

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 2**



**HERRAMIENTA ADMINISTRATIVA PARA LA PLATAFORMA PLATEL**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS**

**Autores:** Orelvis Barrera Zumaquero

Reynier Santiago González León

**Tutor:** Ing. Eduardo Javier Roig Sastre

Ciudad de La Habana, junio de 2009

## **DECLARACIÓN DE AUTORÍA**

Declaramos ser los únicos autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 23 días del mes de junio del año 2009.

**Autores:**

\_\_\_\_\_  
**Orelvis Barrera Zumaquero**

\_\_\_\_\_  
**Reynier Santiago González León**

**Tutor:**

\_\_\_\_\_  
**Ing. Eduardo Javier Roig Sastre**

***Dedicatoria:***

*A mis padres y mi hermano: Por todo el afecto, amor, apoyo y cariño que me han brindado en el transcurso de mi carrera.*

*A mis abuelos, tíos y tías: por toda la confianza y cariño que depositaron en mí.*

*A toda mi familia que de una manera u otra me apoyaron en cada momento.*

*A mis amigos y amigas: por estar a mi lado en los buenos y malos momentos.*

*Reynier Santiago González León*

*A mis padres: Por todo el apoyo, amor y cariño que me han brindado a lo largo de todos estos años, por confiar en mí y ayudarme a realizar este sueño.*

*A mi hijo: Por ser lo más hermoso que me ha dado la vida.*

*A mis hermanos: Por estar siempre presente en todo momento.*

*A mis demás familiares y amigos que siempre han estado a mi lado.*

*A todas las personas que han deseado de una forma u otra la realización de este sueño.*

*Orelvis Barrera Zumaquero*

## ***Agradecimientos***

*Va dirigido a todas aquellas personas que de una forma u otra han contribuido al desarrollo de este trabajo y a nuestra formación como profesionales, en especial a nuestro Comandante en Jefe por ser el creador de la Universidad de las Ciencias Informáticas; esa gran obra de la Revolución que nos abrió sus puertas para convertirnos en hombres de ciencia y de bien.*

*A nuestros amigos que han vivido cada momento de tristeza y alegría a nuestro lado.*

*A nuestro tutor Ing. Eduardo Javier Roig Sastre.*

*A la Ing. Maidelis Milanés Luque.*

## Resumen

Debido al incremento de las relaciones comerciales de la Universidad de las Ciencias Informáticas en el mercado del software, surge la necesidad de crear un centro de atención de llamadas para brindar información sobre los productos que se desarrollan en la institución y mantener abiertos los canales de comunicación con los clientes.

El presente trabajo de diploma tiene como objetivo el desarrollo de una herramienta de administración para la plataforma telefónica PLATEL: software para centros de contactos basado en VoIP (voz sobre el protocolo de internet).

Para ello se realiza un estudio de los sistemas de administración de plataformas telefónicas existentes basadas en IP (protocolo de internet). Se lleva a cabo, además, el desarrollo de la aplicación, así como las pruebas realizadas para asegurar la calidad del software.

En el desarrollo de la aplicación se utilizaron herramientas libres. Como lenguaje de programación se utilizó Python, TurboGears y Dojo Toolkit como *framework* de desarrollo. Para la publicación del sitio se eligió el servidor web Apache. Mientras que de todos los gestores de base de datos analizados PostgreSQL fue el seleccionado. Se aplicaron además las metodologías ágiles de desarrollo Programación Extrema (XP) y Scrum; con el objetivo de construir técnicamente un software en el menor tiempo posible, económico, fiable y que funcione eficientemente.

Con la instalación y puesta en práctica de la herramienta administrativa, se podrá automatizar un proceso de alta prioridad en el mecanismo de un centro de contactos; permitiendo realizar la gestión de las extensiones y dar seguimiento al comportamiento de las llamadas telefónicas en la plataforma.

# Índice de Contenido

## Índice de Contenido

<b>Introducción .....</b>	<b>1</b>
<b>Capítulo I: Fundamentación Teórica.....</b>	<b>4</b>
<b>1.1 Introducción .....</b>	<b>4</b>
<b>1.2 Sistemas de uso actual para la administración de Asterisk .....</b>	<b>4</b>
1.2.1 FreePBX.....	4
1.2.2 Destar.....	5
1.2.3 Asterisk GUI .....	5
<b>1.3 Aplicación web.....</b>	<b>6</b>
<b>1.4 Tecnologías y lenguajes a utilizar .....</b>	<b>7</b>
1.4.1 Modelo Cliente – Servidor .....	7
1.4.2 Tecnologías del lado del cliente.....	8
1.4.2.1 Lenguaje de Marca de Hipertextos .....	9
1.4.2.2 JavaScript .....	9
1.4.2.3 JavaScript asíncrono y XML.....	10
1.4.3 Tecnologías del lado del servidor .....	10
1.4.3 .1 Gestor de Base de Datos .....	10
1.4.3.2 MySQL.....	11
1.4.3 .3 Oracle.....	12
1.4.3 .4 PostgreSQL .....	12
1.4.3 .5 Gestor de Base de Datos seleccionado (PostgreSQL) .....	13
1.4.3 .6 Servidor web Apache.....	14
1.4.3 .7 Servidor web seleccionado (Apache).....	14
1.4.4 Lenguajes de Programación.....	15
1.4.4 .1 Java.....	15
1.4.4.2 Python .....	15
1.4.4.3 Perl.....	16
1.4.4.4 PHP .....	17
1.4.4.5 Lenguaje de Programación seleccionado (Python) .....	17

# Índice de Contenido

<b>1.5 Framework</b> .....	<b>18</b>
1.5.1 TurboGears .....	18
1.5.2 Dojo Toolkit.....	20
<b>1.6 Fundamentación de la Metodología</b> .....	<b>20</b>
1.6.1 Extreme Programing (XP) .....	21
1.6.2 Scrum.....	23
1.6.3 Proceso Unificado de Desarrollo de Software (RUP) .....	24
1.6.4 Metodología seleccionada (XP y Scrum).....	25
<b>1.7 Conclusiones</b> .....	<b>25</b>
<b>Capítulo II Características del Sistema</b> .....	<b>26</b>
<b>2.1 Introducción.</b> .....	<b>26</b>
<b>2.2 Objeto de Automatización.</b> .....	<b>26</b>
<b>2.3 Propuesta del sistema a desarrollar.</b> .....	<b>26</b>
<b>2.4 Arquitectura del sistema propuesto:</b> .....	<b>27</b>
<b>2.5 Interfaces de usuario</b> .....	<b>28</b>
<b>2.6 Requisitos no funcionales del sistema.</b> .....	<b>29</b>
<b>2.7 Conclusiones</b> .....	<b>30</b>
<b>Capítulo III: Desarrollo ágil de la herramienta administrativa para la plataforma Platel</b> .....	<b>31</b>
<b>3.1 Introducción</b> .....	<b>31</b>
<b>3.2 Fase de Exploración</b> .....	<b>31</b>
3.2.1 Personas relacionadas con el sistema.....	31
3.2.2 Historias de Usuario. ....	32
<b>3.3 Planificación.</b> .....	<b>37</b>
3.3.1 Estimación de esfuerzo por Historia de Usuario. ....	38
3.3.2 Iteraciones. ....	38
3.3.3 Plan de duración de las Iteraciones. ....	39

# Índice de Contenido

3.3.4 Plan de entrega.....	40
<b>3.4 Conclusiones .....</b>	<b>40</b>
<b>Capítulo IV: Implementación y Pruebas.....</b>	<b>41</b>
<b>4.1 Introducción .....</b>	<b>41</b>
<b>4.2 Diseño del Sistema.....</b>	<b>41</b>
4.2.1 Tarjeta CRC de la clase controladora.....	41
4.2.2 Tarjeta CRC de la clase manejadora de datos.....	42
4.2.3 Tarjeta CRC de la clase acceso a datos.....	43
<b>4.3 Primera Iteración.....</b>	<b>43</b>
4.3.1 Tareas de Ingeniería que se abordan en la iteración.....	44
<b>4.4 Segunda Iteración.....</b>	<b>48</b>
4.4.1 Tareas de Ingeniería que se abordan en la iteración.....	48
<b>4.5 Tercera Iteración .....</b>	<b>54</b>
4.5.1 Tareas de Ingeniería que se abordan en la iteración.....	55
<b>4.6 Casos de Pruebas.....</b>	<b>61</b>
4.6.1 Caso de Prueba Historia de Usuario: Gestionar extensiones SIP.....	61
4.6.2 Caso de Prueba Historia de Usuario: Gestionar buzón de voz de extensiones SIP.....	64
4.6.3 Caso de Prueba Historia de Usuario: Gestionar extensiones IAX2.....	66
4.6.4 Caso de Prueba Historia de Usuario: Gestionar buzón de voz de extensiones IAX2.....	68
4.6.5 Caso de Prueba Historia de Usuario: Mostrar datos del estado del sistema.....	70
4.6.6 Caso de Prueba Historia de Usuario: Configurar plan de marcación.....	70
<b>4.7 Conclusiones .....</b>	<b>71</b>
<b>Conclusiones Generales.....</b>	<b>72</b>
<b>Recomendaciones.....</b>	<b>73</b>
<b>Referencias Bibliográficas.....</b>	<b>74</b>
<b>Bibliografía .....</b>	<b>76</b>



# Índice de Contenido

Glosario de Términos.....	78
Anexos .....	80

# Índice de Tablas

## Índice de Tablas

Tabla 3.2.1 Personas relacionadas con el sistema.....	32
Tabla 3.2.2.1 Historia de usuario Gestionar extensiones SIP.....	32
Tabla 3.2.2.2 Historia de usuario Gestionar extensiones IAX2. ....	33
Tabla 3.2.2.3 Historia de usuario Gestionar buzones de voz para extensiones SIP.....	34
Tabla 3.2.2.4 Historia de usuario Modificar clases de servicios para extensiones.....	34
Tabla 3.2.2.5 Historia de usuario Gestionar clases de servicios para los buzones de voz.....	34
Tabla 3.2.2.6 Historia de usuario Plan de numeración.....	35
Tabla 3.2.2.7 Historia de usuario Gestionar buzones de voz para extensiones IAX2. ....	36
Tabla 3.2.2.8 Historia de usuario Estado del sistema.....	36
Tabla 3.2.2.9 Historia de usuario Listar extensiones SIP, IAX2 con los datos más significativos.....	37
Tabla 3.2.2.10 Historia de usuario Diseñar y crear base de datos del sistema.....	37
Tabla 3.3.1 Estimación de esfuerzo por Historia de Usuario.....	38
Tabla 3.3.2 Plan de duración de las Iteraciones.....	40
Tabla 3.3.3 Plan de entrega.....	40
Tabla 4.2.1 Tarjeta CRC de la clase controladora. ....	42
Tabla 4.2.2 Tarjeta CRC de la clase manejadora de datos. ....	43
Tabla 4.2.3 Tarjeta CRC de la clase acceso a datos.....	43
Tabla 4.3.1 Historias de Usuario que se abordan en la Primera Iteración.....	44
Tabla 4.3.1.1 Tarea de ingeniería Diseñar y crear interfaz gráfica para insertar extensiones SIP.....	44
Tabla 4.3.1.2 Tarea de ingeniería Diseñar y crear interfaz gráfica para eliminar extensiones SIP.....	45
Tabla 4.3.1.3 Tarea de ingeniería Diseñar y crear interfaz gráfica para eliminar extensiones SIP.....	45
Tabla 4.3.1.4 Tarea de ingeniería Programar las funcionalidades para insertar, eliminar y modificar extensiones SIP.....	46

## Índice de Tablas

Tabla 4.3.1.4 Tarea de ingeniería Realizar verificaciones de errores.....	46
Tabla 4.3.1.5 Tarea de ingeniería Diseñar y crear interfaz gráfica para modificar las clases de servicios para extensiones SIP.....	47
Tabla 4.3.1.6 Tarea de ingeniería Realizar verificaciones de errores.....	47
Tabla 4.3.1.7 Tarea de ingeniería Diseñar y crear base de datos.....	48
Tabla 4.4.1 Historias de Usuario que se abordan en la Segunda Iteración.....	48
Tabla 4.4.1.1 Tarea de ingeniería “Diseñar y crear interfaz gráfica para insertar el buzón de voz de extensiones SIP”.....	49
Tabla 4.4.1.2 Tarea de ingeniería “Diseñar y crear interfaz gráfica para eliminar el buzón de voz de extensiones SIP”.....	49
Tabla 4.4.1.3 Tarea de ingeniería “Diseñar y crear interfaz gráfica para modificar el buzón de voz de extensiones SIP”.....	50
Tabla 4.4.1.4 Tarea de ingeniería “Programar las funcionalidades para insertar, eliminar y modificar los buzones de voz de extensiones SIP”.....	50
Tabla 4.4.1.5 Tarea de ingeniería “Realizar verificaciones de errores”.....	51
Tabla 4.4.1.6 Tarea de ingeniería “Diseñar y crear interfaz gráfica para modificar clases de servicio para el buzón de voz”.....	51
Tabla 4.4.1.7 Tarea de ingeniería “Programar las funcionalidades para modificar clases de servicio para el buzón de voz”.....	52
Tabla 4.4.1.8 Tarea de ingeniería “Diseñar y crear interfaz gráfica para insertar extensiones IAX2”.....	52
Tabla 4.4.1.9 Tarea de ingeniería “Diseñar y crear interfaz gráfica para eliminar extensiones IAX2”.....	53
Tabla 4.4.1.10 Tarea de ingeniería “Diseñar y crear interfaz gráfica para modificar extensiones IAX2”.....	53
Tabla 4.4.1.11 Tarea de ingeniería “Programar las funcionalidades para insertar, eliminar y modificar extensiones IAX2”.....	54
Tabla 4.4.1.12 Tarea de ingeniería “Realizar verificaciones de errores”.....	54
Tabla 4.5.1 Historias de Usuario que se abordan en la Tercera Iteración.....	55

## Índice de Tablas

Tabla 4.5.1.1 Tarea de ingeniería “Diseñar y crear interfaz gráfica para insertar el buzón de voz de extensiones IAX2”	55
Tabla 4.5.1.2 Tarea de ingeniería “Diseñar y crear interfaz gráfica para eliminar el buzón de voz de extensiones IAX2”	56
Tabla 4.5.1.3 Tarea de ingeniería “Diseñar y crear interfaz gráfica para modificar el buzón de voz de extensiones IAX2”	56
Tabla 4.5.1.4 Tarea de ingeniería “Programar las funcionalidades para insertar, eliminar y modificar los buzones de voz de extensiones IAX2”	57
Tabla 4.5.1.5 Tarea de ingeniería “Realizar verificaciones de errores”	58
Tabla 4.5.1.6 Tarea de ingeniería “Diseñar y crear interfaz gráfica para mostrar datos del estado del sistema”	58
Tabla 4.5.1.7 Tarea de ingeniería “Programar las funcionalidades para mostrar datos del estado del sistema”	59
Tabla 4.5.1.8 Tarea de ingeniería “Diseñar y crear interfaz gráfica para modificar plan de marcación”	59
Tabla 4.5.1.9 Tarea de ingeniería “Programar las funcionalidades para modificar plan de marcación”	60
Tabla 4.5.1.10 Tarea de ingeniería “Diseñar y crear interfaz gráfica para mostrar los datos de las extensiones”	60
Tabla 4.5.1.11 Tarea de ingeniería “Programar las funcionalidades para consultar los datos de las extensiones”	60
Tabla 4.6.1.1 Prueba de Aceptación “Insertar extensión SIP”	61
Tabla 4.6.1.2 Prueba de Aceptación “Insertar extensión SIP”	62
Tabla 4.6.1.3 Prueba de Aceptación “Eliminar extensión SIP”	62
Tabla 4.6.1.4 Prueba de Aceptación “Probar que la extensión SIP a eliminar es inexistente”	63
Tabla 4.6.1.5 Prueba de Aceptación “Modificar extensión SIP”	64
Tabla 4.6.1.6 Prueba de Aceptación “Modificar extensión SIP inexistente”	64
Tabla 4.6.2.1 Prueba de Aceptación “Insertar buzón de voz de extensiones SIP”	65

## Índice de Tablas

Tabla 4.6.2.2 Prueba de Aceptación “Gestionar buzón de voz de extensiones SIP” .....	65
Tabla 4.6.3.1 Prueba de Aceptación “Insertar extensión IAX2” .....	66
Tabla 4.6.3.2 Prueba de Aceptación “Eliminar extensión IAX2” .....	67
Tabla 4.6.3.3 Prueba de Aceptación “Modificar extensión IAX2” .....	67
Tabla 4.6.3.4 Prueba de Aceptación “Insertar extensión IAX2 ya existente” .....	68
Tabla 4.6.4.1 Prueba de Aceptación “Gestionar buzón de voz de extensiones IAX2” .....	69
Tabla 4.6.4.2 Prueba de Aceptación “Gestionar buzón de voz de extensiones IAX2” .....	69
Tabla 4.6.5.1 Prueba de Aceptación “Mostrar datos del estado del sistema” .....	70
Tabla 4.6.6.1 Prueba de Aceptación “Modificar plan de marcación” .....	71

## Introducción

Las ciencias informáticas, las telecomunicaciones y paralelamente la producción de software, han alcanzado en la actualidad un elevado auge e importancia a nivel mundial, lo que permite el surgimiento de los centros de contacto y atención de llamadas, los cuales constituyen un fenómeno que está dando lugar a un cambio radical en el ámbito operacional de las empresas y sus relaciones con los clientes. Estos centros son entidades cuyo objetivo es mantener abiertos los canales de comunicación cliente-empresa. Las más reconocidas e importantes empresas del mundo brindan servicios a través de centros de contacto.

En Cuba también se han notado avances en este sentido, ya que la vinculación de todas sus ramas: económicas, políticas y sociales con el mundo de la informática y las telecomunicaciones son de primordial interés para el Estado Cubano, no sólo por los beneficios que trae desde el punto de vista del desarrollo de sistemas para el uso interno, sino también con el objetivo de introducirse en el mercado a escala mundial aprovechando su perspectiva económica.

Surgida al calor de la Batalla de Ideas la Universidad de las Ciencias Informáticas (UCI) avanza con mucha firmeza en la industria del software y las telecomunicaciones, lo que inicialmente pudo ser una idea de producir programas informáticos se ha convertido en un proceso bastante profundo y cada vez más complejo. Se trata de una universidad productiva en crecimiento, donde se gestiona mucha información y se ofrecen varios servicios al personal interno y adjunto. La UCI cuenta además con una Infraestructura Productora de Software cada vez más solidificada que responde de manera creciente a las necesidades de informatización de la sociedad cubana y que elabora software para la exportación a varios países.

Teniendo en cuenta las perspectivas de crecimiento en la producción y el mercado, es necesario centralizar y administrar toda la información referente a la universidad y al proceso productivo de la misma, por lo que urge la creación de un centro de contacto donde se brinde suficiente información referente a la UCI.

La compra de uno de estos centros conlleva a costosas pérdidas para el país, por lo que se hace necesaria la creación de un producto totalmente nacional e independiente con capacidad autónoma de soporte técnico. Comienza así el desarrollo de la plataforma telefónica PLATEL; una solución nacional orientada a centros de contacto y PMGE (pequeñas medianas y grandes empresas). Dicha plataforma telefónica realiza un conjunto de tareas para brindar servicios telemáticos, garantizar el rendimiento,

supervisión y administración de un centro de contacto. El núcleo de la plataforma PLATEL es Asterisk, un software PBX (centralita telefónica) basada en arquitectura PC (computadora personal) y el concepto de software libre (GPL).

La administración de esta PBX se realiza desde una consola por líneas de comandos y ficheros de configuración, aunque varias empresas se han dado a la tarea de realizar interfaces para humanizar más este proceso. Con respecto al tema de administración vía interfaz se han obtenido algunos logros: Asterisk-Now, Free-PBX, Destar, y otros software realizados. Mediante las aplicaciones web se puede administrar Asterisk de manera remota sin necesidad de utilizar una aplicación cliente; el navegador se encarga de todo el trabajo, pero aún así a la hora de realizar configuraciones complejas, continúan las dependencias con la consola y los ficheros de configuración.

El estudio realizado a las diferentes herramientas de administración de la PBX Asterisk arrojó las siguientes irregularidades:

Los sistemas son inestables, no existe seguridad de que las herramientas existentes sean compatibles con nuevas versiones de Asterisk.

Ninguna de las herramientas existentes logra satisfacer todas las necesidades de la plataforma telefónica PLATEL. Además existe una insuficiente capacidad para realizar configuraciones avanzadas o personalizadas.

Se puede definir entonces el siguiente **problema científico**: ¿Cómo facilitar el proceso de administración de la plataforma telefónica PLATEL?

Este problema se enmarca en el **objeto de estudio**: el proceso de administración de una plataforma telefónica y el **campo de acción** es el proceso de administración de la plataforma telefónica PLATEL.

El **objetivo general** que se persigue con la realización de este trabajo es: desarrollar una aplicación web interactiva para administrar en tiempo real la plataforma telefónica PLATEL.

**Con los siguientes objetivos específicos:**

- Implementar y probar una aplicación web que permita administrar la PBX Asterisk de forma remota.
- Desarrollar un paquete de servicios telefónicos.
- Obtener un producto flexible y altamente configurable.

Para encaminar la investigación en vista a resolver el problema planteado se propone la siguiente **Idea a Defender**: con el desarrollo de la herramienta administrativa para la plataforma telefónica orientada a centros de contacto, se alcanzará un alto nivel de abstracción y transparencia entre el administrador del sistema y la PBX Asterisk.

Para cumplir con el objetivo trazado se proponen las siguientes **tareas**:

- Realizar un estudio del arte de las aplicaciones existentes para la administración de Asterisk, con el objetivo de capturar información relevante que pueda ayudar en el desarrollo de la herramienta administrativa.
- Estudiar profundamente el software Asterisk con el fin de lograr un uso óptimo de las funcionalidades que implementa el mismo, y una correcta integración con la herramienta de administración.
- Estudiar los *frameworks* más utilizados en el desarrollo de aplicaciones web con Python y seleccionar el más indicado para el desarrollo de la herramienta administrativa.
- Elaborar documentación correspondiente al análisis, diseño e implementación de la aplicación web con el fin de crear una fuente ingenieril consultable para el desarrollo de futuras versiones.



## Capítulo I: Fundamentación Teórica

### 1.1 Introducción

En el presente capítulo se realiza un acercamiento a los conceptos asociados con el problema, así como una investigación sobre las herramientas de administración existentes para la PBX Asterisk y las tendencias actuales a nivel mundial. También resume un análisis descriptivo de la arquitectura Cliente-Servidor, las principales herramientas y tecnologías utilizadas para dar cumplimiento al problema planteado.

### 1.2 Sistemas de uso actual para la administración de Asterisk

LA PBX Asterisk es un software potente y altamente configurable, sin embargo, el proceso de administración-configuración se realiza por medio de una consola de comandos personalizada (CLI) y editando los ficheros de configuración. Para hacer más amigable el proceso de supervisión de la centralita de manera remota, se han desarrollado algunas GUIs (interfaz gráfica de usuario), que interactúan con el servidor Asterisk a través de AGI y AMI. La interfaz de Gateway o pasarela de Asterisk (AGI) es un estándar para que aplicaciones externas puedan controlar el plan de discado y la lógica avanzada de su funcionamiento, por otro lado, la interfaz de administración (AMI) es una poderosa capa de servicios programable que permite el control y supervisión remoto de la PBX. Los sistemas con interfaz de usuario actuales han sido concebidos para múltiples propósitos según la esfera en que se desee implantar. A continuación se describen las GUIs más usadas:

#### 1.2.1 FreePBX

FreePBX es una GUI fácil de utilizar para el control y gestión de la PBX Asterisk. Es muy popular por ser un software de telefonía de fuente bajo licencia GPL, con funciones completas de aplicación web. Cuenta con una interfaz de usuario amigable que le permite tener un Asterisk completamente funcional casi de inmediato, evitando tener que editar los ficheros de configuración manualmente.

Algunas de las características de FreePBX son: [1]

- Reporte detallado de llamadas.
- Configuración de los usuarios del sistema.
- Música en espera (a través de archivos MP3, o streaming de la Internet).

# Capítulo I: Fundamentación Teórica

- Módulo para el buzón de voz.
- Configuraciones transparentes al usuario.
- Soporte nativo de SIP (Protocolo de Inicio de Sesiones) e IAX2 (Protocolo de intercambio entre Asterisk).
- Soporta todas las tecnologías de Asterisk incluyendo troncos.
- Buscar en el directorio empresa, sobre la base de nombre o apellido.
- Compartir las tareas administrativas.
- Guardar grabaciones del audio de las llamadas.
- Integración con MySQL, PostgreSQL, SQLite, entre otros.
- Información detallada de llamadas.

## 1.2.2 Destar

Es una GUI de licencia GPL-V2 que realiza la administración de Asterisk a través de AMI. Cuenta con una interfaz web para la interacción con el usuario. Posibilita crear objetos, editarlos y borrarlos, por ejemplo, Teléfonos SIP, IAX2 y teléfonos o líneas RDSI (Red Digital de Servicios Integrados). No hay necesidad de escribir directamente en: sip.conf, extensions.conf, voicemail.conf y otros ficheros de configuración. Estos archivos son manejados por Destar dinámicamente, los carga en memoria, los procesa y vuelve a crearlos automáticamente cada vez que guarde su configuración. [2]

## 1.2.3 Asterisk GUI

Esta suite de software está diseñada para extender la funcionalidad de Asterisk a través de la web independientemente del cliente de aplicaciones, la cual incluye monitorización de llamadas entrantes y salientes a la suite, además escalable a través de múltiples servidores de Asterisk. Avalado por licencias GPL, BSD, Comercial y Software Libre. [3]

### Principales funcionalidades:

- Configuración de Asterisk.
- Sincronización de mensajes de voz.

# Capítulo I: Fundamentación Teórica

- Información detallada de llamadas.
- Sistema de facturación.
- Solución para la Gestión de Centros de Contactos.
- Integración con MySQL, PostgreSQL y SQLite.

## Observaciones:

En nuestro país no existe aplicación alguna para gestionar la centralita Asterisk, ya que solo es utilizada en el ámbito de investigación, mientras que el uso de PBX propietarias de software embebido en Cuba, restringen el uso de herramientas incluidos en los respectivos paquetes de aplicaciones que brinda el proveedor, por tal motivo se convierte en una barrera de desarrollo al no poder adaptar el software a los requerimientos que van surgiendo en el tiempo con la experiencias de cada día.

Las PBX clásicas con software embebido poseen herramientas de administración y mantenimiento que comparten elementos de configuración comunes entre ellas, como una especie de protocolo; el menú de opciones se encuentra organizado de tal forma, que los usuarios pueden tener un PBX completamente operativo siguiendo una secuencia de pasos lógicos, aunque la forma de realizar las tareas varían respecto a las normas del fabricante. La capa de abstracción que brindan estas aplicaciones, permiten gestionar las centralitas sin necesidad de conocer con detalles el funcionamiento interno del software que se encuentra empotrado en ellas. La tendencia de los fabricantes está orientada al usuario, brindando facilidades de manejo de las consolas de mantenimiento y la migración de tecnologías.

## 1.3 Aplicación web

Una aplicación web es un sitio web donde la interacción a través del sitio y la incorporación de datos por parte de los usuarios afectan el estado de la lógica del negocio del mismo; es decir, el sitio web procesa los datos introducidos por los usuarios y les muestra a estos un resultado en dependencia de la información introducida. Si no existe un procesamiento diferente en el servidor en dependencia de la petición del usuario, el sitio web no puede ser considerado como aplicación web.

Las aplicaciones web son hoy en día uno de los sistemas más comunes que podemos encontrar en la red a través de Internet. Ejemplo de estas aplicaciones web son los sistemas como GMAIL y HOTMAIL que brindan la posibilidad de comunicarnos a través de correo electrónico o mensajería instantánea (o ambos)

# Capítulo I: Fundamentación Teórica

con personas en todo el mundo, y existen otros como Amazon, que permiten comprar productos en Internet o Google para buscar información.

Las aplicaciones web constan de tres niveles fundamentales:

- Interfaz de usuario.
- Lógica de negocio.
- Datos.

El nivel de **interfaz de usuario** está compuesto por documentos escritos en lenguaje HTML con los datos que se han obtenido de un servidor web y se muestran a los usuarios mediante un cliente web (generalmente un navegador web).

El nivel de **lógica de negocio** está constituido por módulos que implementan la lógica de negocio. Esta se divide en dos: lógica de la aplicación y lógica del dominio la cual se ejecuta en un servidor de aplicaciones. En la actualidad los servidores web funcionan a la vez como servidores de aplicaciones.

El nivel de **datos** está integrado por la información, que generalmente está manejado por un sistema de gestión de base de datos, el cual es accedido mediante el servidor de aplicaciones.

Las aplicaciones web constan de un cliente (navegador), un servidor (servidor web) y una red de comunicación que posibilita la comunicación entre el cliente y el servidor, lo que es perfectamente aplicable a la solución que se pretende lograr.

## 1.4 Tecnologías y lenguajes a utilizar

### 1.4.1 Modelo Cliente – Servidor

Se puede definir el modelo Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma. Las comunicaciones entre el cliente y el servidor son generalmente bidireccionales, el cliente envía un mensaje solicitando un determinado servicio (petición) a un servidor y este envía uno o varios mensajes con la respuesta (servicio).

La utilización de las diferentes aplicaciones o servicios de Internet se lleva a cabo respondiendo a este modelo, que es una forma de especializar terminales y programas para que las actividades y tareas se

# Capítulo I: Fundamentación Teórica

ejecuten con la mayor eficiencia posible. Mediante esta arquitectura el usuario puede acceder a la información sin tener en cuenta su ubicación física.

Usualmente la mayor carga de trabajo se hace en el proceso llamado servidor y los procesos clientes sólo se ocupan de la interacción con el usuario. De esta forma un servidor da servicio a múltiples clientes de forma concurrente y los cambios realizados en las plataformas de los clientes o de los servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final. (Ver Anexo No.7)

## **Características de la arquitectura Cliente/Servidor: [4]**

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del disco y dispositivos de entrada y salida.
- Las funciones del cliente y el servidor pueden estar en plataformas separadas, o en la misma plataforma.
- La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.
- La estructura inherentemente modular facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.

### **1.4.2 Tecnologías del lado del cliente**

Las tecnologías de lado del cliente son las que están insertadas en la página HTML, siendo interpretadas y ejecutadas por el navegador, encargándose de gestionar la comunicación con el servidor, de solicitar un servicio concreto y de recibir los datos enviados por éste, siendo a su vez la herramienta que presenta al usuario los datos en pantalla y que le ofrece los comandos necesarios para utilizar las presentaciones que ofrece el servidor. Su correcta funcionalidad depende del navegador utilizado por el usuario.

## 1.4.2.1 Lenguaje de Marca de Hipertextos

HTML (Lenguaje de Marca de Hipertextos), es un lenguaje de programación muy sencillo utilizado para crear páginas web. Está compuesto por etiquetas que definen la estructura y el formato del documento que será mostrado al usuario. Permite que se creen enlaces dentro del mismo documento o entre distintas fuentes de información a través de hiperenlaces o hipervínculos e incluso, insertar otros elementos como imágenes y sonidos. Es interpretado por los navegadores para mostrar las páginas web al usuario, brindando la posibilidad de intercambiar información en forma de hipertexto.

Una de las principales ventajas que tiene HTML es la gran variedad de navegadores y exploradores que lo soportan, convirtiéndose en el formato más usado para la transmisión de documentos electrónicos a través de Internet.

## 1.4.2.2 JavaScript

JavaScript es un lenguaje de programación, -tipo script- , interpretado, multiplataforma y parcialmente orientado a objetos desarrollado por Netscape para incrementar las funcionalidades del lenguaje HTML. Permite crear efectos especiales para interactuar con el visitante y funciones básicas que son soportadas por la mayoría de los navegadores que se utilizan diariamente. [5]

### Ventajas de JavaScript:

- Permite elaborar aplicaciones web que simulen características de aplicaciones escritorios.
- Es un código “interpretado” por el cliente.
- Es un código integrado a HTML.
- Reutilización de código de programación.
- El lenguaje de scripting es seguro y fiable.
- El código es visible y puede ser leído por cualquier usuario aún estando protegido con las leyes del copyright.

Uno de los inconvenientes que presenta el lenguaje es que requiere estar activado en los navegadores para que su uso sea posible por estos. Por otro lado no le proporciona al programador control total de la página web. En ocasiones los desarrolladores deben realizar diferentes implementaciones de código

# Capítulo I: Fundamentación Teórica

JavaScript para sus aplicaciones, debido a que el código no es interpretado de igual modo por todos los navegadores.

## 1.4.2.3 JavaScript asíncrono y XML

**AJAX**, acrónimo de JavaScript asíncrono y XML, no es una tecnología en sí, sino una técnica de desarrollo web para crear aplicaciones interactivas. En realidad, se trata en la unión de varias tecnologías independientes. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad, usabilidad en las aplicaciones y mejor respuesta a las acciones del usuario. Es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como HTML o XHTML, JavaScript, DOM, XML, XSLT, y el objeto XMLHttpRequest. [6]

## 1.4.3 Tecnologías del lado del servidor

### 1.4.3.1 Gestor de Base de Datos

Una BD (Base de Datos) es un conjunto de datos interrelacionados entre sí, almacenados con carácter persistente o no, en la computadora. O sea, que puede considerarse una colección de datos variables en el tiempo, conjunto de información almacenada en memoria auxiliar que permite acceso directo a determinados programas que manipulan esos datos. Los sistemas de gestión de bases de datos son un tipo de software que funciona como interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta.

Se denomina SGBD (Sistema de Gestión de Bases de Datos) a la aplicación que permite una utilización y la actualización de la información almacenada en una BD, cuyo objetivo fundamental es brindar al usuario las herramientas que le permitan manipular los datos de manera abstracta, sin que necesite conocer dónde y cómo está almacenada la información en la computadora.

En la actualidad los SGBD facilitan el uso de técnicas para gestionar convenientemente la información a almacenar o recuperar, según el caso, de forma fácil de interpretar y útil para el usuario, con facilidad y fiabilidad; es por ello que se han convertido en el instrumento o soporte básico más utilizado en la gestión

# Capítulo I: Fundamentación Teórica

de los sistemas informáticos. Los SGBD más utilizados en el mundo presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional como son: Oracle, MySQL, Microsoft SQL Server, PostgreSQL, Interbase, entre otros.

## 1.4.3.2 MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Es muy utilizado en aplicaciones web, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores. MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger los datos. Está escrito en C y C++ y probado con multitud de compiladores y dispone de APIs (Interfaz de Programación de Aplicaciones) para muchas plataformas diferentes. Permite manejar varias columnas y registros de longitud fija o variable, además de posibilitar las conexiones entre diferentes ordenadores con distintos sistemas operativos utilizando el lenguaje de consulta estructurado (SQL). Es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. Es una base de datos muy rápida en la lectura, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. [7]

### Ventajas:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de APIs en gran cantidad de lenguajes de los que se pueden citar: C, C++, Java y PHP.
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un alto nivel de seguridad en los datos.

### Desventaja:



# Capítulo I: Fundamentación Teórica

- Actualmente Sum Microsystems tiene en su poder a MySQL, prohibiendo la descarga de futuras versiones desde Cuba.

## 1.4.3 .3 Oracle

Oracle es un sistema de administración de base de datos disponible para un ancho rango de plataformas. Es fácil de desarrollar y administrar, puede manipular todo tipo de administración de datos y ofrece una excepcional disponibilidad, escalabilidad, fiabilidad y seguridad. La seguridad de las aplicaciones y bases de datos de Oracle permiten la administración completa de los usuarios, el monitoreo de potenciales ataques y la asignación de privilegios de acceso a través de toda la empresa.

### **Ventajas:** [8]

- Permite implementar diseños "activos", con triggers (Disparadores de fila) y procedimientos almacenados, con una integridad referencial declarativa bastante potente.
- Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.
- El software del servidor puede ejecutarse en multitud de sistemas operativos.

### **Desventaja:**

- El precio de las patentes y licencias del equipamiento que requiere Oracle, con respecto a otros sistemas gestores de bases de datos y la necesidad de ajustes constituyen unos de sus principales problemas.

## 1.4.3 .4 PostgreSQL

PostgreSQL es un servidor de base de datos objeto relacional libre. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo, dicha comunidad es denominada el PGDG (Grupo de desarrollo global de PostgreSQL). Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido mas tarde en otros sistemas de gestión comerciales. Es un sistema objeto-relacional, ya que incluye características propias de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones,

# Capítulo I: Fundamentación Teórica

disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos. [9]

## Ventajas:

- Puede ser utilizado en los principales sistemas operativos: Linux, Unix, Mac OS, Beos, Windows.
- Soporte para los lenguajes más populares: PHP, C, C++, Perl, Python, entre otros.
- Soporte de protocolo de comunicación encriptado por SSL.
- Extensiones para alta disponibilidad, nuevos tipos de índices, datos espaciales y minería de datos.
- Implementación del estándar SQL92/SQL99.
- Incorpora una estructura de datos array.
- Incorpora funciones de diversa índole: manejo de fechas, unicote, reglas y soporte para SSL (Protocolo de Capa de Conexión Segura).
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Puede ser utilizado, modificado y distribuido por cualquiera gratuitamente, para cualquier propósito ya sea con fines privados, comerciales o académicos.

### 1.4.3.5 Gestor de Base de Datos seleccionado (PostgreSQL)

Luego del estudio de los gestores de base de datos anteriores se decide utilizar para el desarrollo del sistema PostgreSQL por las siguientes razones:

- Puede ser utilizado en los principales sistemas operativos: Linux, Unix, Mac OS, Beos, Windows.

# Capítulo I: Fundamentación Teórica

- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos.
- Requiere pocos recursos de hardware y la simplificación del proceso de administración de licencias de software, que no es necesario cuando se usa software libre.

## 1.4.3 .6 Servidor web Apache

El servidor web es un programa que se aloja en un servidor que escucha peticiones HTTP las cuales son atendidas y procesadas por este programa. Atendiendo a las peticiones hechas, gestiona las páginas web o ejecuta códigos en el servidor para dar respuesta, ya sea información o un mensaje detallado de error.

El servidor Apache se desarrolla dentro del proyecto HTTP Server de la Apache Software Foundation. Apache presenta entre otras características informes de errores altamente configurables, índice de directorios, un directorio de alias; negociación de contenidos, gestión de recursos multihilo, integración de imágenes del lado del servidor; reescritura de las URL; comprobación de la ortografía de las URL y manuales online, bases de datos de autenticación y negociado de contenido, aunque carece de una interfaz gráfica que ayude en su configuración. Tiene amplia aceptación en la red por ser altamente configurable de diseño modular lo que aumenta sus capacidades, es el más usado, empleado en el 48% de los sitios web en el mundo. Es un servidor de distribución libre, código abierto (Open source) y multiplataforma. [10]

## 1.4.3 .7 Servidor web seleccionado (Apache)

Se decidió utilizar el servidor Apache para publicar la aplicación por todas las bondades que ofrece este software, por ser extensible, popular y brindar las siguientes ventajas:

- Ayuda en la mejora del posicionamiento.
- Es un software libre.
- Multiplataforma.
- Open Source, modular y presenta mensajes de error altamente configurables.

## 1.4.4 Lenguajes de Programación

La AGI (del inglés: Asterisk Gateway Interface), proporciona una interfaz estándar externa por la que los programas pueden controlar el plan de discado de Asterisk. Por lo general, los scripts de la AGI son utilizados para desarrollar lógica avanzada, comunicarse con bases de datos relacionales (PostgreSQL, MySQL entre otras), y acceder a otros recursos externos que de otra manera resultaría difícil o imposible. La PBX Asterisk brinda una interfaz estándar para los scripts de la AGI, los cuales pueden ser escritos en casi todos los lenguajes de programación modernos, pero los lenguajes más comúnmente utilizados y que se destacan en la programación AGI son: Perl, Java, PHP, Python.

### 1.4.4.1 Java

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

Con la programación en Java, se pueden realizar distintos aplicativos, como son applets, que son aplicaciones especiales, que se ejecutan dentro de un navegador al ser cargada una página HTML en un servidor web. Por lo general los applets son programas pequeños y de propósitos específicos.

La independencia de plataforma es una de las razones por las que Java resulta interesante a los desarrolladores de aplicaciones para Internet, ya que muchas personas deben tener acceso con ordenadores distintos, pero no se queda ahí, Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador, como móviles y agendas.

### 1.4.4.2 Python

**Python** es un lenguaje de programación creado en el año 1990 por Guido Van Rossum, es el sucesor del lenguaje de programación ABC. Los usuarios lo consideran como un lenguaje más limpio para programar; tiene una sintaxis muy visual, gracias a una notación con márgenes de obligado cumplimiento. Es un lenguaje fácil de aprender y potente, multiplataforma, de propósito general. La elegante sintaxis de Python, su gestión de tipos dinámica y la naturaleza interpretada del lenguaje lo hace ideal para guiones (scripts) y desarrollo rápido de aplicaciones en muchas áreas y en la mayoría de las plataformas; desde aplicaciones Windows a servidores de red o incluso, páginas web. Dispone de eficaces estructuras de datos de alto

# Capítulo I: Fundamentación Teórica

nivel y óptimas soluciones de programación. Es Interactivo; cuenta con un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución. Su código no necesita ser compilado, por lo que se dice que el código es interpretado. Es un lenguaje de programación multiparadigma, lo cual obliga a que los programadores adopten un estilo de programación particular:

- Programación orientada a objetos.
- Programación estructurada.
- Programación funcional.
- Programación orientada a aspectos.
- Es un lenguaje muy popular y utilizado por varias razones:
- Contiene gran cantidad de librerías, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- Su código es sencillo lo que permite crear los programas rápidamente. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- Se puede desarrollar en diversas plataformas, como Unix, Windows, OS/2, Mac y otros.
- Es gratuito, incluso para propósitos empresariales.

## 1.4.4.3 Perl

**Perl** (Lenguaje Práctico para la Extracción e Informe) es un lenguaje que toma características del lenguaje de programación C, shell (sh), AWK, sed y Lisp. Una diferencia fundamental de Perl con respecto a los otros lenguajes es que no limita el tamaño de los datos con los que trabaja, el límite lo pone la memoria que en ese momento se encuentre disponible. Permite resolver los problemas fácilmente y hacer una pequeña aplicación web en pocos minutos. Posee una gran flexibilidad y posee varios módulos ya implementados. El principal inconveniente del lenguaje se basa en que en una sola línea de código se agrupan gran cantidad de funcionalidades. Si los script son bien escritos su código se asemeja a los de

PHP.

## 1.4.4.4 PHP

PHP (Hypertext Pre-processor) es un lenguaje de script interpretado en el lado del servidor que se utiliza para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor.

Es un lenguaje que para poder ser ejecutado no es necesario compilarlo y para que pueda funcionar tiene que estar instalado en la computadora el servidor web Apache con librerías del propio lenguaje.

### Ventajas:

- Es un lenguaje libre y multiplataforma.
- Puede establecer conexión con la mayoría de los gestores de base de datos, tales como: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Tiene gran cantidad de funciones ejemplificadas y una amplia documentación.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

### Desventajas:

- El grosor del trabajo es realizado en del servidor sin delegar ninguna funcionalidad al cliente, por lo que es más ineficiente a medida que las solicitudes aumenten de número.
- Como se mezclan sentencias HTML y PHP la legibilidad del código es afectada.
- La programación orientada a objetos es aún muy deficiente para grandes aplicaciones.
- Dificulta la modularización y la organización por capas de aplicación.
- Para su funcionamiento es necesario tener instalado un servidor web.

## 1.4.4.5 Lenguaje de Programación seleccionado (Python)

De los lenguajes tratados anteriormente Python es el más indicado para el desarrollo de la aplicación web, cuenta con gran cantidad de funciones y librerías, posee integración con los frameworks más utilizados para la web, es un lenguaje sencillo de propósito general y rápido de programar, versátil, ágil y multiplataforma. Su código bien organizado proporciona un mayor entendimiento para un posterior desarrollo y mantenimiento de la aplicación por cualquier programador.

# Capítulo I: Fundamentación Teórica

## 1.5 Framework

El concepto *framework* viene aparejado con el desarrollo de software, es una estructura de soporte robusta y bien definida en la cual otro proyecto de software puede ser organizado y desarrollado. Los *frameworks* se han convertido en la piedra angular de la moderna ingeniería del software ya que están compuestos por componentes personalizables e intercambiables para el desarrollo de una aplicación, en otras palabras, es una aplicación genérica incompleta y/o configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un *framework* son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Un *framework* web podemos definirlo como un conjunto de clases y componentes que forman un diseño reutilizable al cual se le pueden incluir módulos para facilitar y agilizar el desarrollo de aplicaciones web como Django (Python), Ruby on Rails (Ruby) y CodeIgniter (PHP). Por lo general pueden ser orientados a la interfaz de usuario, a aplicaciones de publicación de documentos y/o a la combinación de estos elementos.

La mayoría de los *frameworks* web incorporan en su arquitectura MVC (modelo-vista-controlador), separando los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres capas distintas, donde la vista es la página HTML y el código que provee de datos dinámicos a la página; el modelo es el SGBD y la lógica del negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.

### 1.5.1 TurboGears

TurboGears constituye un *framework* diseñado para desarrollar aplicaciones Web 2.0 con Python, está compuesto por una serie de componentes que le permiten al desarrollador hacer aplicaciones web de forma rápida. Todos estos componentes están empaquetados por separado, y pueden ser descargados y utilizado sin TurboGears. Posee amplia documentación, es multiplataforma, cuenta con una numerosa comunidad de desarrolladores y usuarios.

Los principales componentes son: [11]

**MochiKit:** Una biblioteca JavaScript que incluye componentes tales como *logging*, arrastrar (*drag*), soltar (*drop*), y otros efectos visuales que son comunes en muchas bibliotecas. Permite añadir comportamiento

# Capítulo I: Fundamentación Teórica

asíncrono (AJAX) a una aplicación web, carga y manipulación de conjuntos de datos JSON y contiene un conjunto de funciones para facilitar la creación dinámica de componentes, llamada Mochikid.DOM.

## Sus principales características son:

- Gestión de tareas asíncronas.
- Funciones de manipulación de objetos, *arrays* y comparaciones.
- Capa de acceso al DOM.
- Iteraciones y enumeradores en JavaScript.
- Capacidades de log.
- Efectos visuales y funciones de color.
- Funciones Python.

**Kid:** Una plantilla de lenguaje que le permite incluir Python incrustado en una página XHTML. Puedes incrustar código Python, explícitamente con una etiqueta, o utilizar un conjunto de atributos que incluyen estructuras condicionales y bucles para definir el contenido dinámico. Las plantillas son constituidas por XML, lenguaje similar a Zope compilado a Python byte-code. Para el procesado utiliza un *parser* basado en ElementTree, soporta herencia de plantillas y emparejado XSLT.

**CherryPy:** Un *framework* orientado a objetos que le permite desarrollar aplicaciones web en Python, incluye mecanismos para hacer simples operaciones centradas en la web tales como mapeo de URL solicitadas a métodos Python usando decoradores, control del servidor y solicitudes de filtrado.

**SQLObject:** Un simple mapeador de objetos relacional (ORM), que incluye un mecanismo de mapeo de objetos de Python a una base de datos relacional; las tablas son clases, las filas son instancias (objetos) y los campos son atributos. También incluye Python basado en mecanismos de abstracción de SQL que permite a los desarrolladores interactuar con bases de datos desde un *script* Python o *shell*. Puede ser usado con: MySQL, PostgreSQL, SQLite, Firebird, Sybase, Oracle (en desarrollo). Puede generar BD desde Python o código Python desde BD.



# Capítulo I: Fundamentación Teórica

**SQLAlchemy:** Un moderno Mapeador Objeto Relacional, que proporciona un potente y flexible sistema de gestión de la conexión entre la memoria y los objetos de Python, almacenamiento de datos relacional que proporciona la persistencia de los objetos.

TurboGears también incluye una amplia variedad de otras funciones y servicios, incluidos los basados en la web y líneas de comandos, servicios de ayuda, administrador de tareas del proyecto, un servidor web incorporado, un marco de prueba y una herramienta basada en web para la administración del modelo de objetos, entre otros. Incorpora la arquitectura MVC de la misma manera que la mayoría de los demás *frameworks* actuales de aplicaciones web (como Ruby on Rails).

## 1.5.2 Dojo Toolkit

Dojo es una caja de herramientas *open source* de JavaScript que te ayuda a construir aplicaciones de internet. Contiene APIs y *widgets* que facilitan el desarrollo de aplicaciones que utilicen AJAX. Es potente, portable y ligero. Tiene herramientas de testeo para construir interfaces dinámicas. Permite crear *widgets* rápidamente, transiciones animadas, peticiones AJAX con la mejor potencia y facilidad. Soporta *bookmarking* (marcador de páginas web), encapsula las molestias “*crossbrowser*” (Script de navegador cruzado) y es compatible con los navegadores más antiguos. [12]

## 1.6 Fundamentación de la Metodología

### Ingeniería de software

La Ingeniería de Software es una tecnología que indica “*COMO*” construir técnicamente un software: económico, fiable y que funcione eficientemente. Dentro de esta se encuentra la metodología que se va a utilizar para el desarrollo del proyecto. [13]

### Razones de la Ingeniería de software

Estudiar ¿cuáles son las actividades que organizadas, haciendo uso racional de los recursos y apoyándose en técnicas y herramientas, logran la mayor eficiencia al construir un software?

### Metodología

La rama de la metodología, dentro de la ingeniería de software, se encarga de elaborar estrategias de desarrollo de software que promuevan prácticas adaptativas en vez de predictivas; centradas en las

# Capítulo I: Fundamentación Teórica

personas o los equipos, orientadas hacia la funcionalidad y la entrega de comunicación intensiva y que requieren implicación directa del cliente.[14]

Para desarrollar un software es necesario seleccionar la metodología adecuada, teniendo en cuenta parámetros como los requerimientos, calidad, límite de tiempo, entre otros.

Existen varias metodologías de desarrollo de software entre las que se encuentran las pesadas o tradicionales y las ágiles o ligeras.

Las tradicionales se enfocan más en la definición de roles, incluyendo el modelado, siendo utilizadas en proyectos a largo plazo, con un numeroso equipo de desarrollo.

Las metodologías ágiles están orientadas especialmente a proyectos pequeños y dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo manteniendo una alta calidad. [15]

## 1.6.1 Extreme Programming (XP)

La Programación Extrema es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corta duración, equipos pequeños y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. [16]

Las cuatro variables de esta metodología son:

- Coste: Máquinas, especialistas y oficinas.
- Tiempo: Total de Entregas.
- Calidad: Externa e Interna.
- Alcance: Intervención del cliente.

XP surgió como respuesta y posible solución a los problemas derivados del cambio en los requerimientos. Se plantea como una metodología a emplear en proyectos de riesgo para aumentar la productividad.

### Características de XP

La metodología del mismo se basa en:

# Capítulo I: Fundamentación Teórica

**Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos.

**Refabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

**Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

## ¿Qué es lo que propone XP?

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantivo del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

## Derechos del Cliente [17]

- Decidir qué se implementa.
- Saber el estado real y el progreso del proyecto.
- Añadir, cambiar o quitar requerimientos en cualquier momento.
- Obtener lo máximo de cada semana de trabajo.
- Obtener un sistema funcionando cada 3 o 4 meses.

## Derechos del Desarrollador [18]

- Decidir cómo se implementan los procesos.
- Crear el sistema con la mejor calidad posible.
- Pedir al cliente en cualquier momento aclaraciones de los requerimientos.
- Estimar el esfuerzo para implementar el sistema.

# Capítulo I: Fundamentación Teórica

- Cambiar los requerimientos sobre la base a nuevos descubrimientos.

## Lo fundamental en este tipo de metodología es:

- La comunicación entre los usuarios y los desarrolladores.
- La simplicidad al desarrollar y codificar los módulos del sistema.
- La retroalimentación concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

### 1.6.2 Scrum

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos donde sus requisitos pueden ser variantes. Sus características principales las constituyen el desarrollo de software que se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días, que se muestra al cliente y las reuniones a lo largo proyecto, destacando la reunión diaria del equipo de desarrollo durante 15 minutos para la coordinación e integración. [19]

Es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software, se basa en una serie de aspectos que lo hacen factible y fácil de utilizar. Entre ellos tenemos:

- Prioriza el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión.
- Está diseñado especialmente para adaptarse a los cambios en los requerimientos.
- Los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos muy cortos y regulares (de esta forma se puede adaptar en tiempo real el producto que se está construyendo a las necesidades del cliente).
- Se busca entregar software que realmente resuelva las necesidades, aumentando la satisfacción del cliente.
- Procura la construcción del software con calidad.
- Define las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y remueve cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo.

# Capítulo I: Fundamentación Teórica

- Busca equipos efectivos y productivos (dentro de las posibilidades).
- Se basa en los principios de inspección continua, adaptación, autogestión e innovación.

## 1.6.3 Proceso Unificado de Desarrollo de Software (RUP)

RUP es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifica a este proceso de desarrollo. [20]

En RUP se han agrupado las actividades en grupos lógicos, definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

Divide en 4 fases el desarrollo del software:

**Inicio:** El objetivo en esta etapa es determinar la visión del proyecto.

**Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.

**Construcción:** En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.

**Transmisión:** El objetivo es llegar a obtener el *re/lease* del proyecto.

RUP es más adaptable para proyectos de largo plazo, con un equipo de desarrollo muy grande. En esta metodología se genera gran volumen de información, lo que posibilita un mayor entendimiento entre el equipo de desarrollo. Tiene abundantes roles y genera un gran número de artefactos siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. Brinda un proceso integrado que utiliza el estándar de notación UML para permitir desarrollar un proceso de forma iterativa e incremental a partir de la identificación e implementación de los casos de uso.

Se basa en tres características esenciales que la distinguen de otras metodologías: Dirigido por Casos de Uso, Centrado en la Arquitectura, Iterativo e Incremental. Esto es lo que hace único al Proceso Unificado.

**Dirigido por Casos de Uso:** Para construir un sistema con éxito debemos conocer lo que sus futuros usuarios necesitan y desean. Un caso de uso es un servicio que el actor requiere del sistema y este le

# Capítulo I: Fundamentación Teórica

proporciona un resultado. Representan los requerimientos funcionales de la aplicación. Los casos de uso guían el diseño, la implementación y prueba, es decir, guían el proceso de desarrollo.

**Centrado en la Arquitectura:** La arquitectura en un sistema de software se describe mediante diferentes vistas del sistema en construcción. Esta se refleja en los casos de uso pues cada uno describe una funcionalidad y una forma. Es a su vez la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

**Iterativo e Incremental:** Es mucho más sencillo y práctico dividir el trabajo en pequeños módulos, los que no son más que iteraciones que resultan en un incremento. Una iteración es una secuencia de actividades con un plan establecido y criterios de evaluación, cuyo resultado es una versión del software.

Los beneficios de las iteraciones son los siguientes:

- Reduce el coste del riesgo al coste de un solo incremento.
- Menos riesgo de no sacar el producto al mercado en fecha.
- Acelera el ritmo de desarrollo.
- Las necesidades del usuario y correspondientes requisitos no pueden definirse completamente al principio. Se requieren iteraciones sucesivas.

## 1.6.4 Metodología seleccionada (XP y Scrum)

Luego del estudio de las metodologías anteriores se determinó que para una mejor organización, eficiencia y eficacia en el desarrollo de la aplicación es necesario utilizar XP y Scrum principalmente por la estructura del proyecto y las grandes ventajas que ofrecen ambas metodologías ágiles.

## 1.7 Conclusiones

Teniendo en cuenta el estudio realizado anteriormente sobre las tendencias que existen actualmente en cuanto a los paradigmas de tecnologías y herramientas que más se utilizan en el campo de la informática, se toma como decisión el desarrollo de una aplicación web utilizando la metodología XP combinada con Scrum, y beneficiados por las ventajas que brinda Python como lenguaje de programación, usando el TurboGears como *framework* de desarrollo y aprovechando las potencialidades que brinda el gestor de BD PostgreSQL.

# Capítulo 2: Características del Sistema

## Capítulo II Características del Sistema

### 2.1 Introducción.

En el presente capítulo ya se está en condiciones de expresar la especificación del objeto de automatización, la información que se maneja y el análisis de las características del sistema de administración a desarrollar.

### 2.2 Objeto de Automatización.

La administración de Asterisk se realiza escribiendo comandos en la consola incluida (CLI), y se hace necesario editar los ficheros para hacer cambios en la configuraciones. Estas tareas requieren acceso físico al ordenador donde esté hospedada la PBX y generalmente se encuentra en servidores dedicados, sin mouse ni teclado, con el hardware y software necesario para un rendimiento óptimo; por lo que se hace necesario el desarrollo de una aplicación web con interfaz de usuario que permita administrar los procesos y la configuración de la plataforma telefónica PLATEL.

### 2.3 Propuesta del sistema a desarrollar.

Para darle solución al problema planteado se propone desarrollar una herramienta administrativa que permita gestionar las extensiones SIP e IAX2, pues son los protocolos VoIP más utilizados en las comunicaciones con Asterisk y encontrarse inundado el mercado, de tecnologías que implementan dichos protocolos. La herramienta incorpora un módulo de CDS (clases de servicios) que permite la asignación genérica de las facilidades que brindará PLATEL a las extensiones. Dispone de una capa de acceso a datos genérica que posibilitará la integración con los SGBD más utilizados en el mundo.

Para la supervisión del sistema, la herramienta muestra en un panel información relacionada con el estado físico del sistema y las extensiones, tiempo de servicio y un informe detallado en tiempo real de las llamadas activas y los canales que intervienen en cada una. Dispone de un plan de marcación para gestionar los números de acceso a las facilidades de la plataforma telefónica.

Los servicios son la razón de existir de las centralitas telefónicas, la plataforma PLATEL cuenta con un paquete de servicios adicionales que son gestionados por la herramienta administrativa, entre los que se destacan:

## Capítulo 2: Características del Sistema

- Desvío de llamadas en no contesta: Cuando se activa este servicio, las llamadas dirigidas a una extensión se desvían hacia otra que el usuario especifique, sólo si la llamada no es atendida en el tiempo de timbrado establecido.
- Desvío de llamadas en ocupado: Cuando se activa este servicio, las llamadas dirigidas a una extensión se desvían hacia otra que el usuario especifique, sólo si el estado de la extensión original es ocupado.
- Desvío de llamadas en incondicional: Cuando se activa este servicio, las llamadas dirigidas a una extensión siempre son desviadas hacia otra que el usuario especifique.
- Chequeo de desvíos de llamadas: Es una función complementaria de las anteriores, permite verificar si la extensión tiene activado algún tipo de desvío de llamadas.
- Fecha y Hora: Facilidad que brinda información al usuario referente al día, mes, año y hora en que se accede al servicio.
- Búsqueda en directorio: Función que realiza una búsqueda por nombres en el directorio telefónico de la PBX.
- Conferencia de voz: Servicio que permite crear una sala de conferencias donde varios abonados pueden conversar al mismo tiempo.
- Supervisión de llamadas: Funcionalidad a través de la cual el supervisor puede escuchar o grabar una conversación determinada.
- Buzón de voz: Permite grabar y almacenar mensajes dejados por las personas que llaman y el abonado no atiende la llamada, pueden ser escuchados cuando el usuario acceda al servicio o reciba una notificación por correo electrónico con el mensaje de voz adjunto.

### 2.4 Arquitectura del sistema propuesto:

Los patrones de diseño son soluciones estándares a problemas comunes que suceden en el desarrollo de programas informáticos, la aplicación de uno u otro depende de cuan flexible sea el lenguaje de programación en que se desea aplicar, aunque el código resultante no identifica al patrón empleado.

El patrón Singleton (Instancia única) plantea que solo se puede realizar una instancia de una clase,



## Capítulo 2: Características del Sistema

restringiendo la creación de otros objetos de dicha clase, pues se obtendrá la misma instancia, mientras que el patrón Facade (Fachada) provee una interfaz abstracta unificada que facilita el intercambio con otras clases o interfaces más complejas, permitiendo el desarrollo de sistemas flexibles. Ambos patrones se evidencian en el proceso de gestión de Asterisk, la clase responsable de la conexión con AMI, procesa los datos e implementa los métodos que proveen la total transparencia a la clase controladora que recibe la información filtrada previamente. El patrón Decorator o Envoltorio consiste en insertar comportamiento adicional a las funciones dinámicamente, es aplicado en el control de acceso a las *urls* (Localizador de Recurso Uniforme), antes de ejecutar la función asociada a una *url* restringida los decoradores verifican que el usuario en la sesión actual tenga los permisos para acceder a dicha función.

Para lograr la disposición operativa del sistema no solo se requiere de una arquitectura robusta; los cuatro elementos fundamentales que lo componen deben estar correctamente configurados y conectados entre sí: cliente, servidor de base de datos, servidor web y PBX Asterisk. (Ver Anexo No. 8)

La aplicación puede ser desplegada tanto en un ambiente centralizado como distribuido. Al encontrarse el sistema de forma centralizado; los tres pilares fundamentales de la plataforma se encuentran hos pedados en el mismo ordenador, lo cual no sólo constituye una limitante desde el punto de vista de recursos y hardware, sino que de ser accedido uno de los elementos por un atacante, quedaría comprometido todo el sistema. La opción recomendada es la distribuida, ya que además de reducir los riesgos de la arquitectura centralizada permite la incorporación de más PBX Asterisk al sistema.

### 2.5 Interfaces de usuario.

**Extensiones:** Interfaz en la cual los usuarios con los privilegios de diseñador agregan las extensiones (SIP, IAX2) con todos los datos requeridos por el sistema y modifican los datos de las extensiones ya creadas en caso de que sea necesario.(Ver Anexo No.1 y 2)

**Configuraciones generales:** Interfaz en la cual los usuarios con permisos de administración realizarán las configuraciones de los parámetros necesarios para obtener un uso óptimo del sistema; teniendo en cuenta requerimientos de hardware, software, y de la red donde se despliegue la plataforma. Se define la forma de acceso a datos. (Ver Anexo No.3)

**Estado del sistema:** Interfaz en la cual se muestran las estadísticas del sistema: canales SIP, IAX2 activos, horas de funcionamiento del sistema y estado de las extensiones SIP, IAX2. (Ver Anexo No.4)

## Capítulo 2: Características del Sistema

**Plan de Numeración:** Interfaz en la cual el administrador asigna el número de acceso a los servicios que brinda la plataforma telefónica. (Ver Anexo No.5)

**Clases de servicios:** Interfaz en la cual se muestran las facilidades (servicios) que ofrece la plataforma a sus abonados, el administrador escoge las opciones que desea, generando una regla que será aplicada a todas las extensiones que pertenezcan a la clase de servicio configurada. (Ver Anexo No.6)

### 2.6 Requisitos no funcionales del sistema.

#### **Interfaz:**

La aplicación es manejable y fácil de usar, la información que se muestra en la interfaz de usuario está bien detallada y organizada, por lo que no se requieren conocimientos avanzados de informática para trabajar con ella. Es una aplicación dinámica y que implementa comunicaciones asincrónicas con el servidor para evitar que se refresque toda la página en cada petición.

#### **Usabilidad:**

A los usuarios del sistema se les dará un adiestramiento básico en el uso de la aplicación en cada uno de sus niveles, mientras que los administradores deberán obtener un conocimiento avanzado para ser capaces de dar respuesta a cualquier incidente que ocurra con el sistema.

#### **Rendimiento:**

Para lograr un rendimiento óptimo de la aplicación se debe configurar después de haber realizado un estudio detallado del hardware y la red que se va a utilizar. Es recomendable desplegarse en un ambiente distribuido, aunque un despliegue centralizado no interviene en la capacidad del sistema. La eficiencia del producto estará determinada en gran medida por el aprovechamiento de los recursos que disponga.

#### **Portabilidad:**

La herramienta es compatible con los navegadores web y los sistemas operativos más utilizados en todo el mundo, desde cualquier versión de Windows NT en adelante hasta cualquier distribución de Linux. Tiene la capacidad de integración con los SGBD más usados y brinda la posibilidad de alojarse en cualquier servidor web, aunque se recomienda PostgreSQL y Apache respectivamente.

#### **Seguridad:**

## Capítulo 2: Características del Sistema

Para prevenir el acceso no autorizado y la disponibilidad de la información, se implementan secciones de manera que cada trabajador solo puede acceder a las interfaces autorizadas por su rol dentro de la plataforma. El uso de protocolos seguros (https) garantiza la integridad de los datos que se manejan entre el cliente y el servidor.

### **Software:**

Para la interacción con la herramienta el usuario sólo debe tener instalado un navegador web y cualquier sistema operativo. Del lado del servidor se requiere un servidor web (Apache u otro), una PBX Asterisk (1.4 o superior) y un servidor de base de datos (PostgreSQL u otro).

### **Hardware:**

EL servidor de base de datos debe tener 256 MB de RAM y 20 GB de disco duro como mínimo, todos los nodos involucrados en la funcionalidad de la aplicación deben estar conectados a una red que requiere 100 Mbps o más de velocidad.

### **Restricciones de diseño e implementación:**

La aplicación debe estar lista para el despliegue en el menor tiempo posible, con la calidad que el sistema requiere. Para el desarrollo de la aplicación se usarán los *framework* TurboGears y DojoToolkit.

### **2.7 Conclusiones**

En este capítulo se realizó una descripción de las características de la herramienta de administración. Se definieron los requisitos funcionales y no funcionales del sistema los cuales sirvieron de base para el comienzo del desarrollo del sistema. Además se dio a conocer la arquitectura del sistema propuesto, así como las interfaces con las que contará la herramienta administrativa.

# Capítulo III: Desarrollo ágil de la herramienta administrativa para la plataforma Platel.

## Capítulo III: Desarrollo ágil de la herramienta administrativa para la plataforma Platel.

### 3.1 Introducción

El contenido del presente capítulo está enmarcado en las fases iniciales que propone la metodología de desarrollo XP; Exploración, Planificación e Iteraciones. Se abordará la dinámica ágil para el desarrollo del sistema propuesto siguiendo las pautas de la metodología seleccionada y quedarán confeccionados los artefactos (historias de usuarios) que registran los requisitos funcionales y no funcionales a implementar, se estima el tiempo duración y envergadura del módulo.

### 3.2 Fase de Exploración.

Es la fase de inicio que propone XP. Durante esta fase el cliente identifica y describe a grandes rasgos los requisitos funcionales y no funcionales, esto queda registrado en las Historias de Usuarios. También se realiza un acercamiento a las herramientas, tecnologías y prácticas que se utilizarán en el desarrollo del software. La duración de esta fase puede oscilar entre unas pocas semanas o pocos meses dependiendo de la envergadura del producto y el nivel de experiencia adquirido por los desarrolladores sobre las tecnologías usadas.

#### 3.2.1 Personas relacionadas con el sistema

Se define como persona relacionada al sistema toda aquella que obtiene un resultado del valor de uno o varios procesos que se ejecutan en el mismo. Además de aquellas que se encuentran involucradas en dichos procesos, pues participan en ellos pero no obtienen ningún resultado de valor.

<b>Personas Relacionadas con el Sistema</b>	<b>Justificación</b>
Administrador	Es la persona con privilegios para la gestión del sistema. Es el encargado de administrar y supervisar los usuarios y procesos dentro de la plataforma telefónica.

# Capítulo III: Desarrollo ágil de la herramienta administrativa para la plataforma Platel.

Tabla 3.2.1 Personas relacionadas con el sistema.

## 3.2.2 Historias de Usuario.

Las historias de usuario son los artefactos que utiliza XP para especificar los requisitos funcionales y no funcionales del sistema. Son plantillas donde el cliente plasma una descripción breve de las acciones que el software debe realizar, puede apoyarse en el criterio de los desarrolladores. El contenido de las historias de usuario debe ser sencillo, dinámico, sintético y flexible, de manera que sean lo suficientemente comprensibles para los programadores.

Historia de Usuario	
<b>Número:</b> 1	<b>Nombre de la Historia de Usuario:</b> Gestionar extensiones SIP.
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 2
<b>Riegos en Desarrollo:</b> Baja	<b>Puntos Reales:</b> 2
<b>Descripción:</b> El administrador podrá crear, modificar, eliminar y consultar las extensiones SIP. Los datos asociados son: el número de la extensión, nombre de usuario, contraseña, asignarle una clase de servicios, definir los privilegios de llamada para la extensión.	

Tabla 3.2.2.1 Historia de usuario Gestionar extensiones SIP.

## Capítulo III: Desarrollo ágil de la herramienta administrativa para la plataforma Platel.

Historia de Usuario	
<b>Número:</b> 2	<b>Nombre de la Historia de Usuario:</b> Gestionar extensiones IAX2.
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 2
<b>Riegos en Desarrollo:</b> Media	<b>Puntos Reales:</b> 2
<p><b>Descripción:</b> El administrador podrá crear, modificar, eliminar y consultar las extensiones IAX2. Los datos asociados son: el número de la extensión, nombre de usuario, contraseña, asignarle una clase de servicios, definir los privilegios de llamadas para la extensión.</p>	

**Tabla 3.2.2.2 Historia de usuario Gestionar extensiones IAX2.**

Historia de Usuario	
<b>Número:</b> 3	<b>Nombre de la Historia de Usuario:</b> Gestionar buzones de voz para extensiones SIP.
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riegos en Desarrollo:</b> Media	<b>Puntos Reales:</b> 1
<p><b>Descripción:</b> El administrador escoge la extensión a la que se quiere asignar un buzón de voz, le define la clase de servicios que contenga las facilidades que se deseen para el buzón, en caso que se desee notificación por correo electrónico, define las direcciones de correo electrónico en el caso que se desee notificación por esta vía.</p>	

## Capítulo III: Desarrollo ágil de la herramienta administrativa para la plataforma Platel.

**Tabla 3.2.2.3 Historia de usuario Gestionar buzones de voz para extensiones SIP.**

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre de la Historia de Usuario:</b> Modificar clases de servicios para extensiones.
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 1
<b>Riegos en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Se agrupan todas las facilidades (servicios) que se deseen para el grupo de extensiones en una clase de servicios.	

**Tabla 3.2.2.4 Historia de usuario Modificar clases de servicios para extensiones.**

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre de la Historia de Usuario:</b> Modificar clases de servicios para los buzones de voz.
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 1
<b>Riegos en Desarrollo:</b> Media	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Agrupar en una clase de servicios, todos las facilidades que se deseen para cada buzón de voz.	

**Tabla 3.2.2.5 Historia de usuario Gestionar clases de servicios para los buzones de voz.**

## Capítulo III: Desarrollo ágil de la herramienta administrativa para la plataforma Platel.

Historia de Usuario	
<b>Número:</b> 6	<b>Nombre de la Historia de Usuario:</b> Configurar plan de numeración
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riegos en Desarrollo:</b> Media	<b>Puntos Reales:</b> 1
<b>Descripción:</b> El administrador define los números de acceso a los servicios y a los abonados de la plataforma telefónica.	

**Tabla 3.2.2.6 Historia de usuario Plan de numeración.**

Historia de Usuario	
<b>Número:</b> 7	<b>Nombre de la Historia de Usuario:</b> Gestionar buzones de voz para extensiones IAX2.
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 1
<b>Riegos en Desarrollo:</b> Media	<b>Puntos Reales:</b> 1
<b>Descripción:</b> El administrador escoge la extensión a la que se quiere asignar un buzón de voz, le define la clase de servicios que contenga las facilidades que se deseen para el buzón, en caso que se desee notificación por correo electrónico, define las direcciones de correo primaria y alternativa en el caso de que lo requiera.	



## Capítulo III: Desarrollo ágil de la herramienta administrativa para la plataforma Platel.

**Tabla 3.2.2.7 Historia de usuario Gestionar buzones de voz para extensiones IAX2.**

Historia de Usuario	
<b>Número:</b> 8	<b>Nombre de la Historia de Usuario:</b> Estado del sistema
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 1
<b>Riegos en Desarrollo:</b> Baja	<b>Puntos Reales:</b> 1
<b>Descripción:</b> El administrador supervisa el estado de la plataforma telefónica, total de llamadas y canales activos y el tiempo de servicio del sistema.	

**Tabla 3.2.2.8 Historia de usuario Estado del sistema.**

Historia de Usuario	
<b>Número:</b> 9	<b>Nombre de la Historia de Usuario:</b> Listar extensiones SIP, IAX2 con los datos más significativos.
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 1
<b>Riegos en Desarrollo:</b> Baja	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Se realiza al iniciar el sistema y consiste en gestionar y crear un listado de los datos de las extensiones que se encuentren en el sistema, este listado se mantendrá en constante actualización.	

## Capítulo III: Desarrollo ágil de la herramienta administrativa para la plataforma Platel.

**Tabla 3.2.2.9 Historia de usuario Listar extensiones SIP, IAX2 con los datos más significativos.**

Historia de Usuario	
<b>Número:</b> 10	<b>Nombre de la Historia de Usuario:</b> Diseñar y crear base de datos del sistema.
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 2
<b>Riesgos en Desarrollo:</b> Baja	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Se debe diseñar y crear la base de datos del sistema en la cual se almacenará la información de las extensiones, los servicios y los buzones de voz.	
<b>Observaciones:</b> Se usará PostgreSQL como gestor de base de datos.	

**Tabla 3.2.2.10 Historia de usuario Diseñar y crear base de datos del sistema.**

### 3.3 Planificación.

En la fase de planificación se establece la prioridad a cada historia de usuario y se estima el esfuerzo de realización. Las estimaciones de esfuerzo asociado a la implementación de las historias de usuarios la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. El tiempo estimado para el desarrollo de cada historia de usuario no debe exceder los tres puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. El tiempo estimado para la implementación de cada historia de usuario no debe exceder las tres semanas, donde los programadores trabajan el tiempo planificado sin interrupciones.

El tiempo de duración de esta fase es de unos pocos días, que bastan para tomar acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses.

## Capítulo III: Desarrollo ágil de la herramienta administrativa para la plataforma Platel.

### 3.3.1 Estimación de esfuerzo por Historia de Usuario.

Realizar una correcta estimación del esfuerzo necesario para llevar a cabo cada tarea dentro de un proyecto, contribuye a la entrega del producto en tiempo y forma así como satisfacción del cliente. A continuación la estimación de las historias de usuarios identificadas.

Historia de Usuario	Puntos Estimados
Gestionar extensiones SIP.	2
Modificar clases de servicios para extensiones.	1
Gestionar buzón de voz de extensiones SIP.	1
Modificar clases de servicio para el buzón de voz.	1
Gestionar extensiones IAX2.	2
Gestionar buzón de voz de extensiones IAX2.	1
Mostrar datos del estado del sistema.	1
Configurar plan de marcación.	1
Diseñar y crear base de datos del sistema.	2
Listar extensiones SIP, IAX2 con los datos más significativos.	1

**Tabla 3.3.1 Estimación de esfuerzo por Historia de Usuario.**

### 3.3.2 Iteraciones.

La fase planificación contiene varias iteraciones del sistema antes de ser entregado. Se define el Plan de Entrega que está compuesto por iteraciones de no más de tres semanas. Es necesario agrupar las historias de usuario de manera tal, que permita la obtención de una arquitectura del sistema al finalizar la primera iteración, que permita la integración de los otros elementos del sistema. Esto se logra priorizando

## Capítulo III: Desarrollo ágil de la herramienta administrativa para la plataforma Platel.

las historias de usuarios críticas, aunque, esto no siempre es posible ya que el cliente es quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.

**Primera Iteración:** En esta iteración se implementan las historias de usuarios 1 y 10 definidas por el cliente con alta prioridad y la 4 con prioridad media. Estas recogen las principales funcionalidades del sistema. Al concluir esta iteración el sistema cuenta con los servicios solicitados por el cliente.

**Segunda Iteración:** En esta iteración se implementan las historias de usuarios 2, 3 y 5, de las cuales las dos primeras fueron definidas con prioridad alta y la última con prioridad media. Al culminar la misma el producto se va tornando más robusto, y se realizan las pruebas correspondientes para confirmar la aceptación por parte del cliente.

**Tercera Iteración:** En esta iteración se implementan las historias de usuarios 7, 8 y 9 con prioridad media y la 6 con prioridad alta para el cliente. Al final de esta iteración se obtendrá la primera versión completamente operativa del producto y se someterá a pruebas todo el sistema para definir si cumple con todos los requerimientos del cliente.

### 3.3.3 Plan de duración de las Iteraciones.

Iteraciones	Orden de las Historias Usuario a implementar	Duración de la estimación
Iteración 1	Gestionar extensiones SIP.	2 semanas
Iteración 1	Modificar clases de servicios para extensiones.	1 semana
Iteración 1	Diseñar y crear base de datos del sistema.	1 semana
Iteración 2	Gestionar buzón de voz de extensiones SIP.	1 semana
Iteración 2	Modificar clases de servicio para el buzón de voz.	1 semana
Iteración 2	Gestionar extensiones IAX2.	2 semanas
Iteración 3	Gestionar buzón de voz de extensiones IAX2.	1 semana

## Capítulo III: Desarrollo ágil de la herramienta administrativa para la plataforma Platel.

Iteración 3	Mostrar datos del estado del sistema	1 semana
Iteración 3	Configurar plan de marcación	1 semana
Iteración 3	Listar extensiones SIP, IAX2 con los datos más significativos.	1 semana

**Tabla 3.3.2 Plan de duración de las iteraciones.**

### 3.3.4 Plan de entrega.

El plan de entrega comprende la fecha estimada de cierre de cada iteración, donde se realizan revisiones y pruebas a las tareas desarrolladas en la iteración que culmina. En la siguiente tabla se refleja las fechas de propuestas para la entrega de la herramienta administrativa.

Herramienta	Final 1ra iteración 4ta semana de febrero	Final 2da iteración 4ta semana de marzo	Final 3ra iteración 4ta semana de abril
Herramienta administrativa.	0.1	0.2	1.0

**Tabla 3.3.3 Plan de entrega.**

### 3.4 Conclusiones

A lo largo del capítulo se desarrolló una primera etapa de la metodología seleccionada para el desarrollo del sistema: la exploración, planificación de entrega e iteraciones. Se realizó el levantamiento de las historias de usuarios por parte del cliente, y se realizó la estimación de tiempo y esfuerzo para la entrega de cada iteración. Finalmente se cuenta con 10 historias de usuarios que serán desarrolladas en 3 iteraciones.

# Capítulo IV: Implementación y Pruebas.

## Capítulo IV: Implementación y Pruebas.

### 4.1 Introducción

En este capítulo se abordarán las iteraciones realizadas en el transcurso de la fase de implementación del módulo detallando las tareas que fueron generadas por cada historia de usuario. Se expone además los casos de pruebas de aceptación realizados al producto en cada una de las iteraciones para lograr una calidad máxima del mismo.

### 4.2 Diseño del Sistema

Para el diseño de las aplicaciones, la metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración). No obstante el uso de estos diagramas puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante.

A continuación se brindan las tarjetas CRC.

#### 4.2.1 Tarjeta CRC de la clase controladora.

AdminControllers	
SaveCDSSIP	
DeleteCDSIP	
SaveCDSVM	
DeleteCDSVM	
SaveIaxBD	
DestroyIAX	
ShowIaxList	
SaveSIP	
DestroySIP	
ShowSipList	
UpdateDialPlan	AmiCommands
ShowSipStatus	AmiCommands

## Capítulo IV: Implementación y Pruebas.

ShowIax2Status	AmiCommands
RealTimeChannels	AmiCommands
CoreChannels	AmiCommands
SaveGeneralSettings	AmiCommands
SipShowPeers	AmiCommands
SipShowChannels	AmiCommands
Iax2ShowChannels	AmiCommands

**Tabla 4.2.1 Tarjeta CRC de la clase controladora.**

### 4.2.2 Tarjeta CRC de la clase manejadora de datos.

AmiCommands	
GeneralSettings	AsteriskAMI
RestartNow	AsteriskAMI
ClearCDSVM	AsteriskAMI
UpdateCDS	AsteriskAMI
DialPlanReload	AsteriskAMI
SipReload	AsteriskAMI
Iax2Reload	AsteriskAMI
Iax2ShowPeers	AsteriskAMI
SipShowPeers	AsteriskAMI
Iax2CoreShowChannels	AsteriskAMI
SipShowCoreChannels	AsteriskAMI
CoreShowChannels	AsteriskAMI

## Capítulo IV: Implementación y Pruebas.

SipShowChannels	AsteriskAMI
IAX2ShowChannels	AsteriskAMI
UPTime	AsteriskAMI

**Tabla 4.2.2 Tarjeta CRC de la clase manejadora de datos.**

### 4.2.3 Tarjeta CRC de la clase acceso a datos.

AsteriskAMI	
connectTo	
sendCommand	
readResult	
Command	

**Tabla 4.2.3 Tarjeta CRC de la clase acceso a datos.**

### 4.3 Primera Iteración

En esta iteración se contempla la realización de las historias de usuarios que establecerán la arquitectura base del sistema con fin de obtener un producto con las funcionalidades primarias para ser mostrado al cliente.

Historia de Usuario	Estimación	Real
Gestionar extensiones SIP.	2	2
Modificar clases de servicios para extensiones SIP.	1	1
Diseñar y crear base de datos del sistema.	2	1



## Capítulo IV: Implementación y Pruebas.

Tabla 4.3.1 Historias de Usuario que se abordan en la Primera Iteración.

### 4.3.1 Tareas de Ingeniería que se abordan en la iteración.

Tarea de Ingeniería	
<b>Número Tarea:</b> 2.1	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para insertar extensiones SIP.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 02/02/09	<b>Fecha Fin:</b> 04/02/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	
<b>Descripción:</b> Se diseñará y creará una ventana que permita al usuario insertar los datos de las extensiones en la base de datos de la plataforma.	

Tabla 4.3.1.1 Tarea de ingeniería Diseñar y crear interfaz gráfica para insertar extensiones SIP.

Tarea de Ingeniería	
<b>Número Tarea:</b> 2.2	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para eliminar extensiones SIP.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 04/02/09	<b>Fecha Fin:</b> 05/02/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	
<b>Descripción:</b> Se diseñará y creará una interfaz que permita al usuario seleccionar la extensión que se	

## Capítulo IV: Implementación y Pruebas.

desea eliminar en la base de datos.

**Tabla 4.3.1.2 Tarea de ingeniería Diseñar y crear interfaz gráfica para eliminar extensiones SIP.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 2.3	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para modificar extensiones SIP.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 05/02/09	<b>Fecha Fin:</b> 06/02/09
<b>Programador Responsable:</b> Reynier Santiago González León	
<b>Descripción:</b> Se diseñará y creará una ventana que permita al usuario seleccionar una extensión determinada para modificar los datos que desee.	

**Tabla 4.3.1.3 Tarea de ingeniería Diseñar y crear interfaz gráfica para eliminar extensiones SIP.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 2.4	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Programar las funcionalidades para insertar, eliminar y modificar extensiones SIP.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.7
<b>Fecha Inicio:</b> 06/02/09	<b>Fecha Fin:</b> 13/02/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	
<b>Descripción:</b> Se realiza la programación todas las funcionalidades para realizar las operaciones con las extensiones SIP.	

## Capítulo IV: Implementación y Pruebas.

Tabla 4.3.1.4 Tarea de ingeniería Programar las funcionalidades para insertar, eliminar y modificar extensiones SIP.

Tarea de Ingeniería	
<b>Número Tarea:</b> 2.5	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Realizar verificaciones de errores.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 13/02/09	<b>Fecha Fin:</b> 14/02/09
<b>Programador Responsable:</b> Reynier Santiago González León	
<b>Descripción:</b> Se verifica que el usuario haya introducido todos los datos correctamente para realizar las operaciones con las extensiones SIP.	

Tabla 4.3.1.4 Tarea de ingeniería Realizar verificaciones de errores.

Tarea de Ingeniería	
<b>Número Tarea:</b> 3.1	<b>Número Historia de Usuario:</b> 4
<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para modificar las clases de servicios para extensiones SIP.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 16/02/09	<b>Fecha Fin:</b> 20/02/09
<b>Programador Responsable:</b> Reynier Santiago González León	
<b>Descripción:</b> Se diseñará y creará una ventana que permita al usuario seleccionar una clase de servicio determinada la cual se busca en la base de datos de la plataforma y para modificar el estado de los	

## Capítulo IV: Implementación y Pruebas.

servicios.

**Tabla 4.3.1.5 Tarea de ingeniería Diseñar y crear interfaz gráfica para modificar las clases de servicios para extensiones SIP.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 3.2	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Realizar validación de errores.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 20/02/09	<b>Fecha Fin:</b> 21/02/09
<b>Programador Responsable:</b> Reynier Santiago González León	
<b>Descripción:</b> Se verifica que el usuario haya seleccionado una clase de servicio correctamente y le haya asignado al menos un servicio.	

**Tabla 4.3.1.6 Tarea de ingeniería Realizar verificaciones de errores.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 4.1	<b>Número Historia de Usuario:</b> 10
<b>Nombre Tarea:</b> Diseñar y crear base de datos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 23/02/09	<b>Fecha Fin:</b> 25/02/09
<b>Programador Responsable:</b> Reynier Santiago González León	
<b>Descripción:</b> Diseñar y crear el modelo de datos que se usará para almacenar la información de las	

## Capítulo IV: Implementación y Pruebas.

extensiones, los servicios y los buzones de voz.

**Tabla 4.3.1.7 Tarea de ingeniería Diseñar y crear base de datos.**

### 4.4 Segunda Iteración

En esta segunda iteración se completará la gestión del buzón de voz de extensiones SIP, la gestión de extensiones IAX2 y la modificación de las clases de servicio para el buzón de voz.

Historia de Usuario	Estimación	Real
Gestionar buzón de voz de extensiones SIP.	1	1
Modificar clases de servicio para el buzón de voz.	1	1
Gestionar extensiones IAX2.	2	2

**Tabla 4.4.1 Historias de Usuario que se abordan en la Segunda Iteración.**

#### 4.4.1 Tareas de Ingeniería que se abordan en la iteración.

Tarea de Ingeniería	
Número Tarea: 5.1	Número Historia de Usuario: 3
Nombre Tarea: Diseñar y crear interfaz gráfica para insertar el buzón de voz de extensiones SIP.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Fecha Inicio: 26/02/09	Fecha Fin: 26/02/09
Programador Responsable: Orelvis Barrera Zumaquero	
Descripción: Se diseñará y creará una interfaz que permita al usuario seleccionar la extensión a la cual se le asignará un correo de voz y se inserta la base de datos del sistema.	

## Capítulo IV: Implementación y Pruebas.

Tabla 4.4.1.1 Tarea de ingeniería “Diseñar y crear interfaz gráfica para insertar el buzón de voz de extensiones SIP”.

Tarea de Ingeniería	
<b>Número Tarea:</b> 5.2	<b>Número Historia de Usuario:</b> 3
<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para eliminar el buzón de voz de extensiones SIP.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 26/02/09	<b>Fecha Fin:</b> 26/02/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	
<b>Descripción:</b> Se diseñará y creará una interfaz que permita al usuario seleccionar una extensión que tenga buzón de voz para ser eliminado de la base de datos.	

Tabla 4.4.1.2 Tarea de ingeniería “Diseñar y crear interfaz gráfica para eliminar el buzón de voz de extensiones SIP”.

Tarea de Ingeniería	
<b>Número Tarea:</b> 5.3	<b>Número Historia de Usuario:</b> 3
<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para modificar el buzón de voz de extensiones SIP.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 27/02/09	<b>Fecha Fin:</b> 27/02/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	
<b>Descripción:</b> Se diseñará y creará una interfaz que permita al usuario seleccionar la extensión a la cual se le quiere modificar el correo de voz, se le modifican los datos pertinentes y se guardan en la base de	

## Capítulo IV: Implementación y Pruebas.

datos.

**Tabla 4.4.1.3 Tarea de ingeniería “Diseñar y crear interfaz gráfica para modificar el buzón de voz de extensiones SIP”.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 5.4	<b>Número Historia de Usuario:</b> 3
<b>Nombre Tarea:</b> Programar las funcionalidades para insertar, eliminar y modificar los buzones de voz de extensiones SIP.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 27/02/09	<b>Fecha Fin:</b> 3/03/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	
<b>Descripción:</b> Se programan todas las funcionalidades para realizar las operaciones con los correos de VOZ.	

**Tabla 4.4.1.4 Tarea de ingeniería “Programar las funcionalidades para insertar, eliminar y modificar los buzones de voz de extensiones SIP”.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 5.5	<b>Número Historia de Usuario:</b> 3
<b>Nombre Tarea:</b> Realizar verificaciones de errores.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 3/03/09	<b>Fecha Fin:</b> 4/03/09
<b>Programador Responsable:</b> Reynier Santiago González León	

## Capítulo IV: Implementación y Pruebas.

**Descripción:** Se verifica que el usuario haya insertado todos los datos del correo de voz correctamente.

**Tabla 4.4.1.5 Tarea de ingeniería “Realizar verificaciones de errores”.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 6.1	<b>Número Historia de Usuario:</b> 5
<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para modificar clases de servicio para el buzón de voz.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 5/03/09	<b>Fecha Fin:</b> 6/03/09
<b>Programador Responsable:</b> Reynier Santiago González León	
<b>Descripción:</b> Se diseñará y creará una interfaz que permita al usuario seleccionar la clase de servicio para el buzón de voz a la cual se le quiere modificar los servicios pertinentes.	

**Tabla 4.4.1.6 Tarea de ingeniería “Diseñar y crear interfaz gráfica para modificar clases de servicio para el buzón de voz”.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 6.2	<b>Número Historia de Usuario:</b> 5
<b>Nombre Tarea:</b> Programar las funcionalidades para modificar clases de servicio para el buzón de voz.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 6/03/09	<b>Fecha Fin:</b> 11/03/09
<b>Programador Responsable:</b> Reynier Santiago González León	
<b>Descripción:</b> Se programan todas las funcionalidades para realizar las operaciones con las clases de	



## Capítulo IV: Implementación y Pruebas.

servicio para el buzón de voz a la cual se le quiere modificar los servicios pertinentes.

**Tabla 4.4.1.7 Tarea de ingeniería “Programar las funcionalidades para modificar clases de servicio para el buzón de voz”.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 7.1	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para insertar extensiones IAX2.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 12/03/09	<b>Fecha Fin:</b> 14/03/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	
<b>Descripción:</b> Se diseñará y creará una interfaz que le permita al usuario insertar las extensiones IAX2 en la base de datos de la plataforma.	

**Tabla 4.4.1.8 Tarea de ingeniería “Diseñar y crear interfaz gráfica para insertar extensiones IAX2”.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 7.2	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para eliminar extensiones IAX2.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 16/03/09	<b>Fecha Fin:</b> 17/03/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	
<b>Descripción:</b> Se diseñará y creará una interfaz que permita al usuario seleccionar la extensión que	

## Capítulo IV: Implementación y Pruebas.

desea eliminar en la base de datos.

**Tabla 4.4.1.9 Tarea de ingeniería “Diseñar y crear interfaz gráfica para eliminar extensiones IAX2”.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 7.3	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para modificar extensiones IAX2.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 17/03/09	<b>Fecha Fin:</b> 18/03/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	
<b>Descripción:</b> Se diseñará y creará una ventana que permita al usuario seleccionar una extensión determinada la cual se busca en la base de datos de la plataforma y se le modifican los datos pertinentes	

**Tabla 4.4.1.10 Tarea de ingeniería “Diseñar y crear interfaz gráfica para modificar extensiones IAX2”.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 7.4	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Programar las funcionalidades para insertar, eliminar y modificar extensiones IAX2.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.7
<b>Fecha Inicio:</b> 18/03/09	<b>Fecha Fin:</b> 25/03/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	

## Capítulo IV: Implementación y Pruebas.

**Descripción:** Se programan todas las funcionalidades para realizar las operaciones con las extensiones IAX2.

**Tabla 4.4.1.11 Tarea de ingeniería “Programar las funcionalidades para insertar, eliminar y modificar extensiones IAX2”.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 7.5	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Realizar verificaciones de errores.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 25/03/09	<b>Fecha Fin:</b> 26/03/09
<b>Programador Responsable:</b> Reynier Santiago González León	
<b>Descripción:</b> Se verifica que el usuario haya introducido todos los datos correctamente para realizar las operaciones con las extensiones IAX2.	

**Tabla 4.4.1.12 Tarea de ingeniería “Realizar verificaciones de errores”.**

### 4.5 Tercera Iteración

Esta última iteración concluirá el sistema con la implementación de las funcionalidades relacionadas con la gestión del buzón de voz de extensiones IAX2, mostrar los datos del sistema, configurar plan de marcación y la visualización de las extensiones SIP, IAX2 con los datos más significativos.

Historia de Usuario	Estimación	Real
Gestionar buzón de voz de extensiones IAX2	1	1

## Capítulo IV: Implementación y Pruebas.

Mostrar datos del estado del sistema.	1	1
Configurar plan de marcación.	1	1
Listar extensiones SIP, IAX2 con los datos más significativos.	1	1

**Tabla 4.5.1 Historias de Usuario que se abordan en la Tercera Iteración.**

### 4.5.1 Tareas de Ingeniería que se abordan en la iteración.

Tarea de Ingeniería	
<b>Número Tarea:</b> 8.1	<b>Número Historia de Usuario:</b> 7
<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para insertar el buzón de voz de extensiones IAX2.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 27/03/09	<b>Fecha Fin:</b> 27/03/09
<b>Programador Responsable:</b> Reynier Santiago González León	
<b>Descripción:</b> Se diseñará y creará una interfaz que permita al usuario seleccionar la extensión a la cual se le asignará un correo de voz y se inserta en la base de datos del sistema.	

**Tabla 4.5.1.1 Tarea de ingeniería “Diseñar y crear interfaz gráfica para insertar el buzón de voz de extensiones IAX2”.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 8.2	<b>Número Historia de Usuario:</b> 7

## Capítulo IV: Implementación y Pruebas.

<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para eliminar el buzón de voz de extensiones IAX2	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 27/0/09	<b>Fecha Fin:</b> 27/03/09
<b>Programador Responsable:</b> Reynier Santiago González León	
<b>Descripción:</b> Se diseñará y creará una interfaz que permita al usuario seleccionar una extensión a la cual se le elimina el correo de voz de la base de datos.	

Tabla 4.5.1.2 Tarea de ingeniería “Diseñar y crear interfaz gráfica para eliminar el buzón de voz de extensiones IAX2”.

Tarea de Ingeniería	
<b>Número Tarea:</b> 8.3	<b>Número Historia de Usuario:</b> 7
<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para modificar el buzón de voz de extensiones IAX2.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 27/03/09	<b>Fecha Fin:</b> 27/03/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	
<b>Descripción:</b> Se diseñará y creará una interfaz que permita al usuario seleccionar la extensión a la cual se le quiere modificar el correo de voz, se le modifican los datos pertinentes y se guardan en la base de datos.	

Tabla 4.5.1.3 Tarea de ingeniería “Diseñar y crear interfaz gráfica para modificar el buzón de voz de extensiones IAX2”.

## Capítulo IV: Implementación y Pruebas.

Tarea de Ingeniería	
<b>Número Tarea:</b> 8.4	<b>Número Historia de Usuario:</b> 7
<b>Nombre Tarea:</b> Programar las funcionalidades para insertar, eliminar y modificar los buzones de voz de extensiones IAX2.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 28/03/09	<b>Fecha Fin:</b> 1/04/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	
<b>Descripción:</b> Se programan todas las funcionalidades para realizar las operaciones con los correos de VOZ.	

**Tabla 4.5.1.4 Tarea de ingeniería “Programar las funcionalidades para insertar, eliminar y modificar los buzones de voz de extensiones IAX2”.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 8.5	<b>Número Historia de Usuario:</b> 7
<b>Nombre Tarea:</b> Realizar verificaciones de errores.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 1/04/09	<b>Fecha Fin:</b> 2/04/09
<b>Programador Responsable:</b> Reynier Santiago González León	
<b>Descripción:</b> Se verifica que el usuario haya insertado todos los datos del correo de voz correctamente.	

## Capítulo IV: Implementación y Pruebas.

Tabla 4.5.1.5 Tarea de ingeniería “Realizar verificaciones de errores”.

Tarea de Ingeniería	
<b>Número Tarea:</b> 9.1	<b>Número Historia de Usuario:</b> 8
<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para mostrar datos del estado del sistema.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 3/04/09	<b>Fecha Fin:</b> 6/04/09
<b>Programador Responsable:</b> Reynier Santiago González León	
<b>Descripción:</b> Se de debe realizar una interfaz gráfica que muestre los datos del sistema.	

Tabla 4.5.1.6 Tarea de ingeniería “Diseñar y crear interfaz gráfica para mostrar datos del estado del sistema”.

Tarea de Ingeniería	
<b>Número Tarea:</b> 9.2	<b>Número Historia de Usuario:</b> 8
<b>Nombre Tarea:</b> Programar las funcionalidades para mostrar datos del estado del sistema.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 6/04/09	<b>Fecha Fin:</b> 10/04/09
<b>Programador Responsable:</b> Reynier Santiago González León	
<b>Descripción:</b> Se programan todas las funcionalidades para obtener los datos del sistema.	

## Capítulo IV: Implementación y Pruebas.

Tabla 4.5.1.7 Tarea de ingeniería “Programar las funcionalidades para mostrar datos del estado del sistema”.

Tarea de Ingeniería	
<b>Número Tarea:</b> 10.1	<b>Número Historia de Usuario:</b> 6
<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para modificar plan de marcación.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 11/04/09	<b>Fecha Fin:</b> 13/04/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	
<b>Descripción:</b> Se diseñará y creará una ventana que permita que el usuario modificar los números de acceso a los servicios.	

Tabla 4.5.1.8 Tarea de ingeniería “Diseñar y crear interfaz gráfica para modificar plan de marcación”.

Tarea de Ingeniería	
<b>Número Tarea:</b> 10.2	<b>Número Historia de Usuario:</b> 6
<b>Nombre Tarea:</b> Programar las funcionalidades para modificar plan de marcación.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 13/04/09	<b>Fecha Fin:</b> 18/04/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	
<b>Descripción:</b> Se programan todas las funcionalidades para modificar los números de acceso a los servicios.	



## Capítulo IV: Implementación y Pruebas.

Tabla 4.5.1.9 Tarea de ingeniería “Programar las funcionalidades para modificar plan de marcación”.

Tarea de Ingeniería	
<b>Número Tarea:</b> 11	<b>Número Historia de Usuario:</b> 9
<b>Nombre Tarea:</b> Diseñar y crear interfaz gráfica para mostrar los datos de las extensiones.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 18/04/09	<b>Fecha Fin:</b> 20/04/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	
<b>Descripción:</b> Se diseñará y creará una ventana que permita visualizar al usuario los datos de las extensiones.	

Tabla 4.5.1.10 Tarea de ingeniería “Diseñar y crear interfaz gráfica para mostrar los datos de las extensiones”.

Tarea de Ingeniería	
<b>Número Tarea:</b> 11.1	<b>Número Historia de Usuario:</b> 9
<b>Nombre Tarea:</b> Programar las funcionalidades para consultar los datos de las extensiones.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 20/04/09	<b>Fecha Fin:</b> 24/04/09
<b>Programador Responsable:</b> Orelvis Barrera Zumaquero	
<b>Descripción:</b> Se programan todas las funcionalidades para consultar los datos de las extensiones.	

Tabla 4.5.1.11 Tarea de ingeniería “Programar las funcionalidades para consultar los datos de las

## Capítulo IV: Implementación y Pruebas.

extensiones”.

### 4.6 Casos de Pruebas

Las pruebas de aceptación tienen como propósito demostrar al cliente el cumplimiento de los requisitos del software, describiendo un escenario (secuencia de pasos) de ejecución o uso del sistema desde la perspectiva del cliente. Pueden estar asociadas a requisitos funcionales o no funcionales y obligan a definir requisitos que sean verificables y valorar adecuadamente el esfuerzo asociado a la incorporación de un requisito.

#### 4.6.1 Caso de Prueba Historia de Usuario: Gestionar extensiones SIP.

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba: 1</b>	<b>Número Historia de Usuario:</b> Gestionar extensiones SIP.
<b>Nombre de la Prueba:</b> Insertar extensión SIP.	
<b>Descripción de la Prueba:</b> Probar que se inserta la extensión SIP de forma correcta.	
<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente y conectada a la base de datos.	
<b>Entrada / Pasos de ejecución:</b> Se insertan los datos correspondientes sin dejar campos vacíos.	
<b>Resultado Esperado:</b> Se inserta de forma satisfactoria la extensión en la base de datos.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Tabla 4.6.1.1 Prueba de Aceptación “Insertar extensión SIP”.

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba: 2</b>	<b>Número Historia de Usuario:</b> Gestionar extensiones SIP.
<b>Nombre de la Prueba:</b> Insertar extensión SIP existente.	

## Capítulo IV: Implementación y Pruebas.

<b>Descripción de la Prueba:</b> Probar que la extensión SIP a insertar ya existe.
<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente y conectada a la base de datos.
<b>Entrada / Pasos de ejecución:</b> Insertar extensión SIP que se encuentre en la lista de extensiones.
<b>Resultado Esperado:</b> Se obtiene mensaje de error informándole al usuario que la extensión ya existe.
<b>Evaluación de la Prueba:</b> Satisfactoria

**Tabla 4.6.1.2 Prueba de Aceptación “Insertar extensión SIP”.**

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 3	<b>Número Historia de Usuario:</b> Gestionar extensiones SIP.
<b>Nombre de la Prueba:</b> Eliminar extensión SIP.	
<b>Descripción de la Prueba:</b> Probar que se elimina la extensión SIP de forma correcta.	
<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente y conectada a la base de datos.	
<b>Entrada / Pasos de ejecución:</b> Se selecciona la extensión SIP a eliminar.	
<b>Resultado Esperado:</b> Se elimina correctamente la extensión de la base de datos.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

**Tabla 4.6.1.3 Prueba de Aceptación “Eliminar extensión SIP”.**

## Capítulo IV: Implementación y Pruebas.

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba: 4</b>	<b>Número Historia de Usuario:</b> Gestionar extensiones SIP.
<b>Nombre de la Prueba:</b> Eliminar extensión SIP inexistente.	
<b>Descripción de la Prueba:</b> Probar que la extensión SIP a eliminar es inexistente.	
<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente y conectada a la base de datos.	
<b>Entrada / Pasos de ejecución:</b> Se introduce una extensión errónea.	
<b>Resultado Esperado:</b> Se obtiene mensaje de error informándole al usuario que la extensión no existe.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

**Tabla 4.6.1.4 Prueba de Aceptación “Probar que la extensión SIP a eliminar es inexistente”.**

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba: 5</b>	<b>Número Historia de Usuario:</b> Gestionar extensiones SIP.
<b>Nombre de la Prueba:</b> Modificar extensión SIP.	
<b>Descripción de la Prueba:</b> Probar que se modifica la extensión SIP de forma correcta.	
<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente y existir la información de la extensión en la base de datos.	
<b>Entrada / Pasos de ejecución:</b> Se selecciona la extensión SIP a modificar, luego se modifican los datos que se deseen.	
<b>Resultado Esperado:</b> Se modifican de forma correcta los datos en la base de datos.	

## Capítulo IV: Implementación y Pruebas.

**Evaluación de la Prueba:** Satisfactoria

**Tabla 4.6.1.5 Prueba de Aceptación “Modificar extensión SIP”.**

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 6	<b>Número Historia de Usuario:</b> Gestionar extensiones SIP.
<b>Nombre de la Prueba:</b> Modificar extensión SIP inexistente.	
<b>Descripción de la Prueba:</b> Probar que la extensión SIP a modificar es inexistente.	
<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente y conectada a la base de datos.	
<b>Entrada / Pasos de ejecución:</b> Se introduce una extensión errónea.	
<b>Resultado Esperado:</b> Se obtiene mensaje de error informándole al usuario que la extensión que desea modificar no existe.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

**Tabla 4.6.1.6 Prueba de Aceptación “Modificar extensión SIP inexistente”.**

### 4.6.2 Caso de Prueba Historia de Usuario: Gestionar buzón de voz de extensiones SIP.

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 1	<b>Número Historia de Usuario:</b> Gestionar buzón de voz de extensiones SIP.
<b>Nombre de la Prueba:</b> Insertar buzón de voz de extensiones SIP.	
<b>Descripción de la Prueba:</b> Probar que se inserta el buzón de voz de extensiones SIP de forma correcta.	

## Capítulo IV: Implementación y Pruebas.

<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente y conectada a la base de datos.
<b>Entrada / Pasos de ejecución:</b> Se selecciona la extensión SIP a la cual se le desea añadir un buzón de voz.
<b>Resultado Esperado:</b> Se inserta de forma correcta en la base de datos el buzón de voz.
<b>Evaluación de la Prueba:</b> Satisfactoria

**Tabla 4.6.2.1 Prueba de Aceptación “Insertar buzón de voz de extensiones SIP”.**

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba: 2</b>	<b>Número Historia de Usuario:</b> Gestionar buzón de voz de extensiones SIP.
<b>Nombre de la Prueba:</b> Eliminar buzón de voz de extensiones SIP.	
<b>Descripción de la Prueba:</b> Probar que se elimina el buzón de voz de extensiones SIP de forma correcta de la base de datos.	
<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente y conectada a la base de datos.	
<b>Entrada / Pasos de ejecución:</b> Se selecciona la extensión a la cual se le desea eliminar el buzón de voz.	
<b>Resultado Esperado:</b> Se elimina de forma satisfactoria de la base de datos el buzón de voz.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

**Tabla 4.6.2.2 Prueba de Aceptación “Gestionar buzón de voz de extensiones SIP”.**

## Capítulo IV: Implementación y Pruebas.

### 4.6.3 Caso de Prueba Historia de Usuario: Gestionar extensiones IAX2.

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba: 1</b>	<b>Número Historia de Usuario:</b> Gestionar extensiones IAX2
<b>Nombre de la Prueba:</b> Insertar extensión IAX2.	
<b>Descripción de la Prueba:</b> Probar que se inserta la extensión IAX2 de forma correcta en la base de datos.	
<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente y conectada a la base de datos.	
<b>Entrada / Pasos de ejecución:</b> Se introducen todos los datos de la extensión de forma correcta sin dejar campos en blanco.	
<b>Resultado Esperado:</b> Se inserta satisfactoriamente en la base de datos la extensión con todos sus datos.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Tabla 4.6.3.1 Prueba de Aceptación “Insertar extensión IAX2”.

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba: 2</b>	<b>Número Historia de Usuario:</b> Gestionar extensiones IAX2
<b>Nombre de la Prueba:</b> Eliminar extensión IAX2.	
<b>Descripción de la Prueba:</b> Probar que se elimina la extensión IAX2 de forma correcta de la base de datos.	
<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente y conectada a la	

## Capítulo IV: Implementación y Pruebas.

base de datos.
<b>Entrada / Pasos de ejecución:</b> Se selecciona la extensión que se desea eliminar.
<b>Resultado Esperado:</b> Se elimina de forma satisfactoria de la base de datos la extensión.
<b>Evaluación de la Prueba:</b> Satisfactoria

**Tabla 4.6.3.2 Prueba de Aceptación “Eliminar extensión IAX2”.**

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 3	<b>Número Historia de Usuario:</b> Gestionar extensiones IAX2
<b>Nombre de la Prueba:</b> Modificar extensión IAX2	
<b>Descripción de la Prueba:</b> Probar que se modifica la extensión IAX2 de forma correcta en la base de datos.	
<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente y existir la información de la extensión en la base de datos.	
<b>Entrada / Pasos de ejecución:</b> Se selecciona la extensión a la cual se le desea modificar los datos.	
<b>Resultado Esperado:</b> Se modifican los datos de forma satisfactoria en la base de datos.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

**Tabla 4.6.3.3 Prueba de Aceptación “Modificar extensión IAX2”.**

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> 4	<b>Número Historia de Usuario:</b> Gestionar extensiones IAX2
<b>Nombre de la Prueba:</b> Insertar extensión IAX2 ya existente.	



## Capítulo IV: Implementación y Pruebas.

<b>Descripción de la Prueba:</b> Probar que no se inserta una extensión IAX2 ya existente en la base de datos.
<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente y conectada a la base de datos.
<b>Entrada / Pasos de ejecución:</b> Insertar una extensión que se encuentre en la lista de extensiones.
<b>Resultado Esperado:</b> Se obtiene mensaje de error informándole al usuario que la extensión que desea insertar ya existe.
<b>Evaluación de la Prueba:</b> Satisfactoria

Tabla 4.6.3.4 Prueba de Aceptación “Insertar extensión IAX2 ya existente”.

### 4.6.4 Caso de Prueba Historia de Usuario: Gestionar buzón de voz de extensiones IAX2.

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba: 1</b>	<b>Número Historia de Usuario:</b> Gestionar buzón de voz de extensiones IAX2.
<b>Nombre de la Prueba:</b> Insertar buzón de voz de extensiones IAX2.	
<b>Descripción de la Prueba:</b> Probar que se inserta el buzón de voz de extensiones IAX2 de forma correcta.	
<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente y conectada a la base de datos.	
<b>Entrada / Pasos de ejecución:</b> Se selecciona la extensión a la cual se le desea insertar un buzón de voz.	
<b>Resultado Esperado:</b> Se inserta de forma correcta en la base de datos el buzón de voz.	

## Capítulo IV: Implementación y Pruebas.

**Evaluación de la Prueba:** Satisfactoria

**Tabla 4.6.4.1 Prueba de Aceptación “Gestionar buzón de voz de extensiones IAX2”.**

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba: 2</b>	<b>Número Historia de Usuario:</b> Gestionar buzón de voz de extensiones IAX2.
<b>Nombre de la Prueba:</b> Eliminar buzón de voz de extensiones IAX2	
<b>Descripción de la Prueba:</b> Probar que se elimina el buzón de voz de extensiones IAX2 de forma correcta de la base de datos.	
<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente y conectada a la base de datos.	
<b>Entrada / Pasos de ejecución:</b> Se selecciona la extensión a la cual se le desea eliminar el buzón de voz.	
<b>Resultado Esperado:</b> Se elimina satisfactoriamente el buzón de voz de la base de datos.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

**Tabla 4.6.4.2 Prueba de Aceptación “Gestionar buzón de voz de extensiones IAX2”.**

## Capítulo IV: Implementación y Pruebas.

### 4.6.5 Caso de Prueba Historia de Usuario: Mostrar datos del estado del sistema.

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba: 1</b>	<b>Número Historia de Usuario:</b> Mostrar datos del estado del sistema.
<b>Nombre de la Prueba:</b> Mostrar datos del estado del sistema.	
<b>Descripción de la Prueba:</b> Probar que se muestran los datos del estado del sistema de forma correcta.	
<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente.	
<b>Entrada / Pasos de ejecución:</b>	
<b>Resultado Esperado:</b> Se muestran los datos exactos del estado del sistema.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Tabla 4.6.5.1 Prueba de Aceptación “Mostrar datos del estado del sistema”.

### 4.6.6 Caso de Prueba Historia de Usuario: Configurar plan de marcación.

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba: 1</b>	<b>Número Historia de Usuario:</b> Configurar plan de marcación.
<b>Nombre de la Prueba:</b> Modificar plan de marcación.	
<b>Descripción de la Prueba:</b> Probar que se modifica el plan de marcación de forma correcta.	
<b>Condiciones de Ejecución:</b> La aplicación debe estar ejecutándose correctamente y conectada a la base de datos.	
<b>Entrada / Pasos de ejecución:</b> Se llenan los campos con los datos correctos.	

## Capítulo IV: Implementación y Pruebas.

<b>Resultado Esperado:</b> Se inserta en la base de datos el plan de marcación satisfactoriamente.
<b>Evaluación de la Prueba:</b> Satisfactoria

**Tabla 4.6.6.1 Prueba de Aceptación “Modificar plan de marcación”.**

### 4.7 Conclusiones

En el presente capítulo se abordaron las etapas de implementación y pruebas del software. Se establecieron y priorizaron las principales tareas de ingeniería asociadas a las historias de usuario para cada iteración y se realizaron los casos de pruebas que guiaron la calidad del sistema, lo que determinó en cada momento cuanto más se permitía avanzar en el desarrollo de la aplicación.

## Conclusiones Generales

Al concluir el desarrollo de la herramienta administrativa se arribó a la conclusión que el cumplimiento de las tareas de investigación planteadas ha permitido cumplir el objetivo satisfactoriamente.

El estudio de otros sistemas existentes en Cuba y el mundo para la administración de Asterisk, arrojó gran cantidad de información que fue adaptada y aprovechada para el desarrollo de la aplicación, partiendo del principio de reutilización de código, en aras de ganar tiempo. Por otra parte, los elementos obtenidos de un profundo estudio de Asterisk, aportaron importantes herramientas para el desarrollo de una aplicación altamente configurable capaz de realizar la administración sin provocar sobrecargas e intervenir en las rutinas de la PBX. A través de la interfaz que brinda la herramienta administrativa, el usuario puede gestionar extensiones y manejar datos que le permiten tomar decisiones o realizar cualquier acción sobre las llamadas, servicios y sucesos que tienen lugar en tiempo real dentro de la plataforma telefónica.

### Recomendaciones

Para mejorar el rendimiento y aumentar el alcance operacional de la aplicación, a continuación se exponen algunas recomendaciones:

- Incorporar otros módulos para la vinculación y gestión de otras aplicaciones.
- Enriquecer los servicios de la plataforma telefónica.

Con el fin de obtener resultados relevantes a corto plazo de tiempo en proyectos productivos, se recomienda:

- La aplicación metodologías ágiles en otros proyectos en la Universidad de las Ciencias Informáticas.
- Utilizar el *framework* TurboGears para el desarrollo de otros proyectos.

## Referencias Bibliográficas

1. FreePBX. [En línea] 2009. <http://www.freepbx.org/>
2. Management Interface for the Asterisk PBX. [En línea] 2009. <http://destar.berlios.de/index.php?c=3>
3. Asterisk-GUI. [En línea] 2009. <http://www.voip-info.org/wiki/view/Asterisk+GUI>
4. Servicios de capa de Transporte. [En línea] 2006. <http://www.textoscientificos.com/redes/tcp-ip/servicios-capa-transporte>.
5. **Escobar Jariton, Nicolás**. Tecnología. [En línea] 2007. <http://www.alexandria.com.mx/tecnologias.php>.
6. **Eguíluz Pérez, Javier**. Libros Web. [En línea] 2008. <http://www.librosweb.es/ajax/index.html>.
7. **Martínez, Daniel Pecos**. PostGreSQL vs. MySQL. NET Pecos. [En línea] [http://www.netpecos.org/docs/mysql\\_postgres/x15.html#AEN30](http://www.netpecos.org/docs/mysql_postgres/x15.html#AEN30).
8. **Ian Marteens**. LA BASE DE DATOS PERFECTA. [En línea] <http://www.intitec.com/articulos/BaseDatosPerfecta.htm>
9. Idem al 7. [En línea]
10. Fundación Apache. [En línea] <http://www.apache.org>.
11. **Moore, Dana, Budd, Raymond y Wright, William**. Professional Python Frameworks Web 2.0
12. **Razo ,López ,José Alberto**.Hacer AJAX utilizando Dojo. [En línea] 2006. [https://www.gruslic.org.mx/component/option,com\\_remository/Itemid,26/func,showdown/id,21/](https://www.gruslic.org.mx/component/option,com_remository/Itemid,26/func,showdown/id,21/)
13. **Jacobson, Ivar y Booch, Grady y Rumbaugh, James**. El proceso unificado de software. s.l. : Pearson Educación, 2000.
14. Metodologías De Desarrollo De Software. 2007.
15. **Canós, José H. , Letelier, Patricio**. Metodologías Ágiles en el Desarrollo de Software.
16. XP, MSF. [En línea] 2004. [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html).

## Referencias Bibliográficas

17. **Ferrer, G. R. Y. J.** Programación eXtrema y Software Libre. 2002.
18. Idem al 15.
19. Idem al 15
20. RUP Conferencia de Ingeniería de Software Introducción a la Ingeniería de Software. 2005-2006.



## Bibliografía

**Asterisknow.**2009 <http://www.asterisknow.org/>

**Ciberaula.** Una Introducción a APACHE. Ciberaula. [En línea] [Citado el: 6 de febrero de 2008.] [http://linux.ciberaula.com/articulo/linux\\_apache\\_intro](http://linux.ciberaula.com/articulo/linux_apache_intro).

**Daly, Liza.** Next-Generation Web Frameworks in Python. s.l: O'Reilly, April 09, 2007.

**Duque, Raúl González.** Python para todos. España: s.n.

**El Proceso Unificado de Desarrollo de Software.** 2000.

**El servidor de web Apache: Introducción práctica. Apache 1.x y 2.0 alpha.** [En línea] 2000.

F. Javier García Castellano, Pedro Angel Castillo Valdivieso, Juan Julián Melero Guervós **Tutorial de Introducción a Perl.** 2002 <http://flanagan.ugr.es/perl/index2.htm>

**FreePbx.**2009 <http://www.freepbx.org/support/documentation>

**Gift, Noah y Jones, Jeremy M.** Python for Unix and Linux System Administration. s.l.: O'Reilly, 2008.

<http://acs.barrapunto.org/articulos/trunk/LinuxActual/Apache/apache.pdf>.

Management Interface for the Asterisk PBX. <http://destar.berlios.de/index.php?c=3>

**Moore, Dana, Budd, Raymond y Wright, William.** Professional Python Frameworks Web 2.0 Programming with Django and TurboGears. s.l.: Wiley, 2007.

**MySQL AB.** Las principales características de MySQL. MySQL.1995 [En línea]

<http://dev.mysql.com/doc/refman/5.0/es/features.html>

**Pecos, Daniel.** PostgreSQL vs. MySQL. www.netpecos.org. [Online]

[http://www.netpecos.org/docs/mysql\\_postgres/x15.html#AEN30](http://www.netpecos.org/docs/mysql_postgres/x15.html#AEN30).

**Proceso Unificado del Modelado de Software.**

<http://www.chaco.gov.ar/UTN/disenodesistemas/apuntes/oo/ApunteRUP.pdf>

**Rodríguez Villar, Malay.** Introducción de procedimientos Ágiles en la producción de software en la Facultad 7 de la Universidad de las Ciencias Informáticas. [En línea] 2007.

[http://bibliodoc.uci.cu/TD/TD\\_0693\\_07.pdf](http://bibliodoc.uci.cu/TD/TD_0693_07.pdf)

Sitio oficial de Python. “About Python”. 2009 [Disponible en]: <http://www.python.org/about>

**Van Meggelen, Jim y Jared Smith, Leif Madsen E.** The Future of Telephony. s.l.: O' Reilly, August, 2007.

¿Qué es un Sistema Gestor de Bases de Datos o SGBD? [En línea]

<http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>

## Glosario de Términos

**AJAX:** Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications).

**Framework:** Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software, para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

**GPL:** La GNU General Public License (inglés: Licencia Pública General) es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es Software Libre.

**HTML:** Siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

**IAX2:** Protocolo de intercambio entre Asterisk versión 2.

**Interfaz de usuario gráfica (GUI):** es el artefacto tecnológico de un sistema interactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático. La interfaz gráfica de usuario (en inglés Graphical User Interface, GUI) es un tipo de interfaz de usuario que utiliza un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Habitualmente las acciones se realizan mediante manipulación directa para facilitar la interacción del usuario con la computadora.

**IP:** es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol).

## Glosario de Términos

**Opensource:** Código abierto (del inglés open source) es el término con el que se conoce al software distribuido y desarrollado libremente. Fue utilizado por primera vez en 1998 por algunos usuarios de la comunidad del software libre, tratando de usarlo como reemplazo al ambiguo nombre original en inglés del software libre (free software).

**PBX:** Centralita (también denominada Central Telefónica para Negocios Privados) – es una central telefónica propiedad de una empresa privada, en contraposición con la central que es propiedad de un operador de telecomunicaciones o de una empresa de telefonía.

**RUP:** Metodología de desarrollo de software basada en UML. Organiza el desarrollo de software en 4 fases.

**SIP:** Protocolo de inicio de sesión – es un protocolo desarrollado por el Grupo de Trabajo IETF MMUSIC y estándar propuesto para iniciar, modificar y terminar una sesión de usuario interactiva que implica elementos multimedia, tal como video, voz, mensajería instantánea, juegos en línea y realidad virtual.

**URL:** Son las siglas de Localizador de Recurso Uniforme, la dirección global de documentos y de otros recursos en la World Wide Web.

**VoIP:** Voz sobre el protocolo de internet (también denominado Telefonía IP, telefonía por Internet y Teléfono Digital) es el enrutamiento de conversaciones de voz a través de Internet o cualquier otra red basada en IP.

Anexos

Anexo No.1 Interfaz " Extensiones SIP"

The screenshot shows the PLATEL-SOFTWARE PARA CENTROS DE CONTACTOS interface. The header displays the date 'Jueves 04 de Junio de 2009' and navigation links for 'Inicio' and 'Ayuda'. The sidebar menu includes 'Sistema', 'Configuraciones Generales', 'Clases de Servicios', 'Plan de Numeración', 'Extensiones', 'Mantenimiento', 'Reportes', and 'IVR'. The main content area is titled 'Extensiones SIP' and features a 'Básico' tab. The form contains the following fields:

- Extensión:** 1001
- Nombre de Usuario:** Pedro Pablo
- Contraseña:** [Redacted]
- Privilegios:** Realizar/Recibir
- Clase de servicios:** 1

Below the form is an 'Aceptar' button. To the right, a table lists existing extensions:

Extensión	Nombre Usuario	CDS			
1000	1000	1	[Edit]	[Delete]	[Reset]
3000	3000	1	[Edit]	[Delete]	[Reset]
3001	Pedro	1	[Edit]	[Delete]	[Reset]
3002	3002	1	[Edit]	[Delete]	[Reset]
3006	Duany	1	[Edit]	[Delete]	[Reset]
3007	3007	1	[Edit]	[Delete]	[Reset]

Figura 1: Interfaz " Extensiones SIP"

Anexo No.2 Interfaz " Extensiones IAX2"

The screenshot displays the 'Extensiones IAX' configuration interface. On the left is a sidebar menu with categories like 'Sistema', 'Extensiones', and 'Mantenimiento'. The top bar shows the application name and the date 'Jueves 04 de Junio de 2009'. The main area contains a form with 'Básico' and 'Avanzado' tabs. The 'Básico' tab is active, showing fields for 'Extensión:', 'Nombre Usuario:', 'Contraseña:', 'Permisos de llamada:' (set to 'Realizar/Recibir'), and 'Clase de servicios:'. Below the form is an 'Aceptar' button. To the right is a table listing existing extensions.

Extensión	Nombre Usuario	CDS			
101	Asterisk-1	1			
102	Asterisk -2	1			
103	Asterisk-3	1			
104	Oficina-1	1			
105	Laboratorio-6	1			
2000	Asterisk1	1			
2001	Asterisk2	1			
2003	Asterisk3	1			
2004	Asterisk5	1			

Figura 2: Interfaz " Extensiones IAX2"

## Anexo No.3 Interfaz "Configuraciones generales"

PLATEL-SOFTWARE PARA CENTROS DE CONTACTOS

Jueves 04 de Junio de 2009 Inicio Ayuda Salir

Menú

- Sistema
  - Configuraciones Generales
  - Clases de Servicios
  - Plan de Numeración
- Extensiones
  - SIP
  - IAX
- Mantenimiento
  - Panel
- Reportes
- IVR

### Configuraciones generales

#### Buzón de voz.

Formato para los mensajes del buzón de voz: Gsn

Máximo número de autenticaciones fallidas: 3

#### Usuarios SIP.

Puerto SIP: 5060

Escuchar: 0.0.0.0

Permitir registros: 0.0.0.0 0.0.0.0

Denegar registros: 0.0.0.0 0.0.0.0

#### Usuarios de IAX

Puerto IAX: 4569

Escuchar: 0.0.0.0

Permitir registros: 0.0.0.0 0.0.0.0

Denegar registros: 0.0.0.0 0.0.0.0

Guardar Guardar y recargar módulo Reiniciar PBX

Figura 3: Interfaz "Configuraciones generales"

Anexo No.4 Interfaz " Estado del Sistema"

PLATEL-SOFTWARE PARA CENTROS DE CONTACTOS

Jueves 04 de Junio de 2009 Inicio Ayuda

Menú

- Sistema
  - Configuraciones Generales
  - Clases de Servicios
  - Plan de Numeración
- Extensiones
  - SIP
  - IAX
- Mantenimiento
  - Panel
- Reportes
- IVR

**Estado del sistema**

Tiempo de servicio de la PBX			
Días	Horas	Minutos	Segundos
0	0	47	45

Canales    Extensiones SIP    Extensiones IAX

SIP				
	Proveedor	Usuario	ID	En espera
	10.12.165.36	1000	9e0000-6747	No
	10.12.165.36	1000	6f0000-7033	Si
	10.12.165.37	3006	4200000-58a	No
	10.12.165.37	3006	3da0000-4de	Si
	10.12.165.37	3006	3290000-327	Si

IAX				
	Canal	Proveedor	Usuario	ID
	IAX2/10.12.165.33:45	10.12.165.33	conferenci	00001/00001
	IAX2/10.12.165.33:45	10.12.165.33	conferenci	00002/00002
	IAX2/10.12.165.33:45	10.12.165.33	conferenci	00004/00004
	IAX2/10.12.165.33:45	10.12.165.33	conferenci	00005/00005
	IAX2/10.12.165.33:45	10.12.165.33	conferenci	00006/00006

Figura 4: Interfaz " Estado del Sistema"



## Anexo No.5 Interfaz " Plan de Numeración"

PLATEL-SOFTWARE PARA CENTROS DE CONTACTOS

Jueves 04 de Junio de 2009 Inicio Ayuda

Menú

- Sistema
  - Configuraciones Generales
  - Clases de Servicios
  - Plan de Numeración
- Extensiones
  - SIP
  - IAX
- Mantenimiento
  - Panel
- Reportes
- IVR

Plan de numeración

Servicios	Código
Desvío de llamadas en ocupado.	*20
Desvío de llamadas en no contesta.	*22
Desvío de llamadas incondicional.	*21
Chequeo de desvío de llamadas.	*24
Búsqueda en el directorio.	*88
Conferencia 600.	500
Conferencia 610.	610
Conferencia 620.	620
Conferencia 630.	630
Conferencia 640.	640
Intrusión silenciosa de llamadas.	*04
Identificador de la extensión que llama.	*13
Fecha y hora.	*12
Acceso al buzón de voz.	100
Despertador.	*77

Aceptar

Figura 5: Interfaz " Plan de Numeración"

Anexo No.6 Interfaz " Clases de Servicios"

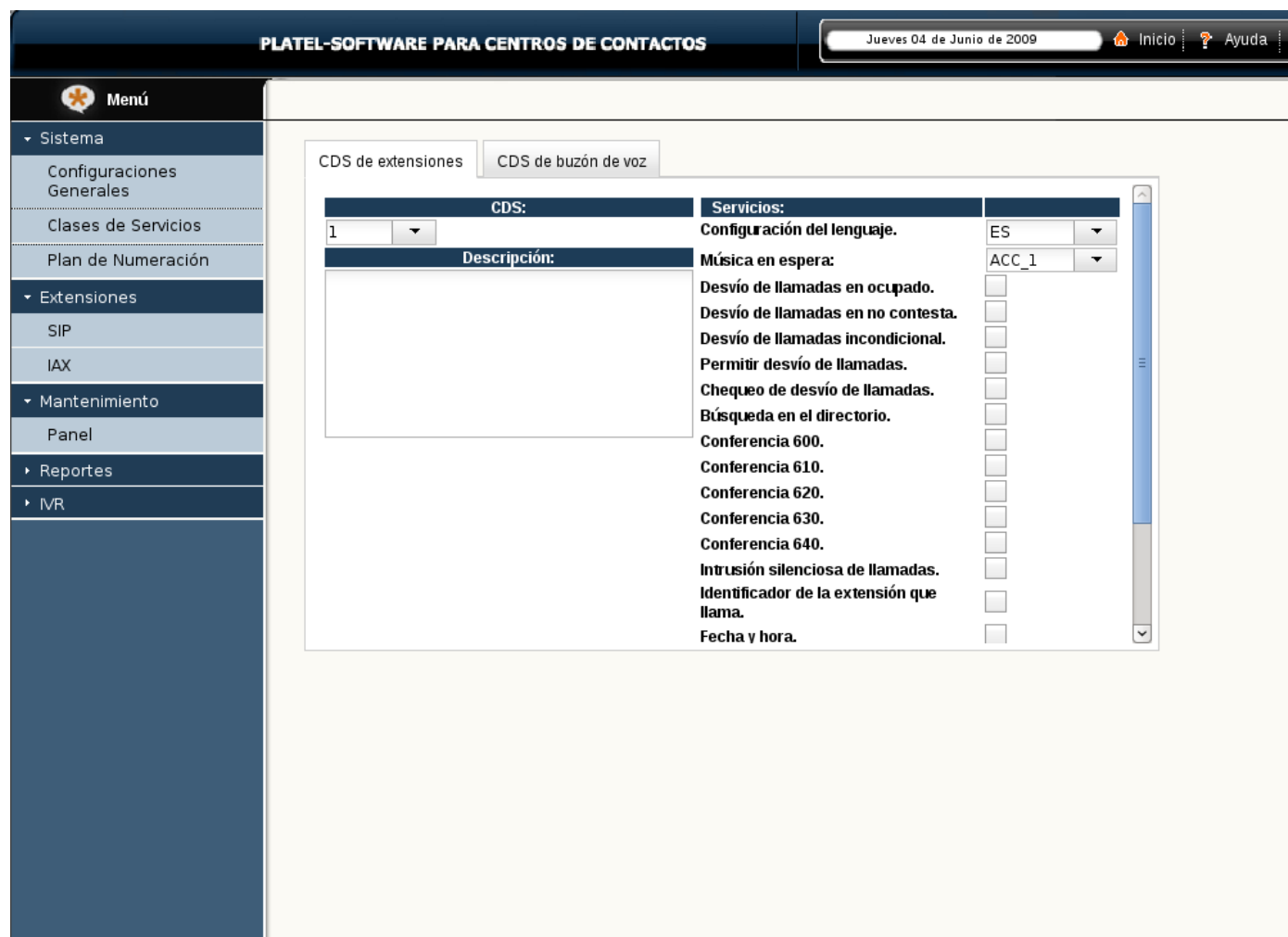


Figura 6: Interfaz " Clases de Servicios"

Anexo No.7 Modelo Cliente-Servidor.

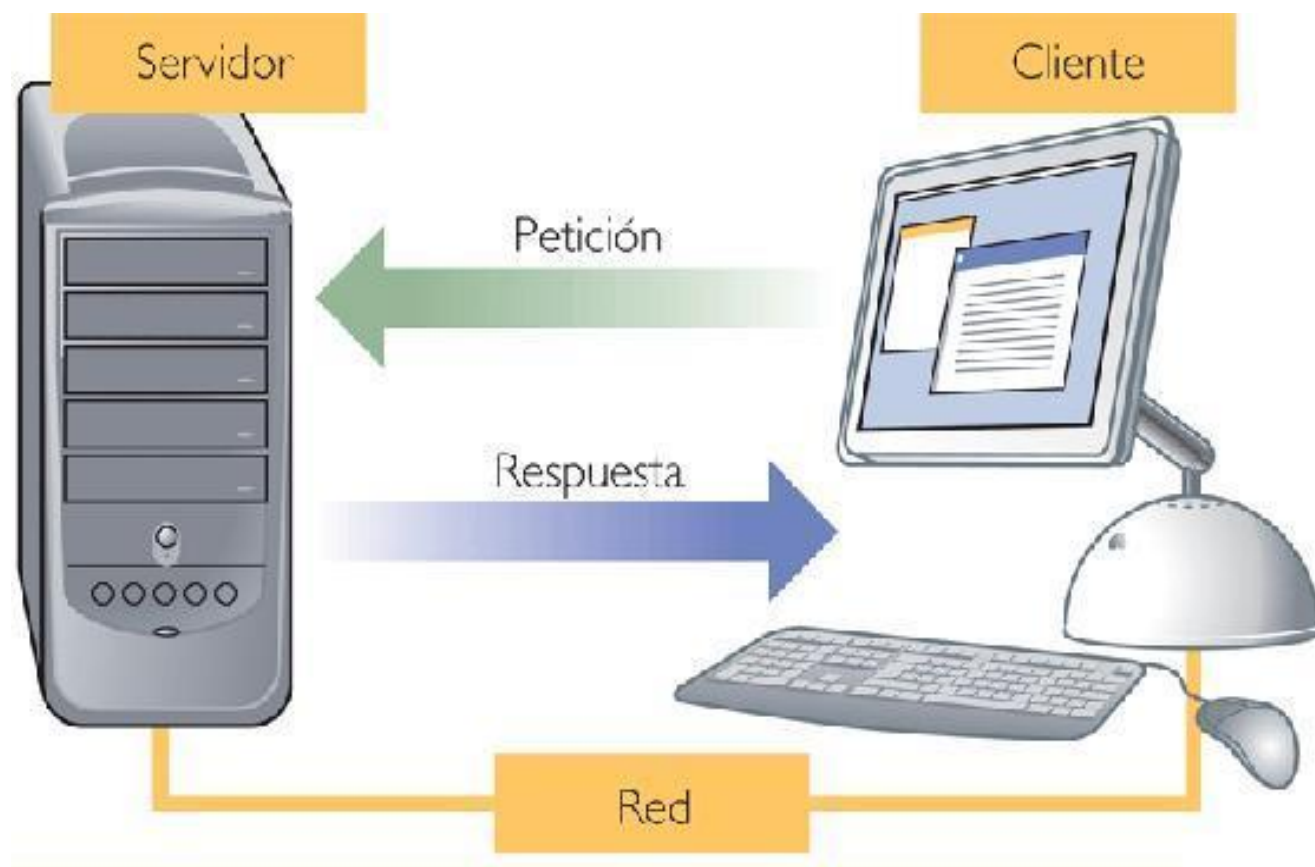


Figura 7: Modelo Cliente-Servidor

Anexo No.8 Arquitectura del sistema.

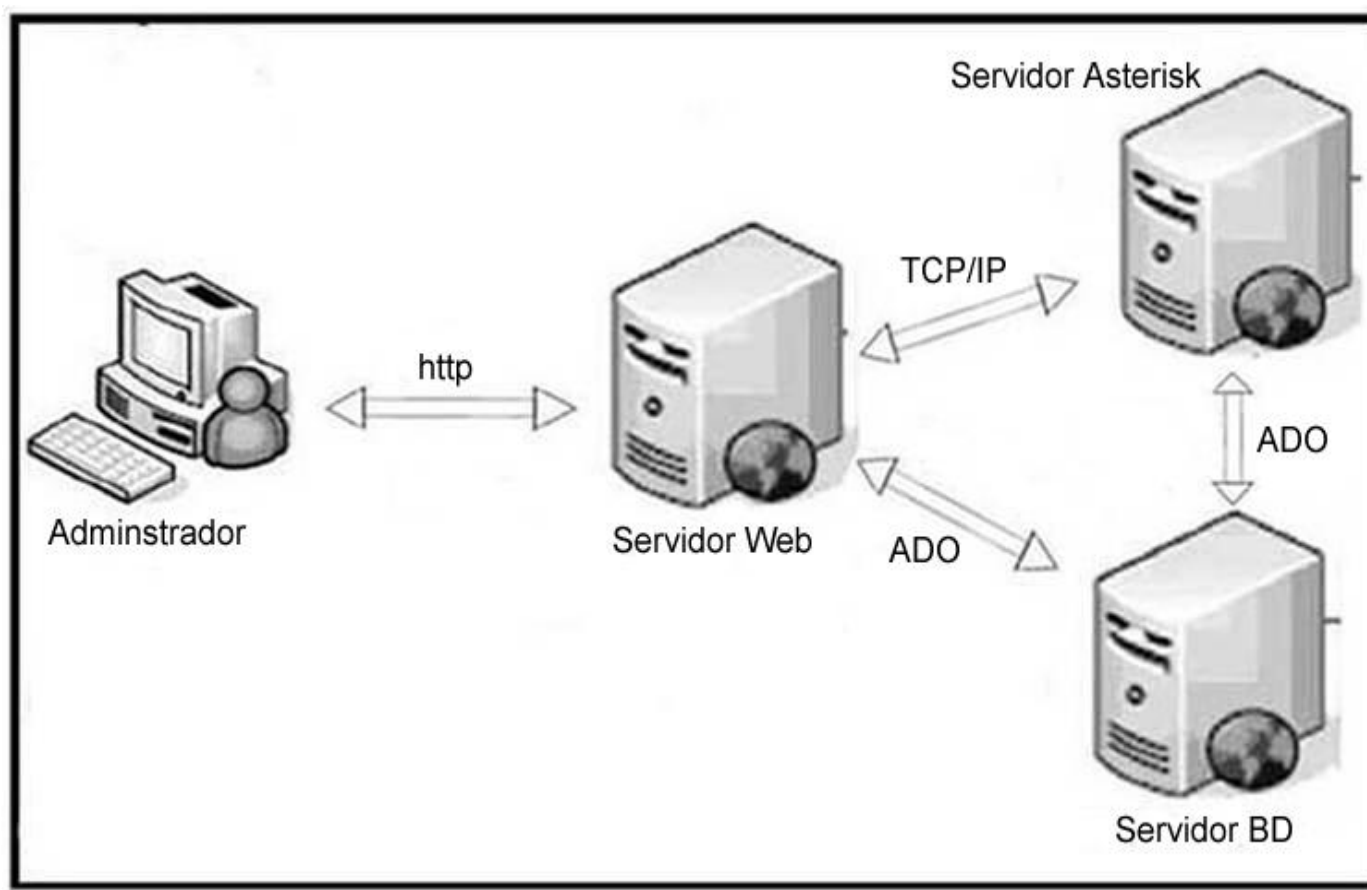


Figura 8: Arquitectura del sistema.