

Universidad de Ciencias Informáticas

# Sistema de Reportes y Facturación de PBX Asterisk

Trabajo de Diploma para optar por el Título de Ingeniero Informático

Autores: Nicolás Montero Puñales  
Luis Eduardo Acosta Aparicio

Tutor: Ing. Yuliesky Bello

Ciudad Habana, Junio 2009

*Dedicado a*

*A mi madre por ser mi fuente de amor y cariño.*

*Luis E.*

*A mi hija por darme la fuerza, la convicción y el motivo.*

*Nicolás*

*Agradecemos a:*

*A mi mamá por ser lo más grande y hermoso que existe en el mundo.*

*A mis abuelos que siempre tengo presentes y quiero con todo mi corazón.*

*A mi prima, a mis tíos por ser mis segundos padres y que además los quiero mucho.*

*A toda mi familia, por preocuparse por este trabajo.*

*A todos mis amigos que han compartido conmigo estos cinco años.*

*A Anni por darme todo lo que necesito para ser feliz.*

*A mi tutor y profesores que han hecho de un sueño la realidad.*

*A mi profesora Francís de Español del Pre.*

*Luis E.*

*A mi madre por ser todo lo que siempre he necesitado, por su amor, su apoyo y su cariño.*

*A mi padre por darme la posibilidad de hacerme un hombre independiente.*

*A mi hermana por soportarme todo este tiempo, y saber apreciar mi extraña forma de demostrar amor y cariño.*

*A mi hermanita por ser tan linda y por quererme tanto.*

*A Jose por todo lo que me ha dado todo este tiempo.*

*A Y@nelis por ser el tesoro al final del arcoíris.*

*A mis abuelos por confiar tanto en mí.*

*A mis demás familiares por brindarme siempre su ayuda.*

*A todos mis amigos por estar ahí cuando los necesitaba y cuando no también.*

*A todos aquellos que de una forma u otra estuvieron a mi lado y me brindaron su apoyo.*

*A mi tutor, profesores y todos aquellos que hicieron posible la realización de este trabajo.*

*Nicolás*

## Resumen

El presente trabajo comprende la elaboración de una herramienta que sea capaz de agilizar el proceso de construcción de reportes y facturación de una PBX (Private Branch Exchange) basada en Asterisk.

La automatización de dicho proceso se hace necesaria debido a que en este momento tanto la configuración como la obtención de información para los reportes y la facturación se hace de forma manual en ficheros de texto que se encuentran en el servidor.

La elaboración de reportes y facturas, es un eslabón esencial dentro de los servicios que brinda cualquier PBX, y si a esto se le agrega la posibilidad de hacerlo de forma visual y remota mediante una aplicación Web, se garantiza un mejor funcionamiento de la Plataforma.

En Cuba no ha sido elaborado un sistema con estas características, y aquellos que existen a nivel mundial son vendidos por grandes sumas de dinero o en ocasiones no satisfacen las necesidades. Es por ello que el presente sistema está diseñado para optimizar dicho proceso y cumplir con la meta de soberanía tecnológica, aspecto importante dentro del desarrollo de nuestra sociedad.

## Índice

<b>Introducción.....</b>	<b>1</b>
<b>Capítulo 1.Fundamentación Teórica .....</b>	<b>4</b>
Introducción .....	4
1. Sistemas de Reportes y Facturación en el mundo .....	4
1.1    Billing IP, Calling Card .....	5
1.1    FreePBX.....	5
1.2    Elastix.....	5
1.3    A2billing .....	6
2. Sistema de Reportes y Facturación en Cuba y en la Universidad de las Ciencias Informáticas .....	6
3. Marco de trabajo, herramientas y metodología utilizada.....	7
3.1 Servidores Web .....	7
3.1.1 Apache .....	7
3.1.2 Lighttpd.....	8
3.3.3 Thttpd .....	8
3.1.4 Servidor Web seleccionado. Apache .....	9
3.2 Gestores de Base de Datos .....	10
3.2.1 MySQL.....	10
3.2.2 Oracle .....	11
3.2.3 PostgreSQL.....	12
3.3.3 Servidor de Base de datos seleccionado. PostgreSQL.....	14
3.3 Metodologías de Desarrollo de Software .....	14
3.3.1 Proceso Unificado de Desarrollo de Software (RUP) .....	15
3.3.2 Programación Extrema (XP).....	16
3.3.3 Scrum.....	18

3.3.4 Metodologías seleccionadas. Programación Extrema y Scrum .....	19
3.3 Entorno de Desarrollo Integrado (IDE) .....	19
3.4 Tecnologías del lado del cliente .....	20
3.4.3 AJAX.....	20
3.4.4 CSS.....	21
3.4.1 HTML .....	22
3.4.2 Javascript .....	22
3.5 Lenguajes de Programación.....	24
3.5.1 Java .....	24
3.5.2 Python.....	25
3.5.3 Perl .....	26
3.5.4 PHP .....	26
3.5.5 Lenguaje seleccionado. Python.....	28
4. Framework.....	28
4.1. Turbogears.....	29
5. Dojo Toolkit .....	31
6. Plataforma .....	31
6.1 Aplicación Web .....	31
6.2 Modelo Cliente/Servidor .....	33
7. Conclusiones.....	35
<b>Capítulo 2.Características del Sistema .....</b>	<b>36</b>
Introducción .....	36
1. Descripción del proceso de Reporte y Facturación en Asterisk PBX.....	36
2. Objeto de Automatización.....	36
3. Información que se maneja.....	37
4. Propuesta del Sistema.....	37
5. Arquitectura Propuesta.....	38
6. Interfaz de usuario .....	38

6.1 Configurar sistema de reportes .....	38
6.2 Configurar sistema de facturas .....	39
6.3 Reportes predefinidos .....	39
6.4 Reportes por criterio de búsqueda .....	39
6.5 Reportes estadísticos evolutivos .....	39
7. Requisitos no funcionales del sistema. ....	40
7.1 Requerimientos de apariencia o interfaz externa .....	40
7.2 Requerimientos de usabilidad .....	40
7.3 Requerimientos de rendimiento .....	40
7.4 Requerimiento de portabilidad .....	41
7.5 Requerimientos de seguridad .....	41
7.6 Requerimientos de hardware .....	41
8. Restricción en el diseño y la implementación .....	42
9. Conclusiones .....	42
<b>Capítulo 3.Exploración, Planificación de la entrega de iteraciones .....</b>	<b>43</b>
Introducción .....	43
1. Fase de exploración .....	43
2. Historias de Usuario .....	43
3. Planificación .....	48
3.1 Estimación de esfuerzo por Historia de Usuario .....	48
4. Iteraciones .....	49
4.1 Plan de duración de las Iteraciones .....	49
4.2 Plan de Entrega .....	50
5. Conclusiones .....	51
<b>Capítulo 4.Iteraciones, Producción y Pruebas .....</b>	<b>52</b>
Introducción .....	52
1. Primera Iteración .....	52
1.1 Tareas de Historia de Usuario realizadas durante la primera iteración .....	53

2. Segunda Iteración .....	56
2.1 Tareas de Historia de Usuario realizadas durante la segunda iteración .....	56
3. Tercera Iteración .....	60
3.1 Tareas de Historia de Usuario realizadas durante la tercera iteración .....	60
4. Pruebas de Aceptación .....	63
5. Conclusiones .....	71
<b>Conclusiones .....</b>	<b>72</b>
<b>Recomendaciones .....</b>	<b>73</b>
<b>Glosario de Términos .....</b>	<b>74</b>
<b>Referencias Bibliográficas .....</b>	<b>75</b>
<b>Anexos .....</b>	<b>77</b>
Anexo 1.Pruebas de Aceptación .....	77
Anexo 2.Historias de Usuario .....	77
Anexo 3.Tarea de Ingeniería.....	78

## Índice de Tablas

<b>Tabla 1:</b> Historia de Usuario Realizar Reportes por Criterio de búsqueda. ....	44
<b>Tabla 2:</b> Historia de Usuario Reportes Predefinidos. ....	45
<b>Tabla 3:</b> Historia de Usuario Reportes Estadísticos Evolutivos por rango de tiempo. ....	46
<b>Tabla 4:</b> Historia de Usuario Realizar Reportes de Registros de Llamadas diario.....	46
<b>Tabla 5:</b> Historia de Usuario Realizar Configuración del Sistema de Reportes.....	47
<b>Tabla 6:</b> Configuración del Sistema de Facturas. ....	47
<b>Tabla 7:</b> Estimación de esfuerzos por Historia de Usuario. ....	48
<b>Tabla 8:</b> Plan de duración de las iteraciones.....	50
<b>Tabla 9:</b> Duración total de las iteraciones. ....	50
<b>Tabla 10:</b> Duración total de las iteraciones.....	50
<b>Tabla 11:</b> Historias de Usuarios realizadas en la primera iteración. ....	53
<b>Tabla 12:</b> Tarea de Ingeniería para la Historia de Usuario Reportes por Criterio de búsqueda. ....	53
<b>Tabla 13:</b> Tarea de Ingeniería para la Historia de Usuario Reportes por Criterio de búsqueda. ....	54
<b>Tabla 14:</b> Tarea de Ingeniería para la Historia de Usuario Reportes por Criterio de búsqueda. ....	54
<b>Tabla 15:</b> Tarea de Ingeniería para la Historia de Usuario Reportes Predefinidos. ....	55
<b>Tabla 16:</b> Tarea de Ingeniería para la Historia de Usuario Reportes Predefinidos. ....	55
<b>Tabla 17:</b> Tarea de Ingeniería para la Historia de Usuario Reportes Predefinidos. ....	56
<b>Tabla 18:</b> Historias de Usuarios realizadas en la segunda iteración.....	56
<b>Tabla 19:</b> Tarea de Ingeniería para la Historia de Usuario Reportes Estadísticos Evolutivos por rango de tiempo. ....	57
<b>Tabla 20:</b> Tarea de Ingeniería para la Historia de Usuario Reportes Estadísticos Evolutivos por rango de tiempo. ....	57
<b>Tabla 21:</b> Tarea de Ingeniería para la Historia de Usuario Reportes Estadísticos Evolutivos por comparación.....	58
<b>Tabla 22:</b> Tarea de Ingeniería para la Historia de Usuario Reportes Estadísticos Evolutivos por comparación.....	58
<b>Tabla 23:</b> Tarea de Ingeniería para la Historia de Usuario Reportes Estadísticos Evolutivos por comparación.....	59
<b>Tabla 24:</b> Tarea de Ingeniería para la Historia de Usuario Reportes Estadísticos Evolutivos por comparación.....	60
<b>Tabla 25:</b> Historias de Usuarios realizadas en la tercera iteración. ....	60
<b>Tabla 26:</b> Tarea de Ingeniería para la Historia de Usuario Configuración del Sistema de Reportes.....	61
<b>Tabla 27:</b> Tarea de Ingeniería para la Historia de Usuario Configuración del Sistema de Reportes.....	61

<b>Tabla 28:</b> Tarea de Ingeniería para la Historia de Usuario Configuración del Sistema de Reportes.....	62
<b>Tabla 29:</b> Tarea de Ingeniería para la Historia de Usuario Configuración del Sistema de Facturas.....	62
<b>Tabla 30:</b> Tarea de Ingeniería para la Historia de Usuario Configuración del Sistema de Facturas.....	63
<b>Tabla 31:</b> Tarea de Ingeniería para la Historia de Usuario Configuración del Sistema de Facturas.....	63
<b>Tabla 32:</b> Caso de Prueba de Aceptación HU1_P1.....	64
<b>Tabla 33:</b> Caso de Prueba de Aceptación HU1_P2.....	65
<b>Tabla 34:</b> Caso de Prueba de Aceptación HU2_P1.....	65
<b>Tabla 35:</b> Caso de Prueba de Aceptación HU2_P2.....	66
<b>Tabla 36:</b> Caso de Prueba de Aceptación HU3_P1.....	66
<b>Tabla 37:</b> Caso de Prueba de Aceptación HU3_P2.....	67
<b>Tabla 38:</b> Caso de Prueba de Aceptación HU4_P1.....	67
<b>Tabla 39:</b> Caso de Prueba de Aceptación HU4_P2.....	68
<b>Tabla 40:</b> Caso de Prueba de Aceptación HU5_P1.....	69
<b>Tabla 41:</b> Caso de Prueba de Aceptación HU5_P2.....	69
<b>Tabla 42:</b> Caso de Prueba de Aceptación HU6_P1.....	70
<b>Tabla 43:</b> Caso de Prueba de Aceptación HU6_P2.....	70

## Índice de Figuras

Figura.1 Modelo Cliente/Servidor .....	34
Figura. 2 Arquitectura Propuesta .....	38

## Introducción

Actualmente la humanidad se encuentra en una era donde las nuevas tecnologías y el avance de las Comunicaciones, implicado por estas, rigen su comportamiento y se vuelve parte indispensable de las actividades cotidianas. Cuba además de estar inmersa en una verdadera revolución educacional, se ha volcado a la labor de alcanzar un nivel de automatización que la sitúe a la vanguardia de la sociedad.

La Universidad de Ciencias Informáticas (UCI), en correspondencia con el actual desarrollo, lleva a cabo una serie de proyectos que pretenden automatizar al máximo todos los procesos, no sólo aquellos relacionados con la actividad educacional dentro de la universidad, sino en todos los sectores de la sociedad, lo cual es un punto de partida para su informatización.

Un punto importante dentro de la informatización de la sociedad es el tema de las telecomunicaciones. La telefonía IP (Internet Protocol) o VoIP (Voz sobre IP) se encuentra a la vanguardia. Actualmente las plataformas telefónicas con estas características juegan un papel indispensable dentro de cualquier empresa o institución, pero tienen un alto precio no sólo de adquisición, sino también de soporte.

Una vía alternativa para muchas empresas o instituciones es el montaje de una plataforma a base de software. Este tipo de plataforma se ha vuelto muy conocida por sus facilidades de uso y sobre todo de expansión. Asterisk PBX es de las más utilizadas por ser Libre, y sobre todo por la gran comunidad con que cuenta a nivel mundial. Esta plataforma a base de software brinda todas las funcionalidades de aquellas desarrolladas por las más renombradas empresas.

Entre las funcionalidades más notables de Asterisk PBX se encuentra un completo sistema de reportes, con el objetivo de dar a los usuarios la posibilidad de controlar el tráfico de llamadas dentro de la plataforma, y realizar el cobro del servicio de telefonía. La administración de este sistema se hace de

forma manual en ficheros de configuración, haciendo muy difícil el trabajo del personal encargado de hacer los cambios pertinentes.

De lo anteriormente expuesto, se deriva la necesidad de un sistema que facilite la administración del sistema de reportes y facturación de Asterisk PBX.

### **Problema Científico**

¿Cómo facilitar la administración del sistema de reportes y facturación en Asterisk PBX?

### **Objetivo General**

Desarrollo de una herramienta que facilite la administración del sistema de reportes y facturación en Asterisk PBX.

### **Objetivos Específicos:**

- Comprender el funcionamiento del Sistema de reportes y facturación de Asterisk PBX.
- Diseñar una herramienta que cumpla con los requerimientos del cliente.
- Implementar las funcionalidades del Sistema.

### **Objeto de Estudio**

Sistema de Reportes y Facturación en una PBX.

### **Campo de Acción**

Administración del sistema de Reportes y Facturación en Asterisk PBX.

### Tareas de la investigación

- Estudiar las herramientas y sistemas de administración de reportes y facturación existentes hasta el momento.
- Instalar y configurar Asterisk PBX para estudiar el funcionamiento de su sistema de reportes y facturación.
- Seleccionar las herramientas para desarrollar el sistema, y elegir la plataforma en la que se desarrollará, fundamentando su elección.
- Seleccionar la metodología de Análisis y Diseño de sistemas informáticos, que facilite y garantice la creación con calidad del sistema.
- Obtener los requerimientos del sistema mediante el análisis de los resultados del estudio realizado.
- Desarrollar el diseño de una herramienta que automatice el sistema de administración de reportes y facturación.
- Implementar un sistema que brinde la solución al problema planteado.

# Capítulo 1

## Fundamentación Teórica

### Introducción

En este capítulo se abarcará el estado actual de los Sistemas de reportes y facturación en las PBX basadas en Asterisk a nivel mundial, nacional y en la Universidad de Ciencias Informáticas. Se presentarán las herramientas, tecnologías, y la metodología utilizada para el desarrollo.

### 1. Sistemas de Reportes y Facturación en el mundo

En el mundo existen sistemas basados en Asterisk como: FreePBX, Elastix, Billing IP Calling Card, A2billing. Estos sistemas proveen la posibilidad de consultar reportes detallados de las llamadas o brindan la posibilidad de poder configurar su facturación; por lo general cuentan con interfaces amigables para el usuario, aunque no disponen de una sólida reportería y facturación en conjunto. Los mencionados sistemas cuentan con ventajas y desventajas que serán expuestas a continuación.

Para realizar un correcto estudio de las herramientas existentes y determinar sus características se utilizan métodos investigativos como:

**Analítico–Sintético:** Durante el proceso de revisión bibliográfica y particularmente para conocer las ventajas y desventajas de estos softwares, asumiendo como centro del análisis el objeto de estudio que se aborda.

**Observación y Experimento:** Durante la instalación y prueba de estos softwares, observando el funcionamiento, características y prestaciones y estableciendo una comparación con las exigencias del cliente.

### **1.1 Billing IP, Calling Card**

Es una solución de software apta para empresas que quieran dar servicio de telefonía, así como de comercialización de telefonía prepaga. El software cuenta con elementos para la administración, el control y la cobranza de los servicios telefónicos que se brindan. En combinación con la plataforma PBX-IP de CallFon representan un verdadero sistema de funcionalidades de telefonía direccionada a la prestación de servicios, permitiendo la facturación de las comunicaciones, pre o post pago, incluido los reportes y estadísticas del funcionamiento de las líneas IP o las TDM, configurables para proveer un amplio rango de servicios, tarifas, facturación, etc. Este sistema está realizado sobre plataforma Libre pero actualmente está en venta por lo que tendría que ser adquirido en el extranjero. (1)

### **1.1 FreePBX**

FreePBX es una herramienta dedicada a la administración de PBX Asterisk. Es muy utilizado 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. Entre sus principales funcionalidades cuenta con un Registro Detallado de Llamadas y la opción de generar reportes dependiendo de criterios. Este sistema no implementa la factura de las llamadas y no cuenta con gráficas que ilustren el comportamiento de las llamadas. (2)

### **1.2 Elastix**

Es un sistema que integra varias herramientas disponibles para centrales telefónicas basadas en PBX Asterisk. Entre otras funcionalidades presenta un módulo para generar varios reportes aunque estos reportes no brindan suficiente información al no ser reportes comparativos, ni evolutivos; conjuntamente implementa la facturación de llamadas. El sistema está realizado sobre plataforma Libre, pero al integrar

varios módulos de otros sistemas, dificulta el soporte, que tiene que ser a través de cursos pagados que se imparten de forma online. (3)

### **1.3 A2billing**

A2billing es un completo sistema de facturación para Asterisk, es decir que muestra el importe a cobrar de acuerdo al tiempo de duración y destino de una llamada. El administrador puede modificar y ejecutar tareas que controlen el sistema a través de una interfaz web. Permite obtener datos estadísticos de la actividad y minutos consumidos por destino. Este sistema está realizado sobre plataforma Libre y posee gran auge gracias a su sencillez, su documentación es abundante aunque con respecto a los reportes no posee gran fortaleza, ya que se centra principalmente en la factura. Otra de las desventajas que presenta es que no permite la integración con base de datos como SQLite y Oracle; además de que muchas de las configuraciones necesarias para su trabajo se hacen sobre ficheros. (4)

Luego de elaborar un análisis que comprende las cuatro aplicaciones anteriormente expuestas, surge la necesidad de la confección de un sistema que reúna las mejores características de los existentes hasta el momento, agregando otras funcionalidades que necesite el cliente, garantizando así que cumpla con las exigencias de este. De esta manera también se contribuye con la ardua lucha por alcanzar la soberanía tecnológica que lleva a cabo nuestro país.

## **2. Sistema de Reportes y Facturación en Cuba y en la Universidad de las Ciencias Informáticas**

Actualmente consultar las llamadas y la facturación de una PBX basada en Asterisk en instituciones cubanas es un proceso complejo, debido a que no se cuenta con una interfaz que muestre los datos almacenados en ficheros de configuración localizados en la PC servidor, haciendo necesaria su búsqueda manual, para posteriormente realizar su consulta de una forma poco eficaz, ya que en el fichero se encuentran las estadísticas de todas las llamadas gestionadas por el servidor, y no es posible efectuar

una búsqueda rápida de las mismas por criterios, con el objetivo de obtener reportes detallados. En la Universidad de las Ciencias Informáticas aún no se dispone de una PBX basada en Asterisk.

### **3. Marco de trabajo, herramientas y metodología utilizada**

#### **3.1 Servidores Web**

##### **3.1.1 Apache**

Apache es un servidor de páginas web de código abierto, multiplataforma y modular, se desarrolla dentro del proyecto HTTP Server de la Apache Software Foundation. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. Se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular, que permite a los administradores de sitios web elegir qué características van a ser incluidas en el servidor, y seleccionar qué módulos se van a cargar, ya sea al compilar o al ejecutar el servidor.

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales, y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache. (5)

##### **Características principales:**

- Trabaja sobre múltiples plataformas (Unix, Linux, MacOSX, Win32, OS2, etc.).
- Incluye módulos que se cargan de forma dinámica.
- Soporta CGI, Perl, PHP.
- Soporte para Bases de datos.
- Soporte SSL para transacciones seguras.

- Incluye soporte para host virtuales.
- Soporta HTTP 1.1.
- Código Abierto.
- Rápido y eficiente.

### 3.1.2 Lighttpd

Lighttpd es un servidor web diseñado para ser rápido, seguro, flexible, y fiel a los estándares. Está optimizado para entornos donde la velocidad es muy importante, y por eso consume menos CPU y memoria RAM que otros servidores. Por todo lo que ofrece, Lighttpd es apropiado para cualquier servidor que tenga problemas de carga. Lighttpd es software libre y se distribuye bajo la licencia BSD. Funciona en GNU/Linux y UNIX de forma oficial. Para Microsoft Windows actualmente hay una distribución conocida como Lighttpd For Windows mantenida por Kevin Worthington. (6)

#### Características:

- Virtual hosting (aloja varios dominios en la misma IP).
- Soporte para PHP, Ruby, y otros.
- Compresión (gzip, bzip2).
- Consumo de memoria constante.
- Redirecciones HTTP, y reescrituras de URL.
- Puede enviar partes de un fichero (rangos).
- Permite módulos externos.
- 

### 3.3.3 Thttpd

Thttpd es un servidor web de código libre disponible para la mayoría de las variantes de Unix. Se caracteriza por ser simple, pequeño, portátil, rápido, y seguro, ya que utiliza los requerimientos mínimos

de un servidor HTTP. Esto lo hace ideal para servir grandes volúmenes de información estática. Se caracteriza por ser:

- Simple, porque maneja sólo el mínimo necesario para poner en práctica el protocolo HTTP, algunas veces un poco más que el mínimo.
- Pequeño, porque tiene un pequeño tamaño de período de explotación, ya que esto no se divide en dos partes y es muy cuidadoso sobre la asignación de memoria.
- Portátil, porque se compila limpiamente sobre la mayoría de sistemas operativos, expresamente incluyendo FreeBSD, SunOS 4, Solaris 2, BSD/OS, Linux, OSF.
- Rápido, porque en el empleo típico es más rápido que los mejores servidores “destacados” (APACHE), y bajo la carga extrema es mucho más rápido.
- Seguro, porque se extiende a grandes longitudes para proteger el servidor WEB contra ataques de otros sitios.

#### **Ventajas:**

- El administrador puede decidir restringir la transferencia de archivos de imagen JPG.
- Los promedios de carga se caen debido a la reducción de la transferencia gráfica.

#### **Desventajas:**

- No posee las mismas aplicaciones que se pueden obtener de un software estándar como lo es el Apache.

#### **3.1.4 Servidor Web seleccionado. Apache**

Luego de hacerse un estudio profundo de los servidores web, se decidió que para el entorno a desarrollar el mejor servidor web es el Apache por las siguientes razones:

**Ventajas:**

- Ayuda en la mejora del posicionamiento.
- Este servidor junto con el módulo mod\_rewrite puede convertirse en una herramienta muy útil para crear páginas con enlaces amigables para los buscadores.
- Es un software libre.
- Multiplataforma.
- Open Source.
- Modular.
- Extensible.
- Presenta mensajes de error altamente configurables.

**3.2 Gestores de Base de Datos**

En la actualidad los SGBD (Sistema de Gestión de Base de Datos) 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 ampliamente usado en la gestión de los sistemas informáticos.

**3.2.1 MySQL**

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario, que funciona en diferentes plataformas. Posee un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor. Soporta a grandes bases de datos. Un índice puede usar prefijos de una columna para los tipos de columna CHAR, VARCHAR, BLOB, o TEXT. MySQL Server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas. (7)

**Ventajas:**

- Consume muy pocos recursos de CPU y memoria.
- Tiene buen rendimiento.
- Posee un tamaño del registro sin límite.
- Control de acceso usuarios-tablas-permisos.

**Desventajas:**

- Carece de soporte para transacciones, rollback y subconsultas.
- El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí que poseen esta característica.
- No es viable su uso en grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.
- Actualmente SUN Microsystem tiene en su poder a MySQL, prohibiendo la descarga de futuras versiones desde Cuba.

**3.2.2 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. Las últimas versiones han sido certificadas para poder trabajar sobre plataforma Linux.

**Ventajas:**

- Permite implementar diseños "activos", con triggers 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.

**Desventajas:**

- El precio de las patentes y licencias del equipamiento que requiere Oracle, con respecto a otros sistemas gestores de bases de datos, constituye uno de sus principales problemas.
- Otro problema es la necesidad de ajustes. Un error frecuente consiste en pensar que basta instalar el Oracle en un servidor y enlazar directamente las aplicaciones clientes. Un Oracle mal configurado puede ser desesperantemente lento.

**3.2.3 PostgreSQL**

PostgreSQL es un servidor de base de datos objeto relacional libre. Como muchos otros proyectos Open Source, el desarrollo de PostgreSQL no es manejado por una sola compañía, sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales que trabajan en su desarrollo, dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group). Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido más tarde en otros sistemas de gestión comerciales.

PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos. (8)

**Ventajas:**

- Soporte de protocolo de comunicación encriptado por SSL.
- Extensiones para alta disponibilidad, nuevos tipos de índices, datos espaciales, minería de datos, etc.
- Incorpora una estructura de datos array.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- 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.

**Desventajas:**

- Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.
- No presenta la rapidez que otros similares.

### 3.3.3 Servidor de Base de datos seleccionado. PostgreSQL

Para el desarrollo del sistema se realizó un detallado estudio de los Gestores de Base de Datos, el cual permitió que se pudiera realizar una elección certera. En dicho estudio se llegó a la conclusión de que el Gestor que se utilizará es PostgreSQL, por las siguientes razones:

- Puede ser utilizado, modificado y distribuido por cualquiera gratuitamente, para cualquier propósito ya sea con fines privados, comerciales o académicos.
- Corre en casi todos los principales sistemas operativos: Linux, Unix, Windows, etc.
- Soporte nativo para los lenguajes más populares: PHP, C, C++, Perl, Python, etc.

Además de las facilidades antes mencionadas se decidió escoger este gestor porque es altamente escalable tanto en la cantidad de datos que puede manipular como en la cantidad de usuarios concurrentes que puede atender, lo cual se ajusta a las características del producto a desarrollar.

### 3.3 Metodologías de Desarrollo de Software

Después de ver y tener en cuenta si es necesario utilizar una metodología específica para el desarrollo de cualquier producto, se impone la tarea de decidir cuál de las metodologías existentes será la más adecuada o la mejor, cuál se corresponde más con el tipo de producto que se quiere obtener, cuál está más acorde con las características del equipo de trabajo, del entorno, del tiempo máximo para la entrega, o sea; deben considerarse aspectos muy importantes antes de adoptar una determinada metodología de desarrollo.

Actualmente existen muchas tendencias de metodologías que brindan diferentes marcos que los desarrolladores pueden emplear para realizar su trabajo, algunas de ellas son:

- RUP (Proceso Unificado de Desarrollo de Software)
- XP (Programación Extrema)

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

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.

#### **RUP se caracteriza por ser:**

- Centrado en la arquitectura: La arquitectura muestra una visión completa del sistema y se describen los elementos más importantes para la construcción del software.
- Dirigido por casos de uso: Los casos de uso muestran los que los usuarios necesitan y desean, esto se obtiene a partir de la modelación del negocio, quedando plasmado en la especificación de requisitos.
- Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Es muy útil dividir el proyecto en mini proyectos ya que el desarrollo de cada uno de ellos es una iteración que contribuye al incremento del proyecto general.
- RUP propone 9 flujos de trabajo: 6 de ingeniería y 3 de apoyo. Los flujos de trabajo ingenieriles son: Modelamiento del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba, y Despliegue.

#### **Ventajas:**

- Evaluación en cada fase que permite cambios de objetivos
- Funciona bien en proyectos de innovación.
- Sigue los pasos intuitivos necesarios a la hora de desarrollar el software.
- Seguimiento detallado en cada una de las fases.

**Desventajas:**

- La evaluación de riesgos es compleja.
- Excesiva flexibilidad para algunos proyectos
- El cliente queda en una situación que puede ser muy incómoda para él.
- El cliente deberá ser capaz de describir y entender a un gran nivel de detalle para poder acordar un alcance del proyecto con él. (9)

**3.3.2 Programación Extrema (XP)**

Actualmente la Programación Extrema se utiliza en proyectos riesgosos cuando se tiene requerimientos dinámicos. Las reglas y prácticas de XP son muy simples, cada una soporta a la otra creando una integración única que forma la metodología de desarrollo de software. Esta metodología pertenece al grupo de metodologías ágiles. Es una metodología basada en la simplicidad, comunicación y retroalimentación. En la Programación Extrema cada integrante es una parte del "TODO". El equipo está compuesto por un representante del negocio, llamado el cliente, que se sienta y trabaja con ellos todos los días, esto permite que cualquier duda a aclarar puede ser resuelta al instante por estar el cliente siempre disponible para el equipo de desarrollo.

**Reglas y prácticas de la Programación Extrema**

Se escriben Historias de Usuario (similares a los casos de uso de RUP), Tareas de la Investigación y Pruebas de Aceptación. El proyecto se divide por iteraciones, se realiza una reunión diaria con el objetivo de analizar y trazarse objetivos. El diseño en XP se caracteriza por ser simple y porque ninguna funcionalidad se añade tempranamente. El código del proyecto debe ser escrito según estándares aprobados. La primera unidad que se implementa es la unidad de prueba, dándosele a este elemento una importancia significativa dentro de la Programación Extrema. En la programación extrema la realización de

pruebas y la experiencia de los clientes adquieren una mayor importancia, haciendo que se trabaje en ciclos de menor tiempo.

**Ventajas:**

- Cuatro ojos ven más que dos. Al trabajar de dos en dos, el código será de mayor calidad desde el mismo momento de crearlo, y tendrá menos fallos.
- Los programadores novatos aprenderán de los expertos al emparejarse con ellos.
- Si una pareja realiza un pedazo de código susceptible de ser reutilizado en el proyecto, hay dos programadores que lo saben y que lo reutilizarán cuando puedan (ya que saben cómo funciona), enseñándolo a sus nuevos compañeros. De esta manera el conocimiento del código ya hecho se propaga de forma natural entre todos los programadores del equipo.
- El estilo de programación tiende a unificarse.
- Apropiado para entornos volátiles.
- Estar preparados para el cambio, significa reducir su coste.
- Planificación más transparente para los clientes, conocen las fechas de entrega de funcionalidades vitales para su negocio.
- Permite definir en cada iteración cuales son los objetivos de la siguiente.
- Permite tener realimentación de los usuarios.
- La presión está a lo largo de todo el proyecto y no en una entrega final.

**Desventajas:**

- Delimitar el alcance del proyecto con el cliente. (10)

### 3.3.3 Scrum

Scrum, más que una metodología de desarrollo software, es una forma de auto-gestión de los equipos de programadores. Un grupo de programadores deciden cómo hacer sus tareas y cuánto van a tardar en ello. Scrum ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, además permite seguir de forma clara el avance de las tareas a realizar, de forma que los "jefes" puedan ver día a día cómo progresa el trabajo.

Sin embargo, Scrum no es una metodología de desarrollo, puesto que no indica qué se debe hacer para hacer el código. Debería, por tanto, complementarse con alguna otra metodología de desarrollo. Se lleva bien con las metodologías ágiles y en concreto, con la programación extrema. (11)

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.

- Busca equipos efectivos y productivos (dentro de las posibilidades).
- Se basa en los principios de inspección continua, adaptación, auto-gestión e innovación.
- Provee un ámbito propicio para desarrollar sus capacidades profesionales (lo que resulta en un incremento en la motivación de los integrantes del equipo).

### **3.3.4 Metodologías seleccionadas. Programación Extrema y Scrum**

Luego del estudio de las metodologías anteriores se decide utilizar XP y Scrum, principalmente porque son metodologías de desarrollo de software ágiles además del complemento que existe entre ambas, se emplean en proyectos de corto plazo, es usada para equipos pequeños y cuyo plazo de entrega era “ayer”. Las metodologías consisten en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final o cliente, pues es uno de los requisitos para llegar al éxito del proyecto.

### **3.3 Entorno de Desarrollo Integrado (IDE)**

El Entorno de Desarrollo Integrado (IDE por sus siglas en inglés) de Eclipse emplea módulos (plugins) para proporcionar toda su funcionalidad, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite o no el usuario. Esta característica le permite a Eclipse extenderse utilizando lenguajes de programación como Ruby, PHP y Python entre otros.

Easy Eclipse empaqueta el entorno de desarrollo Eclipse junto con una cuidada selección de módulos Open Source para obtener un IDE final excepcionalmente bueno para el desarrollo de aplicaciones en PHP, Python, Ruby o Java. Con todos los plugins ya instalados y configurados se puede comenzar el trabajo de forma sencilla y rápida. Easy Eclipse dispone de varias distribuciones para desarrollo entre ellas una optimizada para el desarrollo en Python.

**Las principales fortalezas de esta herramienta son:**

- Es Opensource y multiplataforma.
- Posee integración con Sistemas Controladores de Versiones (SVN por sus siglas en inglés).
- Posee un buen completamiento de código que contribuye al desarrollo ágil de la aplicación.
- Posee una extensa documentación y soporte en internet.
- El consumo es aceptable lo cual no interfiere con el desarrollo.

**3.4 Tecnologías del lado del cliente**

Son las que están insertadas en la página HTML del cliente, siendo interpretadas y ejecutadas por el navegador. Por una parte, se encargan de gestionar la comunicación con el servidor, de solicitar un servicio concreto y de recibir los datos enviados por este, y por otra parte, es 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 soporte de la versión del navegador a ser utilizado por el usuario.

**3.4.3 AJAX**

AJAX, acrónimo de JavaScript asíncrono y XML, no es una tecnología en sí misma; es 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. [8]

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, CCS, XSLT, y el objeto XMLHttpRequest.

### 3.4.4 CSS

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación; y es imprescindible para crear páginas web complejas.

Dicho lenguaje se utiliza para definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, la tabulación con la que se muestran los elementos de una lista.

#### **Ventajas:**

- Obliga a crear documentos HTML/XHTML bien definidos y con significado completo.
- Mejora la accesibilidad del documento.
- Reduce la complejidad de su mantenimiento.
- Permite visualizar el mismo documento en infinidad de dispositivos diferentes.

#### **Desventaja:**

Incompatibilidad entre diferentes navegadores, por lo que si algo se ve excelente en un navegador es posible que en otro no se vea igual, o que varíe la visualización incluso en las diferentes versiones de un mismo navegador.

### 3.4.1 HTML

HTML, Hyper Text Markup Language (Lenguaje de Marca de Hipertextos), es un lenguaje de programación muy sencillo que se utiliza para crear los textos y las páginas web. Es justamente un lenguaje que se basa en las marcas para crear los hipertextos. Está compuesto por etiquetas que definen la estructura y el formato del documento que verá el usuario a través de la web. Esas etiquetas son leídas por el navegador o visualizador, es decir el programa que se utiliza para navegar, y que es el que ejecuta las funciones creadas en HTML permitiendo que puedan ser visibles en nuestra máquina. Es el lenguaje en que se escriben los millones de documentos que hoy existen en internet. Una de las características es que, además del texto, este lenguaje permite que se creen enlaces entre distintas partes del mismo documento o entre distintas fuentes de información a través de hiperenlaces o hipervínculos, e incluso insertar otros elementos como imágenes y sonidos. La principal ventaja que tiene HTML es la gran variedad de navegadores y exploradores que lo soportan. Debido a esto, se ha convertido en el formato más usado para la transmisión de documentos electrónicos a través de Internet. Actualmente existe un gran número de navegadores por lo que a veces se complica que todos puedan interpretar el código HTML de la misma manera. Por esa razón es que quienes crean las páginas, chequean que pueda ser leída por los menos por los navegadores más conocidos.

### 3.4.2 Javascript

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

**Ventajas:**

- Tiempo de transferencia : El mantener la ejecución de lógica en el servidor de páginas implica que deben llevarse a cabo varias requisiciones, esto es, si se necesita llevar a cabo un cálculo o se le están solicitando datos a una persona, esta información debe ser transferida hasta el servidor de páginas para ser ejecutada o validada , en cambio, si se utiliza un "Scripting Language" es posible realizar esta lógica instantáneamente en el cliente , sin incurrir en una transferencia de información adicional.
- Permite elaborar aplicaciones Web que simulen características de aplicaciones escritorios.
- Es un código "interpretado" por el cliente.
- Es un código orientado a objetos.
- 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 cualquiera, incluso si está protegido con las leyes del copyright.

Unos de los inconvenientes que tiene este lenguaje es que tiene que 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 JavaScript para sus aplicaciones, debido a que no todos los navegadores interpretan de la misma manera el código.

### 3.5 Lenguajes de Programación

La AGI (Asterisk Gateway Interface del inglés), 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. Pasando por el control del dial plan a un externo script de la AGI de Asterisk se pueden realizar fácilmente tareas que de otra manera serían difíciles o imposibles.

La PBX Asterisk brinda una interfaz estándar para los scripts de la AGI, los cuales pueden ser escritos en casi cualquier lenguaje de programación moderno. Pero los lenguajes más comúnmente utilizados y que se destacan en la programación AGI son: Perl, Java, PHP, Python.

#### 3.5.1 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 es interesante 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, agendas y en general para cualquier cosa que se le ocurra a la industria.

### 3.5.2 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 indentada (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 su naturaleza interpretada hacen de él, el lenguaje 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 nivel y una solución de programación. (12)

La implementación de Python es bajo la licencia de código abierto que permite el uso libre y distribución para un uso comercial. La licencia de Python es administrada por la Fundación Python de Software. [10]

En los últimos años el lenguaje se ha hecho muy conocido, gracias a varias razones como:

- La cantidad de librerías que contiene, 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.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- La cantidad de plataformas en las que se puede desarrollar, como Unix, Windows, OS/2, Mac, Amiga y otros.

#### **Ventajas:**

- Desarrollo más rápido: Puedes escribir un programa, salvarlo y ejecutarlo. En un lenguaje compilado tienes que pasar por los pasos de compilar y ligar el software, lo cual puede ser un proceso lento.

- **Multiplataforma:** El mismo código funciona en cualquier sistema operativo, la única condición es que disponga del intérprete del lenguaje.

**Desventajas:**

- **Lentitud:** Los programas interpretados son más lentos que los compilados. Sin embargo los programas interpretados suelen ser cortos (como los scripts), en los que la diferencia es inapreciable.

**3.5.3 Perl**

Es un lenguaje que toma características del C, del lenguaje interpretado Shell. 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.

Perl permite resolver los problemas fáciles fácilmente, y resolver también problemas difíciles. Es rápido hacer una pequeña aplicación web.

Perl es la alternativa más popular a PHP, seguramente porque es el lenguaje más antiguo también dentro de las alternativas. Perl tiene una ventaja y es que es muy flexible, y también tiene una gran cantidad de módulos ya escritos.

La principal causa de la sucia apariencia de Perl es por la afición de sus desarrolladores a la escritura en "una línea", empaquetando numerosas funcionalidades en una sola línea de código. Bien escritos los scripts en Perl se asemejan bastante a PHP. (13)

**3.5.4 PHP**

Es un lenguaje de programación utilizado para la creación de sitio web. PHP es un acrónimo recursivo que significa "PHP Hypertext Pre-Processor", (inicialmente se llamó Personal Home Page). Surgió en 1995, desarrollado por PHP Group.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php). (14)

**Ventajas:**

- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

**Desventajas:**

- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- Dificulta la modularización.
- Dificulta la organización por capas de la aplicación.

- Se necesita instalar un servidor web.

### 3.5.5 Lenguaje seleccionado. Python

De los lenguajes tratados anteriormente el 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.

## 4. Framework

El concepto framework viene aparejado con el desarrollo de software. Un framework es una estructura de soporte robusta y bien definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework 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.

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 se le puede añadir 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, por tanto, puede definirse 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, aplicaciones de publicación de documentos y o a la combinación de estos elementos.

La mayoría de los Frameworks Web incorporan en su arquitectura MVC, 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.

#### 4.1. Turbogears

TurboGears constituye un framework diseñado para desarrollar aplicaciones Web 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:

**MochiKit:** Una biblioteca JavaScript que incluye componentes tales como logging (del inglés), arrastrar (drag: del inglés), soltar (drop: del inglés), y otros efectos visuales que son comunes en muchas bibliotecas. Permite añadir comportamiento 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. Incluye 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 usa un parser basado en ElementTree, soporta herencia de plantillas y emparejado XSLT.

**CherryPy:** Un framework orientado a objetos que permite desarrollar aplicaciones web en Python. CherryPy 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.

**SQLAlchemy:** Es 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).

## 5. Dojo Toolkit

Dojo es un framework que contiene APIs y Widgets (controles) para facilitar el desarrollo de aplicaciones Web que utilicen tecnología AJAX. Contiene un sistema de empaquetado inteligente, abstracción de eventos, almacenamiento de APIs en el cliente, e interacción de APIs con AJAX. Resuelve asuntos de usabilidad comunes como pueden ser la navegación y detección del navegador, soportar cambios de URL en la barra de URLs para luego regresar a ellas y la habilidad de degradar cuando AJAX/JavaScript no es completamente soportado en el cliente. Es conocido como "la navaja suiza del ejército de las bibliotecas JavaScript". Proporciona una gama más amplia de opciones en una sola biblioteca JavaScript y es compatible con navegadores antiguos.

## 6. Plataforma

### 6.1 Aplicación Web

Una aplicación Web es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet. Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero. La facilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. Aplicaciones como los web mails, wikis o tiendas en línea son ejemplos bien conocidos de aplicaciones Web. Una ventaja principal es que deberían funcionar igual independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes para Windows, Mac OS X, GNU/Linux, y otros sistemas operativos, la aplicación es escrita una vez y es mostrada casi en todos lados.

#### **Ventajas:**

- Desarrollo barato, sencillo y rápido.
- Acceso ubicuo, sin necesidad de distribución, con pocos requerimientos técnicos.

- Datos centralizados y fácil integración de datos de múltiples fuentes.
- Permiten el desarrollo de comunidades que dan valor a las aplicaciones (software social).
- Una empresa puede migrar de sistema operativo o cambiar el Hardware libremente sin afectar el funcionamiento de las aplicaciones de servidor.
- No se requieren complicadas combinaciones de Hardware/Software para utilizar estas aplicaciones. Solo un computador con un buen navegador Web.
- Actualizar o hacer cambios en el Software es sencillo y sin riesgos de incompatibilidades. Existe solo una versión en el servidor lo que implica que no hay que distribuirla entre los demás computadores. El proceso es rápido y limpio.
- Se facilita el trabajo a distancia. Se puede trabajar desde cualquier PC o computador portátil con conexión a Internet o a una red interna o privada.
- Al funcionar en un navegador, se requiere un conocimiento básico de informática para utilizar una aplicación Web.

Todas estas ventajas dejan claro el potencial de las aplicaciones Web. La utilización de esta tecnología conlleva a reducir costos y complicaciones, y proporciona mayor libertad a la hora de realizar cualquier tipo de cambios. Las aplicaciones Web utilizan la arquitectura cliente-servidor que es una forma específica de diseño de aplicaciones, aunque también se conoce con este nombre a los ordenadores en los que estas aplicaciones son ejecutadas. Por un lado, el cliente es el ordenador que se encarga de efectuar una petición o solicitar un servicio. El cliente no posee control sobre los recursos, sino que es el servidor el encargado de manejarlos. Por otro lado, el ordenador remoto que actúa como servidor evalúa la petición del cliente y decide aceptarla o rechazarla consecuentemente. Una vez que el servidor acepta el pedido la información requerida es suministrada al cliente que efectuó la petición, siendo este último el responsable de proporcionar los datos al usuario con el formato adecuado. El cliente y el servidor no tienen que estar necesariamente en ordenadores separados, sino que pueden ser programas diferentes que se ejecuten en el mismo ordenador.

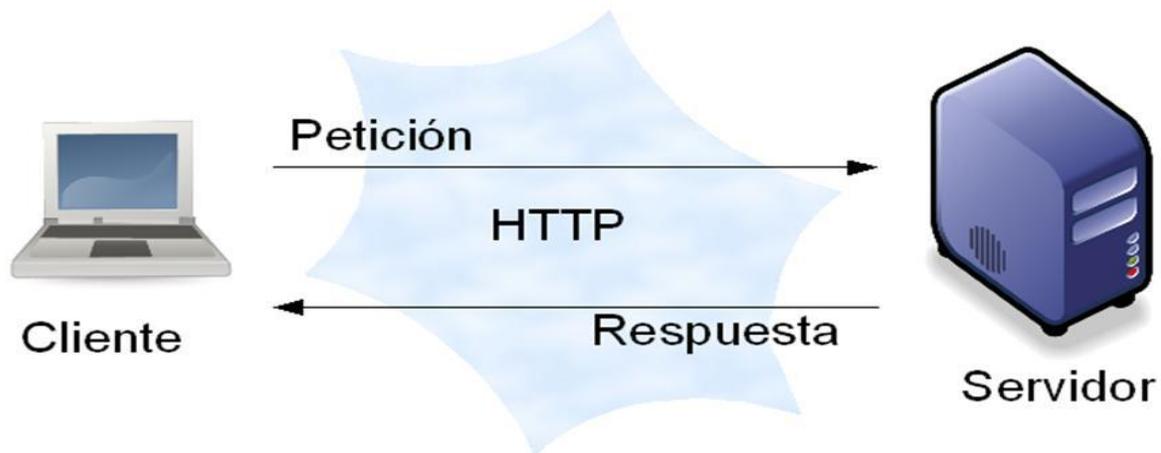
## 6.2 Modelo Cliente/Servidor

Se puede definir la computación 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 en 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. En un sistema distribuido cada máquina puede cumplir el uno u otro rol.

En otras palabras la arquitectura Cliente/Servidor es una extensión de programación modular en la que la base fundamental es separar una gran pieza de software en módulos con el fin de proveer usabilidad, flexibilidad, interoperabilidad y escalabilidad en las comunicaciones y hacer más fácil el desarrollo y mejorar su mantenimiento, cuenta con tres elementos fundamentales; cliente, servidor y red de comunicación.

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 ejecuten con la mayor eficiencia posible. Mediante esta arquitectura el usuario puede acceder a la información sin tener en cuenta su ubicación física y donde pueda estar alojada la misma.

La idea es tratar a una computadora como un instrumento, que por sí sola pueda realizar muchas tareas, pero con la consideración de que realice aquellas que son más adecuadas a sus características. Esto se aplica tanto a clientes como servidores, la forma más estándar de aplicación y uso de estos sistemas es mediante la explotación de las PC's a través de interfaces gráficas de usuario; mientras que la administración de datos y su seguridad e integridad se deja a cargo de computadoras centrales. Usualmente la mayoría del trabajo pesado se hace en el proceso llamado servidor y el o los procesos cliente 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.



**Figura.1 Modelo Cliente/Servidor**

- El servidor presenta a todos sus clientes una interface única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interface 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.

## 7. Conclusiones

En este capítulo se ha hecho un análisis de lenguajes, metodologías y herramientas propuestos para darle solución al sistema siendo elegidos los mejores candidatos para darle solución a la propuesta, después de un profundo análisis se ha llegado a la conclusión que es importante realizar una aplicación Web teniendo en cuenta la facilidad de acceso a la misma desde distintas partes. Es de vital importancia el dominio de estas herramientas seleccionadas y analizar la mejor forma de aplicarlas, para desarrollar un sistema de máxima calidad que cumpla con los requisitos propuestos y de al cliente una versión del producto que satisfaga sus intereses.

## Capítulo 2

### Características del Sistema

#### Introducción

En este capítulo se realizará un análisis de las características del sistema a desarrollar, se hará hincapié en el objeto de automatización, la arquitectura del sistema propuesto y las interfaces de usuario que integran el mismo.

#### 1. Descripción del proceso de Reporte y Facturación en Asterisk PBX

Al ser efectuada una llamada a través de Asterisk PBX se almacenan los datos de esta en un fichero llamado CDR (Call Detail Record). Este fichero brinda toda la información necesaria para confeccionar los reportes, así como los datos para facturar las llamadas. La factura se lleva a cabo después de hacer coincidir el destino de la llamada con el código de la zona que le corresponde, calculando de esta forma el importe para esta llamada. La ubicación del CDR en Asterisk PBX se encuentra configurada en un fichero llamado `cdr_pgsql.conf`.

#### 2. Objeto de Automatización

El proceso de configuración del sistema de reporte y facturación de Asterisk PBX, así como crear reportes con distintos criterios, y la facturación de las llamadas que sean seleccionadas por el usuario.

### **3. Información que se maneja**

La información que se maneja es la brindada el servidor de Asterisk PBX, ya sea a través de los ficheros de configuración o la generada por el mismo.

### **4. Propuesta del Sistema**

El sistema tendrá como principal objetivo brindarle al usuario toda la información que necesite con respecto a cada llamada efectuada en la plataforma, así como la opción de facturar las mismas. Dentro de los reportes que se obtendrán se encuentran los estadísticos o evolutivos: los cuales mostrarán gráficas estadísticas de la evolución de las llamadas en la, reportes por criterio: estos mostrarán la información de las llamadas que el usuario seleccione mediante un criterio de búsqueda introducido por este, y los predefinidos: creados para facilitar el trabajo evitando que el usuario tenga que introducir la información reiteradamente. El sistema también se encargará de facilitar la configuración de los ficheros que conciernen a los reportes y la facturación dentro de la plataforma.

## 5. Arquitectura Propuesta

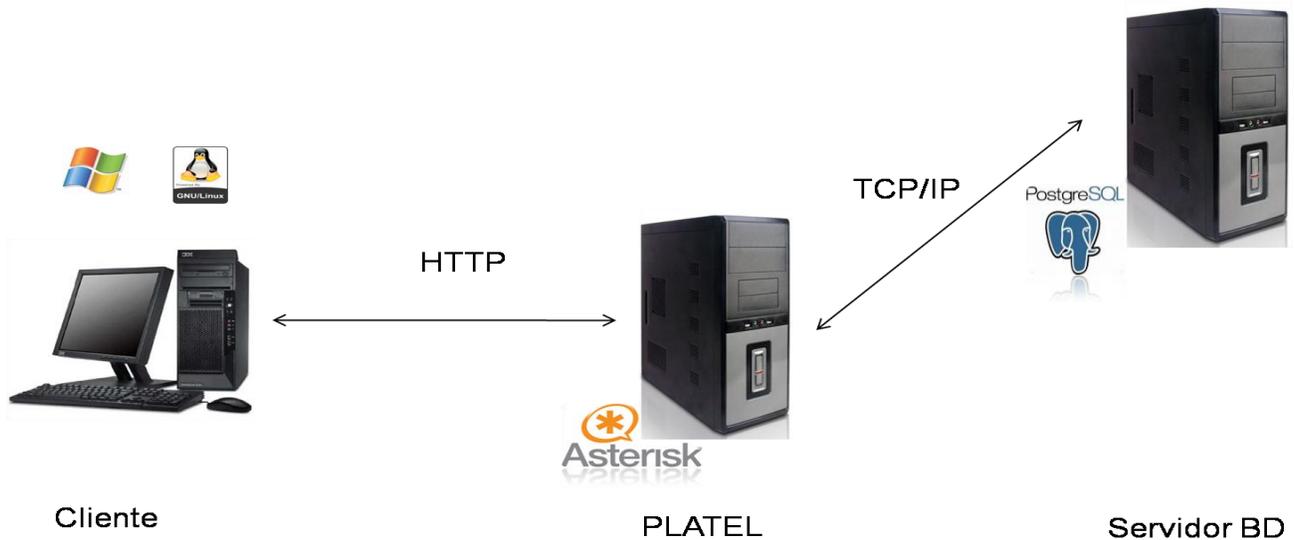


Figura. 2 Arquitectura Propuesta

## 6. Interfaz de usuario

### 6.1 Configurar sistema de reportes

Interfaz mediante la cual el usuario del sistema configura los reportes, los parámetros que serán configurados son los siguientes: ubicación de los reportes, estos se localizan en una base de datos, se configura la ubicación de esta así como los parámetros a mostrar en las tablas de reportería que serán seleccionados según la necesidad del usuario, entre los cuales pueden estar: extensión fuente, extensión destino, factura, duración de la llamada, contexto, fecha y hora en que fue realizada entre otros, contará con la opción de poder guardar su configuración.

## **6.2 Configurar sistema de facturas**

Interfaz mediante la cual el usuario del sistema configura todo lo concerniente a la factura, contará con la posibilidad de agregar o eliminar las zonas, y los parámetros a ser configurados son los siguientes: la zona del destino, el código del destino y las tarifas para facturar las llamadas.

## **6.3 Reportes predefinidos**

Interfaz donde el usuario contará con la opción de escoger entre una serie de reportes predefinidos el que desea mostrar, también se dará la posibilidad de ingresar un nuevo reporte predefinido según las necesidades del usuario. Esta interfaz dará la posibilidad de imprimir o guardar el reporte mostrado.

## **6.4 Reportes por criterio de búsqueda**

Interfaz donde el usuario puede mediante un criterio confeccionar un reporte. Entre los criterios se encuentran: extensión fuente, extensión destino, fecha de la(s) llamada(s), hora de la(s) llamada(s), duración y costo. Esta interfaz dará la posibilidad de imprimir o guardar el reporte mostrado.

## **6.5 Reportes estadísticos evolutivos**

Interfaz donde el usuario puede seleccionar entre varios criterios para obtener un reporte estadístico compuesto por datos y gráficas ilustrativas donde se mostrará el comportamiento de la cantidad de llamadas en la plataforma. Entre las gráficas que se mostrarán se encuentra: las horas de un día seleccionado, los días de un mes seleccionado, los meses del año en curso y el contexto del total de llamadas gestionadas en la plataforma. Esta interfaz dará la posibilidad de imprimir o guardar el reporte mostrado.

## **7. Requisitos no funcionales del sistema.**

### **7.1 Requerimientos de apariencia o interfaz externa**

La aplicación propuesta podrá ser usada por personas que tengan conocimientos básicos de informática, la interfaz debe ser amigable y fácil de usar, de manera que no sea una dificultad para los usuarios el trabajo con la misma.

### **7.2 Requerimientos de usabilidad**

A los administradores del sistema se les dará un adiestramiento básico en el uso de la aplicación. Estas personas tendrán un nivel de acceso amplio en la aplicación para poder darle respuesta a cada incidente ocurrido.

### **7.3 Requerimientos de rendimiento**

La eficiencia del producto estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo cliente/servidor, la rapidez de las consultas de la base de datos y la velocidad del servidor Asterisk.

#### 7.4 Requerimiento de portabilidad

Las herramientas utilizadas podrán ser usadas bajo cualquier sistema operativo Windows NT en adelante o cualquier distribución de Linux. El servidor Web y el servidor de Base de Datos pueden estar en la misma PC sin ocasionar problema alguno.

#### 7.5 Requerimientos de seguridad

**Confiabilidad:** La información manejada por el sistema debe estar protegida de acceso no autorizado.

**Integridad:** La información manejada por el sistema debe ser objeto de cuidadosa protección contra la corrupción y estados de inconsistencia.

**Disponibilidad:** La aplicación deberá estar disponible en todo momento para aquellas personas con acceso a la información, y los mecanismos utilizados para lograr la seguridad, no deben ser un obstáculo.

#### 7.6 Requerimientos de hardware

En el cliente se requiere una máquina con 256 MB de RAM como mínimo, un microprocesador Pentium 4 a más de 1.6 GHz y un disco duro de más de 10 GB, el servidor web junto con el servidor de base de datos debe tener al menos 512 MB de RAM y 20 GB de disco duro mínimo, todas las máquinas implicadas en la funcionalidad de la aplicación deben estar conectadas a la red de al menos 100 Mbps de velocidad.

## **8. Restricción en el diseño y la implementación**

Se debe realizar una aplicación que permita dar respuesta en el menor tiempo posible. Para garantizar el desarrollo de la aplicación se utilizarán las metodologías XP y Scrum.

## **9. Conclusiones**

Tanto las características del sistema a desarrollar, la arquitectura propuesta, las interfaces que integran al mismo, como los distintos requerimientos del software, darán como resultado un sistema amigable, adaptable y fácil de usar, la arquitectura propuesta nos muestra que una empresa de poco presupuesto puede adquirir el hardware requerido para la instalación de la plataforma.

## Capítulo 3

### Exploración, Planificación de la entrega de iteraciones

#### Introducción

La Programación Extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Esta metodología la definen 6 fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

#### 1. Fase de exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

#### 2. Historias de Usuario

Las Historias de Usuario son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy

dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

Durante esta fase, para el desarrollo del sistema se elaboraron cinco Historias de Usuario: *Reportes por Criterio de búsqueda*. (Tabla 1), *Reportes Predefinidos* (Tabla 2), *Reportes Estadísticos Evolutivos por rango de Tiempo* (Tabla 3), *Reportes Estadísticos Evolutivos por comparación* (Tabla 4), *Configuración del Sistema de Reportes* (Tabla 5), *Configuración del Sistema de Facturas* (Tabla 6), a continuación se explican cada una de ellas:

Historia de Usuario	
<b>Número:</b> 1	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Reportes por Criterio de búsqueda.	
<b>Prioridad en el Negocio:</b> Alta	<b>Riesgo en Desarrollo:</b> Alto
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador(es) responsables:</b> Luis E. Acosta Aparicio, Nicolás Montero Puñales	
<b>Descripción:</b> El usuario obtiene un reporte una vez que selecciona el criterio deseado. El criterio puede ser: por extensión, fecha, duración o costo. Los datos obtenidos son mostrados en una tabla.	
<b>Observaciones:</b>	
<b>Firma:</b>	

**Tabla 1:** Historia de Usuario Realizar Reportes por Criterio de búsqueda.

Historia de Usuario	
<b>Número:</b> 2	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Reportes Predefinidos.	
<b>Prioridad en el Negocio:</b> Alta	<b>Riesgo en Desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 1
<b>Programador(es) responsables:</b> Luis E. Acosta Aparicio, Nicolás Montero Puñales	
<b>Descripción:</b> El usuario selecciona un reporte predefinido. Los datos de la búsqueda realizada son mostrados en una tabla. Inicialmente el sistema contiene el reporte diario y el reporte mensual como Reportes Predefinidos, pero también el usuario tiene la posibilidad de crear su propio Reporte Predefinido.	
<b>Observaciones:</b> En el caso del reporte diario y mensual por defecto tienen el valor del día y mes actual respectivamente, pero también se puede seleccionar otro día o mes.	
<b>Firma:</b>	

**Tabla 2:** Historia de Usuario Reportes Predefinidos.

Historia de Usuario	
<b>Número:</b> 3	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Reportes Estadísticos Evolutivos por rango de tiempo.	
<b>Prioridad en el Negocio:</b> Media	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 2
<b>Programador(es) responsables:</b> Luis E. Acosta Aparicio, Nicolás Montero Puñales	
<b>Descripción:</b> El usuario seleccionará un período de tiempo y obtendrá un reporte donde se muestra el comportamiento de parámetros como son: la cantidad y distribución de las llamadas realizadas en este período. El reporte	

estará compuesto por datos y gráficas donde se ilustra dicho comportamiento.
<b>Observaciones:</b>
<b>Firma:</b>

**Tabla 3:** Historia de Usuario Reportes Estadísticos Evolutivos por rango de tiempo.

Historia de Usuario	
<b>Número:</b> 4	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Reportes Estadísticos Evolutivos por comparación.	
<b>Prioridad en el Negocio:</b> Media	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 2
<b>Programador(es) responsables:</b> Luis E. Acosta Aparicio, Nicolás Montero Puñales	
<b>Descripción:</b> El usuario obtendrá un reporte, en el cual se mostrará una comparación entre meses, días y horas. El reporte obtenido muestra una comparación del comportamiento de las llamadas. Esta comparación estará compuesta por datos y gráficos que ilustran dicho comportamiento.	
<b>Observaciones:</b>	
<b>Firma:</b>	

**Tabla 4:** Historia de Usuario Realizar Reportes de Registros de llamadas diario.

Historia de Usuario	
<b>Número:</b> 5	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Configuración del Sistema de Reportes.	
<b>Prioridad en el Negocio:</b> Baja	<b>Riesgo en Desarrollo:</b> Bajo

<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 3
<b>Programador(es) responsables:</b> Luis E. Acosta Aparicio, Nicolás Montero Puñales	
<b>Descripción:</b> El usuario puede personalizar los reportes, seleccionando los datos que serán mostrados en las tablas de reportes predeterminados y reportes por criterio de búsqueda, además contara con la opción de salvar sus cambios.	
<b>Observaciones:</b>	
<b>Firma:</b>	

**Tabla 5:** Historia de Usuario Realizar Configuración del Sistema de Reportes.

Historia de Usuario	
<b>Número:</b> 6	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Configuración del Sistema de Facturas.	
<b>Prioridad en el Negocio:</b> Baja	<b>Riesgo en Desarrollo:</b> Bajo
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 3
<b>Programador(es) responsables:</b> Luis E. Acosta Aparicio, Nicolás Montero Puñales	
<b>Descripción:</b> El usuario ingresará la zona del destino, el código de la provincia, y el importe de la zona. Contará con la posibilidad de eliminar y agregar nuevas zonas.	
<b>Observaciones:</b>	
<b>Firma:</b>	

**Tabla 6:** Configuración del Sistema de Facturas.

### 3. Planificación

En la fase de planificación se establece por parte del cliente la prioridad de cada Historia de Usuario y los programadores obtienen la estimación del esfuerzo necesario. La medida para calcular el esfuerzo necesario es el *punto*. Un punto, equivale a una semana ideal de programación, generalmente una historia no excede los 3 puntos.

En esta fase se obtiene el número de iteraciones necesarias para la implementación del producto.

#### 3.1 Estimación de esfuerzo por Historia de Usuario

Se hizo un estudio con el objetivo de estimar el esfuerzo necesario para realizar cada una de las Historias de Usuario, los cuales se reflejan en la tabla que se muestra a continuación (**Tabla 7**):

Historia de Usuario	Puntos Estimados
<i>Reportes por Criterio de búsqueda.</i>	1
<i>Reportes Predefinidos</i>	2
<i>Reportes Estadísticos Evolutivos por rango de Tiempo</i>	2
<i>Reportes Estadísticos Evolutivos por comparación</i>	2
<i>Configuración del Sistema de Reportes</i>	1
<i>Configuración del Sistema de Facturas</i>	1

**Tabla 7:** Estimación de esfuerzos por Historia de Usuario.

## 4. Iteraciones

Se decidió dividir el trabajo en tres iteraciones, las cuales se detallan a continuación:

**Primera Iteración:** Se implementa la Historia de Usuario número uno y dos, o sea las de alta prioridad. Al concluir se contará con una primera versión (1.0) que será mostrada al cliente como versión de prueba, para comprobar si satisface sus expectativas.

**Segunda Iteración:** En esta iteración se implementa la Historia de Usuario número tres y cuatro, de prioridad media, al terminar dicha iteración se contará con una nueva versión del producto (1.1), a la cual se le realizarán pruebas para verificar si cumple con las funcionalidades requeridas. El cliente observará dichas pruebas con el objetivo de realizar cambios en el sistema en proceso de desarrollo, si lo opina conveniente.

**Tercera Iteración:** En esta iteración se implementa la Historia de Usuario número cinco y seis de prioridad baja, al finalizarla se contará con la primera versión del producto final (2.0) y se le harán pruebas al sistema completo para definir si cumple con todos los requerimientos necesarios.

### 4.1 Plan de duración de las Iteraciones

Como parte del ciclo de vida de cualquier proyecto que haga uso de XP se crea el plan de duración de las iteraciones, mostrando: las iteraciones involucradas en el proceso de desarrollo, las Historias de Usuario a desarrollar en cada iteración y el tiempo de duración estimado para cada una. El orden que poseen las Historias de Usuarios en la **Tabla 8**, será el orden a seguir para la implementación de las mismas. El tiempo de duración estimado de cada iteración se muestra en la **Tabla 9**.

<b>Iteraciones</b>	<b>Orden de las Historias Usuario a implementar</b>	<b>Duración de la estimación</b>
<b>Iteración 1</b>	<i>Reportes por Criterio de búsqueda.</i>	<b>1 semana</b>
<b>Iteración 1</b>	<i>Reportes Predefinidos</i>	<b>2 semanas</b>
<b>Iteración 2</b>	<i>Reportes Estadísticos Evolutivos por rango de Tiempo</i>	<b>2 semanas</b>
<b>Iteración 2</b>	<i>Reportes Estadísticos Evolutivos por comparación</i>	<b>2 semanas</b>
<b>Iteración 3</b>	<i>Configuración del Sistema de Reportes</i>	<b>1 semana</b>
<b>Iteración 3</b>	<i>Configuración del Sistema de Facturas</i>	<b>1 semana</b>

**Tabla 8:** Plan de duración de las iteraciones.

<b>Iteraciones</b>	<b>Duración de la estimación</b>
Iteración 1	3 semanas
Iteración 2	4 semanas
Iteración 3	2 semanas

**Tabla 9:** Duración total de las iteraciones.

## 4.2 Plan de Entrega

<b>Iteraciones</b>	<b>Fecha de Finalización</b>	<b>Versión del Producto</b>
1ra Iteración	3ra Semana de Marzo 2009	1.0
2da Iteración	4ta Semana de Abril 2009	1.1
3ra Iteración	3ra Semana de Mayo 2009	2.0

**Tabla 10:** Duración total de las iteraciones.

## 5. Conclusiones

En este capítulo se desarrollaron las fases de Exploración, Planificación de la Entrega e Iteraciones, se generaron las Historias de Usuarios necesarias para el correcto desarrollo del sistema, se organizó el trabajo por iteraciones, se estimó el tiempo necesario para desarrollar el producto y se elaboró un plan de entrega. Al finalizar la última iteración descrita en este capítulo, el sistema estará listo para entrar en producción.

## Capítulo 4

### Iteraciones, Producción y Pruebas

#### Introducción

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. Se procederá a abordar las Historias de Usuarios definidas en cada Iteración. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores. Las iteraciones quedan expuestas a cambios, luego de cada iteración, se hará una revisión de las mismas y de ser necesario se realizarán los cambios correspondientes, al final de cada iteración debe quedar listo un producto para el cliente. 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. Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior.

#### 1. Primera Iteración

En la primera iteración se abordarán las Historias de Usuarios de alta prioridad en el negocio, el equipo de programadores decidirá las tareas para llevar a cabo la implementación de las Historias de Usuarios.

Historia de Usuario	Puntos Estimados	Real
Reportes por Criterio de búsqueda	1	1
Reportes Predefinidos	2	2

<b>Total</b>	3	3
--------------	---	---

**Tabla 11:** Historias de Usuarios realizadas en la primera iteración.

### 1.1 Tareas de Historia de Usuario realizadas durante la primera iteración

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Diseño de la interfaz para realizar reportes por criterio de búsqueda	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.6
<b>Fecha Inicio:</b> 7/03/2009	<b>Fecha Fin:</b> 9/02/2009
<b>Programador Responsable:</b> Luis E. Acosta Aparicio	
<b>Descripción:</b> Se diseña la interfaz que permite la realización de los reportes por criterio de búsqueda	

**Tabla 12:** Tarea de Ingeniería para la Historia de Usuario Reportes por Criterio de búsqueda.

Tarea de Ingeniería	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Validación de datos y peticiones al servidor	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 9/03/2009	<b>Fecha Fin:</b> 10/03/2009

<b>Programador Responsable:</b> Nicolás Montero Puñales
<b>Descripción:</b> Validar los datos entrados por el usuario con el objetivo de ahorrar peticiones innecesarias al servidor, si estos son válidos se realiza la petición al servidor y se maneja la respuesta de este.

**Tabla 13:** Tarea de Ingeniería para la Historia de Usuario Reportes por Criterio de búsqueda.

Tarea de Ingeniería	
<b>Número Tarea:</b> 3	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Implementación del controlador para criterios de búsqueda	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 11/03/2009	<b>Fecha Fin:</b> 13/03/2009
<b>Programador Responsable:</b> Nicolás Montero Puñales	
<b>Descripción:</b> Se implementa el controlador en el servidor para devolverle al cliente los datos asociados al criterio de búsqueda seleccionado.	

**Tabla 14:** Tarea de Ingeniería para la Historia de Usuario Reportes por Criterio de búsqueda.

Tarea de Ingeniería	
<b>Número Tarea:</b> 4	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Diseño de la interfaz para realizar reportes predefinidos	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 16/03/2009	<b>Fecha Fin:</b> 18/03/2009

<b>Programador Responsable:</b> Luis Eduardo Acosta Aparicio
<b>Descripción:</b> Se diseña la interfaz que permite la realización de los reportes predefinidos.

**Tabla 15:** Tare de Ingeniería para la Historia de Usuario Reportes Predefinidos.

Tarea de Ingeniería	
<b>Número Tarea:</b> 5	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Validación de datos y peticiones al servidor	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.6
<b>Fecha Inicio:</b> 18/03/2009	<b>Fecha Fin:</b> 20/03/2009
<b>Programador Responsable:</b> Nicolás Montero Puñales	
<b>Descripción:</b> Validar los datos entrados por el usuario con el objetivo de ahorrar peticiones innecesarias al servidor, si estos son válidos, se realiza la petición al servidor y se maneja la respuesta de este.	

**Tabla 16:** Tarea de Ingeniería para la Historia de Usuario Reportes Predefinidos.

Tarea de Ingeniería	
<b>Número Tarea:</b> 6	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Implementación del controlador para reportes predefinidos	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.8
<b>Fecha Inicio:</b> 23/03/2009	<b>Fecha Fin:</b> 27/03/2009
<b>Programador Responsable:</b> Nicolás Montero Puñales	

**Descripción:** Se implementa el controlador en el servidor para devolverle al cliente los datos asociados al reporte predefinido seleccionado.

**Tabla 17:** Tarea de Ingeniería para la Historia de Usuario Reportes Predefinidos.

## 2. Segunda Iteración

En la segunda iteración se abordarán las Historias de Usuarios de media prioridad en el negocio, el equipo de programadores decidirán las tareas para una llevar a cabo la implementación de las Historias de Usuarios.

Historia de Usuario	Puntos Estimados	Real
Reportes Estadísticos Evolutivos por rango de tiempo	2	2
Reportes Estadísticos Evolutivos por comparación	2	2
<b>Total</b>	4	4

**Tabla 18:** Historias de Usuarios realizadas en la segunda iteración.

### 2.1 Tareas de Historia de Usuario realizadas durante la segunda iteración

Tarea de Ingeniería	
<b>Número Tarea:</b> 7	<b>Número Historia de Usuario:</b> 3
<b>Nombre Tarea:</b> Diseño de la interfaz para los reportes evolutivos por rango de tiempo	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.6
<b>Fecha Inicio:</b> 6/04/2009	<b>Fecha Fin:</b> 9/04/2009

<b>Programador Responsable:</b> Luis Eduardo Acosta Aparicio
<b>Descripción:</b> Se diseña la interfaz que permite la realización de un reporte estadístico evolutivo por rango de tiempo.

**Tabla 19:** Tarea de Ingeniería para la Historia de Usuario Reportes Estadísticos Evolutivos por rango de tiempo.

Tarea de Ingeniería	
<b>Número Tarea:</b> 8	<b>Número Historia de Usuario:</b> 3
<b>Nombre Tarea:</b> Validación de datos y peticiones al servidor	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.6
<b>Fecha Inicio:</b> 9/04/2009	<b>Fecha Fin:</b> 13/04/2009
<b>Programador Responsable:</b> Nicolás Montero Puñales	
<b>Descripción:</b> Validar que los datos entrados por el usuario con el objetivo de ahorrar peticiones innecesarias al servidor, si estos son válidos se realizan la petición al servidor y se maneja la respuesta de este.	

**Tabla 20:** Tarea de Ingeniería para la Historia de Usuario Reportes Estadísticos Evolutivos por rango de tiempo.

Tarea de Ingeniería	
<b>Número Tarea:</b> 9	<b>Número Historia de Usuario:</b> 3

<b>Nombre Tarea:</b> Implementación del controlador para reportes estadísticos evolutivos por rango de tiempo	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.8
<b>Fecha Inicio:</b>	<b>Fecha Fin:</b>
<b>Programador Responsable:</b> Nicolás Montero Puñales	
<b>Descripción:</b> Se implementa el controlador en el servidor para devolverle al cliente los datos asociados reporte predefinido seleccionado.	

**Tabla 21:** Tarea de Ingeniería para la Historia de Usuario Reportes Estadísticos Evolutivos por comparación.

Tarea de Ingeniería	
<b>Número Tarea:</b> 10	<b>Número Historia de Usuario:</b> 4
<b>Nombre Tarea:</b> Diseño de la interfaz para los reportes estadísticos evolutivos por comparación	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.6
<b>Fecha Inicio:</b> 20/04/2009	<b>Fecha Fin:</b> 23/04/2009
<b>Programador Responsable:</b> Luis Eduardo Acosta Aparicio	
<b>Descripción:</b> Se diseña la interfaz que permite la realización de un reporte estadístico evolutivo por comparación.	

**Tabla 22:** Tarea de Ingeniería para la Historia de Usuario Reportes Estadísticos Evolutivos por comparación.

Tarea de Ingeniería	
<b>Número Tarea:</b> 11	<b>Número Historia de Usuario:</b> 4
<b>Nombre Tarea:</b> Validación de datos y peticiones al servidor	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.6
<b>Fecha Inicio:</b> 24/04/2009	<b>Fecha Fin:</b> 28/04/2009
<b>Programador Responsable:</b> Nicolás Montero Puñales	
<b>Descripción:</b> Validar los datos entrados por el usuario con el objetivo de ahorrar peticiones innecesarias al servidor, si estos son válidos se realiza la petición al servidor y se maneja la respuesta de este.	

**Tabla 23:** Tarea de Ingeniería para la Historia de Usuario Reportes Estadísticos Evolutivos por comparación.

Tarea de Ingeniería	
<b>Número Tarea:</b> 12	<b>Número Historia de Usuario:</b> 4
<b>Nombre Tarea:</b> Implementación del controlador para reportes estadísticos evolutivos comparación	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.8
<b>Fecha Inicio:</b> 29/04/2009	<b>Fecha Fin:</b> 4/05/2009
<b>Programador Responsable:</b> Nicolás Montero Puñales	

**Descripción:**

Se implementa el controlador en el servidor para devolverle al cliente los datos asociados reporte predefinido seleccionado.

**Tabla 24:** Tarea de Ingeniería para la Historia de Usuario Reportes Estadísticos Evolutivos por comparación.

**3. Tercera Iteración**

En la tercera iteración se abordarán las Historias de Usuarios de baja prioridad en el negocio, el equipo de programadores decidirán las tareas para una llevar a cabo la implementación de las Historias de Usuarios perteneciente a la iteración.

Historia de Usuario	Puntos Estimados	Real
Configuración del Sistema de Reportes	1	1
Configuración del Sistema de Facturas	1	1
<b>Total</b>	2	2

**Tabla 25:** Historias de Usuarios realizadas en la tercera iteración.

**3.1 Tareas de Historia de Usuario realizadas durante la tercera iteración**

Tarea de Ingeniería	
<b>Número Tarea:</b> 13	<b>Número Historia de Usuario:</b> 5
<b>Nombre Tarea:</b> Diseño de la interfaz para la configuración del sistema de reportes	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.6

<b>Fecha Inicio:</b> 6/05/2009	<b>Fecha Fin:</b> 11/05/2009
<b>Programador Responsable:</b> Luis Eduardo Acosta Aparicio	
<b>Descripción:</b> Se diseña la interfaz que permite la configuración del sistema de reportes.	

**Tabla 26:** Tarea de Ingeniería para la Historia de Usuario Configuración del Sistema de Reportes.

Tarea de Ingeniería	
<b>Número Tarea:</b> 14	<b>Número Historia de Usuario:</b> 5
<b>Nombre Tarea:</b> Validación de datos y peticiones al servidor	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.6
<b>Fecha Inicio:</b> 5/05/2009	<b>Fecha Fin:</b> 7/05/2009
<b>Programador Responsable:</b> Nicolás Montero Puñales	
<b>Descripción:</b> Validar los datos entrados por el usuario con el objetivo de ahorrar peticiones innecesarias al servidor, si estos son válidos se realiza la petición al servidor y se maneja la respuesta de este.	

**Tabla 27:** Tarea de Ingeniería para la Historia de Usuario Configuración del Sistema de Reportes.

Tarea de Ingeniería	
<b>Número Tarea:</b> 15	<b>Número Historia de Usuario:</b> 5
<b>Nombre Tarea:</b> Implementación del controlador para la configuración de sistema de reportes	

<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.8
<b>Fecha Inicio:</b> 8/05/2009	<b>Fecha Fin:</b> 15/05/2009
<b>Programador Responsable:</b> Nicolás Montero Puñales	
<b>Descripción:</b> Se implementa el controlador que permita la configuración de los parámetros necesarios para el trabajo con los reportes.	

**Tabla 28:** Tarea de Ingeniería para la Historia de Usuario Configuración del Sistema de Reportes.

Tarea de Ingeniería	
<b>Número Tarea:</b> 16	<b>Número Historia de Usuario:</b> 6
<b>Nombre Tarea:</b> Diseño de la interfaz para la configuración del sistema de facturas	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.6
<b>Fecha Inicio:</b> 12/05/2009	<b>Fecha Fin:</b> 15/05/2009
<b>Programador Responsable:</b> Luis Eduardo Acosta Aparicio	
<b>Descripción:</b> Se diseña la interfaz que permite la configuración del sistema de facturas.	

**Tabla 29:** Tarea de Ingeniería para la Historia de Usuario Configuración del Sistema de Facturas.

Tarea de Ingeniería	
<b>Número Tarea:</b> 17	<b>Número Historia de Usuario:</b> 5
<b>Nombre Tarea:</b> Validación de datos y peticiones al servidor	

<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.6
<b>Fecha Inicio:</b> 18/05/2009	<b>Fecha Fin:</b> 22/05/2009
<b>Programador Responsable:</b> Nicolás Montero Puñales	
<b>Descripción:</b> Validar los datos entrados por el usuario con el objetivo de ahorrar peticiones innecesarias al servidor, si estos son válidos se realiza la petición al servidor y se maneja la respuesta de este.	

**Tabla 30:** Tarea de Ingeniería para la Historia de Usuario Configuración del Sistema de Facturas.

Tarea de Ingeniería	
<b>Número Tarea:</b> 18	<b>Número Historia de Usuario:</b> 6
<b>Nombre Tarea:</b> Implementación del controlador para la configuración del sistema de facturas	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.8
<b>Fecha Inicio:</b> 25/05/2009	<b>Fecha Fin:</b> 29/05/2009
<b>Programador Responsable:</b> Nicolás Montero Puñales	
<b>Descripción:</b> Se implementa el controlador que permita la configuración de las facturas que se llevaran a cabo en el sistema.	

**Tabla 31:** Tarea de Ingeniería para la Historia de Usuario Configuración del Sistema de Facturas.

#### 4. Pruebas de Aceptación

Caso de Prueba de Aceptación
------------------------------

<b>Código:</b> HU1_P1	<b>Historia de usuario:</b> Reportes por Criterio de búsqueda.
<b>Nombre:</b> Obtención de un reporte por criterio de búsqueda.	
<b>Descripción:</b> Probar que se genera el reporte de forma correcta.	
<b>Condiciones de ejecución:</b> El sistema se comporta de forma correcta mostrando los datos del reporte deben corresponderse con la entrada.	
<b>Entrada / Pasos de ejecución:</b> Se introduce un criterio de búsqueda valido, compuesto por uno o varios de los siguientes parámetros: extensión que realizó la llamada, extensión a la cual se le realizó, fecha y hora en que se realizó, costo y tiempo de la llamada.	
<b>Resultado esperado:</b> Se obtiene un reporte que contiene los datos de las llamadas realizadas correspondientes con el criterio introducido.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 32:** Caso de Prueba de Aceptación HU1\_P1.

Caso de Prueba de Aceptación	
<b>Código:</b> HU1_P2	<b>Historia de usuario:</b> Reportes por Criterio de búsqueda.
<b>Nombre:</b> Obtención de un reporte con un criterio de búsqueda incorrecto.	
<b>Descripción:</b> Probar que la validación de los datos se hizo de forma correcta.	
<b>Condiciones de ejecución:</b> El sistema detecta la falla en la entrada mostrando una alerta.	
<b>Entrada / Pasos de ejecución:</b> Se introduce un criterio de búsqueda incorrecto, se trata de generar el reporte con esta entrada.	
<b>Resultado esperado:</b> Se obtiene mensaje de error alertando al usuario, en este se le dará información sobre los campos incorrectos.	

**Evaluación de la prueba:** Satisfactoria.

**Tabla 33:** Caso de Prueba de Aceptación HU1\_P2.

Caso de Prueba de Aceptación	
<b>Código:</b> HU2_P1	<b>Historia de usuario:</b> Reportes Predefinidos.
<b>Nombre:</b> Obtención de un reporte predefinido.	
<b>Descripción:</b> Probar que se genera el reporte de forma correcta.	
<b>Condiciones de ejecución:</b> El sistema se comporta de forma correcta, mostrando los datos asociados al reporte seleccionado.	
<b>Entrada / Pasos de ejecución:</b> Es seleccionado uno de los reportes predefinidos que se encuentran en una lista de estos.	
<b>Resultado esperado:</b> Se obtiene un reporte un reporte que contiene los datos de las llamadas realizadas que corresponden con la entrada.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 34:** Caso de Prueba de Aceptación HU2\_P1.

Caso de Prueba de Aceptación	
<b>Código:</b> HU2_P2	<b>Historia de usuario:</b> Reportes Predefinidos.
<b>Nombre:</b> Obtención de un reporte predefinido inexistente.	
<b>Descripción:</b> Probar que solo sean seleccionados reportes predefinidos válidos.	
<b>Condiciones de ejecución:</b> El sistema detecta la falla en la entrada mostrando una alerta.	

<b>Entrada / Pasos de ejecución:</b> Se introduce un reporte predefinido que no coincide con ninguno en la lista de estos, se trata de generar el reporte con esta entrada.
<b>Resultado esperado:</b> Se obtiene mensaje de error indicando al usuario que debe seleccionar un reporte predefinido de la lista.
<b>Evaluación de la prueba:</b> Satisfactoria.

**Tabla 35:** Caso de Prueba de Aceptación HU2\_P2.

Caso de Prueba de Aceptación	
<b>Código:</b> HU3_P1	<b>Historia de usuario:</b> Reportes Estadísticos Evolutivos por rango de tiempo.
<b>Nombre:</b> Obtención de un reporte estadístico por rango de tiempo.	
<b>Descripción:</b> Probar que se genera el reporte de forma correcta.	
<b>Condiciones de ejecución:</b> El sistema se comporta de forma correcta mostrando los datos asociados al rango de tiempo seleccionado.	
<b>Entrada / Pasos de ejecución:</b> Se introduce un rango de tiempo y se intenta obtener el reporte.	
<b>Resultado esperado:</b> Se obtiene un reporte con los datos estadísticos correspondientes a las llamadas realizadas dentro del rango de tiempo.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 36:** Caso de Prueba de Aceptación HU3\_P1.

Caso de Prueba de Aceptación	
<b>Código:</b> HU3_P2	<b>Historia de usuario:</b> Reportes Estadísticos Evolutivos por rango de tiempo.

<b>Nombre:</b> Obtención de un reporte estadístico con un rango incorrecto.
<b>Descripción:</b> Probar que solo sean seleccionados rangos de tiempo correctos.
<b>Condiciones de ejecución:</b> El sistema detecta la falla en la entrada mostrando una alerta.
<b>Entrada / Pasos de ejecución:</b> Se introduce un rango de tiempo incorrecto y se intenta obtener el reporte.
<b>Resultado esperado:</b> Se obtiene mensaje de error indicando al usuario que debe seleccionar un rango de tiempo correcto.
<b>Evaluación de la prueba:</b> Satisfactoria.

**Tabla 37:** Caso de Prueba de Aceptación HU3\_P2.

Caso de Prueba de Aceptación	
<b>Código:</b> HU4_P1	<b>Historia de usuario:</b> Reportes Estadísticos Evolutivos por comparación.
<b>Nombre:</b> Obtención de un reporte estadístico por comparación.	
<b>Descripción:</b> Probar que se genera el reporte de forma correcta.	
<b>Condiciones de ejecución:</b> El sistema se comporta de forma correcta mostrando los datos asociados a la comparación realizada.	
<b>Entrada / Pasos de ejecución:</b> Se selecciona el criterio a comparar, este criterio puede estar conformado por dos o más días, dos o más meses, dos o más extensiones.	
<b>Resultado esperado:</b> Se obtiene un reporte con los datos estadísticos correspondientes a la comparación realizada.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 38:** Caso de Prueba de Aceptación HU4\_P1.

Caso de Prueba de Aceptación	
<b>Código:</b> HU4_P2	<b>Historia de usuario:</b> Reportes Estadísticos Evolutivos por comparación.
<b>Nombre:</b> Obtención de un reporte estadístico con un criterio a comparar incorrecto.	
<b>Descripción:</b> Probar que se realizó correctamente la validación de los datos.	
<b>Condiciones de ejecución:</b> El sistema detecta la falla en la entrada mostrando una alerta.	
<b>Entrada / Pasos de ejecución:</b> Se introduce un criterio a comparar incorrecto y se intenta obtener el reporte.	
<b>Resultado esperado:</b> Se obtiene mensaje de error indicando al usuario que debe seleccionar un criterio a comparar correcto así como información de los datos incorrectos introducidos.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 39:** Caso de Prueba de Aceptación HU4\_P2.

Caso de Prueba de Aceptación	
<b>Código:</b> HU5_P1	<b>Historia de usuario:</b> Configuración del Sistema de Reportes.
<b>Nombre:</b> Configurar el sistema de reportes de forma satisfactoria.	
<b>Descripción:</b> Probar que la configuración de los parámetros necesarios es la correcta para el funcionamiento del sistema de reportes.	
<b>Condiciones de ejecución:</b> El sistema se comporta de forma correcta configurando los parámetros del sistema de reportes.	
<b>Entrada / Pasos de ejecución:</b> Se introducen los datos necesarios para configurar el sistema de reportes tales como: Servidor de base de datos donde se encuentra el Registro Detallado de Llamadas, datos que serán mostrados en las tablas.	
<b>Resultado esperado:</b> Se configura de forma correcta el sistema de reportes.	

**Evaluación de la prueba:** Satisfactoria.

**Tabla 40:** Caso de Prueba de Aceptación HU5\_P1.

Caso de Prueba de Aceptación	
<b>Código:</b> HU5_P2	<b>Historia de usuario:</b> Configuración del Sistema de Reportes.
<b>Nombre:</b> Configurar el sistema de reportes con parámetros incorrectos.	
<b>Descripción:</b> Probar que se realizó correctamente la validación de los parámetros del sistema de reportes.	
<b>Condiciones de ejecución:</b> El sistema detecta la falla en la entrada mostrando una alerta.	
<b>Entrada / Pasos de ejecución:</b> Se introducen parámetros incorrectos y se intenta configurar el sistema de reportes.	
<b>Resultado esperado:</b> Se obtiene mensaje de error indicando al usuario que ha introducido valores incorrectos, se le da información sobre los parámetros que presentan problemas.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 41:** Caso de Prueba de Aceptación HU5\_P2.

Caso de Prueba de Aceptación	
<b>Código:</b> HU6_P1	<b>Historia de usuario:</b> Configuración del Sistema de Facturas.
<b>Nombre:</b> Configurar el sistema de factura de forma satisfactoria.	
<b>Descripción:</b> Probar que se configuran correctamente los parámetros necesarios para el correcto funcionamiento del sistema de factura.	

<b>Condiciones de ejecución:</b> El sistema se comporta de forma correcta configurando los parámetros del sistema de factura.
<b>Entrada / Pasos de ejecución:</b> Se introducen los datos necesarios para configurar el sistema de reportes tales como: tarifa de facturación, llamadas que serán facturadas.
<b>Resultado esperado:</b> Se configura de forma correcta el sistema de facturas.
<b>Evaluación de la prueba:</b> Satisfactoria.

**Tabla 42:** Caso de Prueba de Aceptación HU6\_P1.

Caso de Prueba de Aceptación	
<b>Código:</b> HU6_P2	<b>Historia de usuario:</b> Configuración del Sistema de Facturas.
<b>Nombre:</b> Configurar el sistema de factura con parámetros incorrectos.	
<b>Descripción:</b> Probar que se realizó correctamente la validación de los parámetros del sistema de factura.	
<b>Condiciones de ejecución:</b> El sistema detecta la falla en la entrada mostrando una alerta.	
<b>Entrada / Pasos de ejecución:</b> Se introducen parámetros incorrectos y se intenta configurar el sistema de factura.	
<b>Resultado esperado:</b> Se obtiene mensaje de error indicando al usuario que ha introducido valores incorrectos, se le da información sobre los parámetros que presentan problemas.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 43:** Caso de Prueba de Aceptación HU6\_P2.

## 5. Conclusiones

Como resultado de las Tareas de Ingeniería quedará listo un sistema que será evaluado a través de Pruebas de Aceptación, si el software es capaz de atravesar sin dificultades todas las pruebas se podrá contar con un sistema con el cual el cliente quedará satisfecho y quedaría listo para su uso.

## Conclusiones

Como resultado de los análisis realizados y la implementación del sistema propuesto, surge una nueva herramienta que permite autonomía en su aplicación como parte del proceso de soberanía tecnológica del país, lo que se avala por las siguientes consideraciones:

- El sistema presentado es el resultado del estudio y evaluación de varios sistemas de facturación y reportes existentes, de los que se tomaron sus principales ventajas.
- Se alcanzó satisfactoriamente los objetivos propuesto: Facilitar la administración de reportes y facturación en Asterisk PBX.
- Se demostró que las Metodologías Ágiles seleccionadas (XP y Scrum) garantizan la creación con calidad del sistema.

## Recomendaciones

Como resultado del sistema implementado se hacen las siguientes recomendaciones:

- Aplicar como experiencia piloto el sistema implementado en la Facultad 2 de la UCI.
- Mantener una constante retroalimentación con los clientes con vistas a perfeccionar el sistema.

## Glosario de Términos

**API:** Application Programming Interface.

**CDR:** Call Detail Record. Fichero donde Asterisk almacena los datos relacionados a cada llamada.

**CPU:** Unidad de Proceso Central.

**GPL:** General Public License. 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.

**HTTP:** HyperText Transfer Protocol.

**IP:** Internet Protocol. 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.

**Libre:** Se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

**Opensource:** Código abierto es el término con el que se conoce al software distribuido y desarrollado libremente.

**PBX:** Private Branch Exchange. 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.

**SSL:** Secure Sockets Layer.

**VoIP:** Voz sobre IP. Término usado en telefonía IP para definir los servicios que se usan para transmitir voz usando el protocolo IP.

**Widgets:** Controles. Pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets o Widget Engine.

## Referencias Bibliográficas

1. **S.A, VirtualFon Argentina.** CallFon - PBX, IVR, Call Center, Soluciones VoIP basadas en Asterisk - Digium. [En línea] 2002. [Citado el: 20 de Mayo de 2009.] <http://www.callfon.com.ar/>.
2. **FreePBX.** [En línea] 2009. [Citado el: 20 de Mayo de 2009.] <http://www.freepbx.org>.
3. **Elastix.** [En línea] 2009. [Citado el: 20 de Mayo de 2009.] <http://www.elastix.org/>.
4. **Software, Edgewall.** a2billing. [En línea] [Citado el: 20 de Mayo de 2009.] <http://www.asterisk2billing.org/>.
5. **José H. Canós, Patricio Letelier y M<sup>a</sup> Carmen Penadés.** *Métodologías Ágiles en el Desarrollo de Software.* Valencia : s.n. pág. 8.
6. **Solutions, PaloSanto.** [En línea] 2006-2008. [Citado el: 20 de Mayo de 2009.] <http://www.elastix.org>.
7. **Solís, Manuel Calero.** *Una explicación de la programación extrema (XP).* Madrid : s.n., 2003. pág. 11.
8. **Goncalvez, Flavio E.** *Asterisk PBX. Guía de la Configuración.* Janeiro : s.n., 2007. pág. 362.
9. **Dana Moore, Raymond Budd, William Wright.** *Professional Web 2.0 Programming with Django Python Frameworks.* Indianapolis, Indiana : Wiley Publishing, Inc., 2007. pág. 459.
10. **Duque, Raúl González.** *Python para todos.* España : s.n. pág. 108.
11. **Jim Van Meggelen, Leif Madsen, Jared Smith.** *Asterisk, The Future of Telephony.* United States of America : O'Reilly Media, Inc, 2007. pág. 604.
12. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison Wesley, 2000. pág. 458.
13. **Foundation, The Apache Software.** [En línea] 2009. <http://www.apache.org/>.
14. **Sun Microsystems, Inc.** [En línea] 2009. [Citado el: 10 de Mayo de 2009.] <http://www.mysql.com>.
15. **PostgreSQL Global Development Group.** [En línea] 2009. [Citado el: 30 de Mayo de 2009.] <http://www.postgresql.org/>.
16. **The Perl Foundation.** [En línea] 2009. [Citado el: 10 de Mayo de 2009.] <http://www.sqlite.org/>.
17. **The PHP Group.** [En línea] 2009. [Citado el: 5 de Mayo de 2009.] <http://www.php.net/>.

18. **Villar, Malay Rodríguez.** Introducción de procedimientos Ágiles en la producción de software en la Facultad 7 de la Universidad de las Ciencias Informáticas. [En línea] Junio de 2007. [Citado el: 5 de Mayo de 2009.] [http://bibliodoc.uci.cu/TD/TD\\_0693\\_07.pdf](http://bibliodoc.uci.cu/TD/TD_0693_07.pdf).
19. **Lighthttpd Group.** [En línea] 2009. [Citado el: 20 de Abril de 2009.] <http://www.lighthttpd.net/>.

## Anexos

## Anexo 1.Pruebas de Aceptación

Caso de Prueba de Aceptación	
<b>Código:</b>	<b>Historia de usuario:</b>
<b>Nombre:</b>	
<b>Descripción:</b>	
<b>Condiciones de Ejecución:</b>	
<b>Entrada/Pasos de Ejecución:</b>	
<b>Resultado Esperado:</b>	
<b>Evaluación de la Prueba:</b>	

## Anexo 2.Historias de Usuario

Historia de Usuario	
<b>Número:</b>	<b>Usuario:</b>
<b>Nombre historia:</b>	
<b>Prioridad en el Negocio:</b>	<b>Riesgo en el Desarrollo:</b>
<b>Puntos estimados:</b>	<b>Iteración asignada:</b>
<b>Programador(es) responsables:</b>	

<b>Descripción:</b>
<b>Observaciones:</b>
<b>Firma:</b>

### Anexo 3. Tarea de Ingeniería

Tarea de Ingeniería	
<b>Número Tarea:</b>	<b>Número Historia de Usuario:</b>
<b>Nombre Tarea:</b>	
<b>Tipo de Tarea :</b>	<b>Puntos Estimados:</b>
<b>Fecha Inicio:</b>	<b>Fecha Fin:</b>
<b>Programador Responsable:</b>	
<b>Descripción:</b>	

### Declaración de Autoría

Nosotros: **Luis Eduardo Acosta Aparicio** y **Nicolás Montero Puñales** nos declaramos como únicos autores de este trabajo y autorizamos a la Universidad de Ciencias Informáticas (UCI) para que haga uso de este trabajo de la manera que estime conveniente.

Y para que así conste, firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ de 2009.

\_\_\_\_\_  
Firma de Autores

\_\_\_\_\_  
Firma del Tutor