

Universidad de las Ciencias Informáticas

Facultad 2



Título: Módulo de interconexión y enrutamiento de llamadas de la
Plataforma Platel.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias
Informáticas.

Autor(es): Yanet Oscary Fuentes y Carlo Carlo Lizano González.

Tutor(es): Ing. Abel Alejandro Yáñez Moncada.

Cotutor(es): Msc. Yadira Ruiz Constanten e Ing. Duany Baró Menéndez

Ciudad de La Habana, 2010.

Año 52 de la Revolución

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autores:

Yanet Oscary Fuentes

Carlo Carlo Lizano González

Tutor:

Ing. Abel Alejandro Yáñez Moncada

DATOS DE CONTACTO

Tutor: Abel Alejandro Yáñez Moncada.

Graduado en el año 2009 por la Universidad de la Ciencias Informáticas (UCI). Desde su graduación hasta la actualidad se ha desempeñado como Arquitecto en el proyecto Plataforma Telefónica (PLATEL) en el centro de desarrollo UCID. Actualmente se encuentra en el período de adiestramiento.

AGRADECIMIENTOS

A todas las personas que hicieron posible con su apoyo y dedicación que este sueño se realizara.

A mi compañero de tesis por todas las cosas que pasamos durante el desarrollo de la misma.

A mis tutores Abel y Fadirra por siempre estar tan pendiente del desarrollo de la tesis.

A Jordán que ha sido importante en mi vida, que no tengo palabras para expresar lo que significa para mí.

A Carlos, Camilet y Casel, Anabel, Made, Liorge, Frank, Alberto, Doris, Bonne, Lexany y

Harlenys que me han apoyado en todo en el transcurso de la carrera por su incondicionalidad.

A Duany por ser mi tutor, compañero y sobre todo amigo, por fajarse tanto conmigo durante el transcurso de la tesis.

A mis amigos, a todos a los de mi grupo 2101, a los de mi grupo actual, a los de mi proyecto, a los que hice en el camino a los que en algún momento me dijeron, "Hola, ¿Cómo va la tesis?"

A Carmen y Omar por ser mis padres, por estar pendientes de mí durante estos 6 años.

A Mindalia que ha sido como una madre para mí en estos 6 años y en este quinto año por favor.

A Ismael, Madelin y a Euclides por apoyarme y por darme tanto amor.

A mis vecinos Paridad, Dorka, María Antonia y Bienvenido que se han preocupado por mi carrera como si fuesen mis padres.

A todos mis profesores a todos los que pasaron por mi vida y en especial a Fanetsy, a Fadirra y a Lester que me dieron mucho apoyo en todo momento.

A mi familia que siempre ha estado unida y que me han dado su apoyo y su amor incondicional, que han sido y son muy importantes para mí.

Muy especial a Jorge Oscar y a Vilma Fuentes, mis padres por tanta devoción, dedicación, porque después de tantos sufrimientos y tantas lágrimas pueden ver su sueño realizado, sus expectativas cumplidas, sus valores inculcados y por encima de todo por la confianza que siempre han tenido en mí.

Ah y a mí, por haber luchado aún sin fuerzas y lograr atrapar el sueño que por momentos se me iba de las manos.

Fanet Oscar Fuentes

DEDICATORIA

A mi abuelita Dora porque este era su sueño.

A mi abuelito Manolo por su apoyo y esperanzas depositadas en mí.

A mi familia que me ha dado su apoyo a lo largo de estos 6 años.

A mi primita María que ha sido mi fuerza para seguir hacia adelante.

A mis padres que son lo más poderoso que tengo.

A mis compañeros, a los que estuvieron y los que ya no están.

Fanet Oscar Fuentes

RESUMEN

A finales del siglo XIX se introdujo el teléfono como dispositivo esencial para las comunicaciones a distancia. Durante el desarrollo de la telefonía fue necesario establecer la comunicación entre dos puntos ya fuera logrando una conexión manual o utilizando sistemas digitales de conmutación. Dada esta necesidad y producto de la evolución de los aparatos telefónicos y las redes para su uso y conexión, surgieron las centrales telefónicas privadas digitales, las cuales brindan una amplia gama de servicios telefónicos tradicionales e incluye otros relacionados con la tecnología VoIP. Las Fuerzas Armadas Revolucionarias necesitan de un sistema de administración capaz de gestionar y controlar las comunicaciones telefónicas y el flujo de llamadas a lo largo de todo el país. Este sistema, entre otras funciones, debe ser capaz de interconectar de forma sencilla y eficiente las distintas plataformas telefónicas, así como enrutar el flujo de llamadas entre ellas. El presente trabajo de diploma propone una solución para esta problemática basada en la implementación de un sistema capaz de interconectar plataformas telefónicas y enrutar llamadas usando Asterisk y administrándolo mediante una interfaz web.

PALABRAS CLAVE

Telefonía, PBX, Asterisk, Sistema de Administración, troncal, ruta entrante, ruta saliente, protocolo.

Tabla de Contenidos

DECLARACIÓN DE AUTORÍA	II
DATOS DE CONTACTO	III
AGRADECIMIENTOS.....	IV
DEDICATORIA.....	V
RESUMEN	VI
TABLA DE CONTENIDOS.....	VII
INTRODUCCIÓN.....	1
CAPÍTULO I:“FUNDAMENTACIÓN TEÓRICA. ESTADO DEL ARTE”.....	6
1.1 CENTRALES TELEFÓNICAS PRIVADAS (PBX).....	6
1.2 PROTOCOLOS DE SEÑALIZACIÓN.....	8
1.2.1 H.323.....	8
1.2.2 SIP.....	9
1.2.3 Dahdi.....	10
1.2.4 IAX.....	10
1.3 SOFTPBX	11
1.3.1 Asterisk.....	12
1.3.2 Elastix.....	14
1.3.3 FreePBX.....	15
1.3.4 TrixBos.....	16
1.3.5 Avnvox.....	16
1.4 PBX BASADAS EN ASTERISK MÁS UTILIZADAS INTERNACIONALMENTE.....	17
1.4.1 Digium.....	17
1.4.2 Fonality.....	17
1.5 PBX BASADAS EN ASTERISK MÁS UTILIZADAS EN CUBA.....	17
1.6 METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	18
1.6.1 <i>Proceso de Desarrollo y Gestión de Software de la Unidad de Compatibilización Integración y Desarrollo de Software para la Defensa (UCID).</i>	19
1.7 FRAMEWORK.....	21
1.7.1 <i>Marco de Trabajo SAUXE 1.5.4</i>	21
1.8 GESTOR DE BASE DE DATOS.....	22

1.8.1 PostgreSQL.....	23
1.9 SERVIDOR WEB.....	24
1.9.1 Apache.....	24
1.10 LENGUAJES DE PROGRAMACIÓN.....	25
1.10.1 PHP.....	25
1.11 MODELADO DE SOFTWARE.....	26
1.11.1 El Lenguaje Unificado de Modelado.....	27
1.11.2 Herramienta seleccionada para el modelado, Visual Paradigm para UML.....	27
CAPÍTULO 2:“CARACTERÍSTICAS DEL SISTEMA”	29
2.1 OBJETO DE ESTUDIO Y SITUACIÓN PROBLÉMICA.....	29
2.2 OBJETO DE AUTOMATIZACIÓN.....	30
2.2.1 Información que se maneja.....	31
2.3 PROPUESTA DE SISTEMA.....	31
2.4 MODELO CONCEPTUAL.....	31
2.5 ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE.....	32
2.5.1 Requerimientos funcionales.....	32
2.5.2 Requerimientos no funcionales.....	33
2.5.3 Descripción de los requisitos funcionales.....	34
2.5.4 Prototipos de interfaz de usuario.....	38
CAPÍTULO 3:“DISEÑO DEL SISTEMA”	41
3.1 DISEÑO.....	41
3.1.1 Diagramas de interacción.....	41
3.1.2 Diagramas de clases.....	45
3.2 DISEÑO DE LA BD.....	48
3.2.1 Modelo de Datos.....	48
3.3 PATRONES DE DISEÑO UTILIZADOS.....	50
3.4 TRATAMIENTO DE ERRORES.....	51
3.5 SEGURIDAD.....	52
CAPÍTULO 4:“IMPLEMENTACIÓN Y PRUEBA.”.....	53
4.1 IMPLEMENTACIÓN.....	53
4.1.1 Diagrama de componentes.....	53
4.1.2 Diagrama de componentes.....	54
4.2 PRUEBA.....	55
4.2.1 Diseños de Casos de Prueba.....	55
4.2.2 Pruebas de Software.....	55
4.2.3 Método de Prueba.....	56
4.2.4 Listado de las No Conformidades.....	56
CONCLUSIONES GENERALES Y RECOMENDACIONES	58

BIBLIOGRAFÍA.....59

REFERENCIAS BIBLIOGRÁFICAS..... 62

GLOSARIO DE TÉRMINOS..... 64

ÍNDICE DE FIGURAS

Figura 1 Funcionamiento de una PBX..... 2

Figura 2 Estructura de la PBX..... 7

Figura 3 Algunos protocolos utilizados en una PBX 8

Figura 4 Algunos ejemplos de SoftPBX libres basados en Asterisk..... 12

Figura 5 Estructura de la plataforma Asterisk 13

Figura 6 Dispositivos de Avanzox..... 17

Figura 7 Etapas del ciclo de vida del proyecto..... 20

Figura 8 Fragmento del fichero de configuración extensions.com..... 30

Figura 9 Modelo Conceptual 32

Figura 10 Interfaz para gestionar troncal..... 38

Figura 11 Interfaz para insertar un troncal 39

Figura 12 Interfaz para modificar un troncal..... 39

Figura 13 Interfaz para eliminar un troncal..... 40

Figura 14 Diagrama de secuencia Gestionar Troncales (sección Adicionar Troncales)..... 42

Figura 15 Diagrama de secuencia Gestionar Troncales (sección Modificar Troncales) 43

Figura 16 Diagrama de secuencia Gestionar Troncales (sección Eliminar Troncales)..... 44

Figura 17 Diagrama de clases del diseño Gestionar Troncales..... 45

Figura 18 Modelo de Datos 48

Figura 19 Diagrama de Componentes Sistema de Administración de la Plataforma Platel 53

Figura 20 Diagrama de Componentes Sistema de Administración de la Plataforma Platel 54

ÍNDICE DE TABLAS

Tabla 1 Especificación del requisito Adicionar Troncal	35
Tabla 2 Especificación del requisito Modificar Troncal	36
Tabla 3 Especificación del requisito Eliminar Troncal	37
Tabla 4 Descripción de la clase GestTroncalController	46
Tabla 5 Descripción de la clase Troncal	47
Tabla 6 Descripción de la clase GestTroncal.phtml.....	47
Tabla 7 Descripción de la tabla dat_troncal.....	49
Tabla 8 Descripción de la tabla dat_rutaentrante	50
Tabla 9 Descripción de la tabla dat_rutasaliente.....	50
Tabla 10 Matriz de Integración de Componentes Internos	54
Tabla 11 Listado de no conformidades y acciones correctivas.....	57

Introducción

La telefonía se ha caracterizado por un desarrollo y un perfeccionamiento continuo y acelerado. Desde su concepción original, se han ido introduciendo mejoras sucesivas tanto en el propio aparato telefónico, como en los métodos y sistemas de explotación de la red. Esto se evidencia en los vertiginosos cambios que han ocurrido desde los primeros teléfonos con membrana de carbón hasta los dispositivos móviles, celulares y la telefonía IP (Protocolo de Internet) en menos de un siglo.

En los orígenes de la telefonía era necesario conectar manualmente cables para establecer la comunicación. Este sistema era conocido como PMBX (PBX Manual) que luego fue reemplazado por un dispositivo electromecánico automático y posteriormente con el avance de la electrónica de microprocesadores, por sistemas digitales de conmutación que se le llamó PABX (Private Automatic Branch Exchange) que desplazó al PMBX hasta hacerlo casi inexistente. A partir de ese momento PABX y PBX (Private Branch Exchange) se convirtieron en sinónimos. La PBX, que no era más que una central telefónica conectada directamente a la red pública de teléfono PSTN (Public Switched Telephone Network) por medio de líneas troncales para gestionar, además de las llamadas internas, las entrantes y/o salientes con autonomía sobre cualquier otra central telefónica. Este dispositivo generalmente pertenecía a la empresa que lo tenía instalado y no a la compañía telefónica, de aquí el adjetivo privado de su denominación. Con el pasar de los años la incorporación de las mismas al trabajo diario de las empresas y entidades ha crecido notoriamente ya que brinda herramientas que hacen más productiva y económica la comunicación telefónica.

La telefonía IP surge como una alternativa a la telefonía tradicional, brindando nuevos servicios al cliente y una serie de beneficios económicos y tecnológicos con características especiales como interoperabilidad con las redes telefónicas actuales, calidad de servicio garantizada a través de una red de alta velocidad, servicios de valor agregado. Es un servicio que permite realizar llamadas desde redes que utilizan el protocolo de comunicación IP (Internet Protocol), o sea, el sistema que permite comunicar computadoras de todo el mundo a través de las líneas telefónicas. Esta tecnología digitaliza la voz y la comprime en paquetes de datos que se reconvierten de nuevo en voz en el punto de destino.

Actualmente las PBX se han convertido en tema de interés con la incursión de los protocolos de Voz sobre IP. Dado que ahora se puede aprovechar la Red de datos para transportar la voz en forma de paquetes y con la evolución de los sistemas operativos basados en UNIX, se ha logrado realizar la conmutación de paquetes con simples ordenadores de escritorio. Esto ha cambiado de forma trascendental el concepto de PBX pues ya se le conoce como IP-PBX. Una de sus principales ventajas es su utilización de convergencia de redes de voz y datos, es decir, el acceso a la Internet al igual que la comunicación por VoIP y la telefonía tradicional son todas posibles con una sola línea de acceso a cada usuario. (1)

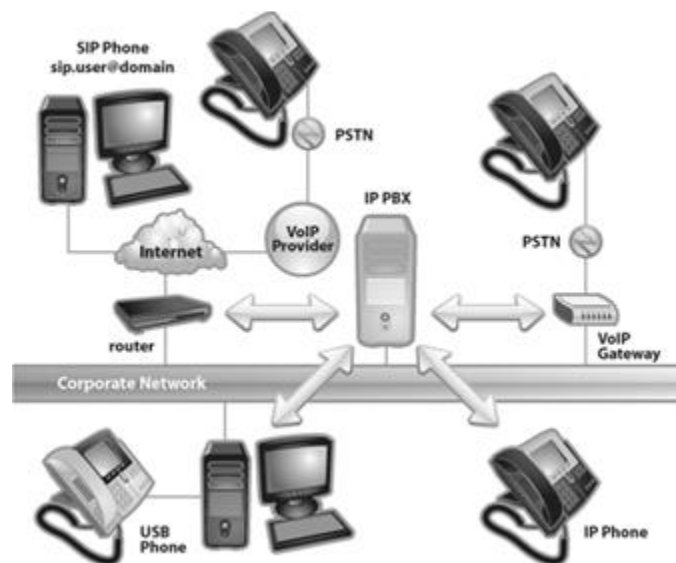


Figura 1 Funcionamiento de una PBX

Actualmente se están desarrollando en el mundo del software libre, programas que realizan las funciones de una central PBX bajo Linux, tal es el caso del programa Asterisk. Este sistema incluye muchas características anteriormente sólo disponibles en costosos sistemas propietarios PBX como buzón de voz, conferencias, distribución automática de llamadas y otras muchas, siendo posible integrar estas y más funciones en un solo computador que brinda comunicación telefónica, Internet, fax, etc. sin costes de licencia asociados, a diferencia de otras centrales digitales que si cobran por sus servicios y funcionalidades, entre ellas las gigantes Ericsson y Panasonic

Quizá lo más interesante de Asterisk es que soporta muchos protocolos VoIP como pueden ser SIP, H.323, IAX y MGCP. Asterisk puede operar con terminales IP actuando como un registrador y como Gateway (puente) entre ambos.

Son muy modestos los avances del país en cuestiones de PBX sobre Asterisk, salvo algunos proyectos investigativos en la Universidad Central de Las Villas “Marta Abreu” y en la Universidad de las Ciencias Informáticas, específicamente en la Facultad 2. La Empresa de Telecomunicaciones de Cuba ETECSA, encargada de las telecomunicaciones en la isla, también ha desarrollado pequeñas centrales telefónicas que resuelven problemas específicos en algunas instalaciones hoteleras y en sus propias oficinas.

Las Fuerzas Armadas Revolucionarias (FAR) tienen desplegado el sistema TrixBos, que es una distribución del sistema operativo GNU/Linux, y tiene la particularidad de ser una PBX por software basada en Asterisk. TrixBos está concebido para usuarios con un alto nivel de conocimientos en temas de informática y telecomunicaciones, lo cual dificulta considerablemente su utilización, ya que en la mayoría de las unidades militares no se dispone del personal calificado para desempeñar este trabajo.

Existen varios sistemas para la administración de Asterisk entre los cuales resalta Elastix como una solución de telefonía unificada que integra servicios de fax, correo electrónico, mensajería instantánea, voz IP, conferencias y colaboración. A pesar de todas las funcionalidades que nos brinda Elastix, resulta engorroso el uso de esta herramienta por personal no calificado o que no posea conocimientos básicos sobre el manejo de la misma. Es por ello que el proyecto Platel ha sido llamado a elaborar un sistema de fácil interpretación y manejo para el personal de las Fuerzas Armadas Revolucionarias que realice las funciones de una PBX.

Entre los servicios que debe facilitar este sistema se encuentra el de interconectarse con otras plataformas telefónicas, facilitando el enrutamiento de llamadas y troncales, dejando al descubierto el siguiente **Problema Científico**: ¿Cómo interconectar y enrutar llamadas telefónicas utilizando distintos protocolos de comunicación? Para dar solución al problema planteado se define como **objeto de estudio**: Sistemas para la administración de PBXs Asterisk y el **campo de acción** se enmarca en: Conexión de Platel con otras plataformas telefónicas.

Se plantea como **Objetivo General**: Desarrollar un sistema de interconexión y enrutamiento de llamadas de la Plataforma Platel.

De aquí se proponen los siguientes **objetivos específicos**:

- ✓ Elaborar el estado del arte que exponga el comportamiento y tendencias actuales de tecnologías para el manejo de la PBX Asterisk y software para su administración.
- ✓ Elaborar la documentación correspondiente al análisis, diseño e implementación del módulo de enrutamiento e interconexión de llamadas con el fin de adicionar una fuente ingenieril consultable para el desarrollo de futuras versiones.
- ✓ Implementar un sistema que permita la interconexión y el enrutamiento de llamadas y troncales de la Plataforma Platel a través de los protocolos estudiados, basado en la documentación generada y siguiendo las pautas del proceso de desarrollo de Software de La Unidad de Compatibilización, Integración y Desarrollo de Software para la Defensa UCID.
- ✓ Probar y validar el sistema elaborado.
- ✓ Integrar la solución propuesta al sistema de administración AdminPlatel.

La **idea a defender** que se establece es:

Con el desarrollo de un sistema de interconexión de la Plataforma Platel se logrará su comunicación con distintas PBX y se garantizará el enrutamiento de llamadas telefónicas entre ellas.

Posibles resultados:

Con el desarrollo de este módulo de la Plataforma Platel, se logrará el enrutamiento de llamadas y la interconexión con distintas plataformas, una de las funcionalidades principales a

alcanzar por cualquier central telefónica, complementando y enriqueciendo la solución que constituye Platel para la problemática existente en el Ministerio de las Fuerzas Armadas Revolucionarias.

Este documento está estructurado en 4 capítulos:

Capítulo 1. “**Fundamentación Teórica. Estado del Arte**”: se especifican conceptos que serán tratados a lo largo de todo el documento y que son de vital importancia para la comprensión del mismo. Se realiza un estudio del estado del arte de sistemas automatizados que usan centrales telefónicas PBX basadas en Asterisk. Además se fundamentan elementos como son las diferentes herramientas, metodologías, lenguaje de programación y plataformas a utilizar en la construcción del software.

Capítulo 2. “**Características del Sistema**”: se realiza una detallada descripción de los requisitos funcionales y no funcionales del sistema. Se realiza el diseño de un prototipo de interfaz inicial en correspondencia a los requisitos funcionales.

Capítulo 3. “**Diseño del Sistema**”: se hará un diseño detallado de cada uno de los artefactos creados en correspondencia con la propuesta del sistema en cuestión. Utilizando los diseños y las plantillas especificadas en el Proceso de Desarrollo del UCID.

Capítulo 4. “**Implementación y prueba**” se detallarán todos los elementos de los diagramas de componentes así como los de los diseños de casos de prueba que contribuyen a la validación de los resultados obtenidos.

Capítulo 1: “Fundamentación Teórica. Estado del Arte”

Introducción

En este capítulo se brinda al lector una panorámica de las herramientas y tecnologías utilizadas para dar solución al problema planteado, estableciendo una comparación cualitativa entre las diferentes opciones existentes y las tendencias actuales a nivel mundial.

1.1 Centrales Telefónicas Privadas (PBX)

Una central de telefonía privada (PBX) es un dispositivo que permite a las empresas conectar sus terminales telefónicos de forma independiente a la red pública o PSTN. Su objetivo fundamental es establecer y mantener la comunicación entre dos puntos finales durante todo el tiempo requerido por los usuarios, y opcionalmente proveer información para contabilizar y facturar las llamadas. Indistintamente muchos fabricantes procuran implementar a sus PBXs la mayor cantidad posible de funcionalidades, de modo que sus productos sean más atractivos para los compradores o clientes, entre otras, se señalan las siguientes:

- ✓ Marcado rápido: Es posible marcar, por ejemplo, números de centros de emergencia, previamente definidos y configurados.
- ✓ Contestador automático: Mensaje de voz para cuando el usuario no esté disponible.
- ✓ Servicio de directorio automatizado: Los usuarios pueden ser enrutados a la extensión deseada tecleando o diciendo verbalmente las iniciales o el nombre del usuario deseado.
- ✓ Desvío de llamadas: Es posible que una extensión reciba las llamadas de otra, previamente definida.
- ✓ Transferencia de llamadas.
- ✓ Llamada en espera

- ✓ Aviso mediante timbre cuando una línea externa o extensión está libre: Es una funcionalidad que se puede acoplar también al servicio de llamada en espera.
- ✓ Conferencia entre 3 o más usuarios.
- ✓ Mensaje de Bienvenida.
- ✓ Marcado de una extensión desde el exterior del sistema: Una extensión puede recibir llamadas desde el exterior de la PBX, así como realizar llamadas al exterior.
- ✓ Sígame (Follow-me): Es posible programar el desvío de llamadas desde cierta extensión a una distinta.
- ✓ Música en espera.
- ✓ Contestador automático de buzón de voz: Tras el mensaje del contestador automático, es posible guardar mensajes de voz. (2)

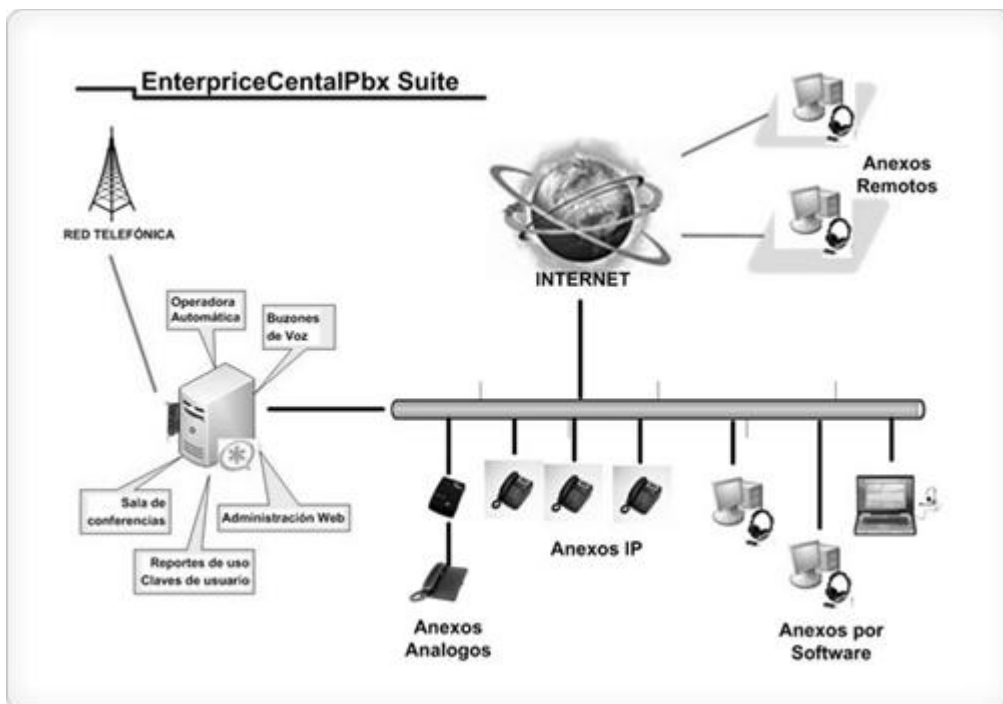


Figura 2 Estructura de la PBX

1.2 Protocolos de Señalización

La señalización a través de protocolos tiene un papel muy importante en la red, ya que es la encargada de establecer, mantener, administrar y finalizar una conversación entre dos puntos. Además de ofrecer funciones de supervisión, marcado, llamada y retorno de tonos de progreso. En los siguientes apartados se describen algunos de los protocolos más importantes utilizados por las PBXs (3). La siguiente imagen muestra el funcionamiento y la relación de varios de ellos con una PBX.

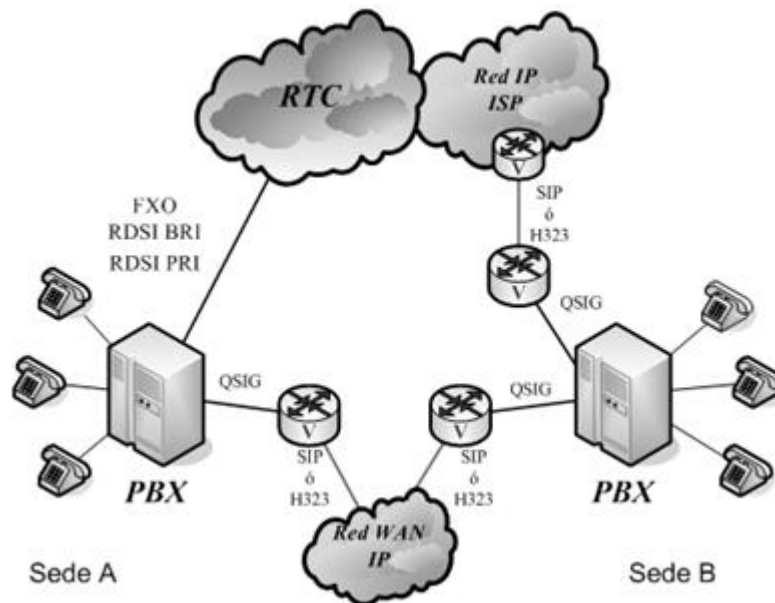


Figura 3 Algunos protocolos utilizados en una PBX

1.2.1 H.323

H.323 es una familia de estándares desarrollado por la ITU (International Telecommunication Union, Unión Internacional de Telecomunicaciones) en 1996 con el objetivo de ofrecer un mecanismo de transporte para servicios multimedia sobre redes que no garantizan calidad de servicio, aunque su uso se ha extendido sobre redes IP. Pese a que inicialmente fue definido como un protocolo de videoconferencia, rápidamente ha evolucionado para cubrir todas las necesidades de la voz sobre IP (VoIP). Se trata de una recomendación bastante cerrada donde se definen los códec a utilizar, tanto en audio como en video, y los protocolos de transporte de la información. Fue el primer estándar en adoptar el Protocolo de Transporte en tiempo Real

(RTP), siendo capaz de aplicar algoritmos de encriptación de la información, evitando de esta manera añadir elementos de seguridad adicionales a los requeridos para la conexión a Internet. Pese a que técnicamente es un protocolo potente y maduro, el interés por parte de los usuarios del protocolo y empresas actualmente ha disminuido debido principalmente a su complejidad.

1.2.2 SIP

SIP (acrónimo de Session Initial Protocol, Protocolo de Inicio de Sesión) es un protocolo desarrollado por el IETF (Internet Engineering Task Force, Grupo de Tareas de Ingeniería de Internet) en 1999 para el control de llamadas multimedia, la implementación de servicios telefónicos avanzados y la modificación y terminación de sesiones de comunicación multimedia entre usuarios. SIP está basado en HTTP (acrónimo de Hypertext Transfer Protocol, Protocolo de Transferencia de Hipertexto) adoptando las características más importantes de este estándar como son la sencillez de su sintaxis y una estructura cliente/servidor basada en un modelo petición/respuesta. Otra de las ventajas de SIP es su sistema de direccionamiento. Las direcciones SIP tienen una estructura parecida a la de un correo electrónico dotando a sus clientes de una alta movilidad facilitando una posible integración en comunicaciones móviles. Cabe destacar que aunque originalmente SIP tenía como objetivo la simplicidad, en su estado actual se ha vuelto tan complejo como H.323. Pero el gran potencial de SIP reside en su flexibilidad ya que ofrece la posibilidad de programar nuevos servicios no definidos por sus creadores. Entornos de programación como CGI (acrónimo de Common Gateway Interface, Interfaz Común de Pasarela) o sencillos lenguajes de programación como CPL (acrónimo Call Processing Language, Lenguaje de Procesamiento de Llamadas), son algunas de las herramientas para la implementación de servicios sin que conlleve a un peligro para la integridad del sistema. Esta es la característica principal por la que SIP actualmente goza de un mayor éxito que H.323. Los clientes SIP llamados peers o user agents, usan el puerto 5060 en TCP (acrónimo de Transmission Control Protocol, Protocolo de Control de Transmisión) y UDP (acrónimo de User Datagram Protocol, Protocolo de Datagrama de Usuario) para conectar con los servidores SIP. Es usado simplemente para iniciar y terminar llamadas de voz y video. El protocolo SIP consta de capacidades tales como:

- ✓ Localización del usuario;

- ✓ Disponibilidad del usuario: Determinación de la voluntad del receptor de la llamada de participar en las comunicaciones;
- ✓ Capacidad del usuario: Determinación del medio y de sus parámetros;
- ✓ Gestión de la sesión: Transferencia, terminación de sesiones, modificación de los parámetros de la sesión desde el propio user agent. (4)

1.2.3 Dahdi

El paquete DADHI (Digium Asterisk Hardware Device Interface) permite cargar los drivers y configurar distintos tipos de tarjetas (para conectar teléfonos analógicos, líneas telefónicas, líneas digitales, etc.) en Asterisk. Anteriormente, esta función de soportar y reconocer las tarjetas de dispositivos telefónicos estaba a cargo de Zaptel, cuyo propietario informó a la empresa Digium que “Zaptel” era una marca registrada ante lo cual fue necesario buscar una palabra que no estuviese registrada y que tuviera relación directa con el software que da soporte a sus tarjetas y el nombre en cuestión fue DADHI. (5)

1.2.4 IAX

Inter-Asterisk Exchange protocol (IAX) fue desarrollado por la empresa norteamericana Digium para la comunicación entre centrales Asterisk aunque actualmente se han implementado clientes que también soportan este protocolo. El principal objetivo de IAX es minimizar el ancho de banda utilizado en la transmisión de voz y vídeo a través de la red IP y proveer un soporte nativo para ser transparente a los NATs (acrónimo de Network Address Translation, Traducción de Dirección de Red). La estructura básica de IAX se fundamenta en la multiplexación de la señalización y del flujo de datos sobre un simple puerto UDP, generalmente el 4569.

El protocolo original ha quedado obsoleto en favor de su segunda versión conocida como IAX2. Permite manejar una gran cantidad de códec y transportar cualquier tipo de datos. (6)

1.3 SoftPBX

En los últimos años gracias al desarrollo de los sistemas operativos de computadoras, se han realizado esfuerzos, por construir software capaz de emular el funcionamiento de una PBX. Un SoftPBX es una PBX diseñada completamente en software. Es capaz de proveer todas las funcionalidades de una centralita, con la posibilidad de extenderlas, ya que no hay dependencia directa con el hardware de telefonía disponible. A pesar de existir tantos desarrollos de hardware en PBX, existen varias razones que aconsejan la adopción de un SoftPBX:

- ✓ Libertad y Gratuidad: En los SoftPBX de código abierto, esta concepción permite la adquisición, instalación y puesta en marcha de sistemas telefónicos sin dependencia a las grandes compañías de la telefonía.
- ✓ Adaptabilidad y Mejora continua: Se trata de software, de códigos informáticos, escritos en lenguajes de programación, por lo que la posibilidad de extender las funcionalidades es mucho mayor frente a las soluciones de hardware. Para la mayor adaptabilidad solo se requiere de personal con conocimientos de telefonía y programación, en lugar de una gran infraestructura de desarrollo de hardware.
- ✓ Disminución de costos contra plataformas privativas: En las variantes de los SoftPBXs de código abierto, en la mayoría de los casos se adquieren los programas en internet de manera gratuita, quedando las mayores inversiones para las plataformas de hardware necesarias, y los dispositivos adicionales.

La siguiente figura muestra algunas de estas aplicaciones que se describen a continuación:



Figura 4 Algunos ejemplos de SoftPBX libres basados en Asterisk

1.3.1 Asterisk

Asterisk es una aplicación con licencia GPL (código abierto) capaz de realizar las funciones de una PBX. Este proyecto fue desarrollado por el ingeniero Mark Spencer, miembro fundador de la compañía Digium y principal desarrollador. Originalmente fue implementado para cualquiera de las diferentes distribuciones Linux existentes. Actualmente se intenta portar a otros sistemas operativos como Mac, Solaris o Microsoft Windows. Además hace voz sobre IP en cuatro protocolos (H.323, SIP, MGCP, IAX) y puede interoperar con casi todos los basados en estándares de telefonía utilizando el equipo de hardware relativamente barato. Asterisk pese a ser una aplicación software ofrece las mismas características y servicios que los más caros sistemas propietarios PBX como puede ser el buzón de voz, salas de conferencia o música en espera entre otros. Son muchas las ventajas que ofrece respecto a las centrales por hardware, algunos de ellos son:

- ✓ **Funcionalidad:** Asterisk dispone de todas las funcionalidades de las grandes centrales propietarias (Cisco, Avaya, Alcatel, Siemens, etc.), desde las más básicas (desvíos, capturas, transferencias, multiconferencias), hasta las más avanzadas (Buzones de voz, IVR (Interactive Voice Response, Respuesta de Voz Interactiva), CTI (Computer

Telephony Integration, Integración Teléfono –Computadora), ACD (Automatic Call Distributor, Distribuidor Automático de Llamadas).

- ✓ Escalabilidad: El sistema puede dar servicio desde 10 usuarios en una sede de una pequeña empresa, hasta 10.000 de una multinacional repartidos en múltiples sedes.
- ✓ Competitividad en coste: No solo por ser un sistema de código abierto sino gracias a su arquitectura de hardware, utiliza plataforma servidor estándar (de propósito no específico) y tarjetas que se conectan como dispositivos periféricos a cualquier placa base (motherboard) de una computadora con puertos PCI (Peripheral Component Interconnect) para las interfaces de telefonía, que por la competencia del mercado se han ido abaratando progresivamente.
- ✓ Interoperabilidad y Flexibilidad: Asterisk ha incorporado la mayoría de estándares de telefonía del mercado, tanto los tradicionales con el soporte de puertos de interfaz analógicos, como los de telefonía IP (SIP, H.323, MGCP, SCCP/Skinny). Eso le permite conectarse a las redes públicas de telefonía tradicional e integrarse fácilmente con centrales tradicionales (no IP) y otras centrales IP.

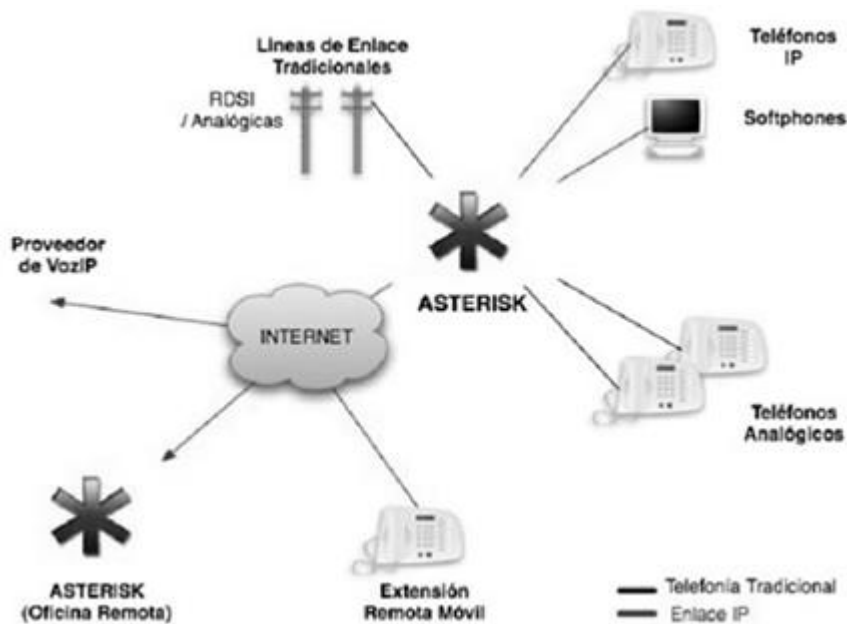


Figura 5 Estructura de la plataforma Asterisk

Adicionalmente provee todas las ventajas de tener una PBX como:

- ✓ Configurar extensiones de su PBX principal en sitios distantes conectados a través de la red Internet y ahorrar costos en llamadas de larga distancia.
- ✓ Conexión con proveedores de Voz sobre IP para realizar llamadas internacionales a bajo costo o asociar un número telefónico de otro país a su sistema de PBX.
- ✓ Cualquier cosa que permita hacer otra planta telefónica de Avaya, Cisco o Panasonic es posible con el sistema de PBX de Asterisk y a una fracción de su precio. (7)

1.3.2 Elastix

Elastix es una distribución basada en Asterisk, promocionada por la empresa ecuatoriana Soluciones Palosanto, es un software aplicativo que integra herramientas disponibles para PBXs basados en Asterisk en una interfaz simple y fácil de usar. Añade su propio conjunto de utilidades integrando varios paquetes de software, en los que incluye su propio conjunto de características y añade nuevas interfaces para el control y reportes de sí mismo. Permite además la creación de módulos de terceros para hacer de éste un conjunto completo de software disponible para la telefonía de código abierto. Sus metas son, la confiabilidad, modularidad y fácil uso, complementadas con la robustez para reportar, lo que lo convierte en una gran opción para implementar un PBX basado en Asterisk. Utiliza FreePBX como sistema de administración. (8)

Algunas de las características provistas por Elastix son:

- ✓ Soporte para video: Se pueden realizar video-llamadas.
- ✓ Soporte para Virtualización: Es posible correr múltiples máquinas virtuales de Elastix sobre la misma caja.
- ✓ Interfaz Web para el usuario, realmente amigable.

- ✓ Fax a email para faxes entrantes: También se puede enviar algún documento digital a un número de fax a través de una impresora virtual.
- ✓ Interfaz para tarifas.
- ✓ Configuración gráfica de parámetros de red.
- ✓ Reportes de uso de recursos.
- ✓ Opciones para reiniciar/apagar remotamente.
- ✓ Reportes de llamadas entrantes/salientes y uso de canales.

1.3.3 FreePBX

FreePBX, es una interface gráfica, vía Web que facilita la gestión y administración de un sistema de telefonía IP, basado en Asterisk fácil de usar y con gran capacidad. Con FreePBX, es posible modificar de una manera sencilla, la configuración del sistema de telefonía, evitando tener que editar los ficheros de configuración manualmente. Puede mantener las bases de datos de usuarios y extensiones, así como todas las funciones de valor añadido. Por citar las más importantes:

- Dial plan de llamadas entrantes y salientes.
- IVR (Recepcionista digital interactiva) – Operadora automática.
- Time conditions – Gestión de llamadas entrantes según horario y fecha.
- Grupo de llamadas (Ring Groups): Round-Robin, todas a la vez, etc.
- Follow-me.
- ACD – Sistema de colas y agentes.
- Monitorización de llamadas.

- Sistema de mensajería.
- Música en espera.
- Conferencias.
- Grabación de las llamadas (sólo recomendado para pequeños volúmenes). (9)

1.3.4 TrixBos

TrixBos es una distribución de Linux que unifica la distribución CentOS, Asterisk y FreePBX, entre otras aplicaciones, que le permite aprovisionar una PBX para Voz sobre IP (VoIP) con recursos de cómputo modestos. TrixBos es una distribución ideal para el usuario novato de Asterisk, empresas caseras que desean utilizar Asterisk, o soluciones que no sobrepasarán la docena de llamadas simultáneas, es una gran fuente de estudio para la administración de Asterisk, ya que utiliza FreePBX con todas las bondades de una interfaz web y facilita su actualización y mantenimiento de esta forma.

La limitante para este sistema radica en que no es una herramienta que soporte una gran cantidad de llamadas simultáneas, aunque cabe destacar algunas características positivas como su flexibilidad con una gran variedad de protocolos y que posee una interfaz web disponible en 6 idiomas distintos. (10)

1.3.5 Avanvox

Es una solución avanzada de centralita IP basada en tecnología Asterisk. Es una solución empaquetada que incluye una plataforma servidor de alto rendimiento, las grandes prestaciones y funcionalidades del reconocido sistema Asterisk y un gestor web para la administración del sistema.

Avanvox acepta la mayoría de los estándares de telefonía del mercado, tanto los tradicionales como los de telefonía IP permitiendo la conectividad a redes públicas de telefonía tradicional e integrarse fácilmente con otras. Esto hace que esta solución sea óptima para empresas que quieren mejorar sus procesos internos y disminuir los costes. (11)



Figura 6 Dispositivos de Avanvox

1.4 PBX basadas en Asterisk más utilizadas internacionalmente

1.4.1 Digium

Digium es el creador y desarrollador primario de Asterisk, la primera de las PBX de código abierto de la industria. Las soluciones de Digium reducen el costo de las implementaciones tradicionales de TDM y VoIP a través de un software basado en estándares de código abierto y gateways de última generación, software de procesamiento de voz y servidores de aplicaciones. El hardware de Digium soporta los protocolos de voz tradicionales incluyendo PRI, RBS, FXS, FXO, E&M, Groundstart, Loopstart, y GR-303. Los protocolos de datos incluyen PPP, Cisco HDLC y Frame Relay. En cuanto a protocolos de compresión de voz Asterisk soporta IAX, (Inter-Asterisk eXchange), SIP, MGCP, Skinny (SCCP), y H.323 (12).

1.4.2 Fonality

Fonality es un líder en sistemas de telefonía IP de código abierto basados en Asterisk®. PBXtra, la galardonada línea de productos IP-PBX de Fonality creada para pequeñas y medianas empresas, es asequible, fácil de usar, confiable, escalable y tiene características de clase empresarial. (13)

1.5 PBX basadas en Asterisk más utilizadas en Cuba

A pesar de que aún es incipiente el uso de la telefonía IP en Cuba, varias son las incursiones hechas en el tema en el país. Tal es el caso del sistema de facturación Asterisk2Billing en la plataforma de telefonía IP, Asterisk PBX, instalado en la Facultad de Ingeniería Eléctrica de la

Universidad Central "Marta Abreu" de Las Villas. Es una solución de código libre y abierto que cumple con las exigencias de la economía y las necesidades potenciales, y lo convierten en un instrumento necesario para el día de hoy, y cualquier sistema de facturación basado en la plataforma de servidor de telefonía IP PBX Asterisk.

En la empresa CIMEX se validó una solución integradora de Asterisk y CCME (Cisco Callmanager Express) y a pesar de tener como desventaja la necesidad de manejar dos sistemas PBXs IP, permitió aprovechar lo mejor de cada uno. Se obtuvo así un modo de agregar nuevas funcionalidades a CCME a un bajo precio. Se verificó el correcto funcionamiento de Asterisk como servidor de correo de voz para CCME y se demostró el ahorro económico que ello implica con respecto a emplear otros productos de la solución de telefonía IP de Cisco. (14)

Por otra parte, ETECSA, la empresa líder de las telecomunicaciones en Cuba, evaluó y seleccionó la solución PBX Asterisk como alternativa al insuficiente nivel de redundancia que presentan las plataformas de los Centros de Llamada de ETECSA en el país, aprovechando las ventajas de la VoIP y el Software Libre para alcanzar este objetivo. La selección de la variante fue el resultado del análisis costo-beneficio de diferentes tecnologías y la propuesta presentada no sólo pretende resolver la problemática detectada, sino que también puede contemplarse para la implementación de nuevos centros o como soporte a crecimientos en arquitecturas en explotación. Esta alternativa puede optimizar, de manera económica, un servicio con importantes retos.

Durante los últimos 3 años en la Facultad 2 de la Universidad de las Ciencias Informáticas, como parte de los proyectos del perfil de Telecomunicaciones, se han dedicado esfuerzos a la producción de software capaz de brindar a los usuarios de ETECSA la mayor cantidad de servicios posibles utilizando las bondades de las centrales telefónicas digitales, primero se realizó el proyecto Call Center con el objetivo de crear una solución de este tipo capaz de competir a nivel internacional, luego el proyecto pasa a llamarse PLATEL Comunicaciones Unificadas e incluye entre sus objetivos la incorporación de la telefonía IP con el uso de la plataforma Asterisk como base y la integración además de servicios de correo electrónico, mensajería instantánea y fax.

1.6 Metodologías de Desarrollo de Software

Para la elaboración de sistemas de cierta envergadura es totalmente necesario apoyarse en una metodología de desarrollo que guíe el proceso de creación del software. Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería.

Hoy en día existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Un ejemplo de ellas son las propuestas tradicionales centradas específicamente en el control del proceso. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Estas metodologías han demostrado ser efectivas y necesarias en un gran número de proyectos, sobre todo aquellos proyectos de gran tamaño (respecto a tiempo y recursos).

Sin embargo la experiencia ha demostrado que las metodologías tradicionales no ofrecen una buena solución para proyectos donde el entorno es volátil y donde los requisitos no se conocen con exactitud, porque no están pensadas para trabajar con incertidumbre. Aunque no puede dejar de reconocerse la profundidad y validez de sus aportes.

Aplicar metodologías tradicionales nos obliga a forzar a nuestro cliente a que tome la mayoría de las decisiones al principio. Luego el coste de cambio de una decisión tomada puede llegar a ser muy elevado si aplicamos metodologías tradicionales. (15)

1.6.1 Proceso de Desarrollo y Gestión de Software de la Unidad de Compatibilización Integración y Desarrollo de Software para la Defensa (UCID).

Diversas tendencias y metodologías de desarrollo de software han aparecido en años recientes, buscando resolver los problemas que proyectos más tradicionales, no han conseguido enfrentar. Entre ellas están los frameworks de proyectos, las metodologías ágiles y los modelos de medición de madurez. Junto con estos marcos de trabajo, ciertas estrategias específicas han permitido a los equipos de desarrollo producir software más robusto, predecible, reutilizable y de fácil mantenimiento.

Todos estos elementos se han tenido en cuenta para el desarrollo del Proceso de Desarrollo y Gestión de Software del UCID, que dicta las pautas a seguir para desarrollar aplicaciones informáticas en dicha entidad, analizando y clasificando los tipos de proyectos así como los interesados en cada uno de ellos.

Las principales características del ciclo de vida expuesto en esta organización son:

- ✓ Las fases son secuenciales y su transferencia debe ser precedida por un proceso de revisión o liberación del Centro de Calidad y su aprobación en Consejo Técnico Formal.
- ✓ El nivel del personal es bajo al comienzo, alcanza su nivel máximo en la fase de construcción y decae rápidamente cuando el proyecto se aproxima a su conclusión.
- ✓ La participación de los interesados es alta en las etapas de Inicio y Modelación, baja en la etapa de Construcción y vuelve a subir en las etapas finales del proyecto.

En el UCID el ciclo de vida del proyecto se considera como parte del ciclo de vida del producto, el cual representa un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en la adquisición, el desarrollo, la explotación y el mantenimiento de un producto.

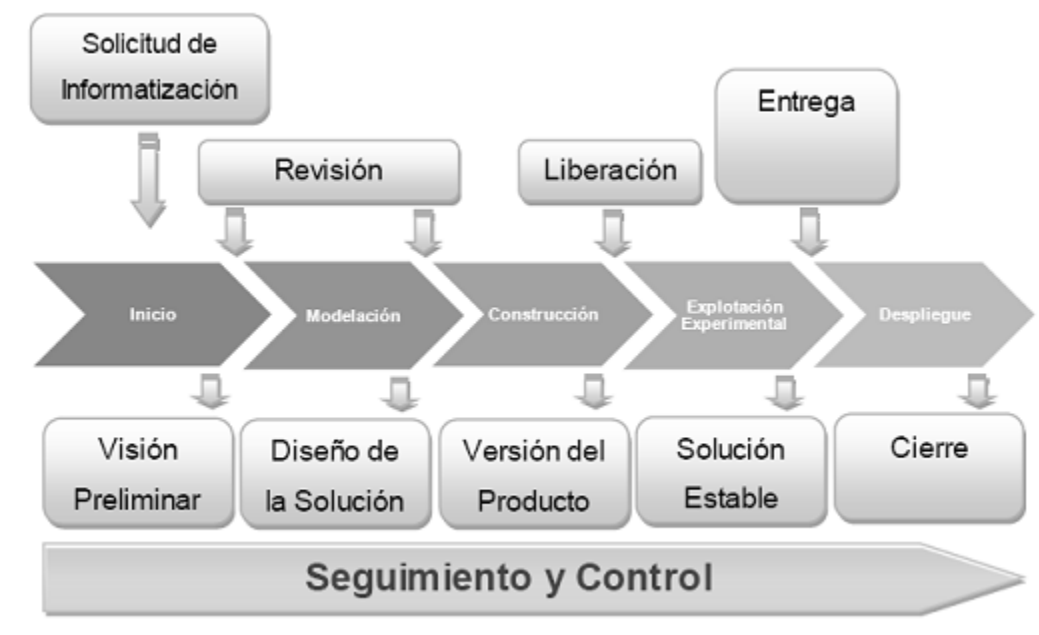


Figura 7 Etapas del ciclo de vida del proyecto

Fases del Ciclo de vida

Las etapas o fases permiten mejorar el planeamiento, ejecución y control del proyecto, se caracterizan por la conclusión y la aprobación de uno o más artefactos entregables siendo estos productos del trabajo, mensurables y verificables, correspondiendo algunos a los

procesos de gestión de proyectos y otros al producto. La consecución exitosa de cada fase es indispensable para poder continuar con el proyecto. El ciclo de vida de un proyecto de software desarrollado en dicha entidad se descompone en el tiempo en cinco fases secuenciales que son: Inicio, Modelación, Construcción, Explotación Experimental, Despliegue. Al final de cada fase los representantes de los grupos de roles presentes en el proyecto realizan una evaluación para determinar si los objetivos se cumplieron y así presentar a Consejo Técnico Formal para su evaluación y dar paso o no a la fase siguiente. (17)

1.7 Framework

Un framework es una plataforma para la implementación de una aplicación. Simplifica el desarrollo mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, proporciona estructura al código fuente, forzando al desarrollador a adicionar código más legible y estético. Por último, facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Ventajas del uso de un framework:

- ✓ El programador no necesita plantearse una estructura global de la aplicación, sino que el framework le proporciona un esqueleto que hay que "rellenar".
- ✓ Facilita la colaboración (todo lo que sea definir y estandarizar va a ahorrar tiempo y trabajo a los desarrollos colaborativos).
- ✓ Es más fácil encontrar herramientas (utilidades, librerías) adaptadas al framework concreto para facilitar el desarrollo. (18)

A continuación se exponen las principales herramientas que se han escogido por parte de la dirección del proyecto para el desarrollo y fructífera implementación del sistema. Estas tecnologías están fuertemente relacionadas con el framework, por esa razón se han insertado en este acápite.

1.7.1 Marco de Trabajo SAUXE 1.5.4

Para el desarrollo del sistema se seleccionó SAUXE 1.5.4, el marco de trabajo del centro UCID para desarrollar aplicaciones web de gestión. SAUXE 1.5.4 se basa en la arquitectura en capas (presentación, negocio, servicio, dominio y acceso a datos, datos) y emplea el MVC (Modelo

Vista Controlador) para llevar a cabo la comunicación entre la capa de presentación y el negocio. Este marco de trabajo emplea como lenguaje de programación PHP, como gestor de base de datos PostgreSQL además define una serie de tecnologías y herramientas, entre las que se encuentran:

ExtJS

ExtJS 2.2 ó 3.1 (Recomendado ExtJS 2.2 modificado por la Línea de arquitectura del Centro de Tecnología de la UCID). Es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas. Posee características positivas como: gran desempeño, componentes de interfaz de usuario personalizables, buen diseño y amplia documentación.

Doctrine

Framework utilizado para el desarrollo de aplicaciones PHP 5.2.3 ó superior que utilicen bases de datos; este framework implementa el patrón ORM (Object Relational Mapping) para desarrollar el dominio de una aplicación y cuenta con una capa de abstracción para el acceso a bases de datos y un lenguaje de consulta propio que se abstrae del gestor que se está utilizando.

Zend Framework

Framework usado para desarrollo de aplicaciones con PHP 5.2.3 ó superior – recomendado PHP 5.2.6 utilizando los componentes Zend_View, Zend_Controller. (19)

1.8 Gestor 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.

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Se trata de un conjunto de programas no visibles al usuario final que se encargan de la privacidad, la integridad, la

seguridad de los datos y la interacción con el sistema operativo. Proporciona una interfaz entre los datos, los programas que los manejan y los usuarios finales. (20)

1.8.1 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 incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional.

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. (21)
- ✓ Corre en casi todos los sistemas operativos: Linux, Unix, Windows, etc.
- ✓ Cuenta con soporte nativo para los lenguajes más populares: PHP, C, C++, Perl, Python, etc.

Este gestor puede ser utilizado, modificado y distribuido por cualquiera gratuitamente, para cualquier propósito ya sea con fines privados, comerciales o académicos.

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 sistema a desarrollar.

1.9 Servidor Web

El servidor web es un programa que se aloja en una computadora que escucha peticiones HTTP las cuales son atendidas y procesadas por este programa. Atendiendo a las peticiones hechas, gestiona las páginas web o ejecuta códigos en el servidor para dar respuesta, ya sea información o un mensaje detallado de error. (22)

1.9.1 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.

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.
- ✓ Rápido y eficiente.

Ventajas:

- ✓ Ayuda en la mejora del posicionamiento.
- ✓ Este servidor junto con el módulo mod_rewrite puede convertirse en una herramienta muy útil para Adicionar páginas con enlaces amigables para los buscadores.
- ✓ Es un software libre.
- ✓ Open Source.
- ✓ Modular.
- ✓ Extensible.
- ✓ Presenta mensajes de error altamente configurables. (23)

1.10 Lenguajes de Programación

La AGI (del inglés: Asterisk Gateway Interface), proporciona una interfaz estándar externa por la que los programas pueden controlar el plan de discado de Asterisk. Por lo general, los scripts de la AGI son utilizados para desarrollar lógica avanzada, comunicarse con bases de datos relacionales (PostgreSQL, MySQL entre otras), y acceder a otros recursos externos que de otra manera resultaría difícil o imposible. La PBX Asterisk brinda una interfaz estándar para los scripts de la AGI, los cuales pueden ser escritos en casi todos los lenguajes de programación modernos, pero los lenguajes más comúnmente utilizados y que se destacan en la programación AGI son: Java Server Page(JSP), Hypertext Pre-processor (PHP), Python, C, C++, Perl, entre otros. (24)

1.10.1 PHP

Es un lenguaje de programación utilizado para la creación de sitios 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 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). (24)

Ventajas:

- ✓ Muy fácil de aprender.
- ✓ 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.
- ✓ Su página oficial posee documentación en 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:

- Se necesita instalar un servidor web.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.

Basado en las ventajas antes descritas se decidió seleccionarlo para la implementación y desarrollo de la herramienta que nos ocupa. Las desventajas que presenta este lenguaje pueden ser solventadas de fácil manera, al instalar Apache como servidor web y evitando mezclar las sentencias HTML y PHP.

1.11 Modelado de software.

El modelado de sistemas de software es una técnica para tratar con la complejidad inherente a estos sistemas. El uso de modelos ayuda al desarrollador a "visualizar" el sistema a construir, los modelos de un nivel de abstracción mayor pueden utilizarse para la comunicación con el

cliente. Además, las herramientas de modelado y las de Ingeniería de Software pueden ayudar a verificar la corrección de modelos que constituyen una fuente de documentación insustituible en el proceso de desarrollo del software.

1.11.1 El Lenguaje Unificado de Modelado

(UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Desde el año 2005 UML es un estándar aprobado por la ISO como ISO/IEC 19501:2005 Information technology — Open Distributed Processing — Unified Modeling Language (UML) Versión 1.4.2. Su evolución se ha visto confirmada con el UML 2.0, que incluye 13 tipos de diagramas y diversas funcionalidades que no se veían en la versión anterior, entre ellas, darle respaldo a BPMN (Business Process Modeling Notation, Notación para el Modelado de Procesos de Negocio).

1.11.2 Herramienta seleccionada para el modelado, Visual Paradigm para UML

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software. Permite modelar todo tipo de diagramas UML, generar código desde diagramas, generar documentación, realizar ingeniería tanto directa como inversa, entre otros. Este software permite crear modelos de gran calidad en poco tiempo y a un menor costo, ya que presenta una licencia gratis para uso no comercial. (26)

Visual Paradigm para UML fue seleccionado como herramienta CASE para el modelado del sistema debido a que además de emplear UML como lenguaje de modelado, soporta la última versión de BPMN y un generador de mapas de objeto-relacional para Java, .NET y PHP. A diferencia de otras herramientas de este tipo es multiplataforma y posee una interfaz amigable que permite un desenvolvimiento mayor por parte del equipo de desarrollo, que además ya se ha familiarizado con ella, pues tiene un papel protagónico en el estudio de la Ingeniería de Software como asignatura.

Conclusiones.

Teniendo en cuenta el estudio realizado anteriormente sobre las tendencias actuales en cuanto a los paradigmas de tecnologías y herramientas más utilizados en el mundo de la informática y las características del centro UCID, así como las especificidades de la herramienta que se pretende implementar, se decide desarrollar una aplicación web empleando el Proceso de Desarrollo de Software de dicho centro a modo de metodología, PHP como lenguaje de programación, PostgreSQL como gestor de bases de datos y usando el framework SAUXE 1.5.4, que contiene ExtJS, Doctrine, Zend Framework. Se determinó utilizar UML como lenguaje de modelado y la herramienta case para representar todos los modelos y diagramas que surjan como artefactos del proceso de desarrollo será Visual Paradigm para UML.

Capítulo 2: “Características del sistema”

2.1 Objeto de estudio y situación problemática.

Las Fuerzas Armadas Revolucionarias de Cuba intentan implementar una Plataforma Telefónica que les permita gestionar y llevar un control de las llamadas y las comunicaciones telefónicas entre sus unidades a lo largo de toda la isla. Para ello utilizan la distribución TrixBox basada en Asterisk, lo cual dificulta en gran medida el trabajo para el personal de las unidades, ya que para poder manejar este sistema se requieren conocimientos avanzados de informática, además de no poseer esta distribución la robustez suficiente para soportar una gran cantidad de conexiones.

Los servicios que brinda cualquier PBX solo pueden funcionar una vez configurados satisfactoriamente en el servidor, este trabajo implica establecer y modificar datos sobre ficheros de configuración manualmente y resulta muy engorroso, máxime cuando no es especialista en la materia la persona que se enfrenta a esta labor.

Un troncal es un enlace que interconecta las llamadas externas de una central telefónica, concentrando y unificando varias comunicaciones simultáneas en una sola señal para un transporte y transmisión a distancia más eficiente (generalmente digital) y poder establecer comunicaciones con otra central o una red entera de ellas.

Dentro de los servicios que debe brindar la PBX se encuentra el enrutamiento de llamadas y la interconexión con troncales a través de distintos protocolos. Cada uno de los protocolos es configurado a través de un fichero de configuración, por ejemplo, en el caso del protocolo SIP, para poder añadir un nuevo troncal, debe configurarse el fichero sip.conf, y escribir correctamente con una sintaxis determinada los parámetros de la nueva conexión, esto en ambos servidores, lo mismo el emisor que el receptor de la llamada. Lo mismo ocurre con los protocolos IAX, que se configura en iax.conf, el H.323, en el h323.conf y el Dahdi en el dahdi.conf. Ocurre de forma similar para el enrutamiento de llamadas, el ruteo se debe indicar en el fichero extensions.conf.

Se hace imprescindible la necesidad de implementar un sistema que sea capaz de garantizar la configuración de estos ficheros de forma cómoda, amigable y segura, teniendo en cuenta que un ligero cambio en el texto de estos archivos puede dar al traste con la configuración de

otros troncales y provocar fallas en las comunicaciones y respetando los distintos niveles de usuarios que pueden interactuar con el sistema.

2.2 Objeto de automatización.

Según lo dicho anteriormente, urge automatizar el proceso de enrutamiento de llamadas, así como la interconexión con troncales por distintos protocolos.

Para el enrutamiento de llamadas el sistema tiene que ser capaz de posibilitar la configuración de forma simple y sencilla, a través de una aplicación web del fichero `extensions.conf`. En este último es necesario escribir sentencias determinadas respetando una sintaxis establecida para ello. A continuación se muestra un ejemplo.

```
; extensions.conf
;
[general]

[globals]

[internos]
exten => _1XX,1,dial(SIP/${EXTEN},50)
exten => _1XX,2,voicemail(u${EXTEN})
exten => _1XX,102,voicemail(b${EXTEN})

exten =>
999,1,voicemailmain(${CALLERIDNUM})
```

Figura 8 Fragmento del fichero de configuración `extensions.com`

En el caso de los troncales, como pueden conectarse a través de disímiles formas, las Fuerzas Armadas han priorizado, dado los intereses del Ministerio, los siguientes protocolos, atendiendo también a la utilidad y uso de estos:

- SIP
- IAX
- H.323
- Dahdi

Asterisk establece para cada uno de estos, un fichero de configuración, mediante el cual se gestionan los troncales de formas bastante similares.

2.2.1 Información que se maneja.

Evidentemente se hace necesaria la manipulación y escritura sobre los ficheros mencionados anteriormente, para lo cual será necesario tener privilegios administrativos o permisos definidos por el administrador del sistema.

2.3 Propuesta de sistema.

El sistema que se pretende desarrollar deberá permitir al usuario la gestión de troncales, así como el enrutamiento de llamadas. Para ello se contará con una interfaz que permitirá realizar estas acciones de manera sencilla e intuitiva.

El sistema mostrará una lista con todos los troncales en pantalla, de forma tal que el usuario podrá elegir el que desea modificar o eliminar. La interfaz de modificación y actualización de los parámetros de una conexión será idéntica a la interfaz para insertar con la diferencia que ya aparecerán en los campos los antiguos datos, de forma tal que no haya que poner todos los datos nuevamente.

De esta forma estaríamos realizando el proceso de configuración de una forma mucho más eficiente y más sencilla que como se hace actualmente, respetando los distintos niveles de usuarios existentes y satisfaciendo esta necesidad de las FAR.

2.4 Modelo conceptual.

Un troncal es una extensión con la propiedad de interconectar dos plataformas telefónicas. Para ello es necesario configurarle la forma en que se conectará con la misma, utilizando las rutas entrantes y salientes. La conexión puede ser a través de distintos protocolos: entre los que se encuentran SIP, IAX, H 323, y Dahdi. En el Anexo 1 se detallan todos los elementos relacionados con el negocio en el Modelo Conceptual.

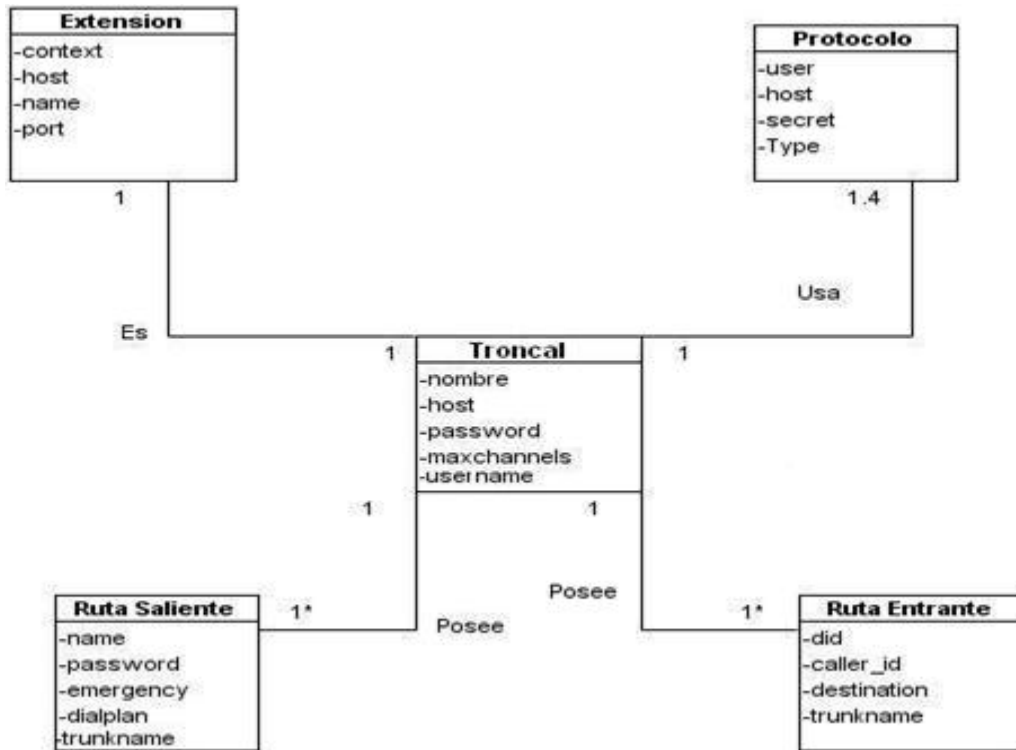


Figura 9 Modelo Conceptual

2.5 Especificación de los requisitos de software.

2.5.1 Requerimientos funcionales

- 2.5.1.1 Adicionar troncal
- 2.5.1.2 Modificar troncal
- 2.5.1.3 Eliminar troncal
- 2.5.1.4 Adicionar ruta entrante
- 2.5.1.5 Modificar ruta entrante
- 2.5.1.6 Eliminar ruta entrante
- 2.5.1.7 Adicionar ruta saliente
- 2.5.1.8 Modificar ruta saliente
- 2.5.1.9 Eliminar ruta saliente

2.5.2 Requerimientos no funcionales.

2.5.2.1 Requerimientos de apariencia o interfaz externa

La apariencia del sistema propuesto cumplirá con los estándares de diseño establecidos por el centro UCID. Se utilizan colores claros, con un contraste agradable a la vista y un tamaño de fuente apropiado para una lectura fácil. La interfaz será amigable, fácil de usar, sencilla, asequible para todo tipo de usuarios, de forma tal que los usuarios sin conocimientos básicos de informática puedan interactuar con ella de forma intuitiva.

2.5.2.2 Requerimientos de usabilidad

Los usuarios con privilegios de administración deberán tener conocimientos avanzados en el manejo de Software de Gestión así como en los temas referentes a Plataformas telefónicas. El resto de los usuarios se recomienda impartirle un curso básico sobre el manejo e interacción con el sistema que se propone.

2.5.2.3 Requerimientos de rendimiento

Para poder lograr un rendimiento óptimo del sistema en cuestión, se recomienda que la máquina que funcione como servidor esté destinada solo para esta actividad, es decir, que no existan otro tipo de procesos corriendo simultáneamente con los servicios del servidor. En el caso de la red, también se debe velar por un correcto funcionamiento de la misma, ya que la eficiencia y la calidad de las comunicaciones dependerán del aprovechamiento de los recursos de la red.

2.5.2.4 Requerimientos de portabilidad

El sistema será multiplataforma, es decir, compatible con los sistemas operativos y distribuciones más usados a nivel mundial, tales como Windows NT en adelante y Linux. Debe ocurrir lo mismo con los navegadores para visualizarlo, podrá usarse cualquiera de los actuales, Opera, Internet Explorer, Firefox, etc. En cuanto al Sistema Gestor de Base de datos aunque se recomienda PostgreSQL, debe presentar capacidad de integración con los gestores mas usados mundialmente, MySQL, Oracle, entre otros.

2.5.2.5 Requerimientos de seguridad

La información manejada por el sistema no será pública, de ahí la necesidad de protegerla de usuarios no autorizados, manipulación inadecuada y estados de inconsistencia. Se implementará de forma tal que cada usuario tenga acceso a la información que esté prevista para su rol. En todo momento la información debe estar disponible para los usuarios con acceso autorizado, y debe preservarse la integridad de los datos.

2.5.2.6 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 requieren como mínimo 512 MB de RAM y 20 GB de disco duro. Todos los nodos involucrados en la funcionalidad de la aplicación deben estar conectados a una red que requiere como mínimo 100 Mbps de velocidad.

2.5.2.7 Requerimientos de software.

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

2.5.3 Descripción de los requisitos funcionales.

2.5.3.1 Especificación del requisito Adicionar Troncal

Conceptos tratados	Conceptos	Atributos
	Troncal	name, host, username, password, max channels, protocol
Precondiciones	Precondiciones	Pre-requisito
	El usuario debe	Autenticar usuario

	estar autenticado como cliente.	
Descripción	<ol style="list-style-type: none"> 1. Se selecciona la opción “Gestionar Troncal”. 2. Se muestra la interfaz correspondiente con las opciones: <ul style="list-style-type: none"> • Adicionar • Modificar • Eliminar 3. Se selecciona la opción “Adicionar”. 4. Se introducen los datos que se solicitan. 5. Se presiona el botón Aceptar. 6. Se verifica que todos los campos estén llenos. 7. Se guardan los datos y se muestra el mensaje de información: “Se ha adicionado correctamente el troncal”. 8. Se presiona el botón Aceptar. 	
Validaciones	Verificar que estén llenos los campos correctamente.	
Post-condiciones	Queda creado el troncal.	
Post-requisito	No procede	

Tabla 1 Especificación del requisito Adicionar Troncal

2.5.3.2 Especificación del requisito Modificar Troncal

Conceptos tratados	Conceptos	Atributos
	Troncal	name, host, username, password, max channels, protocol
Precondiciones	Precondiciones	Pre-requisito

	El usuario debe estar autenticado como cliente.	Adicionar Troncal.
Descripción	<ol style="list-style-type: none"> 1. Se selecciona la opción "Gestionar Troncal". 2. Se muestra la interfaz correspondiente con las opciones: <ul style="list-style-type: none"> • Adicionar • Modificar • Eliminar 3. Se selecciona el troncal que desea modificar. 4. Se selecciona la opción "Modificar". 5. Se muestra la interfaz correspondiente con los datos llenos para que puedan ser modificados. 6. Se introducen los cambios que se deseen. 7. Se presiona el botón Aceptar. 8. Se verifica que todos los campos estén llenos. 9. Se guardan los datos y se muestra el mensaje de información: "Ha sido modificado correctamente el troncal". 10. Se presiona el botón Aceptar. 	
Validaciones	Verificar que se llenen todos los campos.	
Post-condiciones	Queda modificado el troncal.	
Post-requisito	No procede	

Tabla 2 Especificación del requisito Modificar Troncal

2.5.3.3 Especificación del requisito Eliminar Troncal

Conceptos	Conceptos	Atributos
-----------	-----------	-----------

tratados	Troncal	name, host, username, password, maxchannels, protocol
Precondiciones	Precondiciones	Pre-requisito
	El usuario debe estar autenticado como cliente.	Adicionar troncal.
Descripción	<ol style="list-style-type: none"> 1. Se selecciona la opción "Gestionar Troncal". 2. Se muestra la interfaz correspondiente con las opciones: <ul style="list-style-type: none"> • Adicionar • Modificar • Eliminar 3. Se selecciona el troncal que se desea eliminar. 4. Se selecciona la opción "Eliminar". 5. Se muestra el mensaje de confirmación, ¿Está seguro que desea eliminarlo? Los datos asociados al mismo se perderán completamente. 6. Se presiona el botón Aceptar. 7. Se muestra el mensaje de información: "Ha sido eliminado satisfactoriamente el troncal". 8. Se presiona el botón Aceptar. 	
Validaciones	Verificar que el troncal no se esté usando en otro lugar.	
Post-condiciones	Queda eliminado un troncal.	
Post-requisito	No procede.	

Tabla 3 Especificación del requisito Eliminar Troncal

La descripción de los requisitos relacionados con la gestión de rutas entrantes y salientes aparecen en el Anexo 2.

2.5.4 Prototipos de interfaz de usuario.

Se muestran seguidamente los prototipos de interfaz de usuario relacionados con la gestión de troncales, la gestión de rutas entrantes y salientes presentan interfaces similares que se pueden apreciar en el Anexo 3.

Name	Username	Max channels	Host	Protocol
Santiago de Cuba	Stgo	5	stgo.cu	SIP
Guantánamo	Gtmo	5	gtmo.cu	IAX
Holguín	holguin	5	hol.cu	H323
Cuba	cuba	8	cuba.cu	SIP
Uci	UCI	876	uci.cu	Dahdi
Cujae	cujae	9	ispjae.cu	H323
UH	Yanet	8	10.51.83.9	IAX

Figura 10 Interfaz para gestionar troncal.

Figura 11 Interfaz para insertar un troncal

The 'Registrar Troncal' dialog box contains a section titled 'Configuración de los Troncales'. It features six input fields: 'Name', 'Host', 'Username', 'Password', 'Max Channels', and 'Protocol'. The 'Protocol' field is a dropdown menu currently showing '[Seleccionar]'. At the bottom, there are three buttons: 'Cancelar', 'Aplicar', and 'Aceptar'.

The 'Modificar Troncal' dialog box contains a section titled 'Configuración de los Troncales'. It features six input fields: 'Name' (containing 'Guantánamo'), 'Host' (containing 'gtmo.cu'), 'Username' (containing 'Gtmo'), 'Password' (empty), 'Max Channels' (containing '5'), and 'Protocol' (a dropdown menu showing 'IAX'). At the bottom, there are two buttons: 'Cancelar' and 'Aceptar'.

Figura 12 Interfaz para modificar un troncal

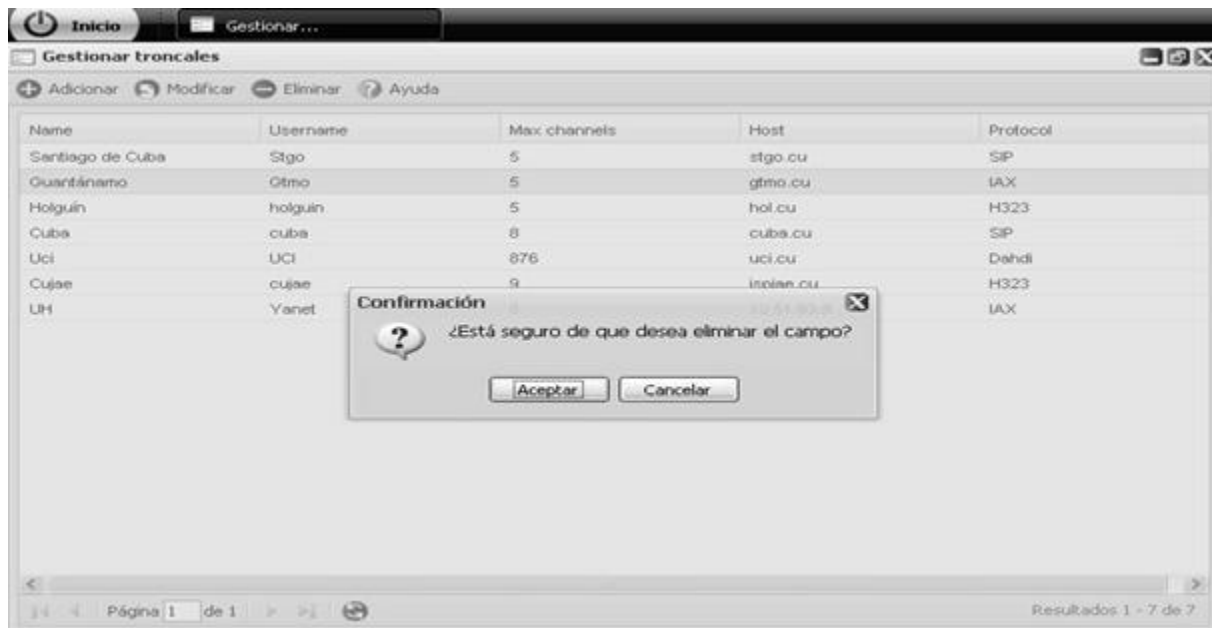


Figura 13 Interfaz para eliminar un troncal

Capítulo 3: "DISEÑO DEL SISTEMA"

Introducción

En este capítulo se hará una descripción detallada del diseño del sistema, apoyada en los diagramas de las funcionalidades más importantes de la aplicación a realizar.

3.1 Diseño

Este momento del proceso de desarrollo de software se propone comprender perfectamente los requisitos del software y transformarlos a un diseño que indique cómo debe ser implementado el software, determinándose las clases necesarias para llevar a cabo las funcionalidades.

El diseño tiene el propósito de formular los modelos que se centran en los requisitos no funcionales y en el dominio de la solución y prepara el ambiente para la implementación y prueba del sistema.

Seguidamente se muestran los diagramas de secuencia y el modelo de clases del diseño de cada una de las secciones que particularizan el comportamiento de las funcionalidades a desarrollar en el módulo. En particular se detalla la funcionalidad gestionar troncales, los artefactos correspondientes a las funcionalidades gestionar rutas entrantes y gestionar rutas salientes se incluirán en los anexos, específicamente en el Anexo 4 y 5.

3.1.1 Diagramas de interacción.

Los diagramas de interacción tienen diferentes formas, basadas todas ellas en una misma información relacionada con las interacciones entre objetos, pero resaltando cada una un punto de vista de la misma: diagramas de secuencia, diagramas de colaboración. Un diagrama de secuencia (que es el que se expondrá en el documento) muestra una interacción que está organizada como una secuencia temporal. En particular, muestra los objetos que participan en la interacción mediante sus líneas de vida y mediante los mensajes que intercambian, organizados en forma de una secuencia temporal.

DIAGRAMA DE INTERACCION

Gestionar Troncales: Sección Adicionar Troncales



Figura 14 Diagrama de secuencia Gestionar Troncales (sección Adicionar Troncales)

DIAGRAMA DE INTERACCION

Gestionar Troncales: Sección Modificar Troncales

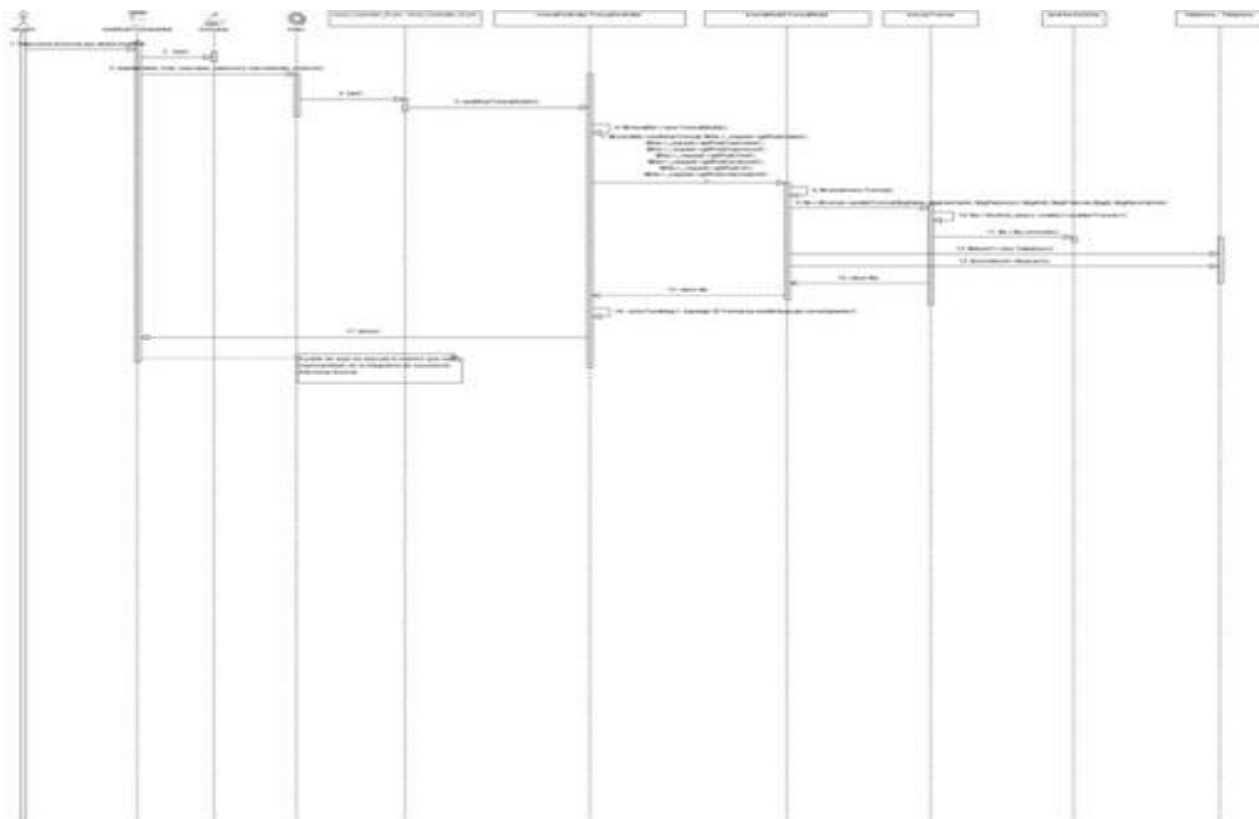


Figura 15 Diagrama de secuencia Gestionar Troncales (sección Modificar Troncales)

DIAGRAMA DE INTERACCION

Gestionar Troncales: Sección Eliminar Troncales

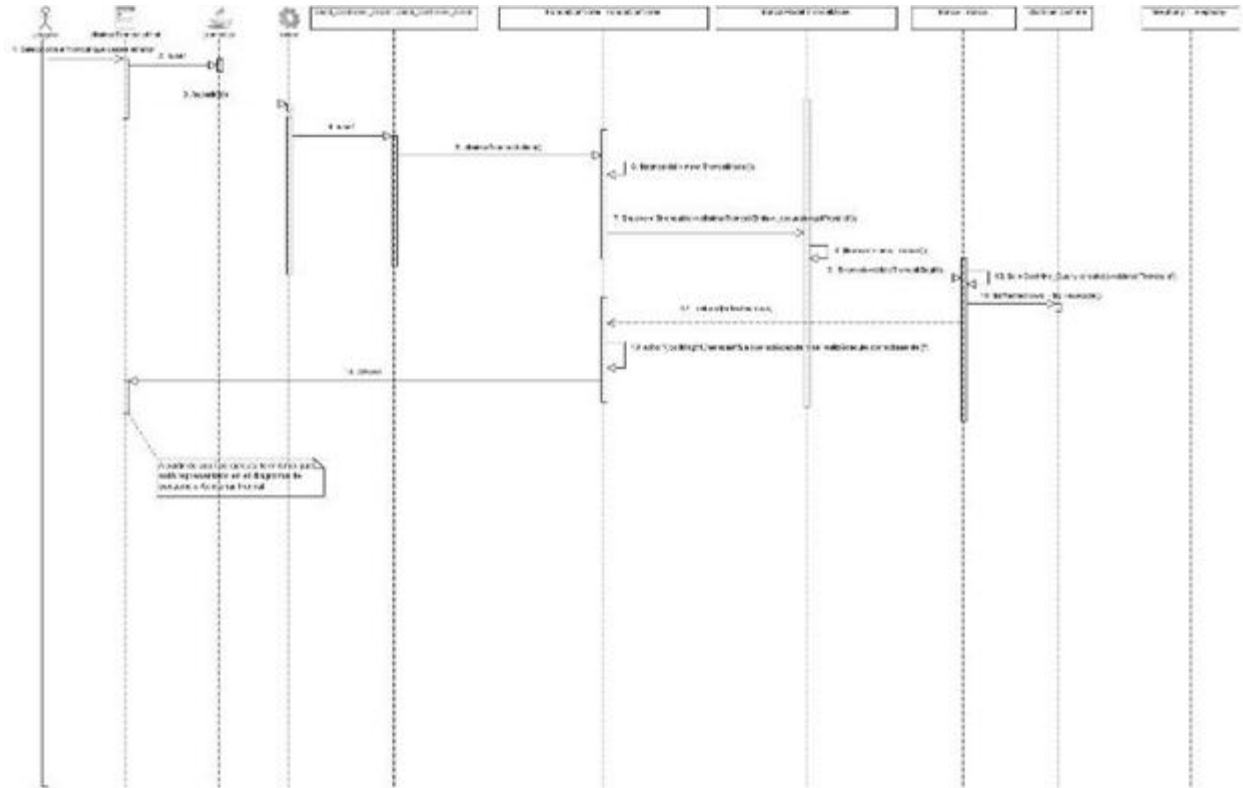


Figura 16 Diagrama de secuencia Gestionar Troncales (sección Eliminar Troncales)

3.1.2 Diagramas de clases.

Los diagramas de clases muestran como se lleva a cabo la colaboración entre las clases para dar cumplimiento a un requisito determinado, para la elaboración de estos diagramas se hace uso de estereotipos que ayudan a representar de manera mas fácil la función y el carácter de las clases dentro de la realización del requisito.

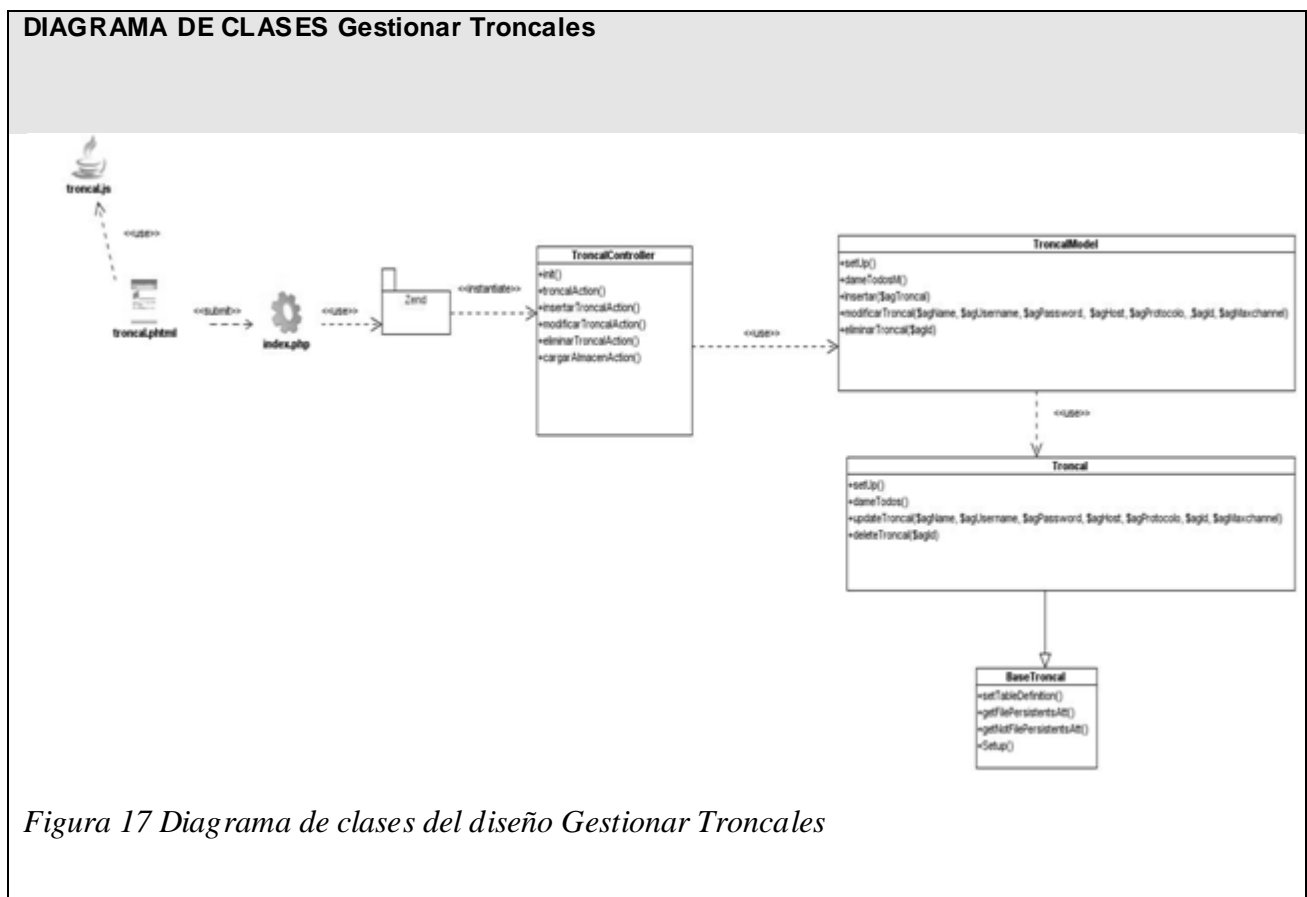


Figura 17 Diagrama de clases del diseño Gestionar Troncales

Descripción de las clases.

A continuación se describen de manera detallada las clases que se definieron en el diseño, destacando los atributos y métodos de cada una de ellas. De manera tal que pueda ser de fácil comprensión para los miembros del equipo de desarrollo.

Nombre: TroncalController	
Tipo de clase (controladora)	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	insertarTroncalAction();
Descripción:	Inserta los troncales.
	modificarTroncalAction();
	Modifica los troncales.
	eliminarTroncalAction();
	Elimina un troncal.
	cargarTroncalAction();
	Se utiliza para obtener los datos guardados y mostrarlos en la interfaz.

Tabla 4 Descripción de la clase GestTroncalController

Nombre: Troncal	
Tipo de clase (entidad)	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	dameTodos()
Descripción:	Devuelve todos los troncales existentes en la base de datos.
	updateTroncal()
	Modifica en la base de datos.

	deleteTroncal(\$aglID)
	Elimina de la base de datos, recibe como parámetro el identificador del troncal a eliminar.

Tabla 5 Descripción de la clase Troncal

Nombre: Troncal	
Tipo de clase (entidad)	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	dameTodos()
Descripción:	Devuelve todos los troncales existentes en la base de datos.
	updateTroncal()
	Modifica el troncal en la base de datos.
	deleteTroncal(\$aglID)
	Elimina de la base de datos, recibe como parámetro el identificador del troncal a eliminar.

Tabla 6 Descripción de la clase GestTroncal.phtml

En el diagrama mostrado en la figura 3.4 aparece el paquete Zend mediando entre la capa que contiene los elementos de la vista y la capa controladora. Este paquete contiene una gran cantidad de clases utilizadas por el framework Zend para su integral funcionamiento, definidas algunas por los creadores del framework y otras por el equipo de arquitectos principales del Centro UCID. El sistema que se elabora utiliza e interactúa con varias de ellas, como son: Doctrine_Exception para el tratamiento de errores, ZendExt_Platel_Telephony, para el manejo y la interacción con Asterisk. Para hacer más directa y sencilla la comprensión de este diagrama, se incluyen solo las clases fundamentales que intervienen en el desarrollo de la aplicación, agrupando en el paquete Zend los elementos generales que no fueron elaborados específicamente para el sistema.

3.2 Diseño de la BD

El principal instrumento para lograr los niveles de abstracciones que conforman la arquitectura de un Sistema de Base de Datos (nivel externo, lógico global o conceptual y nivel interno), es precisamente el Modelo de Datos; que es una representación de la realidad que conserva sólo los detalles relevantes de esa realidad que se está modelando. Es un conjunto de conceptos que pueden servir para describir la estructura de una BD, esto se refiere a tipos de datos, sus vínculos y las restricciones que deben cumplir estos datos.

3.2.1 Modelo de Datos

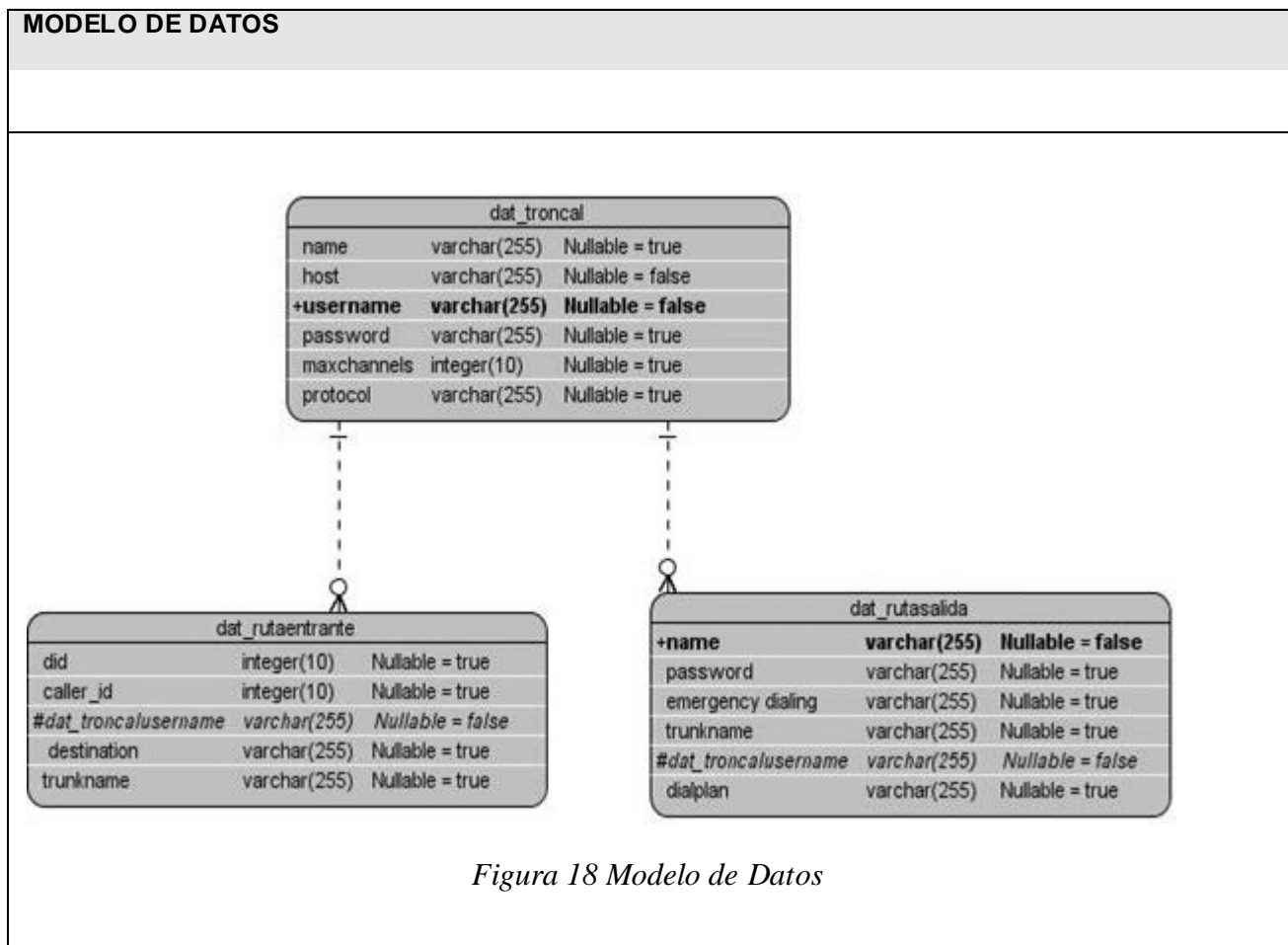


Figura 18 Modelo de Datos

Descripción de las tablas.

Nombre: dat_troncal		
Descripción: Permite la interconexión entre centrales o plataformas telefónicas.		
Atributo	Tipo	Descripción
id	int	Es el identificador de la tabla.
name	varchar	Expresa el nombre del troncal.
host	varchar	Indica la dirección del servidor donde se encuentra el troncal.
username	varchar	Indica el nombre de usuario requerido para acceder a conectar el troncal.
password	varchar	Indica la contraseña de usuario requerida para acceder a conectar el troncal.
maxchannels	int	Indica la cantidad de canales de llamadas permitidos del troncal.
protocol	varchar	Expresa el protocolo de señalización que soporta el troncal.

Tabla 7 Descripción de la tabla dat_troncal

Nombre: dat_rutaentrante		
Descripción: Permite configurar la ruta entrante de un troncal.		
Atributo	Tipo	Descripción
id	int	Es el identificador de la tabla.
did	varchar	Expresa el número de cuenta.
Caller_id	varchar	Expresa el número de identificación de la llamada entrante.
destination	varchar	Indica el destino de la llamada.

trunkname	varchar	Expresa el nombre del troncal al que pertenece esta ruta.
-----------	---------	---

Tabla 8 Descripción de la tabla dat_rutaentrante

Nombre: dat_ruta saliente		
Descripción: Permite configurar la ruta saliente de un troncal para poder establecer comunicaciones con otra central telefónica.		
Atributo	Tipo	Descripción
id	int	Es el identificador de la tabla.
name	varchar	Indica el nombre para autenticarse con dicha ruta.
password	varchar	Indica la contraseña para autenticarse con dicha ruta.
emergency dialing	varchar	Indica si será una ruta emergencia.
trunkname	varchar	Expresa el nombre del troncal al que pertenece esta ruta.
Dialplan	varchar	Como debe ser configurado el patrón de marcado.

Tabla 9 Descripción de la tabla dat_rutasaliente

3.3 Patrones de Diseño Utilizados.

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. Se deben tener presente los siguientes elementos de un patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios).

En las diferentes capas del framework utilizado para el desarrollo de la aplicación se utilizan varios patrones, ellos son:

- ✓ Singleton (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.
- ✓ Factory Method (Método de fabricación): Define una interfaz para crear un objeto, pero deja que sean las subclases quienes decidan qué clase instanciar. Permite que una clase delegue en sus subclases la creación de objetos.
- ✓ Front Controller: implica que todas las solicitudes son dirigidas a un único script PHP que se encarga de instanciar al controlador frontal y redirigir las llamadas.
- ✓ Decorator (Envoltorio): Añade funcionalidad a una clase dinámicamente.
- ✓ Experto: Este se encarga de asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.
- ✓ Creador: Este patrón se encarga de asignarle a la clase B la responsabilidad de crear una instancia de clase A. B es un creador de los objetos A.
- ✓ Alta Cohesión: Este patrón se encarga de asignar una responsabilidad de modo que la cohesión siga siendo alta.
- ✓ Bajo Acoplamiento: Este patrón se encarga de asignar una responsabilidad para mantener bajo acoplamiento. Las clases deben comunicarse con un número pequeño de clases tanto como sea posible.

3.4 Tratamiento de errores.

En un sistema en ocasiones ocurren errores que atentan contra su estabilidad y su correcto funcionamiento a la hora de procesar los datos introducidos desde el exterior, en este caso por un usuario; por lo que se hace necesario realizar acciones concretas con el objetivo de mitigar dichos errores. En el lado cliente se validan los datos a través del uso de expresiones regulares para que el usuario introduzca solo caracteres válidos. Desde el servidor, específicamente en las clases controller, se instancia la clase Doctrine_Exception, para controlar cualquier error que haya pasado la frontera de la interfaz impregnándole una mayor robustez al sistema.

3.5 Seguridad

La seguridad está dada por el control del nivel de acceso al módulo, pues se requiere de una autenticación para los usuarios con acceso al mismo. Los permisos son asignados luego por un administrador del sistema o por alguien que tenga permisos para adicionar un nuevo rol en el subsistema de seguridad. Cualquier usuario que quiera acceder a la aplicación debe estar registrado, en caso contrario se procedería a adicionarlo al sistema, para que pueda acceder al modulo que desee.

Conclusiones

En este capítulo se detallaron los elementos pertenecientes al diseño, de manera tal que quedara descrito todas las relaciones entre las clases existentes, así como el diseño del Modelo de Datos del módulo en cuestión.

Capítulo 4: "Implementación y prueba."

Introducción

En este capítulo se muestran los elementos correspondientes a las etapas de implementación y pruebas del sistema. Destacándose el diagrama de componentes y los casos de prueba como los artefactos fundamentales.

4.1 Implementación.

4.1.1 Diagrama de componentes

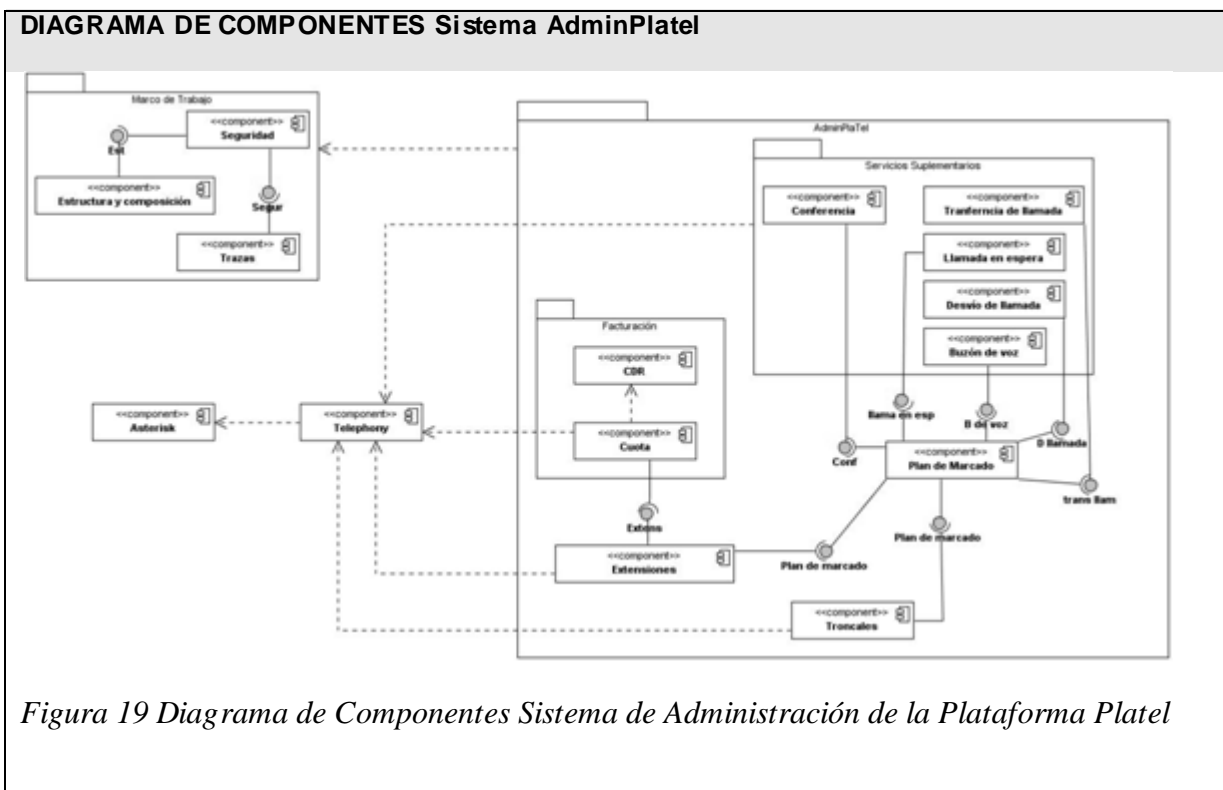
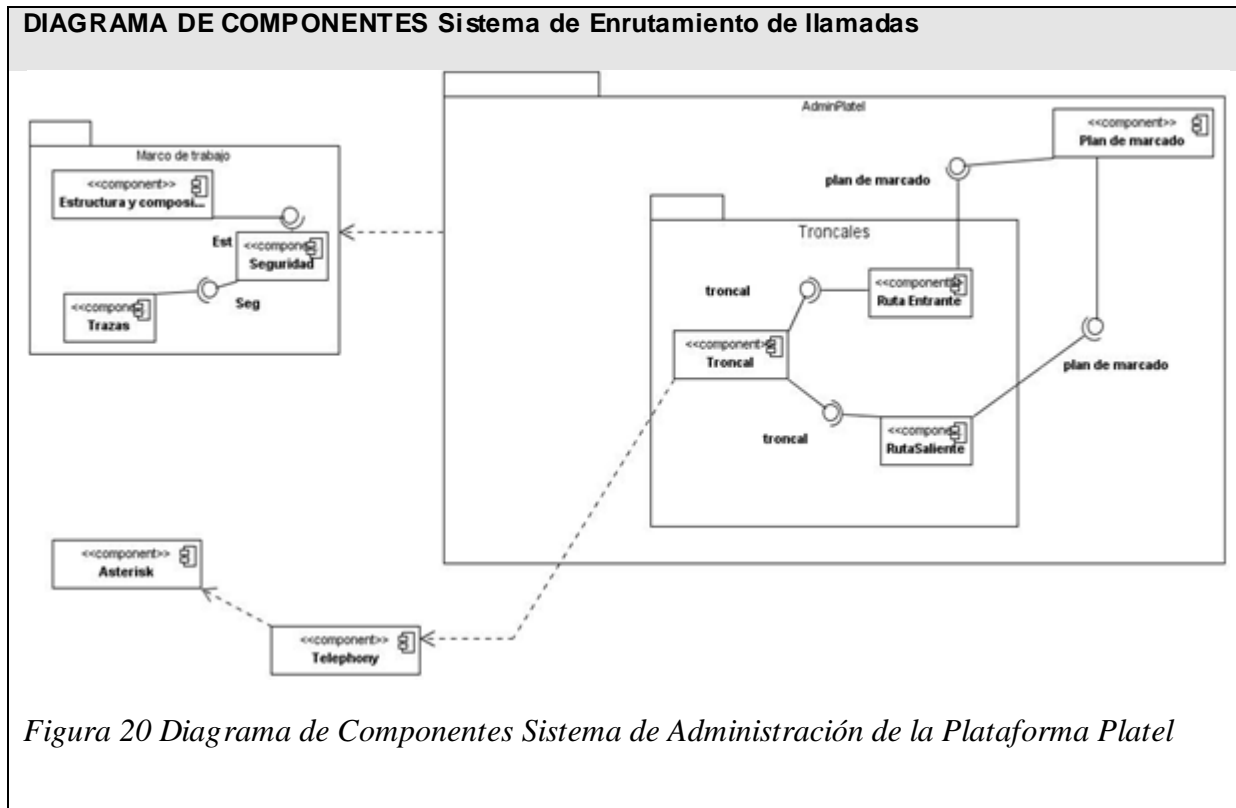


Figura 19 Diagrama de Componentes Sistema de Administración de la Plataforma Platel

4.1.2 Diagrama de componentes



Matriz de Integración de Componentes Internos.

	Componentes Internos	
Componentes Internos	Plan de marcado.	Troncales
Rutas Entrantes	listarcontext();	listarTroncales();
Rutas Salientes	listarContext();	listarTroncales();

Tabla 10 Matriz de Integración de Componentes Internos

Descripción del diagrama de componentes.

Un componente es la parte modular de un sistema, desplegable y reemplazable que encapsula la implementación, un conjunto de interfaces y proporciona la realización de los mismos. Un componente típicamente contiene clases y puede ser implementado por uno o más artefactos (ficheros ejecutables, binarios, etc.). Son las piezas reutilizables de alto nivel a partir de las cuales se pueden construir los sistemas.

El diagrama de componentes del módulo interconexión y enrutamiento de llamadas empaqueta en componentes la seguridad por la cual se podrá acceder al sistema. Relacionado con el marco de trabajo definido por el centro UCID, el módulo interconexión y enrutamiento de llamadas empaqueta un componente por cada información almacenada en esta, o sea, los troncales, las rutas entrantes y las salientes, entre los cuales va a existir una relación de uso.

4.2 Prueba

El desarrollo del software implica una serie de actividades de producción en las que las posibilidades de que aparezca una falla humana son comunes. Los errores pueden empezar a darse desde el primer momento del proceso en el que los objetivos pueden estar especificados de forma errónea e imperfecta; así en los posteriores pasos del diseño y desarrollo. Debido a la imposibilidad humana de trabajar y comunicarse de forma perfecta, el desarrollo del software ha de ir acompañado de una actividad que garantice la calidad. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

4.2.1 Diseños de Casos de Prueba.

Los diseños de casos de prueba se realizaron con el objetivo de demostrar que el sistema reacciona establemente ante fallas de escritura y errores que pueden ser introducidos por los usuarios al interactuar con la interfaz. Las funciones respondieron de forma adecuada y se obtuvo el resultado esperado en cada uno de los escenarios. Estos artefactos(diseños de casos de prueba) pueden apreciarse en el Anexo 6.

4.2.2 Pruebas de Software.

Las pruebas de software se aplican en la fase de explotación experimental, y tienen una gran relevancia, pues a través de ellas afloran una gran cantidad de errores que serían imposibles de mitigar en caso de no aplicarlas. Cada producto a entregar debe ser probado en profundidad para alcanzar un alto nivel de aceptación por parte del cliente.

Como parte de las pruebas antes mencionadas están las unitarias. Van principalmente enfocadas a los elementos más pequeños del software. Mediante ella se prueba una sola unidad, como puede ser una clase o un subsistema determinado. Consiste en una prueba estructural (o caja blanca) y una prueba de especificación (o caja negra).

4.2.3 Método de Prueba.

El método utilizado fue el de caja negra o prueba de especificación, que se basa principalmente en los requerimientos funcionales de la aplicación, obviando la estructura interna del sistema, es decir, realiza pruebas sobre la interfaz y permite descubrir:

- Ausencia y/o errores de algunas funciones.
- Errores de interfaz.

Dentro de las pruebas de caja negra se usó la técnica “Partición de Equivalencia” con la cual se puede:

- ✓ Examinar los valores válidos e inválidos de las entradas existentes en el software.
- ✓ Detectar de forma rápida una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico.

4.2.4 Listado de las No Conformidades

En la tabla 11 se pueden observar un listado con los errores encontrados mediante la aplicación de los casos de pruebas y las medidas tomadas para mitigarlos.

No Conformidades	Acciones Correctivas
La aplicación no carga la pantalla principal.	Se informó al desarrollador de interfaz y fue resuelto al instante.

El sistema no adiciona un nuevo troncal.	Se arregló de forma inmediata después de detectado dicho error.
El sistema no elimina una ruta saliente.	Se arregló de forma inmediata después de detectado dicho error.
Se detectaron parámetros de la interfaz en idioma inglés.	No procede. Es parte de los requisitos.

Tabla 11 Listado de no conformidades y acciones correctivas.

Conclusiones

En este capítulo se describieron los elementos de las etapas de implementación y prueba del módulo. Se elaboró el diagrama de componentes, así como los casos de prueba a aplicarle al sistema para determinar su efectividad. Se realizaron pruebas de caja negra a la aplicación y se detectaron fallas que impedían el funcionamiento correcto del sistema.

Conclusiones Generales y Recomendaciones

Conclusiones Generales

A través de este trabajo de diploma se pudo profundizar en los aspectos referentes a la variedad de plataformas telefónicas existentes así como los servicios que brindan estas. La investigación se centró principalmente en los métodos de interconexión de plataformas telefónicas, el enrutamiento de llamadas y la gestión de troncales. Se elaboraron los artefactos ingenieriles correspondientes a las fases de inicio, modelación, construcción y explotación experimental propuestos por el proceso de desarrollo y gestión de Software del centro UCID, para el módulo de enrutamiento e interconexión de llamadas de la Plataforma Platel. Se adquirieron los conocimientos necesarios para la elaboración e implementación de dicho módulo, logrando un producto terminado, probado y validado. Se pusieron en práctica conocimientos adquiridos durante el transcurso de la carrera.

Recomendaciones

Luego de haber desarrollado una primera versión del Sistema de Interconexión y Enrutamiento de llamadas se recomienda:

- ✓ Continuar el desarrollo de la aplicación dándole cumplimiento a los requisitos que no se han enmarcado en esta primera versión.
- ✓ Desarrollar una herramienta en lenguaje PHP que permita configurar el plan de marcado con expresiones regulares.

Bibliografía

1. **Rojano, Elio.** www.sinologic.net/blog/2008-05/adios-zaptel-hola-dahdi. [En línea] 20 de mayo de 2008.
2. **Certain, Alfred.** *TrixBox al descubierto*. Colombia : Gecko Networks, 2006. 63.
3. **Barrera Zumaquero, Orelvis y González León, Reynier Santiago.** *Trabajo de Diploma: Herramienta administrativa para la plataforma PLATEL*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2009. 87.
4. **Suárez Pérez, Adilenys y Dorvigny Dorvigny, Darvis.** *Trabajo de Diploma Redes de Voz sobre IP mediante PBX basado en Software*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2008. 62.
5. **Molina Sotolongo, Maykel.** Solución a insuficiente nivel de redundancia en la plataforma de los Centros de Llamada. Ciudad Habana, Cuba : s.n., 2008.
6. **Unidad de Compatibilización Integración y Desarrollo de Software para la Defensa.** *Proceso de Desarrollo y Gestion de Proyectos de Software*.
7. **Espinoza, Humberto.** *PostgreSQL Una alternativa de DBMS Open Source*. 2005. pág. 26. Disponible en : http://www.lgs.com.ve/pres/PresentacionES_PSQL.pdf.
8. Nocion Digital. [En línea] [Citado el: 8 de febrero de 2010.] <http://www.nociondigital.com/webmaster/html-tutorial-introduccion-a-dhtml-detalle-285.html>.
9. **Mendoza Sánchez, María A.** *Metodologías de Desarrollo De Software*. 2005.
10. **Hinostroza, Raúl Rodas.** Linux Centro. [En línea] 22 de febrero de 2007. [Citado el: 17 de febrero de 2010.] <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
11. Librosweb.es. [En línea] [Citado el: 5 de febrero de 2010.] http://www.librosweb.es/symfony/capitulo2/el_patron_mvc.html.
12. **White, Stephen A.** *Introduction to BPMN*. pág. 11. Disponible en: http://www.bpmn.org/Documents/Introduction_to_BPMN.pdf.
13. **Álvarez, Sara.** Introducción a este concepto y características especiales. 2007.
14. **Pérez, Javier Eguíluz.** *Introducción a AJAX*. 2008. pág. 251. Disponible en: <http://www.librosweb.es/ajax/index.html>.
15. **Vivar Mesa, Judith, Acosta Viciado, Jaqueline y Alfonso Robaina, Damián.** *Integración de Asterisk y Cisco Callmanager Express: una solución a valorar*.

16. **Ulloa Rojas, Alexander.** <http://todoasterisk.blogspot.com/2009/07/instalacion-de-freepbx-en-un-servidor.html>. [En línea] 2009.
17. Herramientas Case. [En línea] [Citado el: 8 de marzo de 2010.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
18. *Grails: Framework para el desarrollo de aplicaciones web (1era parte).* **López, Esteban Saavedra.** 2009.
19. FreePBX. [En línea] [Citado el: 2 de febrero de 2010.] <http://www.freepbx.org>.
20. Free Download Manager. [En línea] [Citado el: 18 de marzo de 2010.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(Iglesia_Anglicana\)_%5BMac_OS_X_cuenta_14717_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(Iglesia_Anglicana)_%5BMac_OS_X_cuenta_14717_p/).
21. **García, Mayer Horna.** *Ext JS Introducción.* [Flash] 2010.
22. Elastix. [En línea] [Citado el: 3 de febrero de 2010.] <http://www.elastix.org>.
23. *Doctrine ORM for PHP (Guide to Doctrine for PHP).* 2009. pág. 407.
24. **Rodríguez, Juan Manuel y Vitullo, Facundo.** *Desarrollo IP-PBX FIUBA.* 2008.
25. Centro de Datos - Sauxe 154 - Sistema de Planificación de la UCID. [En línea] http://redmine.ucid.uci.cu/projects/plantilpq/wiki/Sauxe_154.
26. **Callfon.** Callfon la PBX-IP de VirtualFon. [En línea] http://www.callfon.com.ar/descripcion_p.php?info=digium.php.
27. Buenas Tareas. [En línea] [Citado el: 23 de febrero de 2010.] <http://www.buenastareas.com/ensayos/Arquitectura-Cliente-Servidor/115409.html>.
28. **Spencer, Mark.** *Asterisk The Future Phone Systems.* Estados Unidos : O'Reilly Media, 2007. 574.
29. **Jarinton, Nicolás Escobar.** Alexandria. [En línea] 2007. [Citado el: 10 de febrero de 2010.] <http://www.alexandria.com.mx/tecnologias.php>.
30. ABCDatos. [En línea] [Citado el: 4 de marzo de 2010.] <http://www.abcdatos.com/webmasters/programa/z2820.html>.
31. A2Billing. [En línea] [Citado el: 5 de febrero de 2010.] <http://www.asterisk2billing.org>.
32. **Delgado Expósito, Eryl.** *Metodologías de desarrollo de software. ¿Cuál es el camino?*

33. **Cuenca, Carlos Luis.** Descripción de la arquitectura en módulos del Apache. *Explicación y enumeración de las funcionalidades asociadas a cada modulo.* [En línea] 2003. <http://www.desarrolloweb.com/articulos/1112.php> .
34. **Fonality.** [En línea] www.fonality.com.
35. **Company, Avavax.** [En línea] www.avavax.com/es/centralita_IP_asterisk_avavax.

Referencias Bibliográficas

1. **Rodriguez, Juan Manuel y Vitullo, Facundo.** *Desarrollo IP-PBX FIUBA.* 2008.
2. A2Billing. [En línea] [Citado el: 5 de febrero de 2010.] <http://www.asterisk2billing.org>.
3. ABCDatos. [En línea] [Citado el: 4 de marzo de 2010.] <http://www.abcdatos.com/webmasters/programa/z2820.html>.
4. **Suárez Pérez, Adilenys y Dorvigny Dorvigny, Darvis.** *Trabajo de Diploma Redes de Voz sobre IP mediante PBX basado en Software.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2008. 62.
5. **Rojano, Elio.** www.sinologic.net/blog/2008-05/adios-zaptel-hola-dahdi. [En línea] 20 de mayo de 2008.
6. **Barrera Zumaquero, Orelvis y González León, Reynier Santiago.** *Trabajo de Diploma: Herramienta administrativa para la plataforma PLATEL.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2009. 87.
7. **Spencer, Mark.** *Asterisk The Future Phone Systems.* Estados Unidos : O'Reilly Media, 2007. 574.
8. Elastix. [En línea] [Citado el: 3 de febrero de 2010.] <http://www.elastix.org>.
9. FreePBX. [En línea] [Citado el: 2 de febrero de 2010.] <http://www.freepbx.org>.
10. **Certain, Alfred.** *TrixBox al descubierto.* Colombia : Gecko Networks, 2006. 63.
11. **Company, Avanvox.** [En línea] www.avanvox.com/es/centralita_IP_asterisk_avanvox.
12. **Callfon.** Callfon la PBX-IP de VirtualFon. [En línea] http://www.callfon.com.ar/descripcion_p.php?info=digium.php.
13. **Fonality.** [En línea] www.fonality.com.
14. **Molina Sotolongo, Maykel.** Solución a insuficiente nivel de redundancia en la plataforma de los Centros de Llamada. Ciudad Habana, Cuba : s.n., 2008.
15. **Delgado Expósito, Eryl.** *Metodologías de desarrollo de software. ¿Cuál es el camino?*
16. **Mendoza Sánchez, María A.** *Metodologías de Desarrollo De Software.* 2005.
17. **Unidad de Compatibilización Integración y Desarrollo de Software para la Defensa.** *Proceso de Desarrollo y Gestion de Proyectos de Software.*

18. *Grails: Framework para el desarrollo de aplicaciones web (1era parte)*. **López, Esteban Saavedra**. 2009.
19. Centro de Datos - Sauxe 154 - Sistema de Planificación de la UCID. [En línea] http://redmine.ucid.uci.cu/projects/plantilpq/wiki/Sauxe_154.
20. **Álvarez, Sara**. Introducción a este concepto y características especiales. 2007.
21. **Espinoza, Humberto**. *PostgreSQL Una alternativa de DBMS Open Source*. 2005. pág. 26. Disponible en : http://www.lgs.com.ve/pres/PresentacionES_PSQL.pdf.
22. Buenas Tareas. [En línea] [Citado el: 23 de febrero de 2010.] <http://www.buenastareas.com/ensayos/Arquitectura-Cliente-Servidor/115409.html>.
23. **Cuenca, Carlos Luis**. Descripción de la arquitectura en módulos del Apache. *Explicación y enumeración de las funcionalidades asociadas a cada modulo*. [En línea] 2003. <http://www.desarrolloweb.com/articulos/1112.php>.
24. **Jarinton, Nicolás Escobar**. Alexandria. [En línea] 2007. [Citado el: 10 de febrero de 2010.] <http://www.alexandria.com.mx/tecnologias.php>.
25. **White, Stephen A**. *Introduction to BPMN*. pág. 11. Disponible en: http://www.bpmn.org/Documents/Introduction_to_BPMN.pdf.
26. Herramientas Case. [En línea] [Citado el: 8 de marzo de 2010.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.

Glosario de términos

ACD: (Distribuidor Automático de Llamadas) Ruta o dirección de la entrada de una llamada telefónica al siguiente operador disponible.

AGI: (Asterisk Gateway Interface) permite extender las funcionalidades de Asterisk mediante el uso lenguajes de programación tales como PHP o PERL. Sirve de enlace entre las aplicaciones externas y el núcleo de Asterisk.

AdminPlate: Nombre del sistema de administración de la Plataforma Platel.

BPMN: (Notación para el Modelado de Procesos de Negocio) Es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo.

C: Lenguaje de programación.

C++: Lenguaje de programación que permite la manipulación de objetos, permite el trabajo con la programación orientada a objetos.

CGI: (Interfaz Común de Pasarela) es una importante tecnología que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web. Especifica un estándar para transferir datos entre el cliente y el programa.

CPL: (Lenguaje de Procesamiento de Llamadas).

Códec: (Es la abreviatura de codificador) Describe una especificación desarrollada en software, hardware o una combinación de ambos, capaz de transformar un archivo con un flujo de datos (stream) o una señal.

CTI: (Integración Teléfono –Computadora) una tecnología que permite la integración y gestión de los diferentes canales de comunicación entre cliente y empresa (siendo el principal el teléfono).

DAHDI: Permite cargar los drivers y configurar distintos tipos de tarjetas (para conectar teléfonos analógicos, líneas telefónicas, líneas digitales, etc.) en Asterisk

Elastix: Es una distribución libre de Servidor de Comunicaciones Unificadas que integra en un solo paquete: VoIP PBX, Fax, Mensajería Instantánea, Correo electrónico, Colaboración.

ETECSA: Empresa de Telecomunicaciones de Cuba S.A.

FAR: Fuerzas Armadas Revolucionarias.

FXO: (Interfaz de Central Externa) es el puerto que recibe la línea analógica. Es un enchufe del teléfono o aparato de fax, o el enchufe de su centralita telefónica analógica. Envía una indicación de colgado/descolgado (cierre de bucle).

FXS: (Interfaz de Abonado Externo) es el puerto que efectivamente envía la línea analógica al abonado. En otras palabras, es el “enchufe de la pared” que envía tono de marcado, corriente para la batería y tensión de llamada es el conector en una central telefónica o en la pared de nuestro hogar, que permite conectar un teléfono analógico.

GPL: (Licencia Pública General) es una licencia orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

HTML: (Lenguaje de Marcado de Hipertexto) es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTTP: (Protocolo de Transferencia de Hipertexto) es protocolo que permite la transferencia de archivos (principalmente en formato HTML) entre un navegador (el cliente) y un servidor web (denominado, entre otros, http en equipos UNIX) localizado mediante una cadena de caracteres denominada dirección URL.

H.323: Es un estándar que describe una familia de protocolos usados para realizar el control de llamadas en una comunicación multimedia a través de redes conmutadas por paquetes.

IAX: (Protocolo Interconexión de Asterisk) es uno de los protocolos utilizado por Asterisk para manejar conexiones VoIP entre servidores Asterisk, y entre servidores y clientes que también utilizan protocolo IAX.

ITU: (Unión Internacional de Telecomunicación) es el organismo especializado de las Naciones Unidas encargado de regular las telecomunicaciones, a nivel internacional, entre las distintas administraciones y empresas operadoras.

IVR: (Respuesta de Voz Interactiva) es una poderosa plataforma de desarrollo de aplicaciones telefónicas, que permite diseñar, integrar, implementar y administrar sistemas de respuesta interactiva de voz, utilizando un amigable lenguaje gráfico y en muy corto tiempo.

MGCP: (Media Gateway Control Protocol) es un protocolo interno de VoIP cuya arquitectura se diferencia del resto de los protocolos VoIP por ser del tipo cliente – servidor.

MVC: (Modelo Vista Controlador) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.

NAT: (Traducción de Dirección de Red) es un proceso que se desarrolló en respuesta a la falta de direcciones de IP.

ORM: (Mapeo Objeto-Relacional) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

PABX: (Central secundaria privada automática) es cualquier central telefónica conectada directamente a la red pública de teléfono por medio de líneas troncales para gestionar, además de las llamadas internas, las entrantes y/o salientes con autonomía sobre cualquier otra central telefónica.

PBX: (Central Telefónica Privada) que no era más que una central telefónica conectada directamente a la red pública de teléfono.

PCI: (Interconexión de Componentes Periféricos) consiste en un bus de ordenador estándar para conectar dispositivos periféricos directamente a su placa base. Estos dispositivos pueden ser circuitos integrados ajustados o tarjetas de expansión que se ajustan en conectores.

PHP: (Pre-procesador de Hipertexto) es un lenguaje de programación diseñado originalmente para la creación de páginas web dinámicas.

PMBX: (Central Telefónica Privada Manual) consiste en lo mismo que una PBX solo que conectada manualmente.

PostgreSQL: es un sistema de gestión de base de datos relacional orientada a objetos y libre.

PSTN: (Red Telefónica Pública Conmutada) es una red de comunicación diseñada primordialmente para la transmisión de voz.

RAM: (Memoria de acceso aleatorio) un tipo de memoria de ordenador a la que se puede acceder aleatoriamente.

RDSI: (Red Digital de Servicios Integrados) es una red que procede por evolución de la red telefónica existente, que al ofrecer conexiones digitales de extremo a extremo permite la integración de multitud de servicios en un único acceso, independientemente de la naturaleza de la información a transmitir y del equipo terminal que la genere.

RTP: (Protocolo de Transporte de Tiempo real) es un protocolo de nivel de sesión utilizado para la transmisión de información en tiempo real, como por ejemplo audio y vídeo en un video-conferencia.

Script: (conjunto de instrucciones) Permiten la automatización de tareas, creando pequeñas utilidades. Es muy utilizado para la administración de sistemas UNIX. Son ejecutados por un intérprete de línea de órdenes y usualmente son archivos de texto.

SERTOD: Empresa de Servicios de Telecomunicación a los Órganos de la Defensa.

SIP: (Protocolo de Inicio de Sesión) es un protocolo con la intención de ser el estándar para la iniciación, modificación y finalización de sesiones interactivas de usuario donde intervienen elementos multimedia como el video, voz, mensajería instantánea, juegos en línea y realidad virtual.

SQL: (Lenguaje de Consulta Estructurado) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.

TCP: (Protocolo de Control de Transmisión) usados para el control de la transmisión en Internet. Permite que diferentes tipos de ordenadores o computadoras se comuniquen a través de redes heterogéneas.

UCID: Unidad de Compatibilización Integración y Desarrollo De Software para la Defensa.

UDP: (Protocolo de Datagrama de Usuario) es un protocolo que permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera.

UML: (Lenguaje Unificado de Modelado) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

URL: (Localizador Uniforme de Recursos) es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación, como por ejemplo documentos textuales, imágenes, videos, presentaciones digitales, etcétera.

UTF: (Formato de Codificación Unicode) es un formato de codificación de caracteres Unicode(proporciona un número único para cada carácter) e ISO 10646 utilizando símbolos de longitud variable.