

Universidad de las Ciencias Informáticas

Facultad 2



Título: Desarrollo del sistema de Conferencias de la plataforma Platel.

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

Autor(es): Yaismery Armenteros Rodríguez.

José Antonio Hernández Martínez.

Tutor: Ing. Jorge Javier García Ruiz.

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 _____.

Yaismery Armenteros Rodríguez

José Antonio Hernández Martínez

Firma del Autor

Firma del Autor

Ing. Jorge Javier García Ruiz.

Firma del Tutor

*“Solamente el esfuerzo propio puede producir
los beneficios que muchos esperan por
milagros”*

J.A. Rosenkranz.

DATOS DE CONTACTO

Ing. Jorge Javier García Ruiz.

Ingeniero en Ciencias Informáticas, graduado en el año 2009 en la Universidad de la Ciencias Informáticas (UCI). Desde su graduación hasta la actualidad se ha desempeñado como implementador del proyecto PLATEL en la línea de telecomunicaciones del centro UCID como parte de su período de adiestramiento.

AGRADECIMIENTOS:

A nuestros familiares por estar presentes en cada momento.

A esas personitas que lloraron con nosotros y no hicieron reír, ustedes nuestros amigos durante 5 años.

A nuestro tutor Jorge, a Duany, Fefa, Abel, Ahmet y todo el equipo de trabajo que nos guió y ayudó a realizar este trabajo.

A todos aquellos que de una forma u otra nos ayudaron con la realización de este trabajo de diploma.

A todos los presentes y los que no pudieron estar:

Nuestros más sinceros agradecimientos.

DEDICATORIA:

A esas personas sin las cuales no estuviera aquí hoy, mis ángeles guardianes, mis amores, mis padres.

A mi abuela Paula, a ti que te lo debo todo.

A mi familia por acompañarme siempre a convertir mis sueños en realidad.

A mi tío Rolando, por ser la base de mi fortaleza, mi ídolo e inspiración.

A todos mis vecinos sin dejar uno, por confiar en mí

Yaismery.

Para toda mi familia por el apoyo incondicional.

José Antonio.

Resumen

El presente trabajo describe el desarrollo del sistema de conferencias de la plataforma Platel: software para centros de contactos basado en VoIP (voz sobre el protocolo de internet), desarrollado en la universidad. El desarrollo de este trabajo se hace necesario ya que, el servicio de conferencia que se brinda actualmente no cuenta con la posibilidad de ser administrado, es decir, que una persona encargada de esta administración sea la que este guiando y prestando los servicios.

Para el desarrollo de la aplicación se utilizaron herramientas libres. Como marco de trabajo se utilizó SAUXE, basado en la arquitectura en capas y emplea el Modelo Vista Controlador cuenta con el lenguaje de programación PHP 5.0, los frameworks Doctrine y Zend Framework y como librería ExtJS. Para la publicación del sitio se eligió el servidor web Apache. Mientras que de todos los gestores de base de datos analizados PostgreSQL fue el seleccionado. Para describir todos los procesos se utilizó el lenguaje de modelado UML con la herramienta Visual Paradigm y la fundamentación del Proceso de Desarrollo de Software de UCID.

Con la implementación de esta aplicación se obtiene una herramienta muy práctica para administrar y gestionar las conferencias de manera más eficaz, automatizada y placentera para el usuario.

Palabras claves:

VoIP, PlaTel.

Índice de contenido.

| | |
|---|-----------|
| Introducción | 1 |
| Capítulo I: Fundamentación Teórica..... | 3 |
| 1.1 <i>Introducción.....</i> | 3 |
| 1.1.1 Telefonía..... | 3 |
| 1.1.2 VoIP..... | 5 |
| 1.1.3 Distribuciones Telefónicas basadas en VoIP..... | 5 |
| 1.1.4 Asterisk PBX..... | 5 |
| 1.1.5 Conferencias como aplicación basada en PBX..... | 8 |
| 1.2 <i>Marco de Trabajo SAUXE.....</i> | 8 |
| 1.2.1 Lenguajes de programación..... | 9 |
| 1.2.2 Gestor de Base de Datos..... | 11 |
| 1.2.3 Modelo Cliente Servidor:..... | 12 |
| 1.2.4 Frameworks y Librerías..... | 13 |
| 1.2.5 Servidor Web Apache..... | 15 |
| 1.2.6 Fundamentación de la Metodología..... | 16 |
| 1.3 <i>IDE de Desarrollo.....</i> | 16 |
| 1.4 <i>Conclusiones.....</i> | 17 |
| Capítulo II Programas y metodologías..... | 18 |
| 2.1 <i>Introducción.....</i> | 18 |
| 2.2 <i>Problema y situación problemática.....</i> | 18 |
| 2.3 <i>Objeto de automatización.....</i> | 18 |
| 2.4 <i>Información que se maneja.....</i> | 18 |

Índice de contenido.

| | |
|---|-----------|
| 2.5 Propuesta del sistema..... | 19 |
| 2.6 Modelo de procesos de negocio..... | 20 |
| 2.7 Especificación de los requisitos de software..... | 21 |
| 2.7.1 Requisitos Funcionales..... | 21 |
| 2.7.2 Requisitos No Funcionales..... | 27 |
| 2.8 Conclusiones..... | 29 |
| Capítulo III: Diseño del sistema..... | 30 |
| 3.1 Introducción..... | 30 |
| 3.2 Definiciones de diseño que se apliquen..... | 36 |
| 3.3 Tratamiento de errores..... | 39 |
| 3.4 Seguridad..... | 39 |
| 3.5 Conclusiones..... | 39 |
| Capítulo IV: Implementación y Pruebas..... | 40 |
| 4.1 Introducción..... | 40 |
| 4.2 Implementación..... | 40 |
| 4.2.1 Diagrama de componentes..... | 40 |
| 4.2.2 Matriz de Integración de Componentes Interna..... | 40 |
| 4.3 Descripción diseño caso de prueba..... | 40 |
| 4.4 Pruebas de software..... | 44 |
| 4.5 Conclusiones..... | 45 |
| Conclusiones Generales..... | 46 |
| Recomendaciones..... | 47 |

Índice de contenido.

| | |
|---|-----------|
| Referencias Bibliográficas | 48 |
| Bibliografía | 49 |
| Anexos..... | 52 |
| <i>Anexo No.1 Especificación de la planilla Modelo conceptual.</i> | <i>52</i> |
| <i>Anexo No.2 DCP Modificar conferencia.</i> | <i>55</i> |
| <i>Anexo No.3 DCP Eliminar conferencia.....</i> | <i>59</i> |
| <i>Anexo No.4 DCP Buscar conferencia.....</i> | <i>61</i> |
| Glosario de Términos | 64 |

Introducción

Las telecomunicaciones hoy en día representan un punto crucial para el desarrollo de la sociedad y su economía perfeccionando y creando un gran avance en áreas como la teleinformática. Estas se han hecho imprescindibles para el desarrollo a nivel mundial ya que significan actualización, comunicación, progreso y son el motor impulsor de la Sociedad de la Información.

La telefonía es una de las ramas de las telecomunicaciones que más se ha desarrollado en los últimos tiempos, muestra de ello es el surgimiento de la tecnología **VoIP** que es un grupo de recursos que permiten el viaje de la voz por la red en forma de paquetes de datos. Contribuyendo a la disminución de los costos de las llamadas debido a que emplea una misma red para transportar voz y datos.

En el mundo actual existen diferentes plataformas telefónicas que se encargan de gestionar las llamadas y brindar otros servicios suplementarios con el propósito de satisfacer las necesidades de comunicación. Una de las más populares en los últimos tiempos es Asterisk, una Central Telefónica libre, cuyo código fuente está disponible para realizar modificaciones y mejoras. Al soportar una mezcla de la telefonía tradicional y los servicios de VoIP, permite a los desarrolladores construir nuevos sistemas telefónicos de forma eficiente o migrar de forma gradual los sistemas existentes a las nuevas tecnologías. Existen varios sistemas de administración que facilitan un poco la gestión y control de esta PBX (del inglés: Private Branch Exchange), entre ellos se encuentra FreePBX. Éste es por excelencia el sistema de administración para Asterisk debido a que su última versión, FreePBXv3 posee una arquitectura más robusta que las anteriores, cuenta con una documentación más elaborada y con una comunidad de desarrollo.

En Cuba es interés nacional el desarrollo y modernización de la infraestructura y los servicios de telecomunicaciones, dada su importancia estratégica para el desarrollo económico social y para la defensa nacional. En los últimos tiempos se ha trabajado en el futuro despliegue de la Plataforma Telefónica Asterisk debido a todas las funcionalidades y facilidades que brinda esta.

Las Fuerzas Armadas Revolucionarias (FAR) está actualmente desarrollando la plataforma telefónica PlaTel, un producto totalmente nacional e independiente con capacidad autónoma de soporte técnico. La misma tiene como objetivo utilizar la VoIP para brindar todos los servicios de una central telefónica, lograr la conectividad e interoperabilidad entre la telefonía IP y la telefonía tradicional y reducir las grandes importaciones que se realizan en la adquisición de PBX. PlaTel emplea como elemento fundamental de su

Introducción.

núcleo la PBX Open Source Asterisk y trabaja bajo el marco de trabajo orientado en el centro UCID llamado SAUXE, el cual integra en un solo paquete: lenguaje de programación, framework y librerías a utilizar.

A pesar de las ventajas que nos brinda FreePBX como herramienta para la administración de Asterisk aún no logra satisfacer todas las necesidades de la plataforma telefónica PlaTel ya que no brinda la posibilidad de realizar una adecuada gestión de las conferencias imposibilitando la optimización de las mismas, logrando solamente esto mediante la escritura en ficheros de configuración y cuenta con una licencia MPL (Licencia Pública de Mozilla) que no permite la reutilización completamente del software. Además, no cuenta con niveles de configuración orientado a personal sin profundos conocimientos de comunicaciones e informática. En otras palabras, existe una insuficiente capacidad para realizar configuraciones avanzadas o personalizadas.

Partiendo de la situación anterior se plantea el siguiente **problema científico**: ¿Cómo gestionar las conferencias en la plataforma PlaTel de manera personalizada?

Por lo cual el **objeto de estudio** de este trabajo de diploma son los sistemas de administración de las plataformas telefónicas. Teniendo como **campo de acción** la gestión de conferencias en el sistema de administración de la plataforma PlaTel. Definiendo como **objetivo general**: desarrollar el sistema de conferencias de la plataforma PlaTel.

Los **objetivos específicos** del trabajo de diploma son los siguientes:

- ✓ Definir el concepto de conferencias.
- ✓ Describir el funcionamiento de las conferencias en la plataforma PlaTel.
- ✓ Describir el funcionamiento de las conferencias en la plataforma Asterisk.
- ✓ Realizar análisis y diseño del sistema de conferencias de la plataforma PlaTel.
- ✓ Implementar los módulos necesarios para configurar las conferencias en la plataforma PlaTel.

Teniendo en cuenta los objetivos planteados se tiene como **idea a defender**: La implementación del sistema de conferencia de la plataforma PlaTel permitirá realizar una adecuada gestión, optimización y personalización de la plataforma.

Capítulo I: Fundamentación Teórica

1.1 Introducción.

En el presente capítulo se realizará un breve análisis de los principales programas y herramientas de administración para la configuración de las salas de conferencias, los conceptos asociados con los mismos, así como una investigación sobre las tendencias actuales en el mundo. También resume un análisis descriptivo de la arquitectura Cliente-Servidor, las principales herramientas y tecnologías de modelado utilizadas para dar cumplimiento al problema planteado.

1.1.1 Telefonía.

Desde tiempos remotos la especie humana sintió la necesidad de comunicarse a largas distancias, empleando métodos muy rústicos hasta su perfeccionamiento en la etapa final del siglo XVIII. No es hasta el 1857 que Antonio Meucci ¹ inventa una máquina capaz de unir la señal eléctrica con la voz humana.

Actualmente existen diferentes tipos de telefonía dentro de los cuales se encuentran: telefonía móvil, telefonía tradicional y telefonía IP.

Telefonía Móvil:

La telefonía móvil ó telefonía celular está basada en el concepto de red celular y está formada por: una red de comunicaciones y los terminales (teléfonos móviles) que permiten el acceso a la red.

Por las facilidades y comodidades que presenta este tipo de telefonía, se puede decir que es una de las más acogidas en el mundo, ya que logra integrar en un solo dispositivo: la conexión a Internet, el correo electrónico, FTP (del inglés: File Transfer Protocol), Telnet y mensajería instantánea, a los cuales se puede acceder con el software adecuado y utilizando los mismos protocolos convencionales.

Telefonía Tradicional:

“La telefonía tradicional no es más que el surgimiento y evolución de la telefonía hasta nuestros días. El

¹ Antonio Meucci: Es el inventor del primer modelo de teléfono, inventó una forma de conexión entre su oficina y el dormitorio de su mujer. Perfeccionó el sistema con una caja de jabón y un diafragma metálico. Sin embargo, le faltaban los medios económicos para mantener sus actividades. En 1871, intenta patentar el invento, pero le falta de dinero. Años después, Bell patenta el sistema.

Fundamentación Teórica

teléfono fue inventado en 1876, originalmente su transmisión era sobre un hilo de hierro con circuitos sin marcación y comunicación punto a punto. Seguidamente se le incorpora el concepto de switch, inicialmente con una persona, luego vía conmutación electrónica” [1].

“Características principales de la telefonía tradicional” [1]:

- ✓ Recursos ocupados durante toda la duración de la llamada.
- ✓ Los precios varían en función del tiempo de uso (tiempo de ocupación del circuito dedicado).
- ✓ Diseñado para “sólo voz”.
- ✓ Sector totalmente regulado.
- ✓ Garantía de disponibilidad: 99,5 %.

Telefonía IP:

“La telefonía IP reúne la transmisión de voz y de datos, lo que posibilita la utilización de las redes informáticas para efectuar llamadas telefónicas. Además, esta tecnología al desarrollar una única red encargada de cursar todo tipo de comunicación, ya sea de voz, datos o video, se denomina red convergente o red multiservicios.”[2]

“La telefonía IP surge como una alternativa a la telefonía tradicional, brindando nuevos servicios al cliente con una serie de beneficios económicos y tecnológicos con características especiales como:

- ✓ **Inter-operatividad con las redes telefónicas actuales:** En el caso de TELMEX se disponen de dos tipos de Interconexión a la red de telefonía pública, desde una central telefónica IP y directamente desde una tradicional.
- ✓ **Calidad de Servicio Garantizada a través de una red de alta velocidad:** En Telefonía IP este concepto posee una serie de aspectos, como son:
 - Red de alta disponibilidad que ofrece hasta de un 99,99% de recursos.
 - Calidad de voz garantizada (bajos indicadores de errores, de retardo, de eco, entre otras).
- ✓ **Servicios de Valor Agregado:** No es más que un servicio brindado, como es el actual prepago, además de nuevos servicios como la mensajería unificada. “[2]

1.1.2 VoIP.

“VoIP (Voz sobre Protocolo de Internet o Telefonía IP), también se conoce como telefonía de Internet, telefonía IP, o voz a través de Internet (VOI). Una categoría de hardware y software que permite a la gente utilizar Internet como medio de transmisión de llamadas telefónicas, enviando datos de voz en paquetes usando el IP en lugar de los circuitos de transmisión telefónicos. Una de sus ventajas es que las llamadas telefónicas a través de Internet no incurren en un gasto añadido a lo que el usuario ya paga por el acceso a Internet” [3].

1.1.3 Distribuciones Telefónicas basadas en VoIP.

TrixBot:

Es una de las distribuciones de Asterisk bajo Linux, basada en CentOS, de sencilla y rápida instalación. Permite administrar la PBX sin necesidad de conocer Linux ni Asterisk de manera avanzada e interconectar teléfonos internos de una compañía y conectarlos a la red telefónica convencional (RTB Red telefónica básica). Utiliza el FreePBX como entorno gráfico de configuración de la PBX.

Incluye muchas características que antes sólo estaban disponibles en caros sistemas propietarios como son la creación de extensiones, envío de mensajes de voz a e-mail, llamadas en conferencia, menús de voz interactivos y distribución automática de llamadas.

Elastix:

“Distribución libre de Servidor de Comunicaciones Unificadas donde se integran en un solo paquete VoIP, PBX, Fax, Mensajería Instantánea, Correo electrónico y Colaboración. Implementa gran parte de su funcionalidad sobre 4 programas de software Asterisk, Hylafax, Openfire y Postfix. Además, añade su propio conjunto de utilidades y permite la creación de módulos de terceros para hacer de este el mejor paquete de software disponible para la telefonía de código abierto. La meta de Elastix son la confiabilidad, modularidad y fácil uso” [4].

1.1.4 Asterisk PBX.

Es una central totalmente de software, compatible, con una gran variedad de operadores de telefonía IP. Presenta soporte para líneas RDSI (Red Digital de Servicios Integrados), RTB (Red Telefónica Básica) con

Fundamentación Teórica

una gran movilidad de los puestos de trabajo sin recablear y ampliaciones instantáneas.

Puede ejecutarse en cualquier ordenador con sistema operativo Linux y provee todas las funciones que se pueden esperar de una pequeña central, también se puede conectar a líneas analógicas, incluso a través de hardware especializado y relativamente asequible. Hace uso de las líneas Voz IP de varios operadores, en cuyo caso no necesita ningún equipamiento adicional al propio ordenador.

Asterisk GUI:

“Esta suite de software está diseñada para extender la funcionalidad de Asterisk a través de la web independientemente del cliente de aplicaciones, la cual incluye monitorización de llamadas entrantes y salientes a la suite, además escalable a través de múltiples servidores de Asterisk. Avalado por licencias GPL (del inglés: General Public License), BSD (Distribución de Software Berkeley), Comercial y Software Libre” [5].

Principales funcionalidades:

- ✓ Configuración de Asterisk.
- ✓ Sincronización de mensajes de voz.
- ✓ Información detallada de llamadas.
- ✓ Sistema de facturación.
- ✓ Solución para la Gestión de Centros de Contactos.
- ✓ Integración con MySQL, PostgreSQL y SQLite.

FreePBX:

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

“Algunas de las características de FreePBX son [6]:

Fundamentación Teórica

- ✓ Agregar o cambiar la extensión y las cuentas de correo de voz en cuestión de segundos.
- ✓ Soporte nativo de SIP (Protocolo de Inicio de Sesiones), IAX (del inglés: Inter-Asterisk eXchange protocol), y los clientes de ZAP (otros criterios de valoración son apoyados a través de extensiones personalizadas).
- ✓ Soporta todas las tecnologías de apoyo troncal para Asterisk.
- ✓ Modular, con un repositorio en línea para añadir / actualizar las funciones de la interfaz.
- ✓ Reduce los costes de larga distancia con LCR (Consultores de Negocios en Línea) y el patrón de gran alcance basado en el enrutamiento de salida.
- ✓ Enruta las llamadas entrantes basándose en la hora del día, identificador de llamadas.
- ✓ Crea los menús del Recepcionista Digital Interactiva (IVR).
- ✓ Grupos de llamada de diseño sofisticado.
- ✓ Es capaz de administrar y aplicar los centros de llamadas con colas.
- ✓ Subida personalizada de la música en espera (MOH).
- ✓ Directorio de la empresa de búsqueda, basado en el nombre o apellido.
- ✓ Detecta y recibe faxes entrantes.
- ✓ Derechos de recurso administrativo.
- ✓ Copia de seguridad y restauración del sistema.
- ✓ Guarda las grabaciones de audio de las llamadas.
- ✓ Muestra detalle de llamadas de informes con asterisco-stat.
- ✓ Muestra las grabaciones de las conversaciones, esto lo consigue mediante Asterisk Recording Interface (ARI.)”

FreePBX cuenta con varias versiones de las cuales se comentará específicamente de la versión 2 y la versión 3. FreePBX versión 2 es caracterizada por contar con una arquitectura muy pobre, lo cual impide a los desarrolladores agregarle nuevos cambios. Por su parte FreePBXv3 está diseñado para solucionar

este problema brindando una mejor documentación que permite continuar el desarrollo de la misma. Posee una mejor estructura interna y es la llamada versión de los desarrolladores ya que ayuda en la utilización de la interfaz gráfica de usuario. Como elemento de arquitectura utiliza el patrón Modelo Vista Controlador, cuenta con los framework Kohana, Doctrine y JQuery. Presenta una licencia MPL. la cual imposibilita enlazar los módulos cubiertos por una licencia GPL con un módulo MPL.

1.1.5 Conferencias como aplicación basada en PBX.

Una conferencia telefónica es una llamada en la cual la persona que llama desea tener más de un receptor escuchando su llamada. Las conferencias telefónicas pueden hacerse de manera que permitan que la parte receptora de la llamada participe durante la misma, o bien la llamada puede hacerse de manera que la parte que llama, adiciona a la llamada a otros participantes. En la mayoría de los casos, los participantes pueden añadirse a la conferencia por sí mismos. En algunos casos la parte que llama, solamente escucha y no puede hablar. Los negocios utilizan las conferencias telefónicas diariamente para reunirse con participantes remotos, tanto dentro como fuera de la compañía.

La conferencia telefónica es considerada como un medio primordial para reducir costos de viaje y para incrementar la productividad de los empleados al no tener que abandonar sus oficinas para asistir a reuniones. Las conferencias telefónicas están siendo cada vez más utilizadas, así como las conferencias en la red, donde se comparten presentaciones y documentos vía internet.

1.2 Marco de Trabajo SAUXE.

Para el desarrollo del módulo de conferencia se utilizó el marco de trabajo utilizado en el centro UCID llamado SAUXE, basado en la arquitectura en capas (presentación, negocio, servicio, dominio y acceso a datos) y emplea el MVC para llevar a cabo la comunicación entre la capa de presentación y el negocio. Este marco de trabajo cuenta con el lenguaje de programación PHP 5.0 y define una serie de tecnologías y herramientas, entre las que se encuentran:

- ✓ Zend Framework
- ✓ Doctrine
- ✓ ExtJS
- ✓ Gestor de base de datos: PostgreSQL.

MVC:

Modelo Vista Controlador ó también llamado por sus siglas en inglés MVC (Model View Controller), es un patrón de diseño que está formado por tres niveles:

- ✓ Modelo: Representa la lógica de negocio, es la información que maneja la aplicación.
- ✓ Vista: Transforma el modelo en una página web que le permite al usuario interactuar con la misma.
- ✓ Controlador: Encargado de procesar las interacciones del usuario con la aplicación y realizar los cambios necesarios en el modelo ó la vista.

El principal objetivo de este patrón es separar la lógica de negocio de la interfaz, lo que permite una mayor robustez de los sistemas que son desarrollados con el MVC. Entre sus ventajas más notables se encuentran:

- ✓ Permite implementar los componentes por separado gracias a la clara separación que existe entre los mismos.
- ✓ La conexión entre el Modelo y sus Vistas es totalmente dinámica, ésta se produce en tiempo de ejecución.
- ✓ Posee un API (del inglés Application Programming Interface) bien definido, lo que permite reemplazar el Modelo, la Vista ó el Controlador sin aparente dificultad.

1.2.1 Lenguajes de programación.

PHP:

PHP (Pre-procesador de Hipertexto) es un lenguaje interpretado en el lado del servidor que se utiliza para generar páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor.

Para poder ser ejecutado no es necesario compilarlo y para su funcionalidad tiene que estar instalado en la computadora el servidor web Apache con librerías del propio lenguaje.

Ventajas:

Fundamentación Teórica

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

La versión que utilizamos en la realización de nuestro trabajo fue PHP 5 que brinda las siguientes ventajas:

- ✓ Soporte sólido para Programación Orientada Objeto.
- ✓ Mejora del soporte con extensiones re-escribibles para MySQL.
- ✓ Mejora del soporte y una suite de herramientas interoperables para XML.
- ✓ Limpiador de errores con excepciones.
- ✓ Posee una primera clase de aplicación SOAP.
- ✓ Soporte para SQLite.
- ✓ Mejoras en el rendimiento del sistema.

HTML:

Lenguaje de marca de Hipertexto, es un lenguaje de programación muy sencillo que se utiliza para crear las páginas web. Está formado por etiquetas o marcas que definen entre otras cosas: el formato del texto, los elementos que conforman la página y disposición. Esas etiquetas son leídas e interpretadas por el navegador o visualizador y ejecutadas, visualizando así la información en pantalla. Una de las características es que a demás del texto, permiten que se creen enlaces entre distintas partes del documento o entre distintas fuentes de información a través del hiperenlace o hipervínculo, y a demás la inserción de elementos de sonido e imagen. El lenguaje HTML puede ser creado y editado por cualquier editor de texto básico como puede ser Gedit, Notepad o cualquier otro editor que admita texto sin formato. La principal ventaja que tiene HTML es la gran variedad de navegadores que lo soportan, siendo así el formato más usado para la transmisión de documentos electrónicos a través de Internet.

Java Script:

“Java Script es un lenguaje de programación, interpretado, multiplataforma y parcialmente orientado a objetos desarrollado por Netscape para incrementar las funcionalidades del lenguaje HTML. Permite crear efectos especiales para interactuar con el visitante y funciones básicas que son soportadas por la mayoría de los navegadores que se utilizan diariamente” [7].

Ventajas de Java Script:

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

Uno de los inconvenientes que presenta el lenguaje es que requiere estar activado en los navegadores para que su uso sea posible por estos. Por otro lado no le proporciona al programador control total de la página web. En ocasiones los desarrolladores deben realizar diferentes implementaciones de código Java Script para sus aplicaciones, debido a que el código no es interpretado de igual modo por todos los navegadores.

1.2.2 Gestor de Base de Datos.

PostgreSQL:

Servidor de base de datos objeto relacional libre. Fue pionero en la introducción de muchos de los conceptos existentes en los sistemas objeto-relacional utilizados en la actualidad, incluido mas tarde en otros sistemas de gestión comerciales. Es un sistema objeto-relacional, ya que incluye características propias de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión

de bases de datos puramente orientado a objetos

Ventajas:

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

1.2.3 Modelo Cliente Servidor:

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

“Características de la arquitectura Cliente/Servidor” [8]:

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

1.2.4 Frameworks y Librerías

ExtJS:

Es una librería de Java Script ligera y de alto rendimiento, compatible con la mayoría de los navegadores que nos permite crear páginas e interfaces web dinámicas. Proporciona un aumento en el rendimiento, la coherencia, flexibilidad y mejoras de la interfaz de usuario. Utiliza tecnologías como AJAX, DHTML (del inglés Dynamic HTML) y DOM (del inglés Document Object Model).

“Originalmente construida como una extensión de la biblioteca YUI, en la actualidad puede usarse como extensión para las bibliotecas JQuery y Prototipo. Incluye soporte para peticiones directas, CRUD (Crear, Obtener, Actualizar, Borrar) y REST (Transferencia de Estado Representacional), nuevos ejemplos y componentes (para gráficas), más de 1,000 mejoras y correcciones, API documentada y CSS (del inglés Cascading Style Sheets) reautorizado y compatibilidad con versiones anteriores” [9].

AJAX:

AJAX, acrónimo de Java Script asíncrono y XML (Lenguaje de Marcas eXtensible), no es una tecnología

Fundamentación Teórica

en sí misma, es una técnica de desarrollo web para crear aplicaciones interactivas. Es la unión de varias tecnologías independientes.

Estas aplicaciones se ejecutan en el navegador de los usuarios, mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios en las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad, usabilidad en las aplicaciones y mejor respuesta a las acciones del usuario.

Es una técnica válida para múltiples plataformas y utilizables en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como HTML o XHTML, Java Script, DOM, XML, CSS y XSLT (del inglés Extensible Stylesheet Language Transformations).

Doctrine:

Es un mapeador de objetos relacionales (ORM) para PHP que se centra en una potente capa de abstracción de bases de datos. Su principal característica es la de escribir las consultas de bases de datos en un lenguaje OO (Orientado a Objeto) denominado DQL (del inglés: Data Query Language) inspirado en el HQL (del inglés: Hibernate Query Language) de Hibernate. Este brinda a los desarrolladores una poderosa alternativa al SQL manteniendo una máxima flexibilidad sin necesidad de duplicar código.

Posee la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir, convertir clases a tablas de una base de datos.[10]

Zend Framework:

Framework para desarrollo de aplicaciones Web y servicios Web con PHP 5, brinda soluciones para construir sitios web modernos, robustos y seguros.

Principales características:

- ✓ Trabaja con MVC (Modelo Vista Controlador).
- ✓ Cuenta con módulos para manejar archivos PDF, canales RSS (del inglés: Really Simple Syndication) y Servicios Web.
- ✓ El Marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.

- ✓ Es una solución para el acceso a las bases de datos que balancean el ORM con eficiencia y simplicidad.
- ✓ Completa documentación y tests de alta calidad.
- ✓ Es un buscador compatible con Lucene.
- ✓ Posee robustas clases para autenticación y filtrado de entrada.

1.2.5 Servidor Web Apache.

Apache es el servidor Web más difundido y utilizado en Internet, con código abierto, multiplataforma y modular. Se desarrolla dentro del proyecto HTTP Server de la Fundación de Software Apache. Presenta alta estabilidad, seguridad y facilidad de expansión.

El servidor Apache es un software que está estructurado en módulos, es decir, está dividido en muchas porciones de código que hacen referencia a diferentes aspectos o funcionalidades del servidor web. Esta modularidad es intencionada ya que la configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Sus módulos se pueden clasificar en tres categorías:

- ✓ **Módulos Base:** Módulo con las funciones básicas del Apache.
- ✓ **Módulos Multiproceso:** Son los responsables de la unión con los puertos de la maquina y de manejar las peticiones.
- ✓ **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor.

Características:

- ✓ Soporta CGI, Perl, PHP.
- ✓ Soporte para Bases de datos.
- ✓ Soporte SSL para transacciones seguras.
- ✓ Soporta HTTP 1.1.
- ✓ Incluye soporte para host virtuales.
- ✓ Mensajes de error altamente configurables.

- ✓ Bases de datos de autenticación y negociado de contenido.

1.2.6 Fundamentación de la Metodología.

Proceso de Desarrollo de Software de UCID:

El objetivo fundamental de este proceso de desarrollo de software es producir un producto de software que satisfaga los requisitos de un cliente con una planificación y una estimación de recursos predecibles. El modelo de desarrollo de software que propone describe la secuencia de actividades a llevar a cabo para obtener una solución más efectiva, esto se logra al combinar los modelos basados en componentes. El ciclo de vida de un proyecto de software desarrollado a partir de este proceso consta de cinco fases: Inicio, Modelación, Construcción, Explotación Experimental y Despliegue. Al finalizar cada una de estas fases los representantes de cada uno de los roles presentes en el proyecto evalúan si se cumplieron los objetivos definidos al iniciar el proceso.

Lenguaje de Modelado UML:

UML ó Lenguaje Unificado de Modelado es un lenguaje de modelado para especificar o describir métodos o procesos. Se emplea para definir un sistema, detallar los artefactos, documentar y construir. Incluye aspectos conceptuales como los procesos de negocio y funciones del sistema, y aspectos concretos como son expresiones de lenguajes de programación, componentes reutilizables y esquemas de bases de datos.

Herramientas de Modelado Visual Paradigm:

“Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación, así como generar código java” [11].

1.3 IDE de Desarrollo.

Un Entorno de Desarrollo Integrado ó IDE de Desarrollo es un entorno de programación empaquetado como un programa de aplicación, en otras palabras, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Estos pueden ser aplicaciones por sí solos o pueden formar

parte de aplicaciones existentes. Un IDE de Desarrollo favorece el desarrollo rápido y factible debido a que provee un marco de trabajo completamente amigable.

Zend Studio:

Editor web orientado a la programación de páginas PHP, con ayudas en la gestión de proyectos y depuración de código.

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene la interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración. [12]

1.4 Conclusiones.

Teniendo en cuenta el estudio realizado previamente sobre las principales tendencias que existen actualmente acerca de las tecnologías y herramientas más usadas en el campo de la informática y la telefonía de VoIP, se tomó como decisión crear una plataforma propia empleando como marco de trabajo SAUXE con el cual actualmente se está trabajando en el centro UCID, sobre el lenguaje PHP 5.0, con la integración del framework Doctrine y la librería ExtJS 3.0. Para el logro de tal resultado se utilizó como herramienta de modelado el Visual Paradigm por las facilidades que estas brindan.

Capítulo II Programas y metodologías.

2.1 Introducción.

En este capítulo se presenta la propuesta de desarrollo del sistema de Conferencias de la plataforma Platel. Se describen los requisitos funcionales y no funcionales del sistema. Se identifican actores y casos de uso (CU), obteniendo como resultado el Modelo de Casos de Uso del Sistema donde se describen detalladamente todos los CU.

2.2 Problema y situación problemática.

Se propone realizar un sistema que brinde a los administradores una vez autenticados crear salas de conferencias, estas a su vez serán insertadas en una Base de Datos para posteriores acciones sobre las mismas (modificación, eliminación y búsqueda). Primeramente para esta creación se necesita editar el fichero **extension.conf** el cual controla todas las llamadas entrantes y salientes que pueden ser enrutadas y manipuladas por el sistema para adicionarle el comando MeetMe (). **MeetMe** es el comando por el cual se brindan las opciones que tendrá la conferencia. Además, contamos con el **MeetMeAdmin** utilizado para administrar la conferencia. Este servicio de administrar conferencia, no es más que la prestación de los servicios habla, escucha y culminación de llamadas de todos los participantes en la sala de conferencia que se esté administrando. Otro de los comandos auxiliares del Meetme es el **MeetMeCount** utilizado para conocer la cantidad de participantes que posee una sala de conferencia determinada.

2.3 Objeto de automatización.

El proceso de gestionar (crear, eliminar, modificar y buscar) conferencias, así como la administración de las mismas.

2.4 Información que se maneja.

Para llevar a cabo la gestión y administración de las conferencias es necesario configurar antes una serie de parámetros:

MeetMe: Aplicación para entrar a una conferencia.

MeetMeAdmin: Aplicación para administrar a una conferencia.

MeetMeCount: Aplicación para conocer el número de usuarios presentes en una conferencia.

Programas y metodologías.

Para configurar el módulo de conferencias de Asterisk deberá editar el fichero meetme.conf.

Ejemplo

```
[rooms]
```

```
; el uso es conf => Numero_de_conferencia,pincode
```

```
conf => 9000
```

```
conf => 101,123456
```

Salón de Conferencia 101 como se definió anteriormente en el fichero meetme.conf con la clave (PIN) 123456.

En el plan de discado podrá utilizar lo siguiente:

```
exten => 500,1,MeetMe(101||123456)
```

Salón de conferencia autenticado

```
exten => 18,1,Answer
```

```
exten => 18,2,Wait(1)
```

```
exten => 18,3,Authenticate(5678)
```

```
exten => 18,4,MeetMe(18|p)
```

```
exten => 18,5,Playback(vm-goodbye)
```

```
exten => 18,6,Hangup
```

2.5 Propuesta del sistema.

El sistema tendrá como objetivo principal brindar la opción de gestionar conferencias: crearlas, eliminarlas, buscarlas y modificarlas. Estas últimas se podrán realizar una vez creadas e insertadas en la Bases de Datos (BD). Además, la aplicación es capaz de mostrar las conferencias que estarán registradas hasta el momento.

Se podrá realizar la administración de una sala de conferencias anteriormente seleccionada por uno de los administradores del sistema. Este proceso se basa en brindarle y administrar los servicios de Habla,

Programas y metodologías.

Escucha y Culminación de llamadas a sus participantes. La interfaz principal muestra todos los participantes, es decir, las extensiones que conforman dicha conferencia. Además de brindar la posibilidad de salir de la aplicación cuando el mismo lo desee.

Realizando una serie de pruebas y comparaciones con otros sistemas de administración de la PBX Asterisk, para el caso de la configuración y administración de conferencias como es el caso de FreePBX en la cual ,esta configuración es muy engorrosa ya que consta de una serie de pasos avanzados, se llegó a la conclusión que este sistema no es capaz de llevar a cabo una administración completa de las conferencias, y se necesita un amplio nivel y conocimiento de la informática para poder realizar este proceso. Por lo tanto, se toma como decisión que el sistema implementado por la plataforma PlaTel cumple con las necesidades anteriormente planteadas.

2.6 Modelo de procesos de negocio.

Después de un extenso análisis, se decidió no realizar la modelación del negocio a través de los diagramas de caso de uso del negocio. Por lo tanto, se decidió realizar un Modelo de Dominio que es un subconjunto del modelo de objetos del negocio donde se representan los conceptos y eventos fundamentales que se expresan como clases con la cardinalidad existente entre ellas (Modelo conceptual). **Ver figura 1.** Los conceptos tratados en este modelo serán explicados en el **Anexo1.**

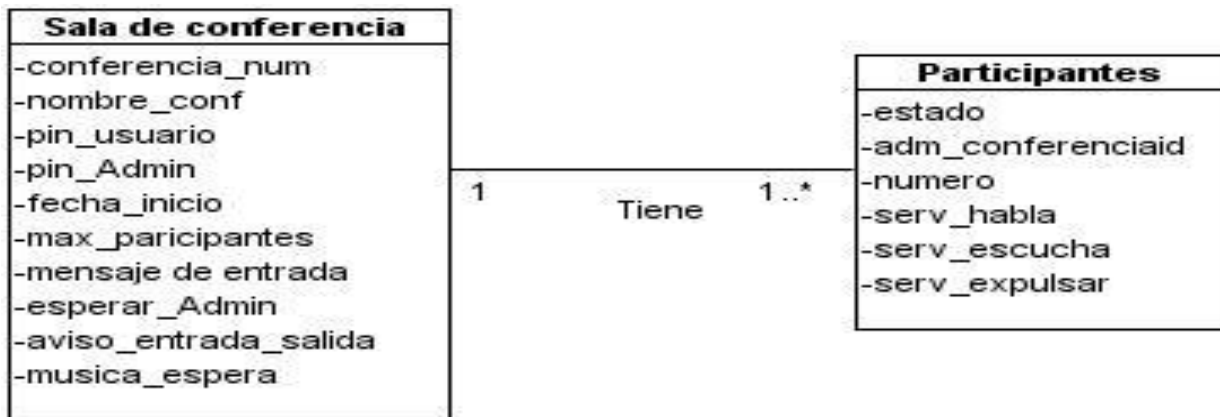


Figura 1 Modelo Conceptual.

2.7 Especificación de los requisitos de software.

2.7.1 Requisitos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Ellos no alteran la funcionalidad del producto, esto quiere decir que los requerimientos funcionales se mantienen invariables sin importar con qué propiedades o cualidades se relacionen.

✓ RF1 Gestionar conferencia.

RF1.1 Crear conferencia.

RF1.2 Modificar conferencia.

RF1.3 Eliminar conferencia.

RF1.4 Buscar conferencia.

✓ RF2 Administrar conferencia.

2.7.1.1 Descripción del requisitos funcional Adicionar Conferencia.

| Conceptos tratados | Conceptos | Atributos |
|--------------------|---|---|
| | Conferencia. | número, nombre, ping_usuario, ping_admin, fecha_inicio, fecha_fin, max_participantes, menentrada, avisoentradasal, esperaradmin, musespera. |
| Precondiciones | Precondiciones | Pre-requisito |
| | El usuario es administrador y se ha autenticado. | No procede. |
| Descripción | 1. Se selecciona en la interfaz el módulo correspondiente a Conferencia. 2. Se muestra en la interfaz las opciones para Gestionar una conferencia y Administrar conferencia. | |

Programas y metodologías.

| | |
|-------------------------|---|
| | <p>3. Se selecciona la opción Gestionar una conferencia (Crear, Modificar, Eliminar, Mostrar)</p> <p>4. Se muestra la interfaz con las conferencias que están registradas en ese momento.</p> <p>5. Se selecciona la opción Crear.</p> <p>6. Se muestra la interfaz con los parámetros necesarios para crear una conferencia.</p> <p>7. Se introducen los datos.</p> <p>8. Se selecciona el botón Aceptar.</p> <p>9. En caso de que se desee salir del formulario, se selecciona el botón Cancelar.</p> |
| Validaciones | El sistema valida los datos según lo descrito en el Modelo Conceptual: Modelo conceptual.doc |
| Post-condiciones | Se adicionó una conferencia. |
| Post-requisito | No procede |

Tabla 1 Descripción RF: Adicionar Conferencia.

2.7.1.2 Descripción del requisitos funcional Modificar Conferencia.

| Conceptos tratados | Conceptos | Atributos |
|---------------------------|-----------------------|---|
| | Conferencia. | número, nombre, ping_usuario, ping_admin, fecha_inicio, fecha_fin, max_participantes, menentrada, avisoentradasal, esperaradmin, musespera. |
| Precondiciones | Precondiciones | Pre-requisito |

Programas y metodologías.

| | | |
|-------------------------|---|------------------------|
| | El usuario es administrador y se ha autenticado. | Adicionar conferencia. |
| Descripción | <ol style="list-style-type: none"> 1. Se selecciona en la interfaz el módulo correspondiente a Conferencia. 2. Se muestra en la interfaz las opciones para Gestionar una conferencia y Administrar conferencia. 3. Se selecciona la opción Gestionar una conferencia (Crear, Modificar, Eliminar, Mostrar) 4. Se muestra la interfaz con las conferencias que están registradas en ese momento. 5. Se selecciona (doble clic) la conferencia que se desea modificar. 6. Se selecciona el botón Modificar. 7. Se muestra una interfaz con los datos de los parámetros con que fue creada la conferencia. 8. Se realizan los cambios en los parámetros deseados. 9. Se selecciona el botón Aceptar. 10. En caso de que se quiera salir del formulario, se selecciona el botón Cancelar. | |
| Validaciones | El sistema valida los datos según lo descrito en el Modelo Conceptual: Modelo conceptual.doc | |
| Post-condiciones | Se modificó una sala de conferencia. | |
| Post-requisito | No procede | |

Tabla 2 Descripción RF: Modificar Conferencia.

Programas y metodologías.

2.7.1.3 Descripción del requisitos funcional Eliminar Conferencia.

| Conceptos tratados | Conceptos | Atributos |
|---------------------------|---|---|
| | Conferencia | número, nombre, ping_usuario, ping_admin, fecha_inicio, fecha_fin, max_participantes, menentrada, avisoentradasal, esperaradmin, musespera. |
| Precondiciones | Precondiciones | Pre-requisito |
| | El usuario es administrador y se ha autenticado. | Adicionar conferencia. |
| Descripción | <ol style="list-style-type: none">1. Se selecciona en la interfaz el módulo correspondiente a Conferencia.2. Se muestra en la interfaz las opciones para Gestionar una conferencia y Administrar conferencia.3. Se selecciona la opción Gestionar una conferencia (Crear, Modificar, Eliminar, Mostrar)4. Se muestra la interfaz con las conferencias que están registradas en ese momento.5. Se selecciona la conferencia que se desea eliminar.6. Se selecciona el botón eliminar.7. En caso de que se desee salir del formulario, se selecciona el botón Cancelar. | |
| Validaciones | El sistema valida los datos según lo descrito en el Modelo Conceptual: Modelo conceptual.doc | |

Programas y metodologías.

| | |
|-------------------------|-------------------------------------|
| Post-condiciones | Se eliminó una sala de conferencia. |
| Post-requisito | No procede |

Tabla 3 Descripción RF: Eliminar Conferencia.

2.7.1.4 Descripción del requisitos funcional Buscar Conferencia.

| Conceptos tratados | Conceptos | Atributos |
|---------------------------|--|---|
| | Conferencia. | número, nombre, ping_usuario, ping_admin, fecha_inicio, fecha_fin, max_participantes, menentrada, avisoentradasal, esperaradmin, musespera. |
| Precondiciones | Precondiciones | Pre-requisito |
| | El usuario se ha autenticado. | Adicionar conferencia. |
| Descripción | <ol style="list-style-type: none">1. Se selecciona en la interfaz el módulo correspondiente a Conferencia.2. Se muestra en la interfaz las opciones para Gestionar una conferencia y Administrar conferencia.3. Se selecciona la opción Gestionar una conferencia (Crear, Modificar, Eliminar, Mostrar)4. Se muestra la interfaz con las conferencias que están registradas en ese momento.5. Se inserta en el buscador el número de la conferencia que se desea buscar.6. Se selecciona el botón Buscar.7. En caso de que se desee salir del formulario, se selecciona el botón | |

Programas y metodologías.

| | |
|-------------------------|--|
| | Cancelar. |
| Validaciones | El sistema valida los datos según lo descrito en el Modelo Conceptual: Modelo conceptual.doc |
| Post-condiciones | Se buscó una sala de conferencia. |
| Post-requisito | No procede |

Tabla 4 Descripción RF: Buscar conferencia.

2.7.1.5 Descripción del requisitos funcional Administrar Conferencia.

| Conceptos tratados | Conceptos | Atributos |
|---------------------------|---|---|
| | Participante | num_conferencia, estado, serv_habla, serv_escucha y exten. |
| | Conferencia | número, nombre, ping_usuario, ping_admin, fecha_inicio, fecha_fin, max_participantes, menentrada, avisoentradasal, esperaradmin, musespera. |
| Precondiciones | Precondiciones | Pre-requisito |
| | El usuario es administrador y se ha autenticado. | Adicionar conferencia. |
| Descripción | <ol style="list-style-type: none"> 1. Se selecciona en la interfaz el módulo correspondiente a Conferencia. 2. Se muestra en la interfaz las opciones para Gestionar una conferencia y Administrar conferencia. 3. Se selecciona la opción Administrar conferencia. 4. Se introduce el número de la conferencia que se desea administrar con la | |

Programas y metodologías.

| | |
|-------------------------|--|
| | <p>contraseña.</p> <ol style="list-style-type: none">5. Se muestra una interface con todos los participantes que tiene dicha sala de conferencia y los servicios (Habla, Escucha, Cuelga, Estado y Número de la extensión).6. Se selecciona el usuario al cuál se le administrará un servicio.7. Se selecciona el servicio a brindar.8. Seleccionar el botón Aceptar.9. En caso de que se desee salir del formulario, se selecciona el botón Cancelar. |
| Validaciones | El sistema valida los datos según lo descrito en el Modelo Conceptual: Modelo conceptual.doc |
| Post-condiciones | Se administró una sala de conferencia. |
| Post-requisito | No procede |

Tabla 5 Descripción RF: Administrar Conferencia.

2.7.2 Requisitos No Funcionales.

2.7.2.1 Requerimientos de apariencia o interfaz externa.

La aplicación propuesta podrá ser utilizada por usuarios con conocimientos básicos en temas de informática y telecomunicaciones, pues estará bien detallada y organizada la información que se mostrará en la interfaz. Será amigable y fácil de usar, de forma tal que no se convierta en un problema para los usuarios el trabajo con la misma.

2.7.2.2 Requerimientos de usabilidad.

A los usuarios del sistema se les dará un adiestramiento básico en el uso de la aplicación en cada uno de sus niveles, mientras que los administradores deberán obtener un conocimiento avanzado para ser

capaces de dar respuesta a cualquier incidente que ocurra con el sistema.

2.7.2.3 Requerimientos de rendimiento.

Realizado el estudio del hardware y la red a emplear, es preciso configurar la aplicación para lograr un rendimiento óptimo de la misma. La eficiencia del producto estará condicionada en gran medida por el aprovechamiento de los recursos con que se dispongan. Teniendo una fuerte influencia la rapidez de las consultas a la base de datos y la velocidad del servidor Asterisk.

2.7.2.4 Requerimiento político-cultural.

El sistema resultante no debe ir contra de las normativas político-culturales establecidas para realizar el proceso de mantenimiento en nuestro país.

El sistema debe tener la flexibilidad necesaria para adaptarse a diferentes requerimientos políticos-culturales, a pesar de haber sido concebida para apoyar el Proceso Docente y la Informatización de la Escuela

2.7.2.5 Requerimientos de portabilidad.

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

2.7.2.6 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.

2.7.2.7 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).

2.8 Conclusiones.

En este capítulo fue descrita la propuesta de solución y las funcionalidades que el sistema debe cumplir a través de un modelo de dominio. Se analizaron y describieron los requisitos funcionales y no funcionales, que fueron de base para el desarrollo del sistema. Además, se da a conocer las interfaces con las cuales contara el sistema para la administración de las conferencias.

Capítulo III: Diseño del sistema.

3.1 Introducción

En este capítulo se diseña la propuesta de solución, para ello se identifican las clases del diseño. Se modelan los diagramas de clases del diseño, el modelo de clases persistentes y el modelo entidad relación. Por último, se modela el diagrama de componentes.

3.1.1.1 Descripción de las clases.

Clase <<ConferenciaController>>

| | |
|-------------------------------------|--|
| Tipo de clase: Controladora. | |
| Para cada responsabilidad: | |
| Nombre: | modificarConfAction() |
| Descripción: | Modifica una conferencia, es decir cambia los datos por los nuevos insertados. |
| Nombre: | eliminarConfAction() |
| Descripción: | Elimina una conferencia con todos sus parámetros. |
| Nombre: | cargarGridDatosAction () |
| Descripción: | Carga el grid, que no es más que el elemento visual en el cual se mostraran los datos. |
| Nombre: | ConferenciaAction() |
| Descripción: | Constructos de la clase. |
| Nombre: | adicionarConfAction() |
| Descripción: | Adiciona las conferencias con todos sus parámetros. |
| Nombre: | existeConferencia() |
| Descripción: | Devuelve verdadero, si encuentra una conferencia con el mismo número que el pasado como parámetro. |

Diseño del sistema.

| | |
|--------------|--|
| Nombre: | cargarModificarAction () |
| Descripción: | Carga y muestra los datos una vez modificados |
| Nombre: | comunAdicionar() |
| Descripción: | Adiciona los datos que sean comunes para las conferencias. |

Tabla 6 Descripción de la clase ConferenciaController.

Clase <<ConferenciaModel>>

| Para cada responsabilidad: | |
|----------------------------|---|
| Nombre: | Insertar() |
| Descripción: | Inserta una conferencia con todos los parámetros que esta necesita. |
| Nombre: | Actualizar(\$idconf, \$datos_conf) |
| Descripción: | Actualiza los datos de la conferencia según el numero de la conferencia pasado por parámetro. |
| Nombre: | Eliminar(\$instance) |
| Descripción: | Elimina la conferencia que se ha seleccionado. |
| Nombre: | EliminarConferencia(\$id) |
| Descripción: | Elimina la conferencia según su número. |
| Nombre: | getConferenciaById(\$numero) |
| Descripción: | Devuelve una conferencia según el número de identificador que esta posee. |
| Nombre: | eliminarConf(\$obj) |
| Descripción: | Método para eliminar las conferencia, según el objeto que se le pase, es decir el recibe la conferencia con todos sus parámetros y la elimina |
| Nombre: | dameTodasC() |
| Descripción: | Devuelve todas las conferencias que están habilitadas. |

Diseño del sistema.

| | |
|--------------|---|
| Nombre: | verif(\$num,\$pass) |
| Descripción: | Verifica que el usuario que está conectado sea administrador de la conferencia. |
| Nombre: | existeCof(\$numero) |
| Descripción: | Verifica si la conferencia que se pasa por parámetros existe. |
| Nombre: | getConferenciaByNumber(\$numero) |
| Descripción: | Devuelve la conferencia que corresponda con el numero pasado por parámetro |

Tabla 7 Descripción de la clase ConferenciaModel.

Clase <<>Conferencia>>

| | |
|-----------------------------------|---|
| Tipo de clase : Entidad | |
| Para cada responsabilidad: | |
| Nombre: | Buscar(\$numero) |
| Descripción: | Busca una conferencia según su número. |
| Nombre: | GetTodos() |
| Descripción: | Devuelve todas las conferencias que estén creadas. |
| Nombre: | GetPorLimite() |
| Descripción: | Devuelve todas las conferencias en un límite de cantidad. |
| Nombre: | GetConferencia() |
| Descripción: | Devuelve una conferencia por su número. |
| Nombre: | GetConferenciaPorId() |
| Descripción: | Devuelve una conferencia por su identificador. |
| Nombre: | dameTodas() |
| Descripción: | Devuelve todas las conferencias que estén creadas para ser utilizadas en la parte |

| | |
|--------------|--|
| | visual. |
| Nombre: | EliminarCf() |
| Descripción: | Elimina una conferencia según su número. |
| Nombre: | verificar(\$nu,\$pa) |
| Descripción: | Verifica que el número de la conferencia corresponda con la contraseña. |
| Nombre: | existe(\$numero) |
| Descripción: | Verifica que existe una conferencia con el mismo número que el pasado por parámetro. |
| Nombre: | getConferenciaByNumber(\$numero) |
| Descripción: | Retorna la conferencia que posea el mismo número que el pasado por parámetro. |

Tabla 8 Descripción de la clase Conferencia.

Clase << conferencia.js >>

Descripción: Es la encargada de visualizar todo lo relacionado con la gestión de conferencias.

Clase <<conferencia.phtml>>

Descripción: Incluye todos los java scripts necesarios para el correcto funcionamiento de la interfaz encargada de gestionar las conferencias.

3.1.1.2 Descripción de las clases.

Clase <<AdmConf>>

| | |
|-----------------------------------|------------------------------|
| Tipo de clase Entidad | |
| Para cada responsabilidad: | |
| Nombre: | setUp() |
| Descripción: | Actualizar la clase entidad. |

Tabla 9 Descripción de la clase AdmConf

Clase << AdmConfController >>

| | |
|------------------------------------|---|
| Tipo de clase: Controladora | |
| Para cada responsabilidad: | |
| Nombre: | confgriddatosadminAction() |
| Descripción: | Configura el grid, que es el elemento visual donde se mostraran los datos del administrador de esa sala de conferencia. |
| Nombre: | modificaradminAction() |
| Descripción: | Modifica los elementos de administración de una conferencia, es decir, los servicios. |
| Nombre: | autenticarAction() |
| Descripción: | Autenticación para saber que los que van a estar administrando la conferencia es personal autorizado. |

Tabla 10 Descripción de la clase AdmConfController.

Clase << AdmConfModel >>

| | |
|-----------------------------------|----------------------|
| Para cada responsabilidad: | |
| Nombre: | update() |
| Descripción: | Actualizar la clase. |

Tabla 11 Descripción de la clase AdmConfModel

Clase << admconf.js >>

Descripción: Es la encargada de visualizar todo lo relacionado con la administración de las conferencias.

Clase << admconf.phtml >>

Descripción: Incluye todos los java scripts necesarios para el correcto funcionamiento de la interfaz

encargada de administrar las conferencias.

Clase << conferencia.js >>

Descripción: Es la encargada de visualizar todo lo relacionado con la gestión de conferencias.

Clase <<conferencia.phtml>>

Descripción: Incluye todos los java scripts necesarios para el correcto funcionamiento de la interfaz encargada de gestionar las conferencias.

3.1.1.3 Descripción de las tablas.

Tabla <<Conferencia>>

Descripción: La tabla almacena todos los datos relacionados con la sala de conferencia para su creación, así como las opciones de entrada y salida que el usuario desee.

| Atributo | Tipo | Descripción |
|-------------------|-----------------------|--|
| numero | Entero. | Número de la conferencia. |
| nombre | Cadena de caracteres. | Nombre de la conferencia. |
| pin_usuario | Cadena de caracteres. | Clave de acceso para entrar en la conferencia del usuario. |
| pin_admin | Cadena de caracteres. | Clave de acceso para entrar en la conferencia del administrador. |
| fecha_inicio | Fecha. | Fecha en que se inicia la conferencia. |
| fecha_fin | Fecha. | Fecha en que se termina la conferencia. |
| duracion | Fecha. | Tiempo que duro la sala de conferencia. |
| max_participantes | Entero. | Cantidad de participantes que posee la sala de conferencia. |

Diseño del sistema.

| | | |
|-----------------|-------|--|
| menentrada | Bool. | Mensaje que se escucha al entrar en la sala de conferencia. |
| esperaradmin | Bool | Si desea esperar por el administrador para entrar en la sala de conferencia. |
| musespera | Bool | Música mientras espera a que se conecten los demás participantes. |
| avisoentradasal | Bool | Aviso a la entrada y a la salida de la conferencia. |

Tabla 12 Descripción de la tabla Conferencia del DER.

Tabla <<Participante>>

| | | |
|---|-----------------|-----------------|
| Descripción: La tabla almacena los datos como tal del participante en la conferencia, los servicios que se le están brindando y su estado con respecto a la misma. | | |
| Atributo | Atributo | Atributo |
| num_conferencia | num_conferencia | num_conferencia |
| estado | estado | estado |
| serv_habla | serv_habla | serv_habla |
| serv_escucha | serv_escucha | serv_escucha |
| exten | exten | exten |

Tabla 13 Descripción de la tabla Participante del DER.

3.2 Definiciones de diseño que se apliquen.

Los patrones de diseño que se utilización fueron los siguientes;

Patrones GRASP (General Responsibility Assignment Software Patterns):

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: Encargado de asignarle responsabilidades a objetos de manera dinámica y configurarlos con

nuevos atributos.

Creador: Encargado de guiar la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento.

Experto: Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

Bajo Acoplamiento: Asignar una responsabilidad para mantener bajo acoplamiento.

Alta Cohesión: Asignar una responsabilidad de modo que la cohesión siga siendo alta.

Patrones GoF:

Table Data Gateway: Posee todas las SQL para tener acceso a una sola tabla o vista: selecciona, inserciones, actualizaciones y eliminaciones.

Row Data Gateway: Permite el acceso a varias bases de datos, y acceder a la fuente de información que se esconde detrás de la interfaz.

Facade o Fachada: Es un patrón de diseño que simplifica la interface entre dos sistemas o componentes de *software* ocultando un sistema complejo detrás de una clase que hace las veces de pantalla o fachada. Una de sus ventajas es aislar los posibles cambios que se puedan producir en alguna de las partes. Implementa métodos convenientes para tareas comunes, puede reducir la dependencia de código externo en los trabajos internos de una biblioteca, y puede envolver una colección mal diseñada de APIs con un solo API bien diseñado.

Singleton o Instancia única: Está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

Factory: Consiste en utilizar una clase constructora abstracta con unos cuantos métodos definidos y otros abstractos: el dedicado a la construcción de objetos de un subtipo de un tipo determinado. Es una simplificación del Abstract Factory, en la que la clase abstracta tiene métodos concretos que usan algunos de los abstractos.

Modelo Vista Controlador: Este patrón está compuesto por tres elementos principales, el modelo, la vista y

Diseño del sistema.

el controlador. Un modelo puede tener diversas vistas, cada una con su correspondiente controlador. Un ejemplo clásico es el de la información de una base de datos, que se puede presentar de diversas formas: diagrama de tarta, de barras, tabular, entre otras. Veamos cada componente:

Modelo: En él se encuentran los datos y las reglas de negocio, es el responsable de:

- ✓ Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- ✓ Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".
- ✓ Lleva un registro de las vistas y controladores del sistema.
- ✓ Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo.

Vista: muestra la información del modelo al usuario y es el encargado de:

- ✓ Recibir datos del modelo y mostrarlos al usuario.
- ✓ Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- ✓ Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

Controlador: gestiona las entradas del usuario y:

- ✓ Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- ✓ Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar ()". Una petición al modelo puede ser " getAdmConfById(\$numero)".

3.3 Tratamiento de errores.

El tratamiento de errores lo tenemos presente en toda la elaboración de la aplicación. Pueden existir errores tanto en la parte del cliente como en la del servidor. En la parte cliente están validados por las expresiones regulares realizadas en el java script.

3.4 Seguridad.

La seguridad en el módulo de PlaTel está regida por ACAXIA diseñado exactamente para gestionarle la seguridad a cualquier aplicación web. Mediante este se pueden gestionar los roles, servicios consumidos y ofertados a otros subsistemas, los usuarios y los permisos que tienen cada uno de estos dentro de las aplicaciones, para garantizar la seguridad de los mismos.

En el módulo de conferencia a parte de utilizar el sistema anteriormente planteado se implementó un pequeño mecanismo a la hora de administrar una conferencia, este consiste en la re-autenticación del usuario para estar seguros de que el mismo es un administrador del sistema y que presenta todos los permisos para realizar esta operación.

3.5 Conclusiones.

En este capítulo se modelaron las clases del diseño que intervienen en la elaboración del módulo de conferencia, se detallaron cada uno de los diagramas de secuencias que trae consigo cada escenario de los requisitos que se tienen. Se realizó el diagrama entidad-relación, el cual tiene la responsabilidad de detallar los tipos de relaciones que existen entre cada una de las entidades que intervienen. Se explicaron los patrones de diseños, el tratamiento de errores y la confección de la ayuda para la comprensión de cada funcionalidad que se brinda.

Capítulo IV: Implementación y Pruebas.

4.1 Introducción

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

4.2 Implementación.

4.2.1 Diagrama de componentes.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones, en ellos se muestran las operaciones que se realizan. Los mismos pueden ser simples archivos, paquetes o bibliotecas.

4.2.2 Matriz de Integración de Componentes Interna.

| | Componentes | | |
|----------------------|------------------|-----------|----------|
| Componentes Internos | Plan de Marcado. | Telephony | Asterisk |
| Conferencia | x | x | x |

Tabla 14 Matriz de Integración de Componentes.

4.3 Descripción diseño caso de prueba.

La descripción de los diseño de casos de prueba (DCP) se realizan para cada uno de los requisitos funcionales, con el objetivo de obtener una documentación por la cual guiarse a la hora de realizarle pruebas al sistema desarrollado. Sin embargo a continuación se colocará el DCP del RF Adicionar conferencia, ya que colocarlos todos sería engorroso y opacaría la calidad del trabajo.

Para ver los restantes DCP remitirse al expediente de proyecto.

| Nombre del requisito | Descripción general | Escenarios de pruebas | Flujo del escenario |
|----------------------|---------------------|-----------------------|---------------------|
|----------------------|---------------------|-----------------------|---------------------|

Implementación y Pruebas.

| | | | |
|-----------------------|---|--|--|
| Adicionar conferencia | El objetivo de este requisito es adicionar una conferencia. | EP 1.1: Adicionar conferencia correctamente. | <ol style="list-style-type: none">1. Se selecciona el menú Inicio.2. Se selecciona la opción Platel3. Se selecciona la opción Conferencia.4. Se selecciona la opción Gestionar Conferencia.5. Se oprime el botón Adicionar.6. Se muestra la interfaz Registrar Conferencia.7. Se introducen el numero, nombre, PIN_usuario, PIN_admin, fecha_inicio, fecha_fin, max_participante.8. Se oprime el botón Registrar, se comprueban los datos, se cierra la interfaz Registrar Conferencia y se actualiza el grid de la interfaz Gestionar Conferencia. |
|-----------------------|---|--|--|

Implementación y Pruebas.

| | | | |
|--|--|--|---|
| | | EP 1.2: Adicionar conferencia incorrectamente. | <ol style="list-style-type: none">1. Se selecciona el menú Inicio.2. Se selecciona la opción Platel3. Se selecciona la opción Conferencia.4. Se selecciona la opción Gestionar Conferencia.5. Se oprime el botón Adicionar.6. Se muestra la interfaz Registrar Conferencia.9. Se introducen el numero, nombre, PIN_usuario, PIN_admin, fecha_inicio, fecha_fin, max_participante.7. El usuario llena los campos con datos incorrectos o deja campos en blanco y oprime el botón Registrar.8. El sistema pone los campos en rojo y muestra el mensaje: <i>Por favor verifique nuevamente que hay campos con valores incorrectos.</i> |
|--|--|--|---|

Implementación y Pruebas.

| | | | |
|--|--|------------------------------------|---|
| | | <p>EP 1.3: Cancelar operación.</p> | <ol style="list-style-type: none"> 1. Se selecciona el menú Inicio. 2. Se selecciona la opción Platel 3. Se selecciona la opción Conferencia. 4. Se selecciona la opción Gestionar Conferencia. 5. Se oprime el botón Adicionar. 6. Se muestra la interfaz Registrar Conferencia. 7. Se introducen el numero, nombre, PIN_usuario, PIN_admin, fecha_inicio, fecha_fin, max_participante. 8. Se oprime el botón Cancelar. 9. El sistema cancela las operaciones y se cierra la ventana. |
|--|--|------------------------------------|---|

Tabla 15 Descripción del DCP: Adicionar Conferencia.

| Id del escenario | Escenario | Numero | Nombre | PIN_usuario | PIN_admin | fecha_inicio | fecha_fin | Max_participante | Respuesta del sistema | Resultado de la prueba |
|------------------|--------------------------------------|--------|--------|-------------|-----------|--------------|-----------|------------------|------------------------------------|------------------------|
| EP 1.1: | Adicionar conferencia correctamente. | 555 | Fac2 | 123 | 1234 | 13/5/2010 | 13/5/2010 | 12 | Se adicionó la conferencia al grid | |
| EP | Adici | l(vací | l(vac | l(va | l(vacío | l(vacío | l(vacío | l(vacío) | Se muestra | |

Implementación y Pruebas.

| | | | | | | | | | |
|-------|--------|-----|------|------|------|--------|--------|------|--|
| 1.2: | onar | o) | ío) | cío) |) |) |) | | el campo en color rojo y se muestra un mensaje indicando que los campos tienen valores incorrectos . |
| | confe | opa | Pep | 123 | 1235 | 13/5/2 | 13/5/2 | 9 | |
| | rencia | | e | 4 | | 009 | 009 | | |
| | incorr | 666 | 325 | asa | asas | CASA | confer | 3 | |
| | ecta | | | s | | S | encia | | |
| | ment | | | | | | | | |
| | e. | 699 | Yais | 587 | 6987 | 5/6/20 | 5/6/20 | para | |
| | | | | 4 | | 10 | 10 | | |
| EP:13 | Cancel | NA | NA | NA | NA | NA | NA | NA | Se cancela la operación y se cierra la ventana. |
| | elar | | | | | | | | |
| | opera | | | | | | | | |
| | ción | | | | | | | | |

Tabla 16 Pruebas por escenario.

4.4 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).

El método utilizado fue el de caja negra o prueba de especificación, que se basa principalmente en los

Implementación y Pruebas.

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.

4.5 Conclusiones.

En el presente capítulo se describieron etapas de implementación y pruebas del software. Se establecieron y priorizaron las principales tareas de ingeniería asociadas a las historias de usuario para cada iteración y se realizaron los casos de pruebas que guiaron la calidad del sistema.

Conclusiones Generales

Conclusiones Generales

Al concluir el desarrollo del módulo administración del sistema de conferencia se arribó a la conclusión que se cumplieron todas las tareas de investigación planteadas, permitiendo cumplir el objetivo general de forma satisfactoria. Se investigaron las principales características y funcionalidades de diferentes distribuciones telefónicas así como del servicio de conferencias, logrando obtener una aplicación funcional que responde a cada uno de los requerimientos funcionales planteados. Además, se obtuvo un software capaz de brindar al usuario la opción de gestionar conferencia y administrarlas de forma sencilla en tiempo real dentro de la plataforma telefónica.

Para lograr este resultado se utilizaron diferentes herramientas que ayudaron a implementar la aplicación. Como lenguaje de programación PHP 5.0 y JavaScript, los frameworks Zend Frameworks y Doctrine, el gestor de base de datos fue PostgreSQL y servidor web Apache. Para realizar un modelado y describir todos los procesos realizados para la realización del presente trabajo se utilizó el lenguaje UML y la herramienta del Visual Paradigm.

Recomendaciones

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

- ✓ Incorporar Video-Web en la aplicación con el objetivo de brindar videoconferencia desde la web.

Referencias Bibliográficas

1. Garcia, Pedro. Telefonía tradicional.
2. Telefonía IP en la Universidad de Chile. ¿Qué es la Telefonía IP? [En línea] [Citado el: 20 de 01 de 2010.] http://www.telefoniaip.uchile.cl/capacitacion_telefonia.htm.
3. VoIP. ¿Qué significa VoIP? [Online] [Cited: 02 20, 2010.] <http://www.masadelante.com/faqs/voip> .
4. Elastix freedom to communicate. Elastix Description. [Online] [Cited: enero 15, 2010.] <http://www.elastix.org/>.
5. Asteris-GUI. [Online] [Cited: 01 6, 2010.] <http://eee.voip-info.org/wiki/view/Asterisk+GUI>.
6. Software, Edgewall. FreePBX. FreePBX feature at a Glance. [Online] [Cited: 01 23, 2010.] <http://www.freepbx.org/trac/>.
7. JavaScript Ya. Que es el jQuery? [Online] [Cited: 02 5, 2010.] <http://www.javascriptya.com.ar/jquery/temarios/descripcion.php?cod=57&punto=1&inicio=0>.
8. TextosCientificos.com. El modelo cliente servidor. [Online] [Cited: 12 14, 2009.] <http://www.textoscientificos.com/redes/tcp-ip/servicios-capa-transporte>.
9. lobo tuerto. Ext JS 3.0 ya está aquí. [Online] [Cited: 02 2, 2010.] <http://lobotuerto.com/blog/2009/07/13/ext-js-3-0-ya-esta-aqui/>.
10. doctrine. *Relacional de objetos Mapper*. [En línea] [Citado el: 21 de 12 de 2009.] http://translate.google.com/cu/translate?hl=es&sl=en&u=http://www.doctrine-project.org/&ei=7sUXTO_2L8H98AbZm4m3CQ&sa=X&oi=translate&ct=result&resnum=1&ved=0CCYQ7gEwAA&prev=/search%3Fq%3Ddoctrine%26hl%3Des.
11. Sitio de descargas de software. Visual Paradigm for UML (ME). [Online] [Cited: 02 9, 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
12. desarrolloweb.com. Zend Studio. [En línea] Miguel Angel Alvarez. [Citado el: 21 de abril de 2010.] <http://www.desarrolloweb.com/articulos/1178.php>.

Bibliografía

garcia, Pedro. Telefonía tradicional.

Sitio de descargas de software. Visual Paradigm for UML (ME). [En línea] [Citado el: 9 de 02 de 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.

VoIP. ¿Qué significa VoIP? . [En línea] [Citado el: 20 de 02 de 2010.] <http://www.masadelante.com/faqs/voip>.

Elastix freedom to communicate. Elastix Descrption. [En línea] [Citado el: 15 de enero de 2010.] <http://www.elastix.org/>.

Asteris-GUI. [En línea] [Citado el: 6 de 01 de 2010.] <http://eee.voip-info.org/wiki/view/Asterisk+GUI>.

Software, Edgewall. FreePBX. FreePBX feature at a Glance. [En línea] [Citado el: 23 de 01 de 2010.] <http://www.freepbx.org/trac/>.

TextosCientificos.com. El modelo cliente servidor. [En línea] [Citado el: 14 de 12 de 2009.] <http://www.textoscientificos.com/redes/tcp-ip/servicios-capa-transporte>.

JavaScript Ya. Que es el jQuery? [En línea] [Citado el: 5 de 02 de 2010.] <http://www.javascriptya.com.ar/jquery/temarios/descripcion.php?cod=57&punto=1&inicio=0>.

lobo tuerto. Ext JS 3.0 ya está aquí. [En línea] [Citado el: 2 de 02 de 2010.] <http://lobotuerto.com/blog/2009/07/13/ext-js-3-0-ya-esta-aqui/>.

Yaismery Armenteros Rodríguez, José Antonio Hernández Martínez. 2010. Especificación de requisitos de software. Ciudad Habana: s.n., 2010.

—. 2010. Glosario de Términos. Ciudad Habana: s.n., 2010.

—. 2010. Historia de usuario. Ciudad Habana: s.n., 2010.

—. 2010. Modelo conceptual. Ciudad Habana: s.n., 2010.

—. 2010. Seguimiento a los requisitos. Ciudad Habana: s.n., 2010.

Grupo de desarrollo de gestión. 2010. Estándar para el diseño de interfaces. Ciudad Habana: s.n., 2010.

Grupo de desarrollo de gestión. 2010. Estándar de documentación. Ciudad Habana: s.n., 2010.

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

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

garcia, Pedro. Telefonía tradicional.

Sitio de descargas de software. Visual Paradigm for UML (ME). [Online] [Cited: 02 9, 2010.]

http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.

VoIP. ¿Qué significa VoIP? . [Online] [Cited: 02 20, 2010.] <http://www.masadelante.com/faqs/voip>.

Elastix freedom to communicate. Elastix Description. [Online] [Cited: enero 15, 2010.] <http://www.elastix.org/>.

Asteris-GUI. [Online] [Cited: 01 6, 2010.] <http://eee.voip-info.org/wiki/view/Asterisk+GUI>.

Software, Edgwall. FreePBX. FreePBX feature at a Glance. [Online] [Cited: 01 23, 2010.]

<http://www.freepbx.org/trac/>.

TextosCientificos.com. El modelo cliente servidor. [Online] [Cited: 12 14, 2009.]

<http://www.textoscientificos.com/redes/tcp-ip/servicios-capa-transporte>.

JavaScript Ya. Que es el jQuery? [Online] [Cited: 02 5, 2010.]

<http://www.javascriptya.com.ar/jquery/temarios/descripcion.php?cod=57&punto=1&inicio=0>.

lobo tuerto. Ext JS 3.0 ya está aquí. [Online] [Cited: 02 2, 2010.]

<http://lobotuerto.com/blog/2009/07/13/ext-js-3-0-ya-esta-aqui/>.

B, Manuel. Diseño de software con patrones. [En línea] 24 de febrero de 2001. [Citado el: 20 de abril de 2010.] <http://www.programacion.com/articulo/disenode-software-con-patrones-parte-4-145>.

Voztelecom, Desarrolladores de Sistemas de. Modelo de configuración básica de Asterisk 1.4 con la plataforma de Voztelecom. Barcelona : Cerdanyola. NIF: B-63102149.

UCID, Grupo de desarrollo del. Propuesta de Guía para la presentación del Trabajo de Diploma en la UCID. Ciudad Habana : s.n., 2009-2010.

Proceso de Desarrollo y Gestión de Proyectos de Software (1ra Versión). Ciudad Habana : s.n., 2009.

UCID, Desarrolladores del. Lista de chequeo. Modelo de diseño. Ciudad Habana : s.n., 2010.

Lista de chequeo. Especificación de Requisitos. Ciudad Habana : s.n., 2010.

Lista de chequeo. Diseños de casos de prueba. Ciudad Habana : s.n., 2010.

Estándar de Ingeniería de Software Versión 1.2. Ciudad Habana : s.n., 2010.

Estándar de Ingeniería de Software Versión 1.1. Ciudad Habana : s.n., 2010.

desarrolloweb.com. Zend Studio. [En línea] Miguel Angel Alvarez. [Citado el: 21 de abril de 2010.]
<http://www.desarrolloweb.com/articulos/1178.php>.

Anexos

Anexo No.1 Especificación de la planilla Modelo conceptual.

| Nombre de la entidad | Conferencia. | | | | | |
|----------------------------------|--|-----------------------|------------------|------------|-----------------------|---------------------------------|
| Descripción de la entidad | Aplicación que permite a los usuarios estar conectados en cadenas en una misma línea telefónica, es decir sala de conferencia. | | | | | |
| Nombre del atributo | Descripción | Tipo | ¿Puede ser nulo? | ¿Es único? | Restricciones | |
| | | | | | Clase válidas | Clases no válidas |
| numero. | Número de la conferencia | Entero | No | Si | Números | Letras. |
| nombre. | Nombre | Cadena de caracteres. | No | Si | Números y letras. | Caracteres especiales. |
| pin_usuario | Contraseña para ingresar como usuario. | Cadena de caracteres. | Si | Si | Números | Letras y caracteres especiales. |
| pin_admin. | Contraseña para ingresar como Admin. | Cadena de caracteres. | Si | Si | Números. | Letras y caracteres especiales. |
| fecha_inicio | Fecha de inicio | fecha | No | No | Números. | Letras y caracteres especiales. |
| fecha_fin | Fecha de culminación de la llamada. | fecha | No | No | Números | Letras y caracteres especiales. |
| duración | Duración de la | Cadena de caracteres. | Si | No | Cadena de caracteres. | Cadena de caracteres. |

| | conferencia | | | | | |
|----------------------------------|--|-----------------------|------------------|------------|---------------|---------------------------------|
| max_participantes | Máximo de participantes | Cadena de caracteres. | No | No | Números | Letras |
| menentrada | Sonido que se escucha al entrar. | bool | Si | No | 1/0 | Letras y caracteres especiales. |
| esperaradmin. | Empieza la conf. Cuando ingresa el Admin | bool | No | No | 1/0 | Letras y caracteres especiales. |
| avisoentradasal | Anuncia el nombre del usuario que entra o sale | bool. | No | No | 1/0 | Letras y caracteres especiales. |
| musespera | Activa o desactiva la música en espera | bool | No | No | 1/0 | Letras y caracteres especiales. |
| | | | | | | |
| Nombre de la entidad | Participantes. | | | | | |
| Descripción de la entidad | Usuarios que van a participar en una sala de conferencias. | | | | | |
| Nombre del atributo | Descripción | Tipo | ¿Puede ser nulo? | ¿Es único? | Restricciones | |
| | | | | | Clase válidas | Clases no válidas |
| adm_conferenciaid | Identificador de la conferencia | Entero. | No. | Si. | Números. | Letras y caracteres especiales |

| | | | | | | |
|---------------|--|---------|-----|-----|----------|---|
| | que se va a administrar a la cual pertenecen. | | | | | |
| num_conf | Numero de la conferencia a la cual pertenece. | Entero. | No. | Si. | Números. | Letras y caracteres especiales |
| estado | Si está conectado o desconectado. | bool | Si | No | 1/0 | Letras, números y caracteres especiales |
| serv_habla | Habilita el servicio de que solo podrá hablar en la conferencia. | bool | Si | No | 1/0 | Letras, números y caracteres especiales |
| serv_escucha | Habilita el servicio de que solo podrá escuchar en la conferencia. | bool | Si | No | 1/0 | Letras, números y caracteres especiales |
| serv_explusar | Se expulsara de la conferencia según el administrador. | bool | Si | No | 1/0 | Letras, números y caracteres especiales |

Tabla 17: Especificación del Modelo conceptual.

Anexo No.2 DCP Modificar conferencia.

| Nombre del requisito | Descripción general | Escenarios de pruebas | Flujo del escenario |
|-----------------------|---|--|--|
| Modificar conferencia | El objetivo de este requisito es modificar una conferencia. | EP 1.1: Modificar conferencia correctamente. | <ol style="list-style-type: none"> 1. Se selecciona el menú Inicio. 2. Se selecciona la opción Platel. 3. Se selecciona la opción Conferencia. 4. Se selecciona la opción Gestionar Conferencia. 5. Se muestra la interfaz Gestionar Conferencia. 6. Se selecciona en el grid de la interfaz la conferencia que se desea modificar. 7. Se oprime el botón Modificar. 8. Se modifican los campos que se deseen y se oprime el botón Modificar. 9. Se comprueban los datos, se cierra la interfaz Modificar conferencia, se actualiza el grid de la interfaz Gestionar Conferencia y se muestra un mensaje de confirmación. |

| | | | |
|--|--|---|---|
| | | <p>EP 1.2: Modificar conferencia incorrectamente.</p> | <ol style="list-style-type: none"> 1. Se selecciona el menú Inicio. 2. Se selecciona la opción Platel. 3. Se selecciona la opción Conferencia. 4. Se selecciona la opción Gestionar Conferencia. 5. Se muestra la interfaz Gestionar Conferencia. 6. Se selecciona en el grid de la interfaz la conferencia que se desea modificar. 7. Se oprime el botón Modificar. 8. El usuario modifica los campos con datos incorrectos y oprime el botón Modificar. 9. El sistema pone los campos en rojo y muestra el mensaje <i>Por favor verifique nuevamente que hay campos con valores incorrectos.</i> |
|--|--|---|---|

| | | | |
|--|--|----------------------------|---|
| | | EP 1.3: Cancelar operación | <ol style="list-style-type: none"> 1. Se selecciona el menú Inicio. 2. Se selecciona la opción Platel. 3. Se selecciona la opción Conferencia. 4. Se selecciona la opción Gestionar Conferencia. 5. Se muestra la interfaz Gestionar Conferencia. 6. Se selecciona en el grid de la interfaz la conferencia que se desea modificar. 7. Se oprime el botón Modificar. 8. Se modifican los campos que se deseen. 9. Se oprime el botón Cancelar. 10. El sistema cancela las operaciones y se cierra la ventana. |
|--|--|----------------------------|---|

Tabla 14 DCP Modificar Conferencia.

| No | Nombre de campo | Clasificación | Puede ser nulo | Descripción |
|----|-----------------|---------------|----------------|---|
| 1. | Numero | Texfield | NO | Permite introducir el número de la nueva conferencia. |
| 2. | Nombre | Texfield | NO | Permite introducir el nombre de la nueva conferencia. |

| | | | | |
|----|------------------|-----------|----|--|
| 3. | PIN_usuario | Textfield | NO | Permite introducir el pin de usuario para entrar a la nueva conferencia. |
| 4. | PIN_admin | Textfield | NO | Permite introducir el pin de administrador para poder entrar a la nueva conferencia. |
| 5. | fecha_inicio | combobox | NO | Permite introducir la fecha de inicio que tendrá la conferencia. |
| 6. | fecha_fin | combobox | NO | Permite introducir la fecha de fin que tendrá la conferencia. |
| 7. | max_participante | Textfield | NO | Permite introducir máximo de participantes que tendrá la conferencia. |

Tabla 15 Pruebas por escenario.

Anexo No.3 DCP Eliminar conferencia.

| Nombre del requisito | Descripción general | Escenarios de pruebas | Flujo del escenario |
|-----------------------|--|---------------------------------|--|
| Eliminar conferencia. | El objetivo de este requisito es eliminar una conferencia. | EP 1.1: Eliminar conferencia | <ol style="list-style-type: none"> 1. Se selecciona el menú Inicio. 2. Se selecciona la opción Platel. 3. Se selecciona la opción Conferencia. 4. Se selecciona la opción Gestionar Conferencia. 5. Se muestra la interfaz Gestionar Conferencia. 6. Se selecciona en el grid de la interfaz la conferencia que se desea eliminar. 7. Se oprime el botón Eliminar. 8. Se muestra un mensaje de confirmación: ¿Está seguro de que desea eliminar el campo? 9. Se oprime el botón Aceptar. 10. Se elimina la cuota del grid de la interfaz Gestionar Conferencia y se muestra un mensaje indicando que se eliminó correctamente. |

| | | | |
|--|--|--|---|
| | | <p>EP 1.2: Cancelar operación.</p> | <ol style="list-style-type: none"> 1. Se selecciona el menú Inicio. 2. Se selecciona la opción Platel. 3. Se selecciona la opción Conferencia. 4. Se selecciona la opción Gestionar Conferencia. 5. Se muestra la interfaz Gestionar Conferencia. 6. Se selecciona en el grid de la interfaz la cuota que se desea eliminar. 7. Se oprime el botón Eliminar. 8. Se muestra un mensaje de confirmación: ¿Está seguro de que desea eliminar el campo? 9. Se oprime el botón Cancelar. 10. El sistema cancela la operación y se cierra la ventana. |
|--|--|--|---|

Tabla 20 DCP Eliminar Conferencia.

Anexo No.4 DCP Buscar conferencia.

| Nombre del requisito | Descripción general | Escenarios de pruebas | Flujo del escenario |
|---------------------------|--|--|---|
| <p>Buscar conferencia</p> | <p>El objetivo de este requisito es buscar una conferencia</p> | <p>EP 1.1: Buscar conferencia de forma correcta.</p> | <ol style="list-style-type: none"> 1. Se selecciona el menú Inicio. 2. Se selecciona la opción Platel. 3. Se selecciona la opción Conferencia. 4. Se selecciona la opción Gestionar Conferencia. 5. Se muestra en la interfaz Gestionar Conferencia un grid con la información de todas las cuotas existentes. 6. Se introduce el número de la conferencia y se oprime buscar. 7. Se muestra un grid con la conferencia. |

| | | | |
|--|--|--|---|
| | | <p>EP 1.2: Buscar conferencia de forma incorrecta.</p> | <ol style="list-style-type: none"> 1. Se selecciona el menú Inicio. 2. Se selecciona la opción Conferencia. 3. Se selecciona la opción Gestionar Conferencia. 4. Se muestra en la interfaz Gestionar Conferencia un grid con la información de todas las cuotas existentes. 5. El usuario introduce el número de la conferencia que se desea buscar con datos incorrectos y oprime el botón Buscar. 6. El sistema muestra un mensaje de error y brinda la posibilidad de volver a buscar. |
|--|--|--|---|

Tabla 21 DCP Buscar Conferencia.

| Id del escenario | Escenario | Numero | Respuesta del sistema | Resultado de la prueba |
|------------------|---|--------|---|------------------------|
| EP 1.1 | Buscar conferencia de forma correcta. | 555 | Se muestra en la interfaz la conferencia que cumple con el número especificado. | |
| EP 1.2 | Buscar conferencia de forma incorrecta. | pepe | Se muestra el mensaje de error. | |

Tabla 22 Tabla de valores.

Glosario de Términos

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

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

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

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

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

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

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

Glosario de Términos

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

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

UML: Lenguaje Unificado de Modelado es un lenguaje de modelado para especificar o describir métodos o procesos.