo:

Requerimientos de Software:

- Sistema Operativo Ceñitos 5.2
- Asterisk 1.4
- Softphone X-Lite

### **1.1.1 Sistema Operativo Ceñitos**

CentOS es una versión libre de Red Hat, basada en su código fuente, que tiene las mismas bondades de éste. Incluye entornos KDE y GNOME, también incluye el instalador Anaconda, y el gestor de actualizaciones PUP.

### **1.1.2 Asterisk**

Asterisk es una aplicación de software libre (bajo licencia GPL) que proporciona funcionalidades de una central telefónica (PBX). Como cualquier PBX, se puede conectar un número determinado de teléfonos para hacer llamadas entre sí e incluso conectar a un proveedor de VoIP o bien a una RDSI tanto básicos como primarios.

### **1.1.3 Softphone X-Lite**

Es un software de telefonía IP que permite la comunicación entre dos servidores Asterisk o con dispositivos que manejen protocolos SIP.

## **1.2 Instalación de Ceñitos 5.2**

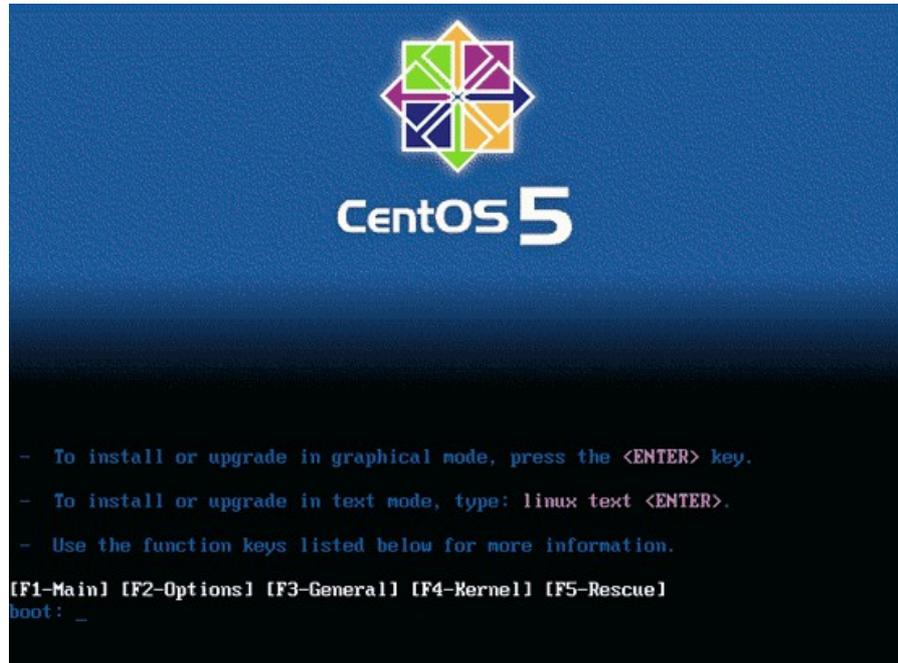
Requerimientos Mínimos:

- 192 MB de Memoria RAM
- 850 MB de disco duro
- Procesador Intel Pentium I

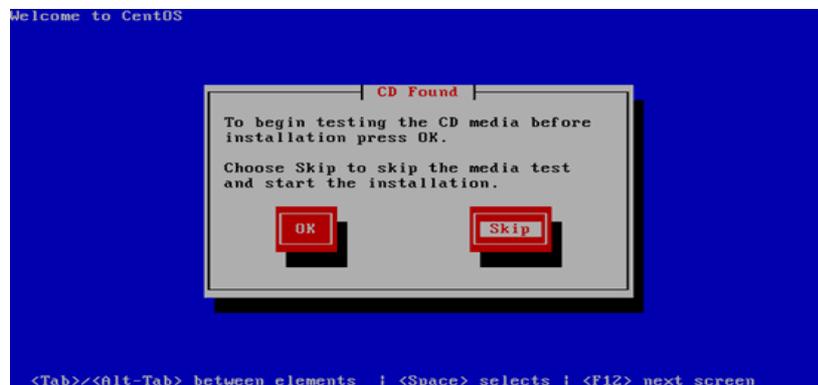
Para descargarlo lo podemos hacer de la siguiente dirección: <http://www.centos.org/>

Pasos para instalar:

Presiona entera y que el mismo CentOS detecte tu tipo de hardware y los controladores apropiados para el mismo.



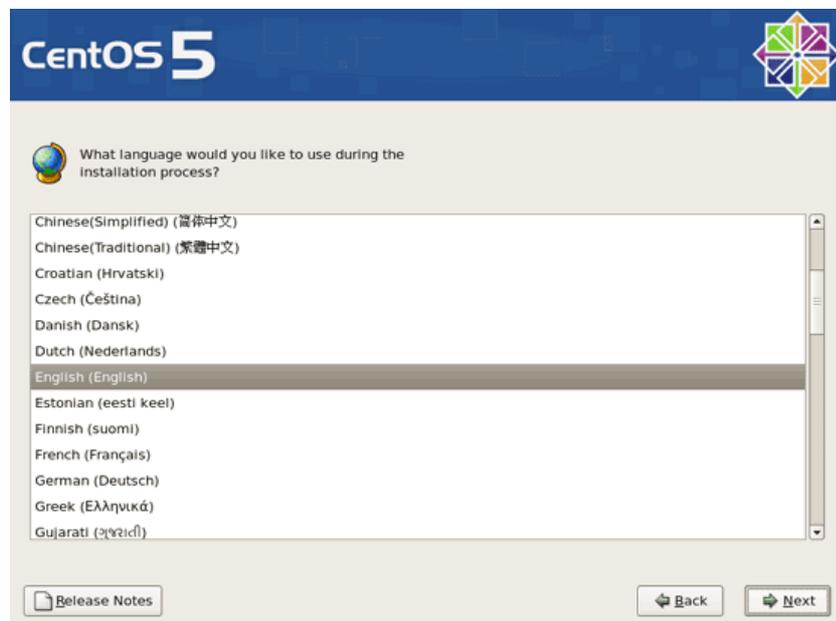
Aquí puedes darle "Skip" (Saltar o Pasar) Para que no busque errores físicos en el CD o DVD que has grabado, asumimos que esta libre de defectos.



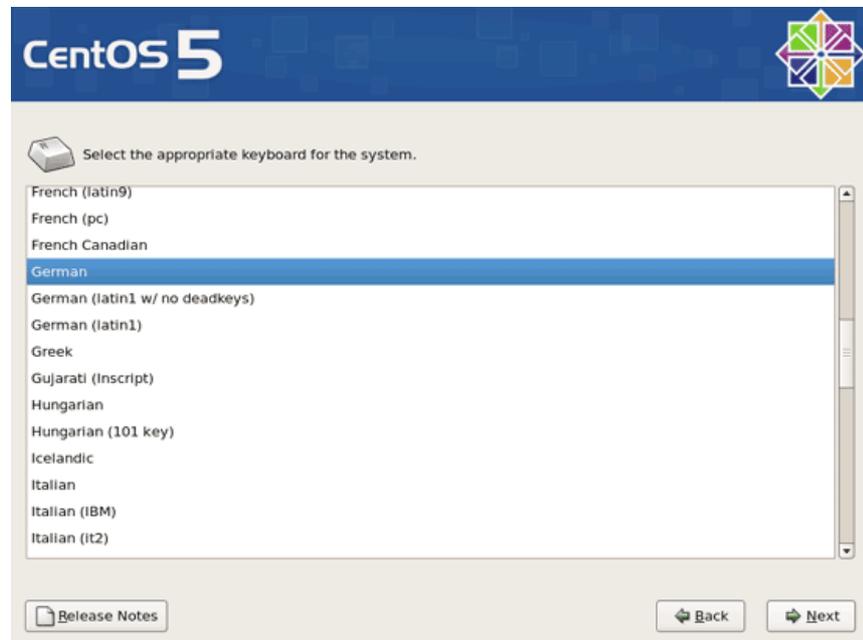
La pantalla de bienvenida de CentOS, presiona "Next"



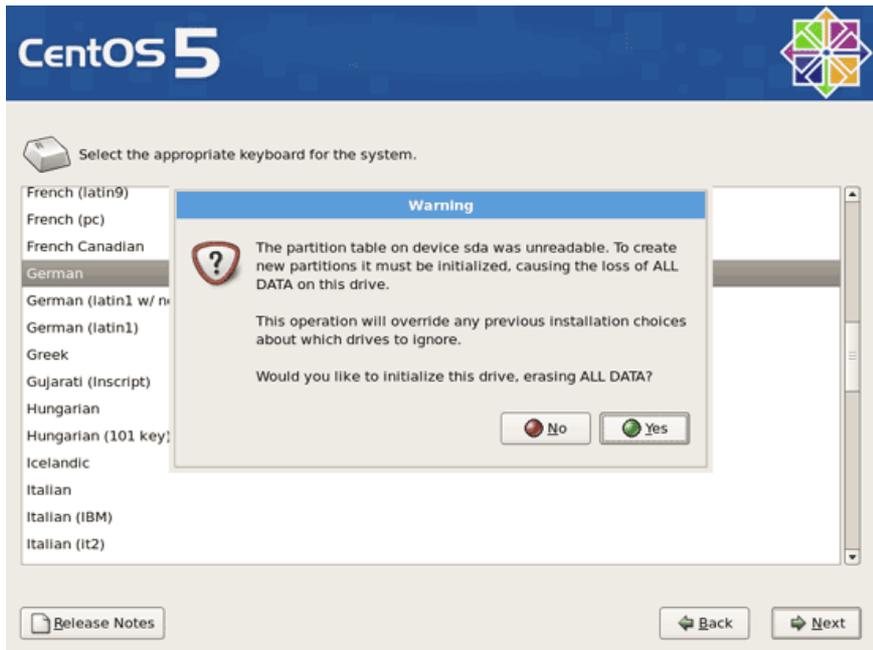
Aquí puedes escoger el idioma, por cuestiones de compatibilidad con caracteres, sugerimos en ingles



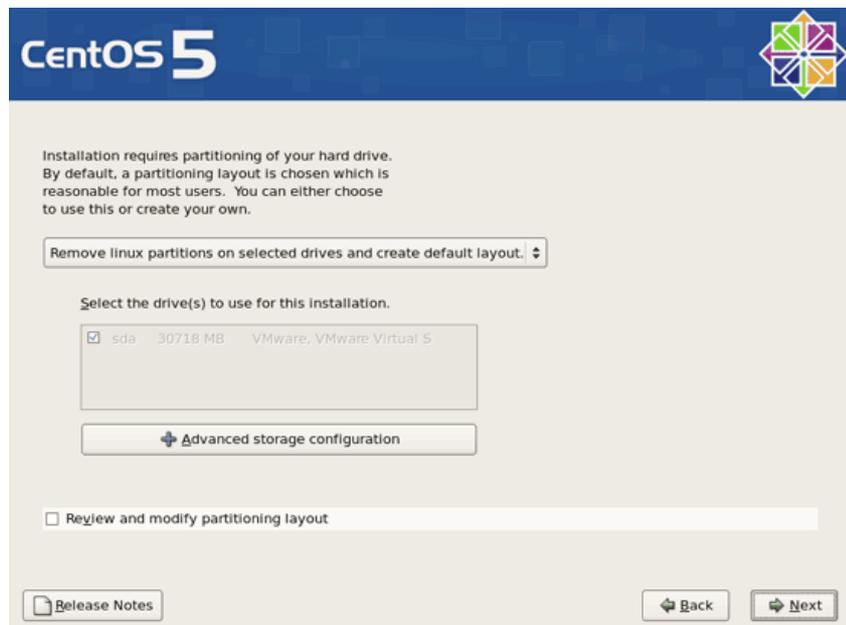
Selecciona el tipo de teclado, (español) o ingles dependiendo de que tipo de teclado que uses.



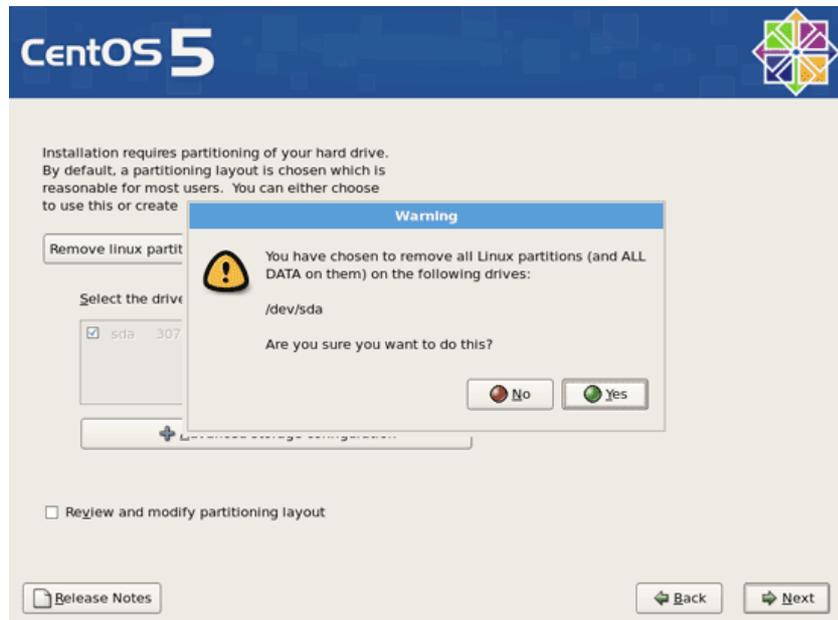
Solo será usado como servidor y que no será usado para otro sistema operativo sea Windows, por lo cual podemos eliminar todas las particiones del disco duro, para instalar CentOS.



Aquí se debe seleccionar el tipo de particionamiento que debes usar, en la mayoría de casos la propuesta por defecto esta bien, sin embargo para otro tipo de instalaciones específicas se requiere otro tipo de partición.



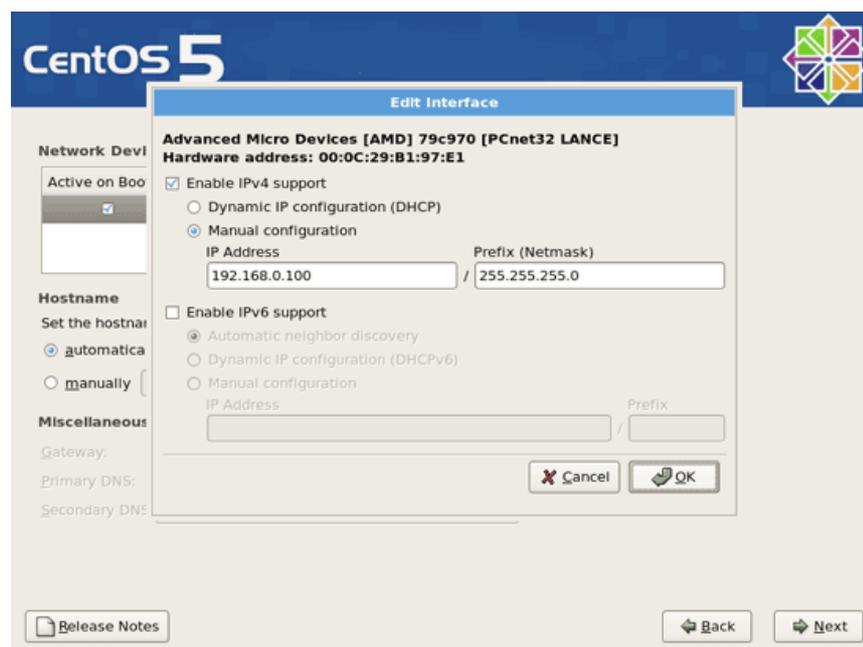
Aquí confirma que estas de acuerdo con eliminar las particiones existentes.



Aquí tienes que dar la configuración de red, por defecto CentOS propone DHCP, pero debido a que este va a ser un servidor dedicado sería convenientes que le configures una IP de tu red interna. Presiona "Edit"

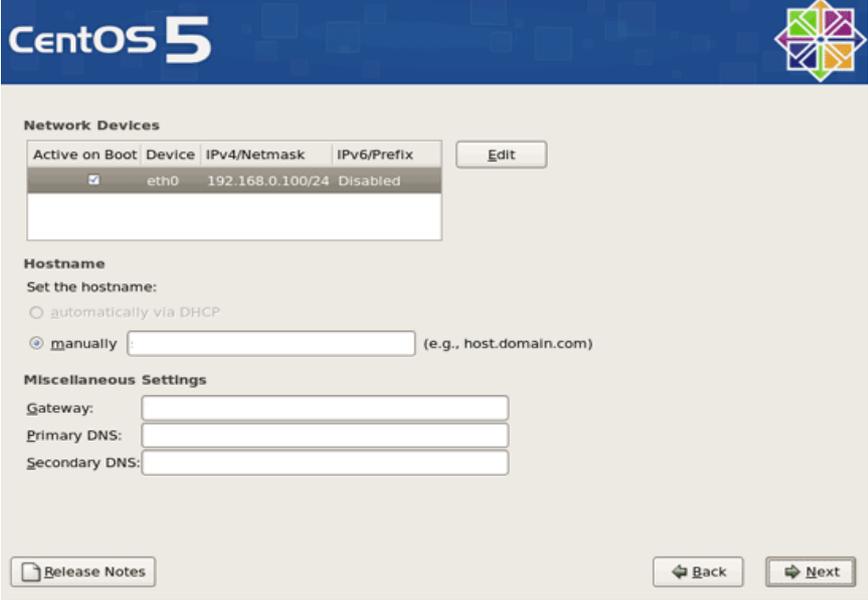


En la ventana que aparece configura los parámetros de red de acuerdo a tu red interna, si no estas seguro puedes contactarnos para determinar cual es la dirección IP de tu red interna y como debes configurarla.



**Figura 1.j**

Configura el nombre de tu servidor



The image shows the CentOS 5 Network Configuration Wizard. The header features the CentOS 5 logo and a colorful geometric icon. The main content area is divided into three sections: Network Devices, Hostname, and Miscellaneous Settings. The Network Devices section contains a table with columns for Active on Boot, Device, IPv4/Netmask, and IPv6/Prefix, and an Edit button. The Hostname section has radio buttons for 'automatically via DHCP' and 'manually', with a text input field and a placeholder '(e.g., host.domain.com)'. The Miscellaneous Settings section includes input fields for Gateway, Primary DNS, and Secondary DNS. At the bottom, there are buttons for 'Release Notes', 'Back', and 'Next'.

Active on Boot	Device	IPv4/Netmask	IPv6/Prefix	Edit
<input checked="" type="checkbox"/>	eth0	192.168.0.100/24	Disabled	

**Hostname**  
Set the hostname:

automatically via DHCP

manually  (e.g., host.domain.com)

**Miscellaneous Settings**

Gateway:

Primary DNS:

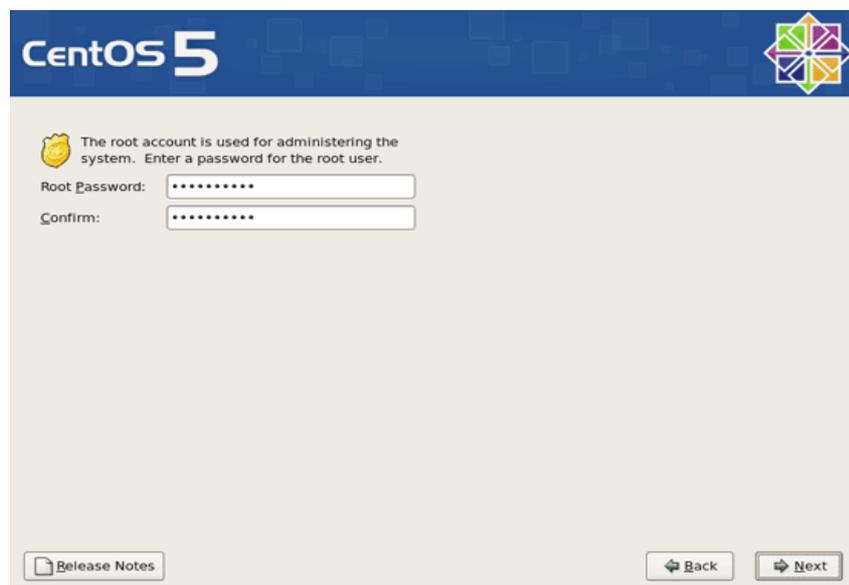
Secondary DNS:

[Release Notes](#) [Back](#) [Next](#)

Selecciona tu zona horaria, en el caso de Ecuador es America/Guayaquil -5 GMT

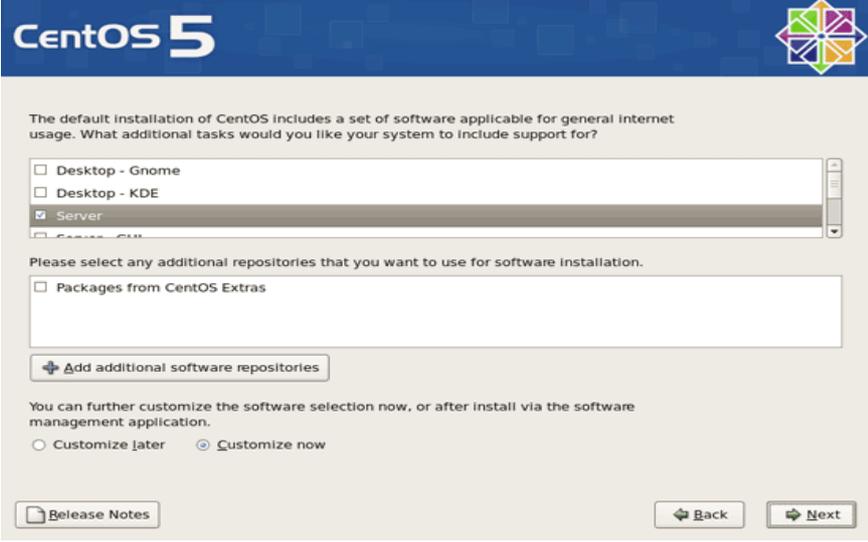


Selecciona un password de 8 dígitos entre números y letras



Ahora seleccionamos que tipo de instalación queremos realizar, en este caso seleccionamos "Server" (quítale el

check a las demás casillas) adicionalmente no es necesario darle check a "Packages from CentOS Extras", Luego dale "Customize Now" (Customizar ahora) y de ahí "Next" o Siguiente.



The screenshot shows the CentOS 5 installation customization interface. At the top, there is a blue header with the "CentOS 5" logo on the left and a colorful geometric logo on the right. Below the header, the text reads: "The default installation of CentOS includes a set of software applicable for general internet usage. What additional tasks would you like your system to include support for?". There is a scrollable list of options: "Desktop - Gnome", "Desktop - KDE", "Server" (which is checked), and "Server - GUI". Below this list, the text says: "Please select any additional repositories that you want to use for software installation." There is a checkbox for "Packages from CentOS Extras" which is currently unchecked. Below the checkbox is a button labeled "Add additional software repositories". At the bottom of the form, there is a message: "You can further customize the software selection now, or after install via the software management application." followed by two radio buttons: "Customize later" and "Customize now" (which is selected). At the very bottom, there are three buttons: "Release Notes" on the left, and "Back" and "Next" on the right.

Debemos seleccionar los grupos de paquetes que queremos instalar. Selecciona: Editors, Text-based Internet, DNS Name Server, Mail Server, Server Configuration Tools, Web Server, Administration Tools, Base (Deselecciona otros grupos de paquetes) y dale click a "Next" o Siguiente



El instalador revisa las dependencias...



Presiona "Next" para iniciar la instalación



El disco duro esta siendo formateado.



La instalación comienza, tomara unos minutos.



Finalmente la instalación es completada y ahora se puede quitar el CD o DVDy reiniciar el sistema



Congratulations, the installation is complete.

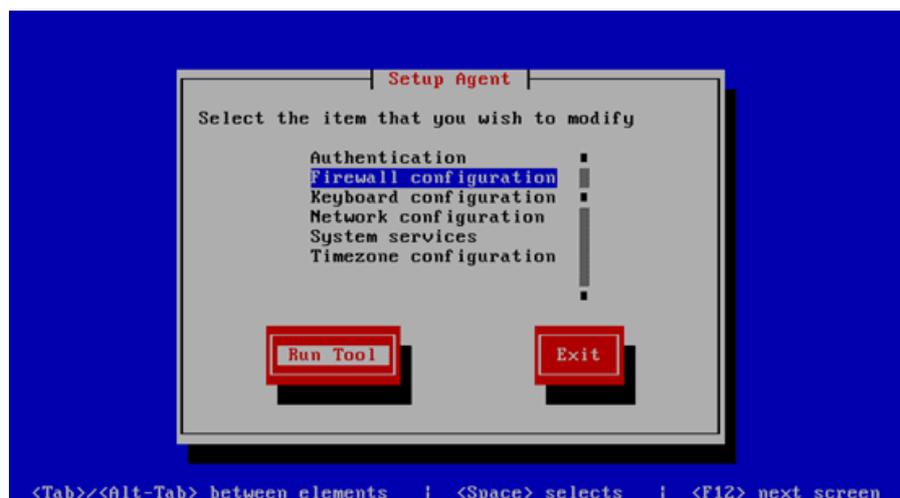
Remove any media used during the installation process and press the "Reboot" button to reboot your system.

Release Notes

Back

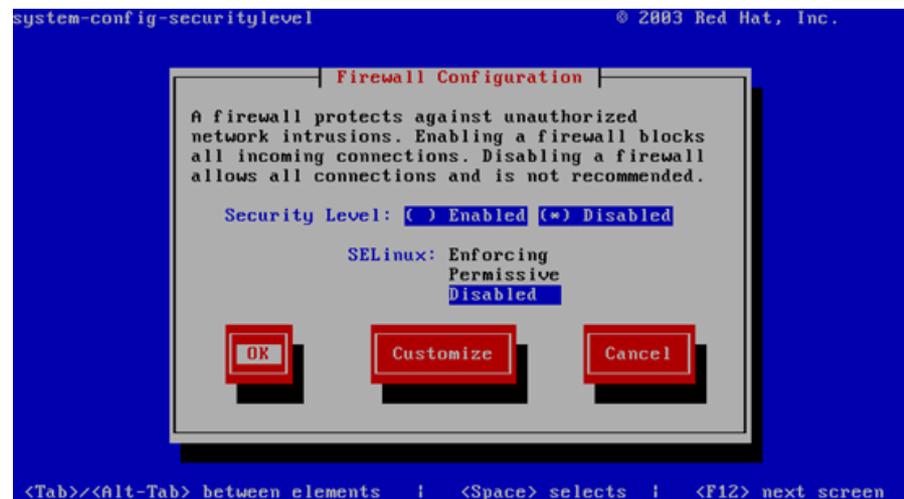
Reboot

Después de reiniciar el servidor, veras "Firewall Configuration" (configuración de Firewall), presiona "Run Tool" (Ejecutar herramienta)

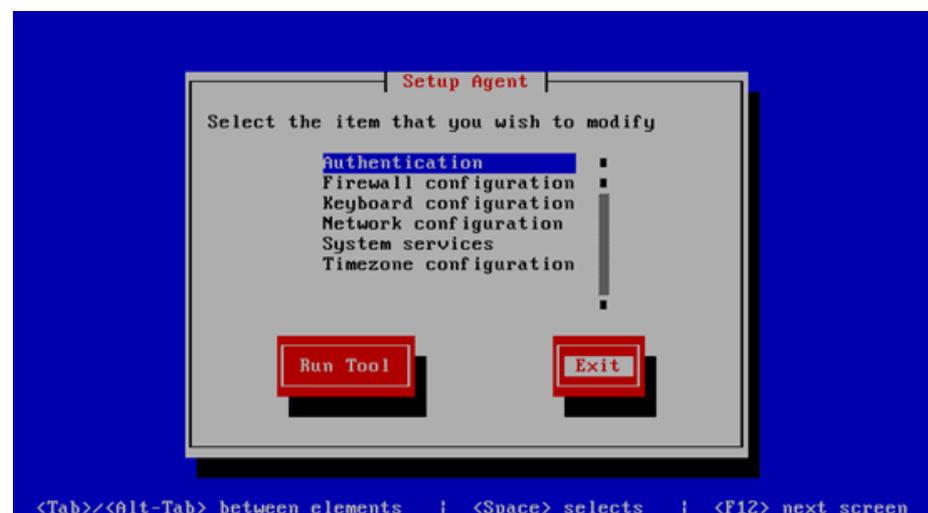


Debido a que se instalaran medidas de seguridad a medida de la implementación, deshabilitaremos el firewall, por ahora será mejor deshabilitado debido que el mismo firewall podría interferir con la configuración e instalación de otras aplicaciones

SELinux es una extensión de seguridad de CentOS que provee seguridad adicional al sistema. Sin embargo no es necesario hasta finalizar la configuración total de la implementación.

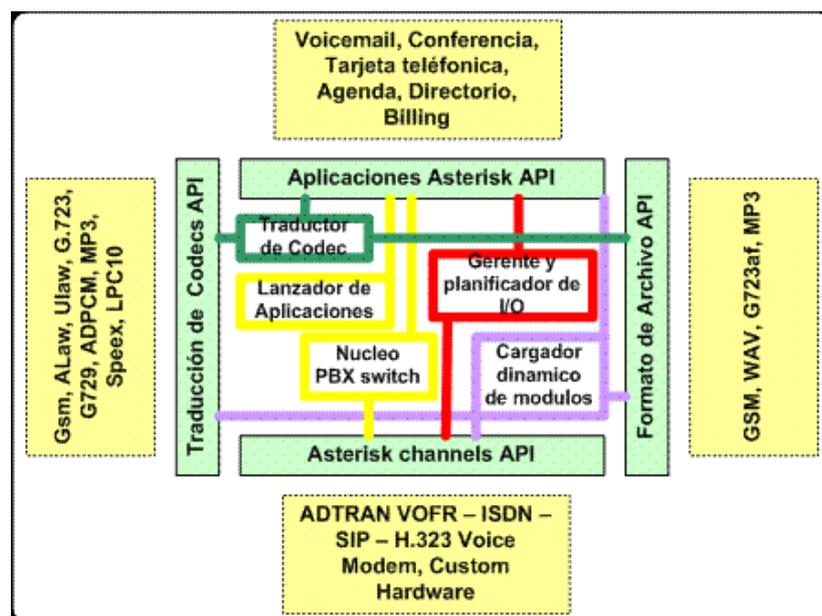


Luego selecciona "Exit" o Salir.



### 1.3 Arquitectura del Asterisk

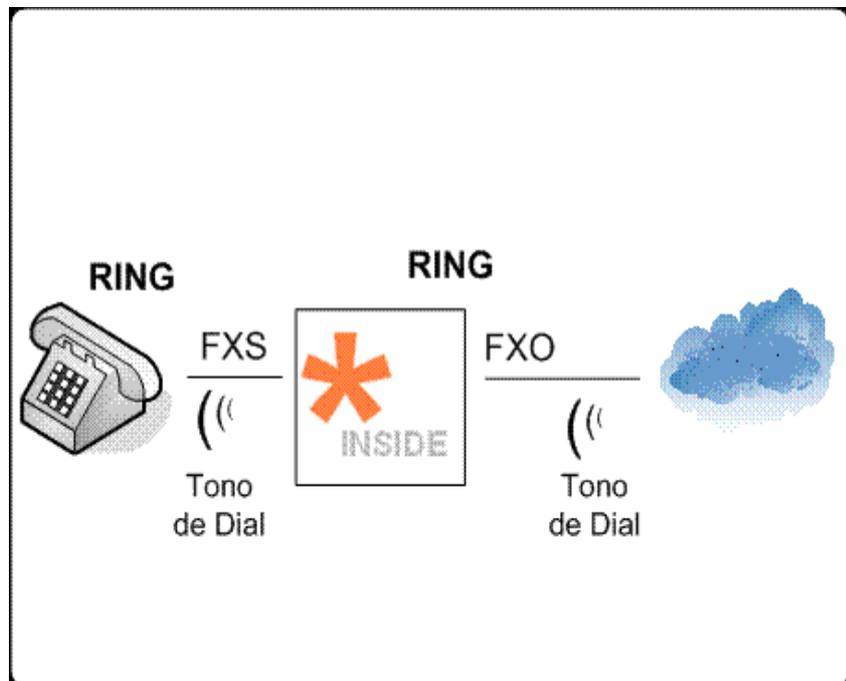
Dentro de una visión general, Asterisk es un PABX híbrido que integra tecnologías como TDM y telefonía IP con funcionalidad de unidades de respuesta automática y distribución automática de llamadas. En este momento de la lectura del libro es probable que usted no este entendiendo todos estos términos, pero a lo largo de los capítulos, estará cada vez mas familiarizado con ellos. En la figura de abajo podemos ver que Asterisk se puede conectar a una operadora de telecomunicaciones o un PABX usando interfaces analógicas o digitales. Los teléfonos pueden ser IP, analógicos o ADSI que es un teléfono analógico con display digital.



Arquitectura de Asterisk

## 1.4 Conceptos previos

**Canal:** Es una conexión que trae una llamada al Asterisk. Un canal podría ser una conexión a un teléfono ordinario fijado a mano o a una línea telefónica ordinaria, o a una llamada lógica (como una llamada telefónica del Internet). El Asterisk no hace ninguna distinción entre los canales del estilo de “FXO” y de “FXS” (es decir, no distingue entre las líneas telefónicas y los teléfonos). Cada llamada se pone o se recibe en un canal distinto.



**FXO(Foreign Exchange Office):** Es una interfaz en un dispositivo VoIP para conectarlo a una central telefónica PBX.

**FXS(Foreign Exchange Station):** Es la interfaz en un dispositivo VoIP que permite conectarlo directamente a teléfonos, faxes en centrales PBXs.

**SIP(Session Initiation Protocol):** SIP es un protocolo para establecer sesiones entre uno o más clientes de VoIP. SIP es un lenguaje que los teléfonos IP utilizan para intercambiar información entre ellos.

**IAX(Inter-Asterisk Exchange Protocol):** Es utilizado para manejar conexiones VoIP entre servidores Asterisk, y entre servidores y clientes que también utilizan protocolo IAX.

**DAHDI(Digium Asterisk Hardware Device Interface):** Antiguo nombre de Zaptel. Zaptel son las APIs de los drivers de las tarjetas que sirven para implementar la plataforma asterisk para la voz.

**Calling Rules:** Son las reglas que se establecen para las llamadas

**DialPlan:** El plan de marcación o “Dial Plan”, es el corazón de toda configuración en asterisk, y de esta configuración dependerá el performace y eficiencia de nuestra central telefónica.

**Contexto:** Un contexto, en Asterisk, es una agrupación de extensiones que comparten las mismas reglas de marcado. Dentro del plan de marcación de Asterisk, los contextos definen diferentes secciones, cada una con propósitos específicos. El uso de contextos nos permite restringir el flujo de llamadas entre extensiones o el acceso a ciertos servicios (p. ej. llamadas de larga distancia, llamadas a celulares, etc.) o números especiales, entonces. En estos casos, los contextos son la mejor manera de organizar nuestras extensiones.

## **1.5 Instalación del Asterisk**

Se explicará paso a paso la configuración del Asterisk pero previo a esto se indicarán los requerimientos mínimos para su respectiva instalación y la configuración de los archivos necesarios para la comunicación.

### Requerimientos Mínimos:

- Procesador a 1.5 GHz (Pentium 4)
- 256 MB en RAM
- 10 GB en disco duro.

### Pasos para instalar:

**a.** Para obtener las fuentes de Asterisk y drivers de Zaptel para el uso con hardware de Digium, usted debe bajar los paquetes de Digium.

Baje los archivos usando el comando wget. Cree el directorio /usr/src si este no existe. En el momento de finalización de este libro estas eran las versiones corrientes, substituya los comandos abajo con los archivos de las versiones más actuales.

```
# cd /usr/src
#wget http://ftp.DIGIUM.com/pub/zaptel/zaptel-1.4.x.tar.gz
#wget http://ftp.DIGIUM.com/pub/libpri/libpri-1.4.x.tar.gz
#wget http://ftp.DIGIUM.com/pub/Asterisk/Asterisk-addons-1.4.x.tar.gz
#wget http://ftp.DIGIUM.com/pub/Asterisk/Asterisk-1.4.x.tar.gz
```

**b.** Luego descomprimir los archivos.

```
# tar xzvf ASTERISK-1.4.x.tar.gz
# tar xzvf libpri-1.4.x.tar.gz
# tar xzvf ASTERISK-addons-1.4.x.tar.gz
# tar xzvf zaptel-1.4.x.tar.gz
```

**c.** Normalmente la compilación de los drivers zaptel es simple. En tanto que pueden ocurrir casos donde usted no posea ninguna placa TDM que use el driver zaptel. Asterisk precisa de una fuente de temporización que normalmente sea provista por una placa con driver zaptel. Si esta placa no existe va ha ser preciso compilar el módulo ztdummy.

Compilando los drivers de Zaptel.

```
cd /usr/src/zaptel-1.4.x/  
make clean  
./configure  
make menuselect  
make install  
make install-udev  
make config  
update-rc.d zaptel defaults
```

Use el comando make menuselect para escoger los módulos necesarios.

Screenshot del comando “make menuselect”:

**d.** Compilando el Asterisk

```
cd /usr/src/libpri-1.4.x/  
make clean  
make  
make install
```

```
cd /usr/src/asterisk-1.4.0
make clean
./configure
make menuselect
make
make install
make samples ;use to create sample configuration files
make config
```

e. Una vez instalado podemos iniciar el Asterix con

```
/usr/sbin/asterisk -vvvc
```

Y lo paramos con:

```
/usr/sbin/asterisk stop now
```

## **1.6 Configuración del Asterisk**

Su funcionamiento se realiza mediante canales y estos canales son drivers para distintos tipos de conexiones para protocolos de VoIP como SIP, IAX, MGCP.

Para el sistema se implementó el uso del protocolo SIP, que es muy parecido a HTTP o a SMTP.

Se recuerda que todos los archivos de configuración se encuentran en la ruta en etc/asterix

## **Archivo manager.conf**

Se puede configurar el acceso remoto hacia el gestor de Asterisk. Se requiere definir un usuario, la correspondiente clave de acceso y los privilegios necesarios, a fin de poder permitir utilizar diversas herramientas para la administración y/o supervisión remota(s).

Su configuración:

```

[;!
;! Automatically generated configuration file
;! Filename: manager.conf (/etc/asterisk/manager.conf)
;! Generator: Manager
;! Creation Date: Wed May 6 00:22:21 2009
;!
;
;
; AMI - The Asterisk Manager Interface
;
;
; Third party application call management support and PBX event supervision
;
; This configuration file is read every time someone logs in
;
;
; Use the "manager list commands" at the CLI to list available manager commands
; and their authorization levels.
;
; "manager show command <command>" will show a help text.
;
; ----- SECURITY NOTE -----
; Note that you should not enable the AMI on a public IP address. If needed,
; block this TCP port with iptables (or another FW software) and reach it
; with IPsec, SSH, or SSL vpn tunnel. You can also make the manager
; interface available over http if Asterisk's http server is enabled in
; http.conf and if both "enabled" and "webenabled" are set to yes in
; this file. Both default to no. httptimeout provides the maximum
; timeout in seconds before a web based session is discarded. The
; default is 60 seconds.
;
[general]
displayssystemname = yes
enabled = yes
webenabled = yes
port = 5038
;httptimeout = 60
; a) httptimeout sets the Max-Age of the http cookie
; b) httptimeout is the amount of time the webserver waits
;    on a action=waitevent request (actually its httptimeout-10)
; c) httptimeout is also the amount of time the webserver keeps
;    a http session alive after completing a successful action
bindaddr = 0.0.0.0
;displayconnects = yes
;
; Add a Unix epoch timestamp to events (not action responses)
;
;timestampevents = yes
;[mark]
;[mark]
;[admin]
;secret = clave
;deny=0.0.0.0/0.0.0.0
;permit=0.0.0.0/0.0.0.0
;
;secret = mysecret
;deny=0.0.0.0/0.0.0.0
;permit=209.16.236.73/255.255.255.0
;
; If the device connected via this user accepts input slowly,
; the timeout for writes to it can be increased to keep it
; from being disconnected (value is in milliseconds)
;
; writetimeout = 100
;
; Authorization for various classes
[admin]
secret = barcelona
permit = 0.0.0.0/0.0.0.0
read = system,call,log,verbose,command,agent,user,config
write = system,call,log,verbose,command,agent,user,config

```

## Archivo sip.conf

Sirve para configurar todo lo relacionado con el protocolo SIP y añadir nuevos usuarios o conectar con proveedores SIP.

Su configuración:

```
;
[general]
port=5060
disallow=all
allow=g726
allow=ulaw
allow=alaw
;[80]
type=friend
host=dynamic
language=es
context=internal
secret=80
username=80
callerid=80
dtmfmode=rfc2833
qualify=yes
;
[81]
type=friend
host=dynamic
language=es
context=internal
secret=81
username=81
callerid=81
dtmfmode=rfc2833
qualify=yes
;
[1300]
type=friend
host=dynamic
language=es
context=internal
secret=1300
username=1300
callerid=1300
dtmfmode=rfc2833
qualify=yes
;
```

## **Archivo extensión.conf**

El archivo `extensions.conf` es el más importante del Asterisk y tiene como misión principal definir el dialplan o plan de numeración que seguirá la centralita para cada contexto y por tanto para cada usuario.

El fichero `extensions.conf` se compone de secciones o contextos entre corchetes `[]` Hay dos contextos especiales que están siempre presentes que son `general` y `globals`

Su configuración

```

;! Automatically generated configuration file
;! Filename: extensions.conf (/etc/asterisk/extensions.conf)
;! Generator: Manager
;! Creation Date: Fri May 15 18:09:56 2009
;!
[general]
static = yes
writeprotect = no
clearglobalvars = yes
:
:
[internal]
:
;!exten => 80,1,Dial(SIP/80,26)
;!exten => 80,n,Hangup
:
:
;!exten => 81,1,Dial(SIP/81,26)
;!exten => 81,n,Hangup
;!exten => 1300,1,Agi(agi://server.thesis.org/hello.agi)
;!exten => 1300,n,Hangup
[globals]
FEATURES =
DIALOPTIONS =
RINGTIME = 20
FOLLOWMEOPTIONS =
PAGING_HEADER = Intercom
CID_6000 = 6000
CID_6001 = 6001
CID_6002 = 6002
trunk_1 = DAHDI/trunk_1
[default]
exten = _#6XXX,1,Set(MBOX=${EXTEN:1}@default)
exten = _#6XXX,n,VoiceMail(${MBOX})
exten = a,1,VoicemailMain(${MBOX})
exten = 9999,1,VoicemailMain(${CALLERID(num)})@default
[macro-stdexten]
exten = s,1,Set(__DYNAMIC_FEATURES=${FEATURES})
exten = s,2,GotoIf("${FOLLOWME_${ARG1}}" = "1"?5:3)
exten = s,3,Dial(${ARG2},${RINGTIME},${DIALOPTIONS})
exten = s,4,Goto(s-${DIALSTATUS},1)
exten = s,5,Macro(stdexten-followme,${ARG1},${ARG2})
exten = s-NOANSWER,1,Voicemail(${ARG1},u)
exten = s-NOANSWER,2,Goto(default,s,1)
exten = s-BUSY,1,Voicemail(${ARG1},b)
exten = s-BUSY,2,Goto(default,s,1)
exten = _s-,1,Goto(s-NOANSWER,1)
exten = a,1,VoicemailMain(${ARG1})
[macro-stdexten-followme]
exten = s,1,Answer
exten = s,2,Dial(${ARG2},${RINGTIME},${DIALOPTIONS})
exten = s,3,Set(__FMCIDNUM=${CALLERID(num)})
exten = s,4,Set(__FMCIDNAME=${CALLERID(name)})
exten = s,5,Followme(${ARG1},${FOLLOWMEOPTIONS})
exten = s,6,Voicemail(${ARG1},u)
exten = s-NOANSWER,1,Voicemail(${ARG1},u)
exten = s-BUSY,1,Voicemail(${ARG1},b)
exten = s-BUSY,2,Goto(default,s,1)
exten = _s-,1,Goto(s-NOANSWER,1)
exten = a,1,VoicemailMain(${ARG1})
[macro-pagingintercom]
exten = s,1,SIPAddHeader(Alert-Info: ${PAGING_HEADER})
exten = s,2,Page(${ARG1},${ARG2})
exten = s,3,Hangup

```

```

[conferences]
[ringgroups]
[queues]
[voicemenus]
[voicemailgroups]
[directory]
exten = 5555,1,Directory(default|default|ef)
[page_an_extension]
[pagegroups]
[asterisk_guitools]
exten = executecommand,1,System(${command})
exten = executecommand,n,Hangup()
exten = record_vmnu,1,Answer
exten = record_vmnu,n,Playback(vm-intro)
exten = record_vmnu,n,Record(${var1})
exten = record_vmnu,n,Playback(vm-saved)
exten = record_vmnu,n,Playback(vm-goodbye)
exten = record_vmnu,n,Hangup
exten = play_file,1,Answer
exten = play_file,n,Playback(${var1})
exten = play_file,n,Hangup
[macro-trunkdial-failover-0.3]
exten = s,1,GotoIf(${LEN(${FMCIDNUM}}) > 6]?1-fmsetcid,1)
exten = s,2,GotoIf(${LEN(${GLOBAL_OUTBOUND_CIDNAME}}) > 1]?1-setgbobname,1)
exten = s,3,Set(CALLERID(num)=${IF(${LEN(${CID_${CALLERID(num)}})} > 2]?${CID_${CALLERID(num)}})})
exten = s,n,GotoIf(${LEN(${CALLERID(num)}}) > 6]?1-dial,1)
exten = s,n,Set(CALLERID(all)=${IF(${LEN(${CID_${ARG3}})} > 6]?${CID_${ARG3}}:${GLOBAL_OUTBOUND_CID}})})
exten = s,n,Goto(1-dial,1)
exten = 1-setgbobname,1,Set(CALLERID(name)=${GLOBAL_OUTBOUND_CIDNAME})
exten = 1-setgbobname,n,Goto(s,3)
exten = 1-fmsetcid,1,Set(CALLERID(num)=${FMCIDNUM})
exten = 1-fmsetcid,n,Set(CALLERID(name)=${FMCIDNAME})
exten = 1-fmsetcid,n,Goto(1-dial,1)
exten = 1-dial,1,Dial(${ARG1})
exten = 1-dial,n,GotoIf(${LEN(${ARG2}}) > 0 ?1-${DIALSTATUS},1:1-out,1)
exten = 1-CHANUNAVAIL,1,Dial(${ARG2})
exten = 1-CHANUNAVAIL,n,Hangup()
exten = 1-CONGESTION,1,Dial(${ARG2})
exten = 1-CONGESTION,n,Hangup()
exten = 1-out,1,Hangup()
[DLPN_DialPlan1]
include = test-zap
include = default
include = parkedcalls
include = conferences
include = ringgroups
include = voicemenus
include = queues
include = voicemailgroups
include = directory
include = pagegroups
include = page_an_extension
[DID_trunk_1]
include = DID_trunk_1_default
[prueba_Noel]
exten =>6001,1,Answer()
exten =6001,2,Wait(3)
exten => 6001,3,Playback(hello-world)
[DID_trunk_1_default]
[test-zap]
exten = _2XXXXXX,1,Dial(Zap/1/${EXTEN},20,r)
exten = _2XXXXXX,2,Hangup
[entrante]
exten = s,1,Dial(SIP/6001,20,r)
exten = s,2,Hangup

```

## Archivo Voicemail.conf

Permitirá grabar las llamadas.

A más de esto permitirá que se envíen correos con el archivo de grabación realizando la respectiva configuración.

Su configuración:

```
[general]
; Escoger el formato del correo de voz. Recomendado usar WAV, por razones de
; compatibilidad.
format=wav
;
; Si se dispone de espacio suficiente en la cuenta de correo, la siguiente
; opción específica que se adjunte el mensaje de voz a un mensaje de correo
; electrónico, de modo que se pueda escuchar al dar clic desde el cliente.
;
attach=yes
;
[default]
; Cada buzón de voz se lista en el siguiente formato:
; buzón => clave de acceso,Nombre de persona,correo electrónico,correo
; electrónico de servicio de localizador. Ejemplos:
101 => secreto1,Nombre,alguien@algo.algo,numero@mi-celular.algo
102 => secreto2,Nombre,otro@algo.algo
103 => secreto3,Nombre,alguien-mas@algo.algo
```

## Archivo Asterisk.conf

Le dice al asterisk los directorios donde está todo, incluyendo el directorio que contiene todos los otros archivos de configuración. Por defecto, el asterisk busca el archivo asterisk.conf en el directorio /etc/asterisk, pero puedes proveer un parámetro por línea de comando para utilizar otro archivo asterisk.conf.

```
[directories]
astetcdir => /etc/asterisk
astmoddir => /usr/lib/asterisk/modules
astvarlibdir => /var/lib/asterisk
astdatadir => /var/lib/asterisk
astagidir => /var/lib/asterisk/agi-bin
astspooldir => /var/spool/asterisk
astrundir => /var/run
astlogdir => /var/log/asterisk
[options]
languageprefix = yes ; Use the new sound prefix path syntax
;verbose = 3
;debug = 3
;alwaysfork = yes ; same as -F at startup
;nofork = yes ; same as -f at startup
;quiet = yes ; same as -q at startup
;timestamp = yes ; same as -T at startup
;execincludes = yes ; support #exec in config files
;console = yes ; Run as console (same as -c at startup)
;highpriority = yes ; Run realtime priority (same as -p at startup)
;initcrypto = yes ; Initialize crypto keys (same as -i at startup)
;nocolor = yes ; Disable console colors
;dontwarn = yes ; Disable some warnings
;dumpcore = yes ; Dump core on crash (same as -g at startup)
;internal_timing = yes
;systemname = my_system_name ; prefix uniqueid with a system name for global
uniqueness issues
;maxcalls = 10 ; Maximum amount of calls allowed
;maxload = 0.9 ; Asterisk stops accepting new calls if the load average exceed this
limit
;cache_record_files = yes ; Cache recorded sound files to another directory during
recording
;record_cache_dir = /tmp ; Specify cache directory (used in conjunction with
cache_record_files)
;transmit_silence_during_record = yes ; Transmit SLINEAR silence while a
channel is being recorded
;transmit_silence = yes ; Transmit SLINEAR silence while a channel is being
recorded or DTMF is being generated
;transcode_via_sln = yes ; Build transcode paths via SLINEAR, instead of directly
;runuser = asterisk ; The user to run as
;rungroup = asterisk ; The group to run as
dahdichannname = no ; Channels created by chan_dahdi will be called 'DAHDI',
otherwise 'Zap'
; Changing the following lines may compromise your security.
;[files]
;astctlpermissions = 0660
;astctlowner = root
;astctlgroup = apache
;astctl = asterisk.ctl
```

## **Archivo Agents.conf**

Estos son los usuarios, que puede utilizar para responder a las llamadas en las colas.

Los agentes pueden ser separados en diferentes grupos llamados contexto. De esta manera usted tiene un mayor flexability en la gestión de las colas, porque se puede elegir el agente que, con el que la cola al trabajo.

```

;
; Agent configuration
;
[general]
;
; Define whether callbacklogins should be stored in astdb for
; persistence. Persistent logins will be reloaded after
; Asterisk restarts.
;
persistentagents=yes
; Enable or disable a single extension from logging in as multiple agents.
; The default value is "yes".
;multiplelogin=yes
[agents]
;
; Define maxlogintries to allow agent to try max logins before
; failed.
; default to 3
;
;maxlogintries=5
;
;
; Define autologoff times if appropriate. This is how long
; the phone has to ring with no answer before the agent is
; automatically logged off (in seconds)
;
;autologoff=15
;
; Define autologoffunavail to have agents automatically logged
; out when the extension that they are at returns a CHANUNAVAIL
; status when a call is attempted to be sent there.
; Default is "no".
;
;autologoffunavail=yes
;
; Define ackcall to require an acknowledgement by '#' when
; an agent logs in using agentcallbacklogin. Default is "no".
; Can also be set to "always", which will also require AgentLogin
; agents to acknowledge calls by pressing '#'.
;
;ackcall=no
;
; Define endcall to allow an agent to hangup a call by '*'.
; Default is "yes". Set this to "no" to ignore '*'.
;
;endcall=yes
;
; Define wrapuptime. This is the minimum amount of time when
; after disconnecting before the caller can receive a new call
; note this is in milliseconds.
;
;wrapuptime=5000
;
; Define the default musiconhold for agents
; musiconhold => music_class
;
;musiconhold => default
;
; Define the default good bye sound file for agents
; default to vm-goodbye
;
;goodbye => goodbye_file
;
; Define updatecdr. This is whether or not to change the source
; channel in the CDR record for this call to agent/agent_id so
; that we know which agent generates the call
;
;updatecdr=no
;
; Group memberships for agents (may change in mid-file)
;
;group=3
;group=1,2
;group=

```

```

;
;
;-----
; This section is devoted to recording agent's calls
; The keywords are global to the chan_agent channel driver
;
;
; Enable recording calls addressed to agents. It's turned off by default.
;recordagentcalls=yes
;
;
; The format to be used to record the calls: wav, gsm, wav49.
; By default its "wav".
;recordformat=gsm
;
; The text to be added to the name of the recording. Allows forming a url link.
;urlprefix=http://localhost/calls/
;
;
; The optional directory to save the conversations in. The default is
; /var/spool/asterisk/monitor
;savecallsin=/var/calls
;
;
; An optional custom beep sound file to play to always-connected agents.
;custom_beep=beep
;
;-----
;
; This section contains the agent definitions, in the form:
;
;
;agent => agentid,agentpassword,name
;
;
;agent => 1001,4321,Mark Spencer
;agent => 1002,4321,Will Meadows

```

## Archivo Dahdi\_guiread.conf

Este archivo es utilizado para las configuraciones de la tarjeta Digium

```

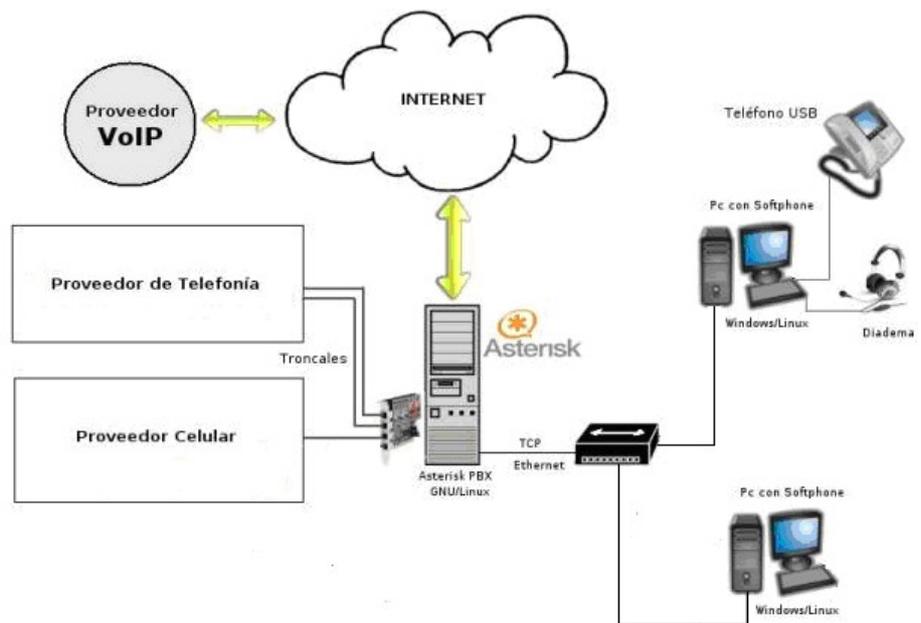
[general]
#include "../dahdi/system.conf"

```

## 1.7 Funcionalidad de Asterisk

Como podemos visualizar en la figura Funcionalidad de Asterisk, permite realizar las llamadas de estaciones que tengan instalado un softphone (Ej: X-Lite).

En IVRIX las llamadas se realizaran mediante aplicación WEB, pasarán por nuestro servidor Asterix y este se encargará de direccionarlas con su dueño en este caso puede ser un teléfono convencional, celular (para esto se necesita que tenga el hardware respectivo: Tarjeta Digium com módulos FXO) o que tenga instalado un softphone ya que esta herramienta hace uso de un protocolo cuyo nombre es el SIP.



Funcionalidad de Asterisk

## **1.8 Instalación del Softphone.**

En esta sección se indicará paso a paso la instalación del softphone, en este sistema se utilizará el X-Lite, pero previo a esto se indicará los requerimientos mínimos:

- Intel Pentium III 700 MHz o equivalente
- 256 MB en RAM
- 30MB de disco duro.

En esta ruta se indican los pasos para la instalación del X-LITE:

[http://www.itson.mx/asterisk/Herramientas\\_files/Manual\\_XLITE.pdf](http://www.itson.mx/asterisk/Herramientas_files/Manual_XLITE.pdf)

# CAPITULO II

## Introducción

A continuación mostraremos la integración de Asterisk con la parte aplicativa hecha en Java (Jsp), la herramienta con la que es desarrollado el sistema es My Eclipse 6.0

## 2.1 Herramientas Utilizadas

A continuación mencionaremos las herramientas de software necesarias para el desarrollo de este módulo:

Requerimientos de Software:

- My Eclipse 6.0
- JDK 1.5
- Tomcat 5.0

### **2.1.1 My Eclipse**

MyEclipse se basa en la plataforma Eclipse, y se integra tanto de propiedad y las soluciones de fuente abierta en el entorno de desarrollo.

Lo podemos descargar de la siguiente dirección:

**<http://www.myeclipseide.com/Article68.html>**

### **2.1.2 JDK**

La base para poder operar cualquier producto que utiliza java es el JDK de la plataforma correspondiente, por lo cual antes de instalar el servidor de Tomcat, se debe instalar el JDK.

Lo podemos descargar de la siguiente dirección:

**<http://java.sun.com/j2se/1.4.2/download.html>**

### **2.1.3 Tomcat**

Apache Tomcat es un contenedor de servlets que soporta tecnologías como Java Servlets y JavaServer Pages.

El proyecto Tomcat es soportado por programadores de todo el mundo, este proyecto es Open Source.

Lo podemos descargar de la siguiente dirección:

**<http://java.sun.com/j2se/1.4.2/download.html>**

## **2.2 Comunicación de Asterisk con Java**

### **2.2.1 Manager API**

El Administrador de la API permite la interacción remota con un servidor de Asterisk. En contraste con el protocolo de Asterisk, FastAGI no explicitamente pasa el control a su aplicación cuando se utiliza el Administrador de la API, pero le permite consultar y cambiar su estado en cualquier momento

El Administrador de la API se compone de tres conceptos: las acciones, respuestas y eventos. Las acciones pueden ser enviadas a Asterisk e indicar que haga algo.

Los eventos son enviados por un asterisco, sin relación directa con las acciones es el envío de su solicitud. Los eventos les informa sobre los cambios en el estado de Asterisk.

La conexión con el servidor de Asterisk a través de Administrador de la API se produce a través de TCP / IP por lo general en el puerto por defecto 5038.

Para habilitar el Administrador de la API de Asterisk debe editar su manager.conf archivo de configuración y reiniciar Asterisk.

El manager.conf contiene también limitaciones en el rango de direcciones IP que puedan conectarse y nombre de usuario y contraseñas para la autenticación

```

;!
;! Automatically generated configuration file
;! Filename: manager.conf (/etc/asterisk/manager.conf)
;! Generator: Manager
;! Creation Date: Wed May 6 00:22:21 2009
;!
;
;
; AMI - The Asterisk Manager Interface;
; Third party application call management support and PBX event supervision
;
;
; This configuration file is read every time someone logs in
;
;
; Use the "manager list commands" at the CLI to list available manager
commands
; and their authorization levels.
;
;
; "manager show command <command>" will show a help text.
;
;
; ----- SECURITY NOTE -----
;
;
;
;
[general]
displayssystemname = yes
enabled = yes
webenabled = yes
port = 5038
;httptimeout = 60
; a) httptimeout sets the Max-Age of the http cookie
; b) httptimeout is the amount of time the webserver waits
;   on a action=waitevent request (actually its httptimeout-10)
; c) httptimeout is also the amount of time the webserver keeps
;   a http session alive after completing a successful action
bindaddr = 0.0.0.0
;displayconnects = yes
;
;
; Add a Unix epoch timestamp to events (not action responses)
;
;
;timestampevents = yes
;[mark]
;[mark]
;[admin]
;secret = clave
;deny=0.0.0.0/0.0.0.0
;permit=0.0.0.0/0.0.0.0
;
;
;secret = mysecret
;deny=0.0.0.0/0.0.0.0
;permit=209.16.236.73/255.255.255.0
;
;
; If the device connected via this user accepts input slowly,
; the timeout for writes to it can be increased to keep it
; from being disconnected (value is in milliseconds)
;
;
; writetimeout = 100
;
;
; Authorization for various classes
[grupo_1]
secret = XTmas31
permit = 0.0.0.0/0.0.0.0
read = system,call,log,verbose,command,agent,user,config
write = system,call,log,verbose,command,agent,user,config

```

# CAPITULO III

## Introducción

En este capítulo trataremos la comunicación de la aplicación con la Base de Datos con sus respectivas tablas que se realizaron para el desarrollo del sistema como las respectivas herramientas que utilizaron para su funcionamiento.

### 3.1 Herramientas Utilizadas

A continuación mencionaremos las herramientas de software necesarias para el desarrollo de este módulo:

Requerimientos de Software:

- My SQL
- Hibernate

#### 3.1.1 My SQL

Es la BD con la que va a trabajar el sistema y ya viene con el S.O CentOS.

Y con la siguiente línea de comando en el prompt lo podemos instalar:

```
yum install mysql-server mysql mysql-devel
```

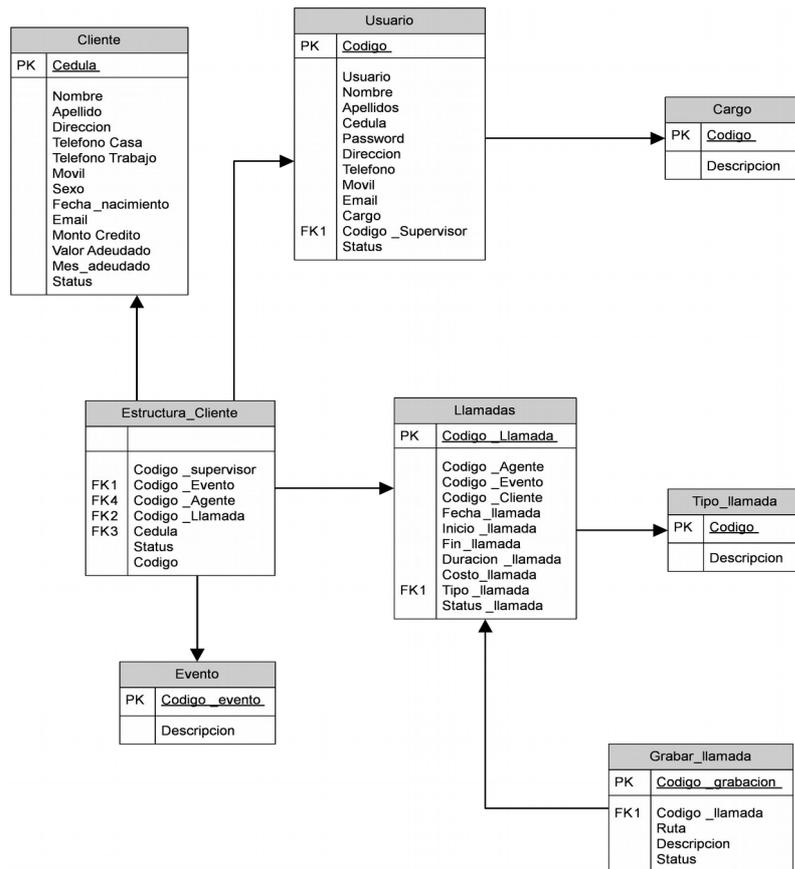
### **3.1.2 Hibernate**

Hibernate es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una BD relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

### **3.2 Diagrama Entidad Relación**

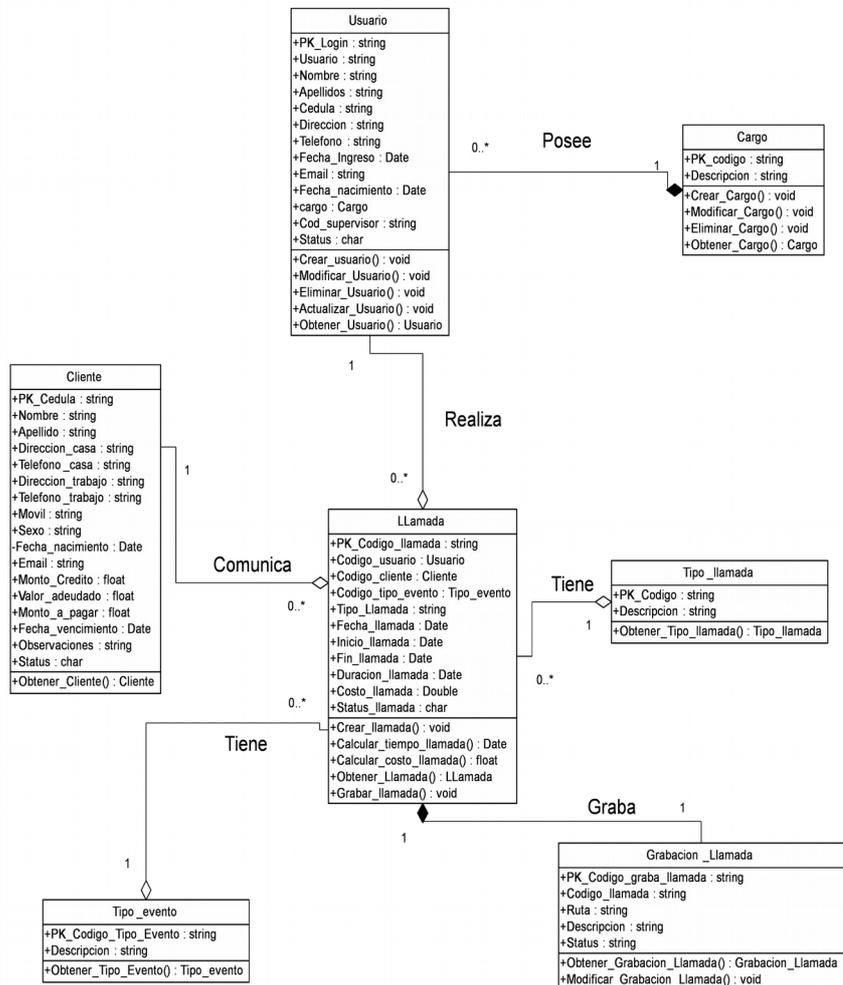
El diagrama entidad-relación pretende visualizar los objetos que pertenecen a la base de datos como entidades.



**Figura 2a DER.**

### 3.3 Diagrama de Clases.

Este diagrama de clases va a representar la estructura de mi aplicación mostrando sus principales clases, atributos, operaciones y las relaciones entre ellos.

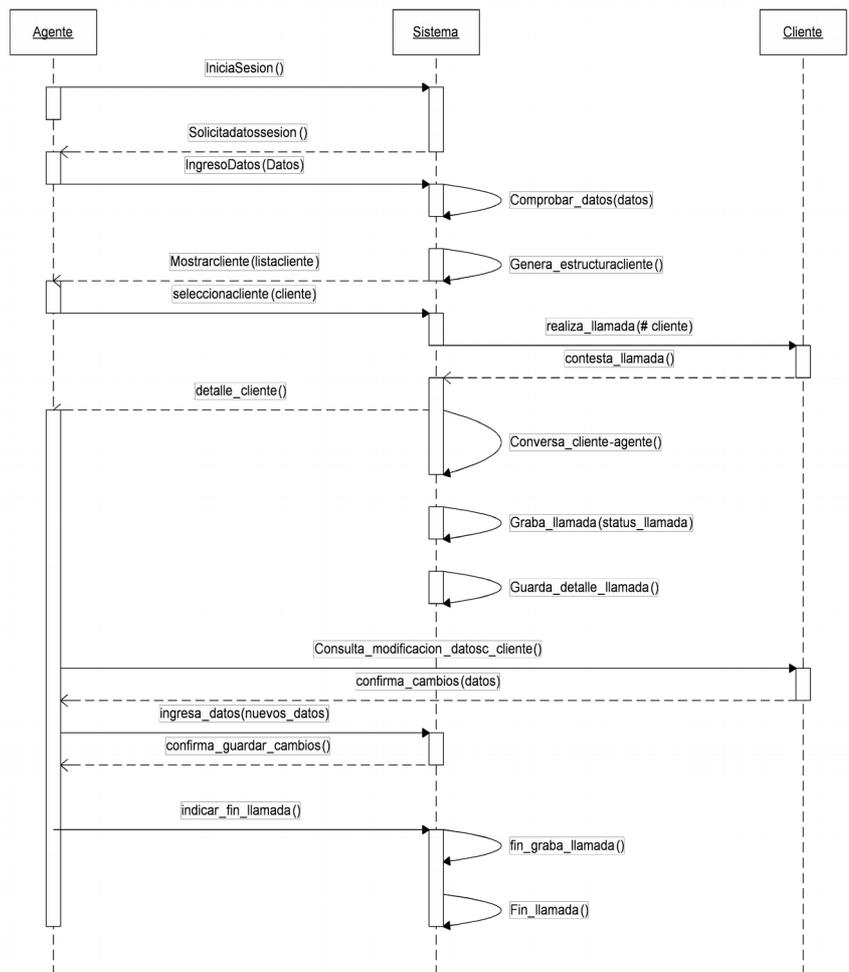


**Figura 3b Diagrama de Clases.**

### 3.4 Diagrama de Secuencia

En este diagrama se muestra la interacción de los objetos que componen el sistema de forma temporal.

#### 3.4.1 Secuencia Agente.



**Figura 3.c Diagrama Secuencia Agente.**

### 3.4.2 Secuencia Supervisor

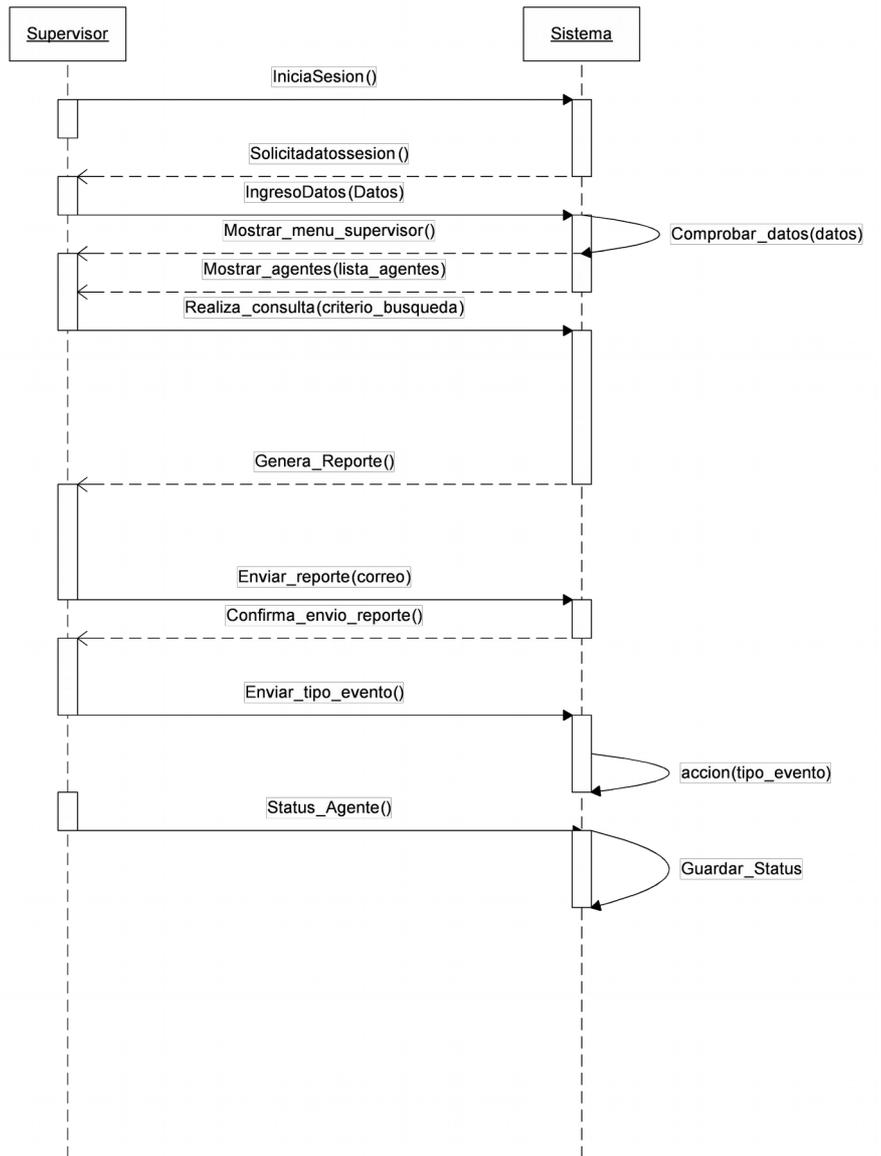
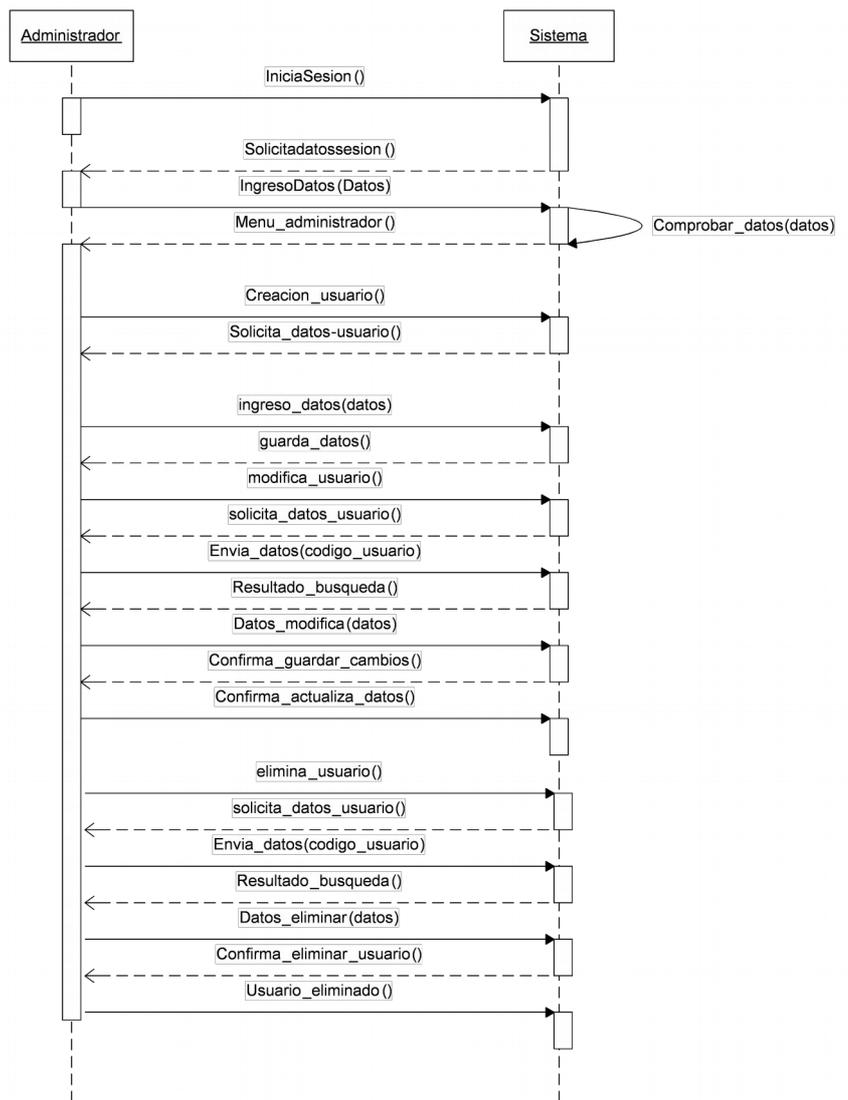


Figura 3.d Diagrama Secuencia Supervisor.

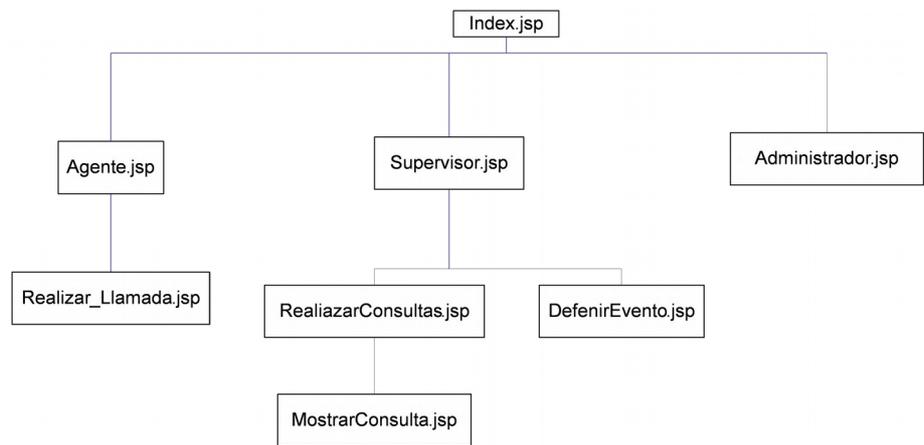
### 3.4.3 Secuencia Administrador



**Figura 3.e Secuencia Administrador.**

### 3.5 Diagrama Flujo de Ventanas

Este diagrama nos ayuda a ver la secuencia de las pantallas de mi aplicación.



**Figura 3.f Diagrama Flujo de Ventanas.**

### 3.6 Creación de objetos en la BD.

#### usuarios

**Objetivo:** Almacenar los usuarios que podrán hacer uso del sistema, ya sean estos Administradores, Supervisores y Agentes.

```

CREATE TABLE `usuarios` (
  `Codigo` char(4) COLLATE latin1_spanish_ci NOT NULL,
  `Login` varchar(15) COLLATE latin1_spanish_ci NOT NULL,
  `Password` varchar(12) COLLATE latin1_spanish_ci NOT NULL,
  `Nombre` varchar(25) COLLATE latin1_spanish_ci NOT NULL,
  `Apellido` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
  `Cedula` varchar(10) COLLATE latin1_spanish_ci NOT NULL,
  `Direccion` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
  `Email` varchar(35) COLLATE latin1_spanish_ci NOT NULL,
  `Status` char(2) COLLATE latin1_spanish_ci NOT NULL COMMENT
'Estatus del Usuario',
  `Grupo` char(2) COLLATE latin1_spanish_ci NOT NULL,
  `Supervisor` char(10) COLLATE latin1_spanish_ci DEFAULT NULL,
  `Fecha_nacimiento` timestamp NOT NULL,
  `Activo` char(3) COLLATE latin1_spanish_ci NOT NULL,
  `Codigo_EstructuraAgente` varchar(3) COLLATE latin1_spanish_ci NOT
NULL,
  `Telefono` varchar(15) COLLATE latin1_spanish_ci NOT NULL,
  PRIMARY KEY (`Codigo`),
  KEY `FK_grupo` (`Grupo`),
  KEY `FK_estructura` (`Codigo_EstructuraAgente`) USING BTREE,
  CONSTRAINT `FK_estructura` FOREIGN KEY (`Codigo_EstructuraAgente`)
REFERENCES
`estructuraagentes` (`Codigo_estructura`),
  CONSTRAINT `FK_grupo` FOREIGN KEY (`Grupo`) REFERENCES
`cargos` (`Codigo_Car`)
)

```

### ***Cuadro 1 Ingreso Usuarios***

#### **clientes**

**Objetivo:** Almacenar información de los respectivos clientes que maneje la empresa.

```

CREATE TABLE `clientes` (
`Cedula` char(10) COLLATE latin1_spanish_ci NOT NULL,
`Nombre_Cliente` varchar(25) COLLATE latin1_spanish_ci NOT NULL,
`Apellido_Cliente` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
`TelefonoCasa_Cliente` varchar(10) COLLATE latin1_spanish_ci NOT
NULL,
`TelefonoCasa2_Cliente` varchar(10) CHARACTER SET latin1 DEFAULT
NULL,
`Movil_Cliente` varchar(10) CHARACTER SET latin1 DEFAULT NULL,
`Sexo_Cliente` varchar(9) COLLATE latin1_spanish_ci NOT NULL,
`FechaNacimiento_Cliente` datetime NOT NULL,
`Email_Cliente` varchar(25) CHARACTER SET latin1 DEFAULT NULL,
`MontoComprado_Cliente` int(11) DEFAULT NULL,
`ValorAdeudado_Cliente` int(11) DEFAULT NULL,
`MesAdeudado_Cliente` varchar(20) CHARACTER SET latin1 NOT
NULL,
`Status_Cliente` char(2) CHARACTER SET latin1 DEFAULT NULL,
`Codigo_estructura` varchar(3) COLLATE latin1_spanish_ci DEFAULT
NULL,
`Nivel_Llamada_Cliente` int(10) unsigned DEFAULT NULL,
`DireccionCasa_Cliente` varchar(45) COLLATE latin1_spanish_ci NOT
NULL,
`ZonaUrbana_Cliente` varchar(10) COLLATE latin1_spanish_ci NOT
NULL,
`DireccionTrabajo_Cliente` varchar(45) COLLATE latin1_spanish_ci
DEFAULT NULL,
PRIMARY KEY (`Cedula`),
KEY `FK_estructura` (`Codigo_estructura`),
KEY `FK_parroquias` (`ZonaUrbana_Cliente`),
CONSTRAINT `FK_estructura` FOREIGN KEY (`Codigo_estructura`)
REFERENCES `estructuraagentes` (`Codigo_estructura`),
CONSTRAINT `FK_parroquias` FOREIGN KEY (`ZonaUrbana_Cliente`)
REFERENCES `parroquias` (`Codigo_parroquia`)
)

```

## **Cuadro 2 Ingreso Clientes**

### **cargos**

**Objetivo:** Almacenar el tipo de usuario en el sistema:

Administrador, Supervisor, Agente.

```

CREATE TABLE `cargos` (
`Codigo_Car` char(2) COLLATE latin1_spanish_ci NOT NULL,
`Descripcion` varchar(46) CHARACTER SET latin1 NOT NULL,
`Status` char(1) CHARACTER SET latin1 NOT NULL,
PRIMARY KEY (`Codigo_Car`) USING BTREE
)

```

### **Cuadro 3 Cargos Definidos**

#### **estructuraagentes**

**Objetivo:** Permite que a los agentes se le asignen un grupo de clientes el cual tiene que llamar y que mantendrán un orden asignado por el sistema.

```
CREATE TABLE `estructuraagentes` (  
  `Codigo_estructura` varchar(3) COLLATE latin1_spanish_ci NOT NULL,  
  `CodigoEvent` varchar(3) COLLATE latin1_spanish_ci DEFAULT NULL,  
  `Descripcion_Estructura` varchar(150) COLLATE latin1_spanish_ci  
  DEFAULT NULL,  
  `Zona_Urbana` varchar(150) COLLATE latin1_spanish_ci DEFAULT  
  NULL,  
  `Status` varchar(2) COLLATE latin1_spanish_ci DEFAULT NULL,  
  PRIMARY KEY (`Codigo_estructura`) USING BTREE,  
  KEY `FK_evento` (`CodigoEvent`) USING BTREE,  
  CONSTRAINT `FK_evento` FOREIGN KEY (`CodigoEvent`)  
  REFERENCES `evento` (`Codigo_Eve`)  
  )
```

### **Cuadro 4 Relación Estructura-Evento**

#### **evento**

**Objetivo:** Es la acción que van a realizar los agentes: Publicidad o Cobros.

```
CREATE TABLE `evento` (  
  `Codigo_Eve` varchar(3) COLLATE latin1_spanish_ci NOT NULL,  
  `Descripcion_Eve` varchar(25) COLLATE latin1_spanish_ci NOT NULL,  
  `Status_Eve` varchar(2) CHARACTER SET latin1 NOT NULL,  
  `Detalle_Eve` varchar(100) COLLATE latin1_spanish_ci DEFAULT NULL,  
  PRIMARY KEY (`Codigo_Eve`)  
)
```

### ***Cuadro 5 Eventos definidos***

## **parroquias**

**Objetivo:** Almacena los diferentes sectores de la ciudad.

```
CREATE TABLE `parroquias`  
  `Codigo_parroquia` varchar(3) CHARACTER SET latin1 COLLATE  
  latin1_spanish_ci NOT NULL,  
  `Descripcion` varchar(45) CHARACTER SET latin1 COLLATE  
  latin1_spanish_ci NOT NULL,  
  `Status` varchar(2) CHARACTER SET latin1 COLLATE  
  latin1_spanish_ci NOT NULL,  
  `Zona` varchar(15) CHARACTER SET latin1 COLLATE  
  latin1_spanish_ci NOT NULL,  
  PRIMARY KEY (`Codigo_parroquia`)  
)
```

### ***Cuadro 6 Parroquias definidas***

## **registro\_ingreso**

**Objetivo:** Almacena la fecha y hora de ingreso y salida del usuario en el sistema.

```
CREATE TABLE `registro_ingreso` (  
  `Numero_Registros` varchar(6) COLLATE latin1_spanish_ci NOT NULL,  
  `cod_usuario` varchar(4) COLLATE latin1_spanish_ci NOT NULL,  
  `Id_Session` varchar(40) COLLATE latin1_spanish_ci NOT NULL,  
  `Hora_ingreso` datetime NOT NULL,  
  `Hora_salida` datetime DEFAULT NULL,  
  PRIMARY KEY (`Numero_Registros`),  
  KEY `FK_usuario` (`cod_usuario`),  
  CONSTRAINT `FK_usuario` FOREIGN KEY (`cod_usuario`)  
  REFERENCES `usuarios` (`Codigo`)  
  )
```

### ***Cuadro 7 Registro de los Ingresos.***

## **registro\_llamadas**

**Objetivo:** Almacenar el registro completo de las llamadas, donde incluye la hora que inicio la llamada, fin de la llamada, agente que realizo la llamada, el número que hizo la llamada, al cliente que se le hizo la llamada.

```

CREATE TABLE `registro_llamada` (
  `Codigo_Registro` varchar(4) COLLATE latin1_spanish_ci
  NOT NULL,
  `Id_Agente` char(4) COLLATE latin1_spanish_ci NOT NULL,
  `Cedula_Cliente` char(10) COLLATE latin1_spanish_ci NOT
  NULL,
  `Status_Llamada` varchar(45) CHARACTER SET latin1 NOT
  NULL,
  `Detalle_Llamada` varchar(500) CHARACTER SET latin1
  NOT NULL,
  `Fecha_Llamada` datetime DEFAULT NULL,
  `Tipo_Evento` varchar(3) COLLATE latin1_spanish_ci
  DEFAULT NULL,
  `Tiempo_Llamada` int(10) unsigned DEFAULT NULL,
  `Llamada_Destino` varchar(10) COLLATE latin1_spanish_ci
  DEFAULT NULL,
  PRIMARY KEY (`Codigo_Registro`),
  KEY `FK_Cliente` (`Cedula_Cliente`),
  KEY `FK_Agente` (`Id_Agente`),
  KEY `FK_tipoEvento` (`Tipo_Evento`),
  CONSTRAINT `FK_Agente` FOREIGN KEY (`Id_Agente`)
  REFERENCES `usuarios` (`Codigo`),
  CONSTRAINT `FK_Cliente` FOREIGN KEY
  (`Cedula_Cliente`) REFERENCES `clientes` (`Cedula`),
  CONSTRAINT `FK_tipoEvento` FOREIGN KEY
  (`Tipo_Evento`) REFERENCES `evento` (`Codigo_Eve`)
)

```

**Cuadro 8 Registro de las Llamadas.**

**Prerrequisito:** Para almacenar el registro de las llamadas(propia del Asterisk) debemos realizar los siguientes pasos:

1. Bajamos en paquete de Asterisk addons en la carpeta /usr/src

```
wget
```

```
http://downloads.digium.com/pub/asterisk/asterisk-  
addons-1.4-current.tar.gz
```

2. Se decompile el paquete:

```
tar -xf asterisk-addons-1.4-current.tar.gz
```

3. Entramos a la carpeta recién creada:

```
cd asterisk-addons-1.4.7
```

4. Se compila:

```
./configure
```

```
make
```

```
make install
```

Una vez instalado el MySQL se configura una contraseña para el usuario root:

Nos ubicamos en el cliente MySQL y ponemos:

```
mysql> grant all privileges on *.* to root@'localhost'  
identified by 'clave_de_root' with grant option;
```

```
mysql> flush privileges;
```

```
mysql> quit
```

Ahora creamos la base de datos:

```
mysqladmin create 'nombre_BD' -u root -p  
'clave_de_root'
```

```
mysql -u root -p 'clave_de_root'
```

```
use 'llamadas'
```

Luego definimos la siguiente estructura:

```
mysql> CREATE TABLE llamadas (  
  calldate datetime NOT NULL default '0000-00-00  
00:00:00',  
  clid varchar(80) NOT NULL default "",  
  src varchar(80) NOT NULL default "",  
  dst varchar(80) NOT NULL default "",  
  dcontext varchar(80) NOT NULL default "",  
  channel varchar(80) NOT NULL default "",  
  dstchannel varchar(80) NOT NULL default "",  
  lastapp varchar(80) NOT NULL default "",  
  lastdata varchar(80) NOT NULL default "",  
  duration int(11) NOT NULL default '0',  
  billsec int(11) NOT NULL default '0',  
  disposition varchar(45) NOT NULL default "",  
  amaflags int(11) NOT NULL default '0',  
  accountcode varchar(20) NOT NULL default "",  
  uniqueid varchar(32) NOT NULL default "",  
  userfield varchar(255) NOT NULL default ""  
);
```

Finalmente, falta agregar unas líneas en el archivo de configuración de Asterisk y realizar esto:

```
[global]
hostname='servidor_BD'
dbname='nombre_BD'
table=llamadas
password=clave_root
user=asterisk
port=3306
sock=/var/lib/mysql/mysql.sock
userfield=1
```

### 3.7 Procesos principales

En esta sección describiremos los procesos principales de Telefonlvrix.

#### 3.7.1 Proceso llamadas

Código de la página **Llamada.jsp** que permite comunicar el agente con el cliente.

```

try {
    var=id_session.getAttribute("var1").toString();
    }catch(Exception e){System.out.println(e.getMessage()
+"llamada.jsp");}
    if ((var==id_session.getId())){
        System.out.println("usu.getStatus()");
        if(WebUtil.getString(request,"evento").equals("Cerrar
Ventana")){
            id_session.setAttribute("status_llamada","Fallida");
            valor_Llamada=false;
        }
        else
            if(WebUtil.getString(request,"evento").equals("Llamada
Contestada")){

id_session.setAttribute("status_llamada","Contestada");
            }else
                if(WebUtil.getString(request,"evento").equals("Llamada
noContestada"))
{id_session.setAttribute("status_llamada","NoContestad");
            }else
                {String retornaLlamada=null;
                String estadodeGrabacion=null;
                int i=0;
                Date fecha=null;
                do {
                    try{
estadodeGrabacion=id_session.getAttribute("estadoGrabacion")
.toString();i=6100;
                    }catch(Exception e)
{System.out.println(e.getMessage()+"ESTADO de GRABACION");i+
+;}
                    }while(i<1000);

                try{
                    Llamada llamadaCliente;
                    String
telefono=id_session.getAttribute("telefono").toString();
                    String
extension=id_session.getAttribute("extension").toString();
                    System.out.println(estadodeGrabacion+"**");
                    if(estadodeGrabacion!=null){
                        llamadaCliente=new
Llamada(telefono,extension,estadodeGrabacion);
                        status=llamadaCliente.runLlamada();
                        fecha=llamadaCliente.getHora_llamada();
                        System.out.println(fecha+"-"+status);
                        id_session.setAttribute("fecha", fecha);
                    }
                    }catch(Exception e){System.out.println(e.getMessage()
+"Error Llamada");status="null";}
                    if(status.equals("Originate failed")){
                        valor_Llamada=false;
                    }
                }
            }else{
                response.sendRedirect("Error_login.jsp");
            }%>

```



```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN">
<%@page import="Web_util.WebUtil;"%>
<html>
<head>
<title>Llamando a Cliente</title>
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="keywords"
content="keyword1,keyword2,keyword3">
<meta http-equiv="description" content="This is my
page">
<!--
<link rel="stylesheet" type="text/css"
href="styles.css">
-->
<script>
function closepopup()
{
if(false == my_window.closed)
{
my_window.close ();
}
else
{alert('Window already closed!');}
}
</script>
</head>
<body>
<form id="form1" name="form1" method="post" action="">
<% if(valor_Llamada) {%>

Realizando Llamada <br>
<IMG ID="Picture554" HEIGHT=150 WIDTH=191
SRC="Imagenes/telefono.gif" VSPACE=0 HSPACE=0 ALIGN="TOP"
BORDER=0 ALT="">

<p>
<input type="submit" name="evento" id="evento"
value="Llamada Contestada" onclick="window.close()" />
</p>

<p>
<input type="submit" name="evento" id="evento"
value="Llamada no Contestada" onclick="window.close()" />
</p>
<%}
else {
%>
La llamada no Ha sido Contestada,
<p>
<input type="submit" name="evento" id="evento"
value="Cerrar Ventana" onclick="window.close()" />
</p>
<%} %>
</form>
</body>
</html>

```

## Clase Asterisk.java

```
package asteriskJava;

import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import
org.asteriskjava.manager.AuthenticationFailedException;
import org.asteriskjava.manager.ManagerConnection;
import org.asteriskjava.manager.ManagerConnectionFactory;
import org.asteriskjava.manager.TimeoutException;
import org.asteriskjava.manager.action.OriginateAction;
//import org.asteriskjava.manager.action.StatusAction;
import org.asteriskjava.manager.response.ManagerResponse;

public class Llamada
{
    private ManagerConnection managerConnection;
    private String Telefono=null;
    private String Extension=null;
    private Date hora_llamada=null;
    private String aux=null;
    private String contextoLlamada=null;

    public Llamada(String fono,String extension,String
contexto) throws IOException
    {
        Telefono=fono;
        aux=extensión;
        extensión="SIP/"+extensión;
        contextoLlamada=contexto;
        ManagerConnectionFactory factory = new
ManagerConnectionFactory(
            "192.168.16.2", "grupo_1", "XTmas31");

        this.managerConnection =
factory.createManagerConnection();
    }
    @SuppressWarnings("deprecation")
    public String runLlamada() throws IOException,
AuthenticationFailedException,
TimeoutException, ParseException
    {
        String status;
        //managerConnection.addEventListener(this);
        //StatusAction statusAction;
        OriginateAction originateAction;
        ManagerResponse originateResponse;
        //statusAction=new StatusAction();
    }
}
```

```

//System.out.println(Telefono+extensión);
//Telefono="ZAP/1/"+Telefono;
originateAction = new OriginateAction();
originateAction.setChannel(Extension);

//originateAction.setContext("DLPN_DialPlan1");

//originateAction.setContext("DLPN_DialPlan2");
originateAction.setContext(contextoLlamada);
originateAction.setExten(Telefono);
originateAction.setActionId(Extension);
originateAction.setCallerId(aux);

originateAction.setPriority(new Integer(1));
try {
originateAction.setTimeout(new
Integer(60000));
}catch(Exception e ) {}

// connect to Asterisk and log in
managerConnection.login();

//System.out.println(originateAction.getChannel());

//System.out.println(managerConnection.getHostname());
;

String
aux=java.net.InetAddress.getLocalHost().getHostAddress();

//System.out.println(aux);
// send the originate action and wait for a
maximum of 30 seconds for Asterisk
// to send a reply
originateResponse =
managerConnection.sendAction(originateAction, 30000);

// print out whether the originate succeeded
or not

System.out.println(originateResponse.getResponse());
System.out.println(originateResponse.getUniqueId());
System.out.println(originateResponse.getActionId());
//originateResponse.setDateReceived(new
Date());//Fijo una hora de llamada
System.out.println(originateResponse.getDateReceived(
));//Me devuelve una hora de llamada

```

```

System.out.println(originateResponse.getMessage())
; //Muestra el mensaje si se contesto la llamada o
me cerro
        status=originateResponse.getMessage();
        // and finally log off and disconnect
        managerConnection.logoff();
        SimpleDateFormat dateformat=new
SimpleDateFormat("yy-MM-dd HH:mm:ss");

        Date
dateLlamada=dateformat.parse(dateformat.format(ori
ginateResponse.getDateReceived()));
        setHora_llamada(dateLlamada);
        return status;
    }
    public Date getHora_llamada() {
        return hora_llamada;
    }
    public void setHora_llamada(Date
hora_llamada) {
        this.hora_llamada = hora_llamada;
    }
}

```

### 3.7.2 Proceso Reportes.

```

<%@ page language="java"
import="java.util.*,Web_util.WebUtil,Tablas.DAO.UsuarioDAO" pageEncoding="ISO-8859-1"%>
<%@ page
import="Tablas.beans.*,java.text.SimpleDateFormat,Tablas.DAO.ConsultasDAO,Tablas.DAO.AgenteClientesDAO,Tablas.DAO.EventoDAO" %>

<%
// En esta pagina se realiza una consulta de tipo
general a la base para obtener un reporte .pdf
// Es necesario Invocar cuatro clases para cumplir
con este proceso
// Las clases son: ConectarBD: Permite realizar la
conexion a la Base de Datos.
//ReportesBD: Es la encargada de tomar los parámetros
y entregarselos al Jasper para que genere el Reporte
Basico.
//ReportesGeneralBD: Es la encargada de tomar los
parámetros y entregarselos al Jasper para que genere
el Reporte General.
UsuarioDAO uDAO=new UsuarioDAO();
AgenteClientesDAO cDAO=new AgenteClientesDAO();
Usuarios usu=new Usuarios();
EventoDAO eDAO=new EventoDAO();
HttpSession id_session=request.getSession();
boolean estado_Alerta=true;
boolean estado_Reporte=false;
String var=null;
String
MensajeError=null;System.out.println(MensajeError);
SimpleDateFormat dateformat=new SimpleDateFormat("dd-
MMMM-yyyy HH:mm:ss");
try {
var=id_session.getAttribute("var1").toString();
}catch (Exception e)
{System.out.println(e.getMessage());}
if (!(var==id_session.getId())){
System.out.println("Session
nueva"+id_session.getId());
response.sendRedirect("Error_login.jsp");
}else
{
usu=uDAO.consulta_Registro(id_session.getAttribute("r
eg").toString()).getUsuario_in();
String codigo=null;
String nombre=null;
Date fechaDesde=new Date();
Date fechaHasta=new Date();
Integer numeroLlamadas=null;
String contestada=null;
String estado_llamada=null;
String tipoEvento=null;

if(WebUtil.getString(request,"evento").equals("Menu
Principal")){
response.sendRedirect("Supervisor.jsp");
}

```

```

else
if(WebUtil.getString(request,"eventoAG").equals("Consultar")){//Consultar por Codigo Agente o

    fechaDesde=WebUtil.getDate(request,"fecha_desde");
    fechaHasta=WebUtil.getDate(request,"fecha_hasta");
    codigo=WebUtil.getString(request,"co");
    nombre=WebUtil.getString(request,"no");
    numeroLlamadas=WebUtil.getInteger(request,"numero");
    contestada=WebUtil.getString(request,"estado");

    tipoEvento=eDAO.retornarCodigo(WebUtil.getString(request,"tipo_evento"));

    if((fechaDesde!=null)&&(fechaHasta!=null)){
        if(fechaDesde.before(fechaHasta)){//Validar Fecha
            if(!(codigo.equals(""))){//Campo codigo en blanco
                if(uDAO.ValidarLogin(codigo)){//Validar al agente esta OK +FECHA+AGENTE
                    if((contestada.equals("co"))){
                        estado_llamada="Contestada";
                    }
                    else
                    if((contestada.equals("noco"))){
                        estado_llamada="NoContestada";
                    }
                    else
                    if((contestada.equals("fa"))){
                        estado_llamada="Fallida";
                    }
                    if(estado_llamada!=null){//Validar si selecciono estado de llamada

//FECHA+AGENTE+ESTADO DE LLAMADA
                        if((tipoEvento!=null)){//VALIDA SI SELECCIONO TIPO DE EVENTO
                            System.out.println("SE SELECCIONO TODOS LOS CAMPOS");//SELECCIONO TODOS LOS CAMPOS EXECTO ULTIMAS LLAMADAS
                            id_session.setAttribute("parametro1","Fecha_Desde");
                            id_session.setAttribute("cadena1",fechaDesde);
                            id_session.setAttribute("parametro2","Fecha_Hasta");
                            else {//no selecciono el tipo de evento

                                System.out.println("no selecciono TIPO DE EVENTO");
                                id_session.setAttribute("parametro1","Fecha_Desde");
                                id_session.setAttribute("cadena1",fechaDesde);
                                id_session.setAttribute("parametro2","Fecha_Hasta");
                                id_session.setAttribute("cadena2",fechaHasta);
                                id_session.setAttribute("parametro3","Login");
                                id_session.setAttribute("cadena3",codigo);
                                id_session.setAttribute("parametro4","Estado_Llamada");
                                id_session.setAttribute("cadena4",estado_llamada);
                                id_session.setAttribute("parametro5","null");

```

```

id_session.setAttribute("cadena5","null");
id_session.setAttribute("ruta","Consulta_General_Excepto_Ev.jasper");
response.sendRedirect("ReportesGenerales.jsp");
    }
    }else{//NO SELECCIONO NADA EN ESTADO DE LLAMADA
System.out.println("No selecciono Estado de llamada");
        if((tipoEvento!=null)){
            //SELECCIONO FECHA , LOGIN Y EVENTO
id_session.setAttribute("parametro1","Fecha_Desde");
            id_session.setAttribute("cadena1",fechaDesde);

id_session.setAttribute("parametro2","Fecha_Hasta");
            id_session.setAttribute("cadena2",fechaHasta);
            id_session.setAttribute("parametro3","Login");
id_session.setAttribute("cadena3",codigo);
id_session.setAttribute("parametro4","null");
id_session.setAttribute("cadena4","null");
id_session.setAttribute("parametro5","CodigoEventos");

id_session.setAttribute("cadena5",tipoEvento);

id_session.setAttribute("ruta","Consulta_General_Excepto_Es.jasper");
response.sendRedirect("ReportesGenerales.jsp");
            }
            else {
                //NO HA SELECCIONADO NI ESTADO DE LLAMADA NI TIPO DE EVENTO
id_session.setAttribute("parametro1","Fecha_Desde");
                id_session.setAttribute("cadena1",fechaDesde);

id_session.setAttribute("parametro2","Fecha_Hasta");
                id_session.setAttribute("cadena2",fechaHasta);
                id_session.setAttribute("parametro3","Login");
                id_session.setAttribute("cadena3",codigo);
                id_session.setAttribute("parametro4","null");
                id_session.setAttribute("cadena4","null");
                id_session.setAttribute("parametro5","null");
                id_session.setAttribute("cadena5","null");
id_session.setAttribute("ruta","Consulta_General_Excepto_Es_Ev.jasper");
response.sendRedirect("ReportesGenerales.jsp");}
}}//Else no selecciono estado de llamada }
else//Valida que el Usuario exista
MensajeError="Usuario Inexistente"; }//Validar que el campo Agente no este en blanco }//Valida que la fecha desde se menor que la hasta
else
MensajeError="Datos de Fechas Erroneos";
}//valida campo fechas que tenga dato
else//Campo fecha esta vacio validar el resto de campos{
if(!(codigo.equals("")) ){//Campo codigo en blanco
        if(uDAO.ValidarLogin(codigo)){
            //Validar al agente esta OK +FECHA+AGENTE
            if(numeroLlamadas > 0 &&
cDAO.RetornarNumeroRegistrosLlamadas() >=
numeroLlamadas){ if((contestada.equals("co"))){
                estado_llamada="Contestada";
            }
        }
    }
}

```

```

}
    else
        if((contestada.equals("noco"))){
            estado_llamada="NoContestada";
        }
        else
            if((contestada.equals("fa"))){
                estado_llamada="Fallida";
            }
            if(estado_llamada!=null){//Validar si
selecciono estado de llamada
                //FECHA+AGENTE+ESTADO DE LLAMADA
                if((tipoEvento!=null)){//VALIDA SI
SELECCIONO TIPO DE EVEN System.out.println("SE
SELECCIONO TODOS LOS CAMPOS");
                //Todos los campos excepto el campo fecha
                id_session.setAttribute("parametro1","null");
                id_session.setAttribute("cadena1","null");
                id_session.setAttribute("parametro2","Login");
                id_session.setAttribute("cadena2",codigo);
                id_session.setAttribute("parametro3","Estado_Llamada"
);
                id_session.setAttribute("cadena3",estado_llamada);

                id_session.setAttribute("parametro4","CodigoEventos")
;
                id_session.setAttribute("cadena4",tipoEvento);
                id_session.setAttribute("parametro5","Limite");

                id_session.setAttribute("cadena5",numeroLlamadas);

                id_session.setAttribute("ruta","Consulta_General_Li_E
xcepto_Fe.jasper");
                System.out.println(codigo+estado_llamada+tipoEvento+n
umeroLlamadas);
                response.sendRedirect("ReportesGenerales.jsp");
            }
            else {//no selecciono el tipo de evento ni fecha

                id_session.setAttribute("parametro1","null");
                id_session.setAttribute("cadena1","null");
                id_session.setAttribute("parametro2","Login");
                id_session.setAttribute("cadena2",codigo);
                id_session.setAttribute("parametro3","Estado_Llamada"
);
                id_session.setAttribute("cadena3",estado_llamada);
                id_session.setAttribute("parametro4","null");
                id_session.setAttribute("cadena4","null");
                id_session.setAttribute("parametro5","Limite");

                id_session.setAttribute("cadena5",numeroLlamadas);

                id_session.setAttribute("ruta","Consulta_General_Li_E
xcepto_Fe_Ev.jasper");
                response.sendRedirect("ReportesGenerales.jsp");
            }
        }else
        { if((tipoEvento!=null){
                //SELECCIONO , LOGIN Y numero de
                llamadas,EVENTO
                id_session.setAttribute("parametro1","null");
                id_session.setAttribute("cadena1","null");

```

```

        id_session.setAttribute("parametro2","Login");
        id_session.setAttribute("cadena2",codigo);

id_session.setAttribute("parametro3","CodigoEventos")
;
id_session.setAttribute("cadena3",tipoEvento);
id_session.setAttribute("parametro4","null");
id_session.setAttribute("cadena4","null");
id_session.setAttribute("parametro5","Limite");
    id_session.setAttribute("cadena5",numeroLlamadas);

id_session.setAttribute("ruta","Consulta_General_Li_E
xcepto_Fe_Es_jasper");

//System.out.println(fechaDesde+"-"+fechaHasta+codigo
+contestada+tipoEvento);
response.sendRedirect("ReportesGenerales.jsp");
    } else
        {//HA SELECCIONADO Login, y numero de
llamadas
id_session.setAttribute("parametro1","null");
id_session.setAttribute("cadena1","null");
id_session.setAttribute("parametro2","Login");
    id_session.setAttribute("cadena2",codigo);
id_session.setAttribute("parametro3","null");
id_session.setAttribute("cadena3","null");
id_session.setAttribute("parametro4","null");
id_session.setAttribute("cadena4","null");
id_session.setAttribute("parametro5","Limite");
    id_session.setAttribute("cadena5",numeroLlamadas);

id_session.setAttribute("ruta","Consulta_General_Li_E
xcepto_Fe_Es_Ev_jasper");

//System.out.println(fechaDesde+"-"+fechaHasta+codigo
+contestada+tipoEvento);
response.sendRedirect("ReportesGenerales.jsp");
    }
} //Else no selecciono estado de llamada
}
//No selecciono Numero de llamadas
else //No selecciono Numero de llamadas
{
    if((contestada.equals("co"))){
        estado_llamada="Contestada";
    }
    else
        if((contestada.equals("noco"))){
            estado_llamada="NoContestada";
        }
    else
        if((contestada.equals("fa"))){
            estado_llamada="Fallida";
        }
if(estado_llamada!=null){
    //Selecciono codigo de agente y Estado de
llamada if((tipoEvento!=null)){
        //Selecciono codigo agente ,estado de
llamada y evento de llamada
id_session.setAttribute("parametro1","null");
        id_session.setAttribute("cadena1","null");

```

```

id_session.setAttribute("parametro2", "Login");
id_session.setAttribute("cadena2", codigo);
id_session.setAttribute("parametro3", "Estado_Llamada"
);
id_session.setAttribute("cadena3", estado_llamada);

id_session.setAttribute("parametro4", "CodigoEventos")
;
id_session.setAttribute("cadena4", tipoEvento);
id_session.setAttribute("parametro5", "null");
id_session.setAttribute("cadena5", "null");
id_session.setAttribute("ruta", "Consulta_General_Exce
pto_Fecha_Nu.jasper");

//System.out.println(fechaDesde+"-"+fechaHasta+codigo
+contestada+tipoEvento);
response.sendRedirect("ReportesGenerales.jsp");
    }
    else
        //Selecciono Codigo agente y estado de
llamada
    {
        id_session.setAttribute("parametro1", "null
");
id_session.setAttribute("cadena1", "null");
id_session.setAttribute("parametro2", "Login");
id_session.setAttribute("cadena2", codigo);
id_session.setAttribute("parametro3", "Estado_Llamada"
);
id_session.setAttribute("cadena3", estado_llamada);
id_session.setAttribute("parametro4", "null");
id_session.setAttribute("cadena4", "null");
id_session.setAttribute("parametro5", "null");
id_session.setAttribute("cadena5", "null");
id_session.setAttribute("ruta", "Consulta_General_Exce
pto_Fecha_Nu_Ev.jasper");
System.out.println("Selecciono Codigo agente y estado
de llamada");
response.sendRedirect("ReportesGenerales.jsp");
    }
    else
        if((tipoEvento!=null)){
            //Selecciono codigo agente y evento de
llamada
id_session.setAttribute("parametro1", "null");
id_session.setAttribute("cadena1", "null");
id_session.setAttribute("parametro2", "Login");
id_session.setAttribute("cadena2", codigo);
id_session.setAttribute("parametro3", "null");
id_session.setAttribute("cadena3", "null");
id_session.setAttribute("parametro4", "CodigoEventos")
;
id_session.setAttribute("cadena4", tipoEvento);
id_session.setAttribute("parametro5", "null");
id_session.setAttribute("cadena5", "null");
id_session.setAttribute("ruta", "Consulta_General_Exce
pto_Fecha_Es_Nu.jasper");

//System.out.println(fechaDesde+"-"+fechaHasta+codigo
+contestada+tipoEvento);
response.sendRedirect("ReportesGenerales.jsp");
        }
    } //Else de seleccion numero de
llamadas
}
}

```

```

}
    else
        MensajeError="Usuario no existe";
    }//Validar que el campo Agente no este en
    blancoelse {
        //El campo fecha y el campo Usuario esta en
    blancoif(numeroLlamadas > 0 &&
    cDAO.RetornarNumeroRegistrosLlamadas() >=
    numeroLlamadas){
        //Selecciona campo Numero de llamadas
        if((contestada.equals("co"))){
            estado_llamada="Contestada";
        }
        else
            if((contestada.equals("noco"))){
                estado_llamada="NoContestada";
            }
        else
            if((contestada.equals("fa"))){
                estado_llamada="Fallida";
            }
        }
        if(estado_llamada!=null){
            //Selecciono Numero de llamada y Estado
de llamada
                if((tipoEvento!=null)){
                    //Selecciono Numero de llamadas ,
estado de llamada y tipo de evento
id_session.setAttribute("parametro1","null");
            id_session.setAttribute("cadena1","null");

id_session.setAttribute("parametro2","Estado_Llamada");
id_session.setAttribute("cadena2",estado_llamada);

id_session.setAttribute("parametro3","CodigoEventos");
id_session.setAttribute("cadena3",tipoEvento);
            id_session.setAttribute("parametro4","Limite");

id_session.setAttribute("cadena4",numeroLlamadas);

id_session.setAttribute("parametro5","null");
            id_session.setAttribute("cadena5","null");

id_session.setAttribute("ruta","Consulta_General_Li
_Excepto_Fe_Lo.jasper");

//System.out.println(fechaDesde+"-"+fechaHasta+codi
go+contestada+tipoEvento);
response.sendRedirect("ReportesGenerales.jsp");

        }
    }
}
}}}}//fin de clase consulta General}
%><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<style type="text/css">
<!--.Estilo1 {font-size: 8px}
-->
</style>

```

```

<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>Consultas Supervisores </title>
<style type="text/css">
<!--
.style3 { color: #EEEEEE;
font-weight: bold;
font-size: 10pt;
}
-->
</style>
<link href="estilos.css" rel="stylesheet"
type="text/css" media="all" />
<style type="text/css">
<!--
.style4 {color: #DDDDDD}
.style6 {color: #EEEEEE; font-weight: bold; }
.style7 {color: #EEEEEE; font-weight: bold; font-
size: 8pt; }
.style8 {font-size: 8pt}
.style9 {font-size: 8.5pt}
.style10 {font-size: 12pt}
.Estilo2 {font-size: 8pt; font-weight: bold; }
.Estilo3 {font-size: 8pt; font-weight: bold;color:
#0000EE }
-->
</style>
<script src="SpryAssets/SpryTabbedPanels.js"
type="text/javascript"></script>
<link href="SpryAssets/SpryTabbedPanels.css"
rel="stylesheet" type="text/css" />
</head>

<body class="oneColElsCtr" >
<table width="33%" border="0" align="center"
cellpadding="2" cellspacing="1"
bordercolor="#0099CC">
<tr> <th colspan="2" scope="col"></th> </tr> <tr>
<td class="h7link"></td>
<td class="h2link">&nbsp;&nbsp;&nbsp;</td> </tr> <tr>
<td width="22%" rowspan="2" valign="top"
class="h7link">
<table width="100%" height="91" border="1"
cellpadding="2" cellspacing="0"
bordercolor="#DDDDDD">
<tr>
<th height="19" bgcolor="#CCCC99"
scope="col"><span class="style8">Menú Supervisor
</span></th>
</tr>
<tr>
<td width="7%" align="center"
bgcolor="#CCCC99"><strong>
<span class="style8"
style="background-color:
#FF8A8A; border-width: 2px; border-style: ridge
ridge ridge ridge; border-color: #FFFF99;" >
<strong><%=MensajeError %></strong>

```

```

</span>
    </strong></td>
    <%>
    else {%>
    <td width="7%" align="center"
bgcolor="#CCCC99"><strong>
    <span class="style8"
        style="background-color:
#FF8A8A; border-width: 2px; border-style: ridge
ridge ridge ridge; border-color: #FFFF99;" >
        <strong>Mensajes</strong>
    </span>
    </strong></td>
    <%> %>
</tr>

    <tr>
    <td>&nbsp;</td>
</tr>
</table></td>
    <td width="78%" class="h2link"><table
width="100%" border="0" cellpadding="2"
cellspacing="0" class="style4">
    <tr>
    <th bgcolor="#6699CC" scope="col"><div
align="right" class="style7">Fecha:</div></th>
    <th bordercolor="#DDDDDD"
bgcolor="#6699CC" scope="col"><div align="left"><
%=dateFormat.format(new Date()) %></div></th>
    </tr>
    <tr>
    <th width="24%" bgcolor="#6699CC"
scope="col"><div align="right" class="style6"><span
class="style8">Supervisor:</span> </div></th>
    <th width="76%" bgcolor="#6699CC"
scope="col"><div align="left"><%=usu.getNombre()+
"+usu.getApellido() %></div></th>
    </tr>
    </table></td>
</tr>
<tr>
    <td valign="top" class="h2link">
    <table width="100%" height="100%"
border="0">
    <tr>
    <th colspan="2" scope="col"><span
class="style8">Consultas General de Llamadas
Realizadas</span></th>
    </tr>
    <tr>
    <td colspan="2" align="center">
    <table width="100%" height="100%"
border="0">
    <tr>
    <th scope="col">
    <div id="apDiv4">
    <div id="TabbedPanels1" class="TabbedPanels">
    div align=left>
    <div id="cabecera" >
    <span class="style10"
style="background-color: #FFFF99; border-width:
2px; border-style: ridge ridge ridge ridge; border-
color: #FFFF99;">
    <strong>ConsultaGeneral</strong>
    </span>

```

```

<div id="cuerpo1" style="background-color: #CCCCCC;
border-width: 2px; border-style: ridge ridge ridge
ridge; border-color: #FFFF99;display: block">

        <form id="form1" name="form1"
method="post" action="" >
        <p align="center"
class="Estilo3"> </p>
        <p></p>
        <p align="center"
class="Estilo3">Ingrese Rango de Fechas
        </p>
        <p align="left"><span
class="style8">Desde:
                <input size="10"
id="fecha_desde" name="fecha_desde" type="text"
READONLY onclick="displayCalendar(this)"
                title="MM/DD/YYYY" >
                Hasta:
                <input size="10"
id="fecha_hasta" name="fecha_hasta" type="text"
READONLY onclick="displayCalendar(this)"
                title="MM/DD/YYYY"
        >
                </span>
        <span class="style8">
        <p align="center"
class="Estilo3">Codigo o Nombre del Agente </p>
        <p></p>
        <p align="left"><span
class="style8">
                Código :
                <input type="text" name="co"
id="co" size="10" />
                Nombre : <input
type="text" name="no" id="no" />
                </span>
                </p>
                <span class="style8">
        <p align="center"
class="Estilo3">Ultimas llamadas Realizadas </p>
        <p></p>
        <p align="left">
                <input type="text"
name="numero" id="numero" />
        </p>
                </span>
        <span class="style8">
        <p align="center"
class="Estilo3">Consultar las Llamadas Realizadas
por su Estado </p>
        <p></p>
        <p align="left">
                <input type="radio"
name="estado" id="co" value="co" />
                <label
for="Cadena">Contestadas</label>
        </p>
        <p align="left">
                <input type="radio"
name="estado" id="noco" value="noco" />
                <label for="25">No
Contestadas</label>

```

```

</p>
        <p align="left">
            <input type="radio"
name="estado" id="fa" value="fa" />
            <label
for="35">Fallidas</label>
        </p>
    </p>
    </span>
    <span class="style8">
        <p align="center"
class="Estilo3">Tipo de Evento </p>
        <p></p>
        <p>Eventos </p>

        <select name="tipo_evento">
            <% java.util.List mostrarEventos=null;
            try {
                mostrarEventos=eDAO.MostrarEventos();
            } catch (Exception e)
{System.out.println(e.getMessage());mostrarEvent
os=null;}
            if(!(mostrarEventos==null)) {
                int i=0;
                for (Object val : mostrarEventos) {
                    Eventos evento=(Eventos)val;
                    if(i > 0){
                        %> <option id="<
%=evento.getCodigo_evento() %>"><
%=evento.getDescripcion() %></option>

                    <% }
                    else {
                        i++;%>
                        <option
id="Seleccione">Seleccione</option>
                    <% }
                    i++;
                } %>
            </select>

        </span>

        <p>
            <input type="submit"
name="eventoAG" id="eventoAG"
value="Consultar"/>
        </p>
    </form>
</div>
    <form id="form4" name="form4"
method="post" action="">
        <div id="cuerpo4"
style="background-color: #CCCCCC; border-width:
3px; border-style: ridge ridge ridge ridge;
border-color: green;display: none">
            <span class="style8">
                <p align="center"
class="Estilo3">Consultar las Llamadas
Realizadas por el Cedula o Nombre del Cliente
</p>
        <p></p>
    </p></p>

```



### 3.7.3 Proceso Estructura Agente-Clientes

```
<%@ page language="java"
import="java.util.*,Web_util.WebUtil,Tablas.DAO.*"
pageEncoding="ISO-8859-1"%>
<%@ page
import="Tablas.beans.*,java.text.SimpleDateFormat"
%>
<%
//Esta pagina me ayuda a la creaci3n de la
estructura Agentes-Clientes
//y con la clase AgentesClienetsDAO puedo armar esa
estructura del agente.
UsuarioDAO uDAO=new UsuarioDAO();
boolean estado=true;
AgenteClientesDAO cDAO=new AgenteClientesDAO();
HttpSession id_session=request.getSession();
Usuarios usu=new Usuarios();
String var=null;
String num_registro=null;
SimpleDateFormat dateformat=new
SimpleDateFormat("dd-MMMM-yyyy HH:mm:ss");
EstructuraAgentes NumEstruc=null;
try {
var=id_session.getAttribute("var1").toString();
} catch(Exception e)
{System.out.println(e.getMessage()+"aa");}
if ((var==id_session.getId())){
System.out.println("Session
viejas"+id_session.getId());
num_registro=id_session.getAttribute("reg").toStrin
g();
//System.out.println(id_session.getAttribute("reg")
.toString());
//cDAO.PruebaParroquia();
usu=uDAO.consulta_Registro(num_registro).getUsuario
_in();
System.out.println(usu.getEstructura_Agentes());
cDAO.AsignacionClientes(usu.getEstructura_Agentes()
);
NumEstruc=usu.getEstructura_Agentes();

if(WebUtil.getString(request,"Salir").equals("Salir
")){
uDAO.CerrarRegistro(num_registro);
uDAO.Desactivo(usu.getLogin());
id_session.invalidate();
response.sendRedirect("index.jsp");
}
}
else{
System.out.println("Session
nueva"+id_session.getId());
estado=false;
response.sendRedirect("Error_login.jsp");}%>
```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<%@page import="Web_util.WebUtil;"%>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=utf-8" />
<title>Agentes Clientes</title>
<style type="text/css">
<!--
.style3 { color: #EEEEEE;
font-weight: bold;
font-size: 10pt;}
--></style>
<link href="estilos.css" rel="stylesheet"
type="text/css" media="all" />
<style type="text/css">
<!--
.style4 {color: #DDDDDD}
.style6 {color: #EEEEEE; font-weight: bold; }
.style7 {color: #EEEEEE; font-weight: bold; font-
size: 8pt; }
.style8 {font-size: 8pt}
.style9 {font-size: 8.5pt}
--></style>
</head>
<body><form id="form1" name="form1" method="post"
action="">





```

```

try {
descripcionEvento=usu.getEstructura_Agentes().getEvento().getDescripcion();
} catch (Exception e)
{System.out.println(e.getMessage());
}
%>
<%=descripcionEvento %>
</th>
</tr>
<tr>
<td>
<div align="center">
<%
String detalleEvento=null;
try {
detalleEvento=usu.getEstructura_Agentes().getEvento().getDetalle();
} catch (Exception e)
{System.out.println(e.getMessage());
}
%>
<%=detalleEvento %>
</div>
</td>
</tr>
</table></td>
<td width="73%" class="h2link">
<table width="100%" border="0"
cellpadding="2" cellspacing="0" class="style4">
<tr>
<th bgcolor="#6699CC" scope="col"><div align="right" class="style7">Fecha:</div></th>
<th bordercolor="#DDDDDD" bgcolor="#6699CC" scope="col"><div align="left"><%=dateformat.format(new Date()) %></div></th>
<th bgcolor="#6699CC" scope="col"><div align="right" class="style7">EXT:<strong><%=usu.getExtension() %></strong></div></th></tr>
<tr>
<th width="24%" bgcolor="#6699CC" scope="col"><div align="right" class="style6"><span class="style8">Agente</span>:</div></th>
<th width="70%" bgcolor="#6699CC" scope="col"><div align="left"><%=usu.getNombre() +usu.getApellido() %></div></th>
<th width="116%" bgcolor="#6699CC" scope="col"><div align="left"></div></th>
</tr>
</table></td>
</tr>
<tr>

```

```

<td class="h2link"><p
align="center"><strong>Listado de
Clientes</strong></p>
<table width="100%" border="0"
cellspacing="0" cellpadding="2">
<tr>
<th colspan="2" bgcolor="#DDDDDD"
scope="col"><span class="style9">Nombre del
Cliente</span></th>
<th width="29%" bgcolor="#DDDDDD"
scope="col"><span
class="style9">Teléfono</span></th>
</tr>
<%
if(estado) {
java.util.List Listar_Clientes=null;
int count=0;
try {

Listar_Clientes=cDAO.ListarClientes(Clientes.class,
NumEstruc.getCodigo_estructura());
} catch (Exception e) {
//System.out.println(e.getMessage()
+"Cargar lista de Clientes");
Listar_Clientes=null;
}
if(!(Listar_Clientes==null))
{
for (Object val :
Listar_Clientes)
{
count++;
Clientes cliente =
(Clientes)val;
%>
<tr>
<td width="42%">
<%if (count==1)
{
%>
<a href="DatosClientes.jsp"><
%= cliente.getNombre_Cli() %></a>
<%
try {

id_session.setAttribute("cedula",cliente.getCedula(
));
} catch (Exception e)
{System.out.println(e.getMessage()+"Traer campo de
la pagina anterior");
} } %>
<td width="29%"><%=
cliente.getNombre_Cli() %></td>
<td>
<div align="center">
<%=cliente.getTelefono_Cli() %>
</div></td>
</tr>

```



### 3.7.4 Proceso Grabaciones Llamadas.

```
%@ page language="java"
import="java.util.*,Web_util.WebUtil,Tablas.DAO.UsuarioDAO" pageEncoding="ISO-8859-1"%>
<%@ page
import="Tablas.beans.*,java.text.SimpleDateFormat" %>
<%
// Esta página es la del supervisor y cuando se
realiza un check en el agente y luego se actualiza
// la llamada del agente sera grabada.
UsuarioDAO uDAO=new UsuarioDAO();
HttpSession id_session=request.getSession();
String var=null;
boolean valorSeccion=false;
Usuarios usu=new Usuarios();
List agentes=null;
String num_registro=null;
SimpleDateFormat dateformat=new SimpleDateFormat("dd-
MMMM-yyyy HH:mm:ss");
try {
var=id_session.getAttribute("var1").toString();
} catch(Exception e)
{System.out.println(e.getMessage());}

    if ((var==id_session.getId())){
    valorSeccion=true;
    //System.out.println("Session
vieja"+id_session.getId());

num_registro=id_session.getAttribute("reg").toString(
);

//System.out.println(id_session.getAttribute("reg").t
oString());
    try {
usu=uDAO.consulta_Registro(num_registro).getUsuario_i
n();
//System.out.println(num_registro+usu.getApellido());
}
catch(Exception e){System.out.println(e.getMessage()
+"ERRORRRRR");}
if(WebUtil.getString(request,"Salir").equals("Salir")
){
valorSeccion=false;
uDAO.CerrarRegistro(num_registro);
uDAO.Desactivo(usu.getLogin());
id_session.invalidate();
response.sendRedirect("index.jsp");
}
}
```

```

else
    if(WebUtil.getString(request,"evento").equals("
Consulta Basica  ")) {
        response.sendRedirect("Consultas.jsp");
    }
    else

if(WebUtil.getString(request,"evento").equals("Definir
Tipo Evento")) {
    response.sendRedirect("Evento.jsp");
}
else
if(WebUtil.getString(request,"evento").equals("
Consulta General  ")) {
    response.sendRedirect("ConsultasGenerales.jsp");
}

else
if(WebUtil.getString(request,"Actualizar").equals("Act
ualizar")) {
try {
    uDAO=null;
    uDAO=new UsuarioDAO();
    agentes=uDAO.ListarAgentes (usu.getCodigo());
    } catch (Exception e)
{System.out.println(e.getMessage()); agentes=null;}
    if (!(agentes==null)) {

        for (Object val : agentes) {
            Usuarios usu2=(Usuarios)val;
            System.out.println(usu2.getCodigo());
            String
estadoGrabacion=WebUtil.getString(request,usu2.getCodi
go());
            System.out.println(estadoGrabacion);
            if (!(estadoGrabacion.equals(""))){

uDAO.ActualizarEstadoGrabacion(estadoGrabacion,"AG");
            }
            else

uDAO.ActualizarEstadoGrabacionTodos(usu2.getCodigo());

        }
    }
} else if(WebUtil.getString(request,"evento").equals("
Grabaciones  ")) {/**allan
    response.sendRedirect("Grabaciones.jsp");
}/**
} else{
System.out.println("Session
nueva"+id_session.getId());
response.sendRedirect("Error_login.jsp");
} %>

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=utf-8" />
<!-- TemplateBeginEditable name="doctitle" -->
<title>Supervisor TelefonIVRX</title>
<!-- TemplateEndEditable -->
<!-- TemplateBeginEditable name="head" --><!--
TemplateEndEditable -->
<link href="estilos.css" rel="stylesheet"
type="text/css" media="all" />
<style type="text/css">
</style></head>
<%if(valorSeccion){//mostrar mensaje de evento
guardado
String valorEventoJSP=null;
try {
valorEventoJSP=(String)id_session.getAttribute("Va
loParro");
}catch(Exception e){System.out.println("Error de
update"+e.getMessage());valorEventoJSP=null;}
//System.out.println(valorEventoJSP);
if(valorEventoJSP==null){
%><body class="oneColElsCtr"
onLoad=window.setTimeout("location.href='Supervis
or.jsp'",10000)>
<%}else{
id_session.setAttribute("ValoParro",null);
%><body class="oneColElsCtr"onLoad="alert('EL
evento ha sido guardado exitosamente');">
<%}} %><form id="form1" name="form1" method="post"
action="">
<table width="33%" border="0" align="center"
cellpadding="2" cellspacing="1"
bordercolor="#0099CC">
<tr><th colspan="2" scope="col"></th></tr> <tr> <td class="h7link"></td>
<td class="h2link">&nbsp;</td></tr><tr>
<td width="27%" rowspan="2" valign="top"
class="h7link">
<table width="100%" height="91" border="1"
cellpadding="2" cellspacing="0"
bordercolor="#DDDDDD">
<tr>
<th height="19" bgcolor="#6699CC"
scope="col"><span class="style9">
Menu Supervisor
</span></th>
</tr>

```

```

<tr>
  <td width="7%" align="center"
bgcolor="#CCCC99"><strong>
  <input type="submit" name="evento"
id="evento" value="Definir Tipo Evento" />
  </strong></td>
</tr>
<tr>
  <td width="7%" align="center"
bgcolor="#CCCC99"><strong>
  <input type="submit" name="evento"
id="evento" value="  Consulta General  " />
  </strong></td>
</tr>
<tr>
  <td width="7%" align="center"
bgcolor="#CCCC99"><strong>
  <input type="submit" name="evento"
id="evento" value="  Consulta Basica  " />
  </strong></td>
</tr>
<tr>
  <td width="7%" align="center"
bgcolor="#CCCC99"><strong>
  <input type="submit" name="evento"
id="evento" value="  Grabaciones  " />
  </strong></td>
</tr>
<tr>
  <td <div align="center" </di> </td>
</tr>
</table></td>

  <td width="73%" class="h2link">
  <table width="100%" border="0"
cellpadding="2" cellspacing="0" class="style4">
  <tr>
  <th bgcolor="#6699CC" scope="col"><div
align="right" class="style7">Fecha:</div></th>
  <th bordercolor="#DDDDDD"
bgcolor="#6699CC" scope="col"><div align="left"><
%=dateformat.format(new Date()) %></div></th>
  </tr><tr><th width="24%" bgcolor="#6699CC"
scope="col"><div align="right"
class="style6"><span class="style8">Agente</span>:
  </div></th>
  <th width="76%" bgcolor="#6699CC"
scope="col"><div align="left"><%=usu.getNombre () +
"+usu.getApellido() %></div></th></tr>

```

```

</table></td>
</tr>
<div id="apDiv2">
    <tr>
        <td class="h2link"><p
align="center"><strong>Listado de
Agentes</strong></p>
        <table width="100%" border="0"
cellspacing="0" cellpadding="4">
            <tr>
                <th width="42%" colspan="1"
bgcolor="#DDDDDD" scope="col">Agente</span></th>
                <th width="12%" bgcolor="#DDDDDD"
scope="col">Estado</span></th>
                <th width="22%" bgcolor="#DDDDDD"
scope="col">Tiempo Actividad</th>
                <th width="24%" bgcolor="#DDDDDD"
scope="col">Grabacion</th>
            </tr>
        </table>
        <table width="135%" height="251" border="3"
align="left" cellpadding="2" cellspacing="1"
class="oneColElsCtr" id="container">
            <caption align="right">
                <br />
            </caption>
            <% String color=null;
int cont=0;
try {
uDAO=null;
uDAO=new UsuarioDAO();
agentes=uDAO.ListarAgentes (usu.getCodigo());
} catch (Exception e)
{System.out.println (e.getMessage());agentes=null;}
if (! (agentes==null)) {
for (Object val : agentes) {
    Usuarios usu2=(Usuarios)val;
cont++;
if (usu2.getActivo().equals ("ON")) {
color="style6";
}
else
color="style5";
%><tr> <th width="310" scope="col"><span
class=<%=color%>><%=usu2.getNombre ()+"
"+usu2.getApellido ()+" (" +usu2.getLogin ()+" ) "
%></span></th>
        <th width="90" scope="col"><strong><
%=usu2.getActivo () %></strong></th>
        <% Date
date=uDAO.retornarHoraIngreso (num_registro);
//System.out.println (date);
%>

        <%if (usu2.getActivo ().equals ("ON")) {
Date
horaConectado=usu2.getHora_conexion ();
long time=new Date ().getTime ()-
horaConectado.getTime ();
//float conectado=time/1000;
int conectadoInt=(int)time/1000;
//System.out.println (time);
if (conectadoInt <=60) {
%>

```

### 3.8 Jasper Reports

JasperReports es una poderosa herramienta de código abierto que tiene la capacidad para entregar contenidos enriquecidos en la pantalla, a la impresora, o en PDF, HTML, XLS, RTF, ODT, CSV, TXT y archivos XML. Está escrito completamente en Java y puede ser usado en una variedad de aplicaciones Java para crear contenido dinámico. Su objetivo principal es ayudar a crear la página orientada lista para imprimir documentos de forma sencilla y flexible.

JasperReports organiza los datos recuperados de una fuente de datos, según un informe de diseño definido en un archivo JRXML. Con el fin de llenar un informe con los datos, el informe de diseño debe ser compilado primero. La elaboración de las JRXML archivo que representa el informe de diseño se realiza por el compileReport () por el método expuesto JasperCompileManager clase.

A través de la compilación, el informe de diseño se carga en un informe de diseño de objetos que se serializa y se almacena en el disco (JasperReport clase). Este objeto serializado se utiliza cuando la solicitud se quiere llenar el informe se especifica con los datos de diseño. De hecho, la elaboración de un informe de diseño implica la recopilación de todas las expresiones de Java se define en el archivo JRXML que representa el diseño de informes. Diversas verificaciones se realizan en tiempo de compilación, el informe para comprobar la coherencia de diseño. El resultado es un listo para llenar informe de diseño que se utiliza para generar documentos en diferentes conjuntos de datos.

Con el fin de llenar un informe de diseño, se puede utilizar el fillReportXXX () los métodos expuestos por el JasperFillManager clase.

```
<td height="32" class="h7link"></td>
<td class="h2link">
  <div align="left">
    <input type="submit" name="Actualizar"
    id="Actualizar" value="Actualizar" />
```



Los parámetros son referencias a objetos que se pasan en el informe de las operaciones de llenado. Son muy útiles para pasar el informe de datos del motor que no puede encontrar normalmente en su fuente de datos. Por ejemplo, podríamos pasar a la memoria del motor el nombre del usuario que ha lanzado el informe de operación de llenado, si queremos que aparezca en el informe, o podemos cambiar dinámicamente el título de nuestro informe.

Un aspecto importante es el uso de parámetros de los informes en la cadena de la consulta del informe, a fin de poder personalizar aún más el conjunto de datos recuperados de la base de datos. Estos parámetros pueden actuar como filtros dinámicos en la consulta que el suministro de datos para el informe.

Declarar un parámetro en un informe de diseño es muy sencillo y sólo requiere que especifique su nombre y su clase:

```
<parameter name="ReportTitle"
class="java.lang.String"/>
<parameter name="MaxOrderID"
class="java.lang.Integer"/>
<parameter name="SummaryImage"
class="java.awt.Image"/>
```

Se debe descargar el JasperReport de la siguiente dirección:

[http://jasperforge.org/website/jasperreportswebsite/trunk/documentation.html?group\\_id=252](http://jasperforge.org/website/jasperreportswebsite/trunk/documentation.html?group_id=252)

# MANUAL USUARIO

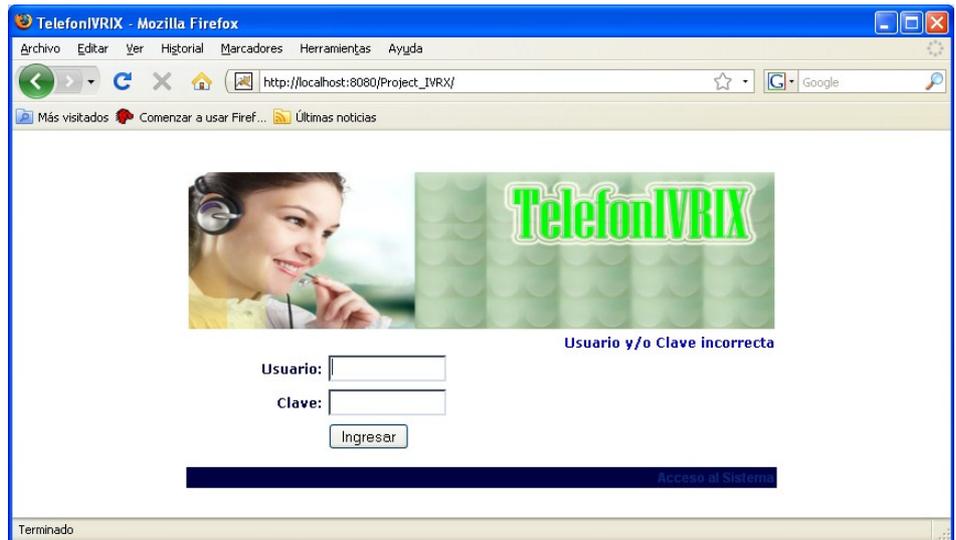
## **Introducción**

A continuación se detallara la parte gráfica del sistema y las diferentes opciones que podemos manejar en la interfaz del sistema.

### **1.1 Ingreso del Sistema**

Tenemos la pantalla principal del sistema solicitando su usuario y contraseña para el acceso al mismo.

Una vez que se ingresan los datos damos click sobre el botón ingresar y se accederá a la página indicada según el tipo de usuario (Administrador, Supervisor, Agente) que acceda, en caso de ser usuario o clave incorrecta se mostrará el mensaje (Usuario y/o Contraseña Incorrecta).



**Figura 1.a Ingreso al sistema.**

*A continuación se mostrará las pantallas usadas por los Agentes.*

## **1.2 Lista de Clientes**

Como se puede apreciar aparecerá la lista de clientes a las que el agente debe llamar.

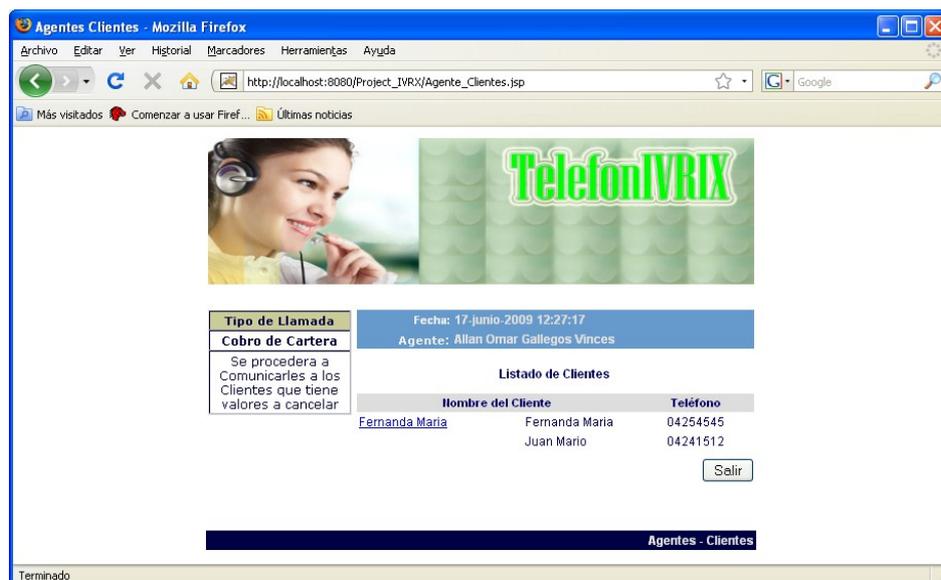
Se maneja por prioridad, esto quiere decir que el sistema ordenará la lista de clientes acorde a la cantidad que adeude el cliente o si al cliente se lo ha llamado y no ha contestado.

Una vez generada la lista se activará como link el usuario que hay que contactar, el resto de clientes permanecerá como texto, adicionalmente hay una columna con el teléfono del cliente.

Entonces cuando se llame a un cliente, se irá a la pantalla de datos del cliente y en el mismo momento se generará la llamada. Una vez culminada la llamada se regresará a la lista de clientes y se activará el siguiente cliente como link y el anterior quedará como texto y así sucesivamente hasta que termine con los clientes asignados a ese agente.

Además donde aparece “**Tipo de Llamada**” nos indicará si la llamada que debe realizarse a los clientes es de cobros o de publicidad, en caso de ser publicidad se enviará un texto para que el agente se lo comunique al cliente.

El botón Salir lo regresará a la pantalla de ingreso al Sistema.



**Figura 1b Lista de Clientes.**

### **1.3 Datos del Cliente**

En esta pantalla se mostrarán todos los datos del cliente, y del cual ciertos campos serán inactivados por el sistema, quedando habilitados: la dirección, los números de teléfono y el email que son los datos que pueden variar.

La llamada se realiza una vez que fue seleccionado el cliente en la página de Lista de Clientes, luego de finalizar la llamada se manejarán varios estados de llamada entre fallidas y exitosas.

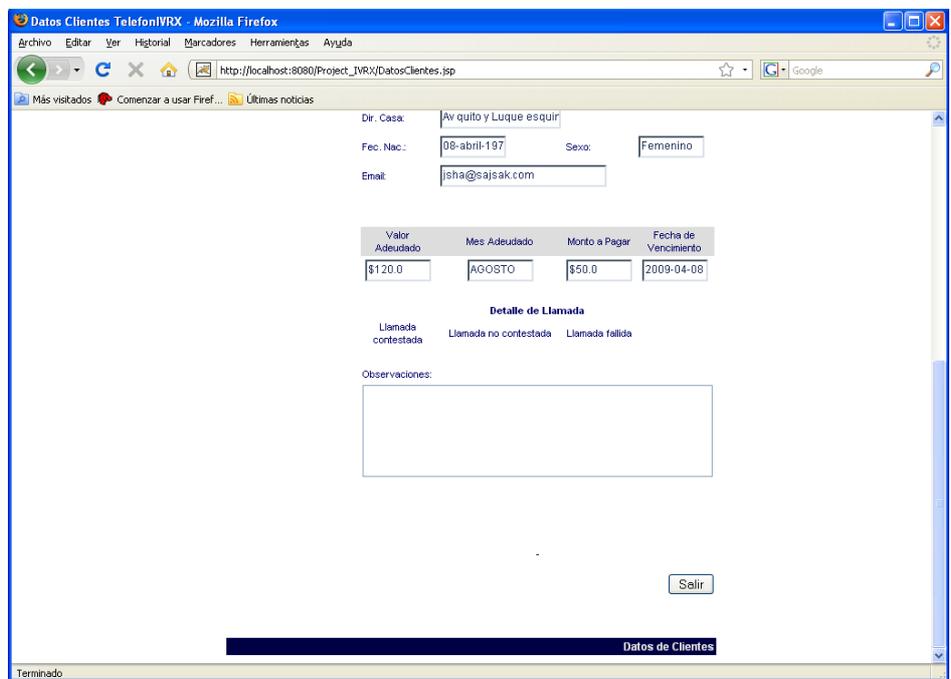
En la observación se detallara el compromiso del cliente para cancelar la deuda o detallar el motivo por el cual la llamada fue fallida.

Realizado esto, damos en Guardar para actualizar los campos y volver a la página de Lista de Clientes.

El botón Salir lo regresará a la pantalla de ingreso al Sistema.

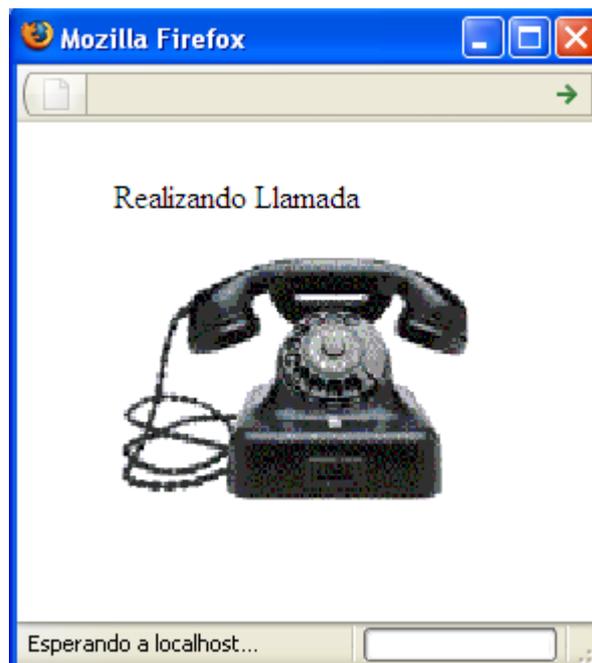


**Figura 1c.1 Datos de Cliente.**



**Figura 1c.2 Datos de Cliente.**

Si damos click en el botón llamar se realizará la llamada al cliente mostrándonos la siguiente pantalla:



*A continuación se mostrarán las pantallas que serán manejadas por el supervisor.*

## **1.4 Lista de Agentes**

En esta pantalla mostrará los agentes conectados al sistema (solo los que han sido asignados a ese supervisor), en caso de que el agente esté conectado en el sistema en la columna estado será de ON, caso contrario será de OFF.

Adicionalmente, la columna de tiempo se muestra el tiempo que un agente ha durado en la llamada actual.

En la columna de Grabación el Supervisor seleccionará esta opción en caso que desee grabar la llamada de un agente.

Tenemos 2 botones que, el primero Realizar Consultas y el otro Definir Tipo de Evento que serán explicados posteriormente.



**Figura 1d Lista de Agentes.//Cambiar**

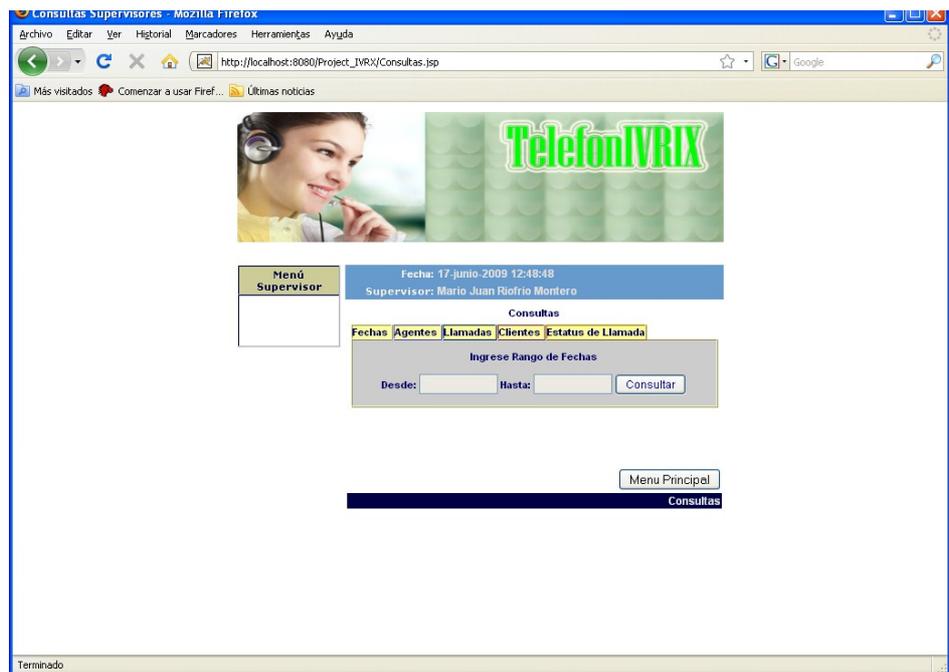
## 1.5 Realizar Consultas

Al dar click en el botón de Realizar Consultas, se abrirá la página mostrada a continuación, la que nos indicará el tipo de consulta que quiera realizar el supervisor acorde al menú que seleccione: Fechas, Agentes, Llamadas, Clientes, Status de Llamada.

Primero tenemos consulta por Fechas para ver, para revisar las llamadas que se han hecho durante las fechas que fueron definidas por el supervisor.

En el botón Consultar nos mostrará los resultados de la consulta, acorde a las fechas que se hayan ingresado.

En el botón Menú Principal retornaremos a la pantalla de Lista de Agentes

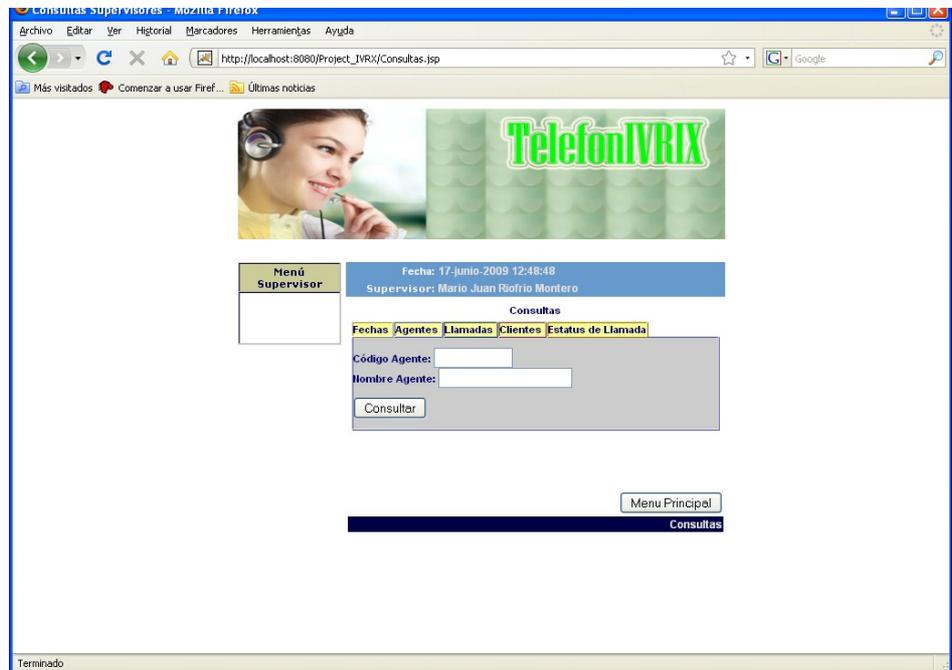


**Figura 1e Consulta por Fechas.**

En la consulta por Agentes se mostrará todas las llamadas realizadas por el agente.

En el botón Consultar nos mostrará los resultados de la consulta, acorde al código del agente o el nombre de l agente que se haya ingresado.

En el botón Menú Principal retornaremos a la pantalla de Lista de Agentes.

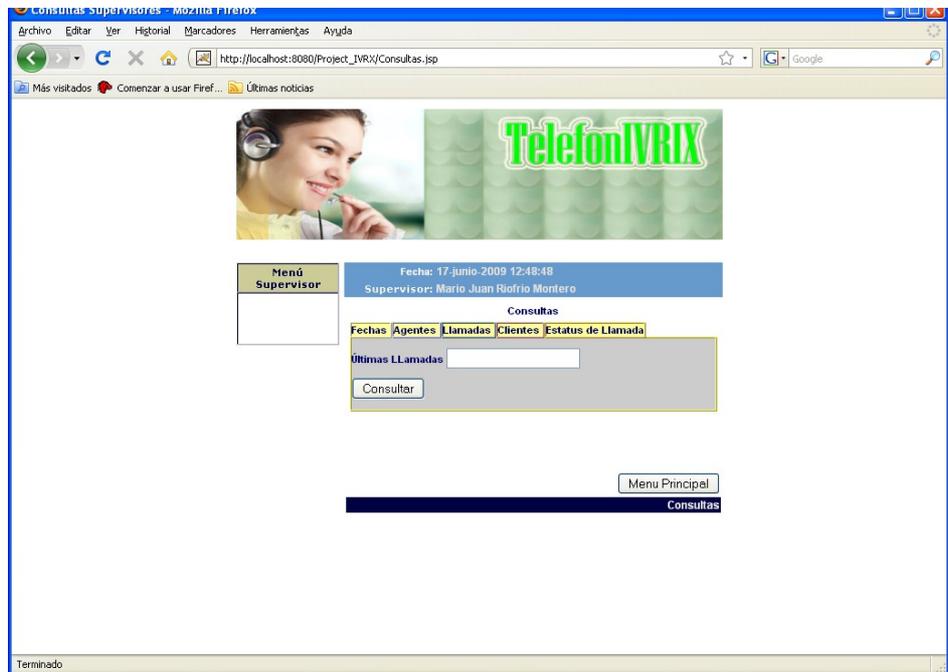


**Figura 1f Consulta por Agentes.**

En la consulta por Llamadas, se mostrará las ultimas llamadas realizadas por los agentes que estén asignados a es supervisor.

En el botón Consultar nos mostrará los resultados de la consulta, acorde al parámetro ingresado.

En el botón Menú Principal retornaremos a la pantalla de Lista de Agentes.



**Figura 1g Consulta por llamadas.**

En la consulta por Clientes, se mostrará las llamadas que han sido realizadas hacia el cliente que se indique.

En el botón Consultar nos mostrará los resultados de la consulta, acorde a los datos del cliente que se haya ingresado.

En el botón Menú Principal retornaremos a la pantalla de Lista de Agentes.



**Figura 1h Consulta por Clientes.**

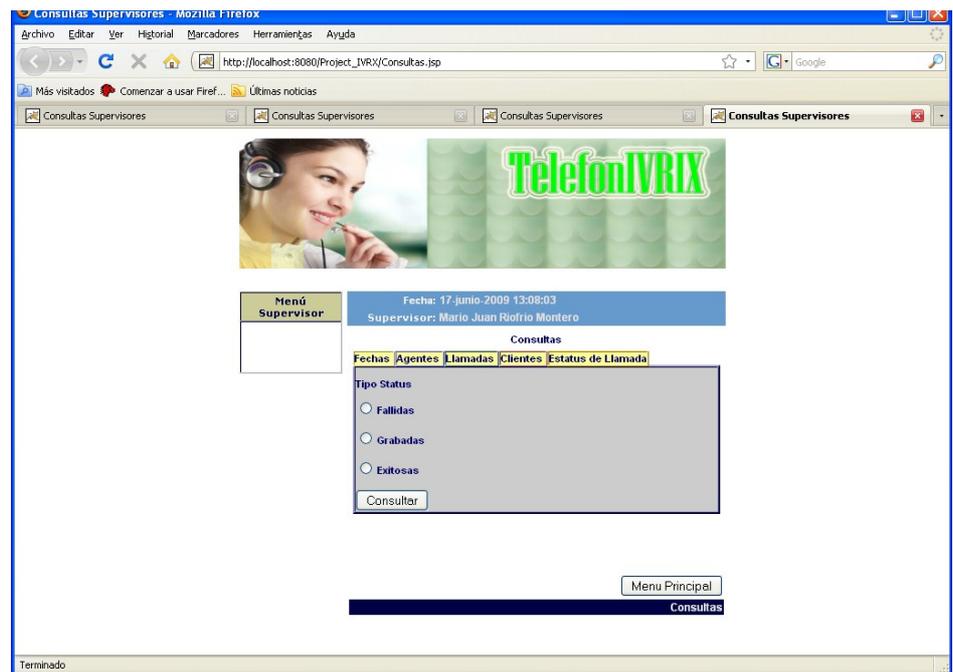
En Consulta por Status de Llamada se mostrará al tipo de llamada que se haya seleccionado.

Se tienen tres estatus de llamadas:

- Fallidas
- Grabadas
- Exitosas

En el botón Consultar nos mostrará los resultados de la consulta, acorde al status de llamada seleccionado.

En el botón Menú Principal retornaremos a la pantalla de Lista de Agentes.



**Figura 1i Consulta por Status de Llamada.**

## 1.6 Definir Tipo de Evento

Al dar click en el botón Definir tipo de Evento en la página de Lista de Agentes, nos mostrará esta pantalla en la que el supervisor definirá el tipo de evento se realizará para sus agentes siendo estos: Cobro de Cartera y Telemercadeo.

Se tiene dividida a la ciudad en Parroquias para que según eso, los supervisores les asignen a los agentes el sector con el cual ellos deban trabajar.

En el cuadro de Detalle de la Llamada es lo que va a aparecer en las pantallas de los respectivos agentes, ya que será el mensaje que los agentes tengan que comunicar a sus clientes en el momento de la llamada.

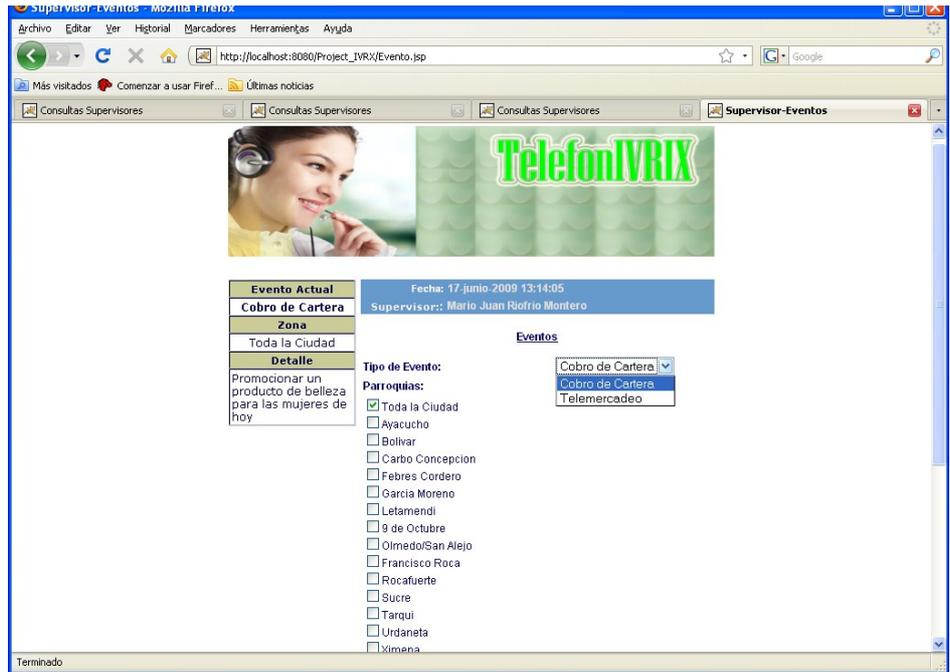
En Evento Actual, se mostrará el tipo de operación que realizarán los agentes que fue definido por el supervisor.

En Zona se mostrará a los agentes el sector que fue definido por el supervisor.

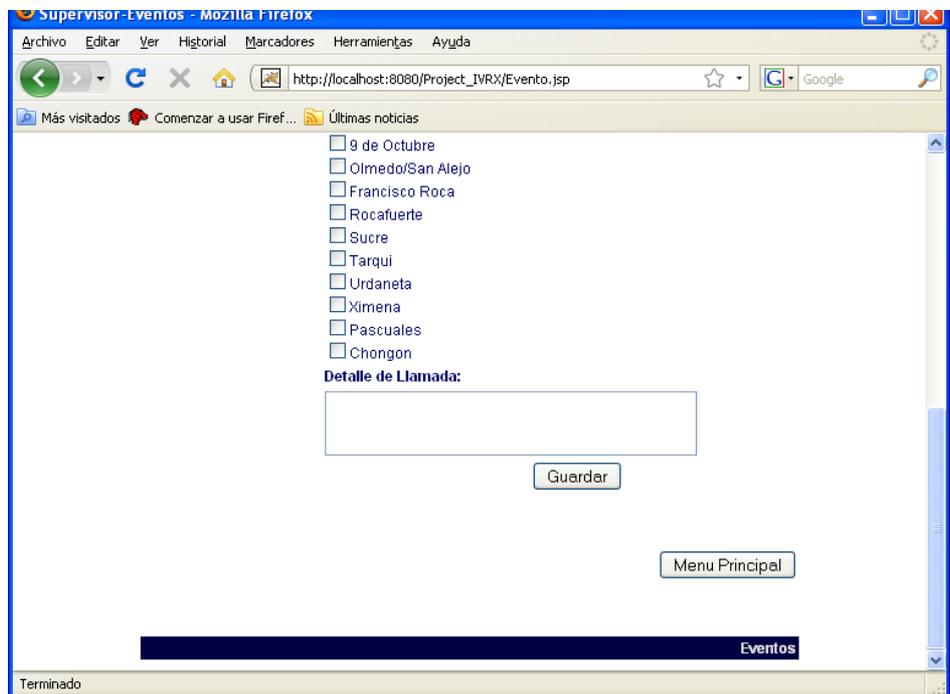
Y en Detalle de la Llamada, se mostrará el mensaje que fue definido por el supervisor

En el botón Guardar, se guardará la información ingresada por el Supervisor.

En el botón Menú Principal retornaremos a la pantalla de Lista de Agentes.



**Figura 1.j.1 Definir Tipo de Evento.**



**Figura 1.j.2 Definir Tipo de Evento.**

*Las pantallas que se mostrará a continuación serán usadas por los Administradores del Sistema.*

## **1.7 Creación y Modificación de Usuarios**

Cuando ingresamos con privilegios de Administrador, nos mostrará la pantalla de Administrador, el cual nos mostrará un menú con dos opciones: Creación y Modificación.

Cuando seleccionamos en Creación, se solicitarán todos los datos necesarios del nuevo usuario.

Donde se indica Lorin y Pass Word son los datos con los cuales accederán al sistema.

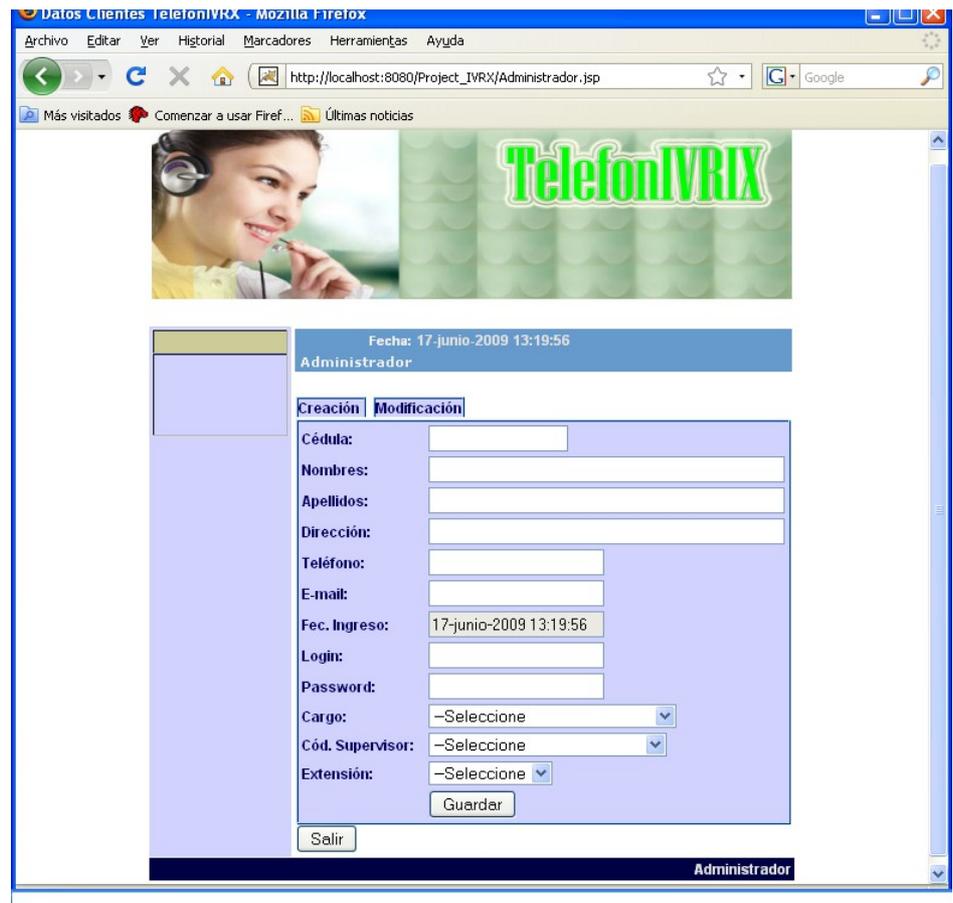
Si el cargo que se selecciona es de Agente se mostrará todos los supervisores que existen en el sistema pero en caso de que se seleccione como cargo Supervisor automáticamente se desaparecerá el listbox de Supervisores.

En extensión será la línea que tendrá asignado el Agente que será la persona encargada de realizar la llamada a los clientes, tanto el Supervisor como el Administrador no se les asignará extensión.

Una vez ingresado los datos procedemos a Guardar los datos, en caso de existir campos vacíos se impedirá almacenar la información, al igual en caso de ingresar datos que no coincide con lo que se solicita como cédula incorrecta, email inválido, etc. y si los datos fueron

correctamente ingresados el sistema limpiará los campos donde se ingreso el texto.

En el botón Salir retornaremos a la pantalla de ingreso al Sistema.



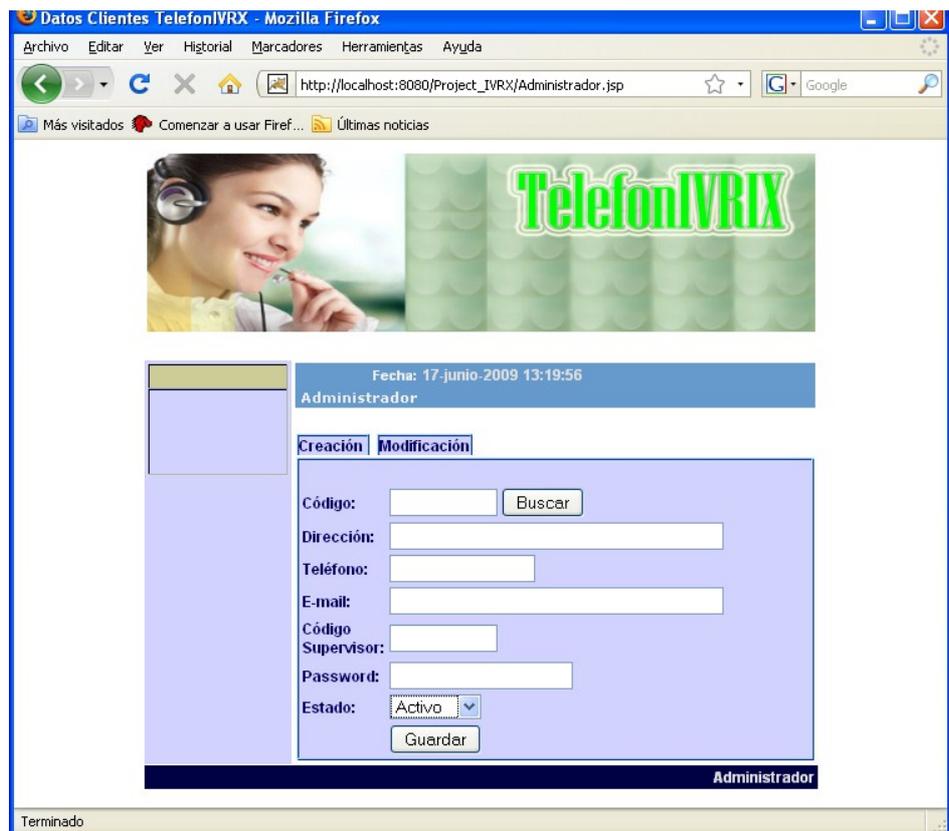
**Figura 1k Creación de Usuarios.**

Cuando seleccionamos Modificación se ingresará el agente que quisiéramos modificar, al momento de dar click en el botón Buscar se mostrará toda la información del agente que se ingresó.

En el Estado se indicará si el empleado esta actualmente laborando en la empresa (Activo) o no (Inactivo).

En el botón Guardar, se guardará la información ingresada por el Administrador.

En el botón Salir retornaremos a la pantalla de ingreso al Sistema.



**Figura 11** Modificación de Usuarios. **//Cambiar**

**Requisitos para preparación del equipo del agente:**

**Requisitos para preparación del equipo del supervisor:**

## **Bibliografía**

[http://www.nettix.com.pe/manuales/instalacion\\_de\\_servidor\\_centos\\_5.2/Page-3.html](http://www.nettix.com.pe/manuales/instalacion_de_servidor_centos_5.2/Page-3.html)

[http://www.contactcentervoip.com/ES/taxonomy\\_menu/9/44](http://www.contactcentervoip.com/ES/taxonomy_menu/9/44)

<http://www.voipforo.com/asterisk/configuracion-sip-conf.php>

<http://www.alcancelibre.org/staticpages/index.php/como-ekiga-asterisk>

[http://www.counterpath.net/assets/files/191/X-Lite3.0\\_UserGuide.pdf](http://www.counterpath.net/assets/files/191/X-Lite3.0_UserGuide.pdf)

<http://asterisk-java.org/development/tutorial.html>

[http://www.contactcentervoip.com/ES/taxonomy\\_menu/9/30](http://www.contactcentervoip.com/ES/taxonomy_menu/9/30)

<http://curso--de-asterisk.blogspot.com/2009/03/dial-plan-features.html>

<http://documentacion.irontec.com/cursoAsteriskVozIP-3-introduccionAsterisk.pdf>

<http://www.voztovoice.org/?q=node/20>