



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**  
**FACULTAD 2 “TELECOMUNICACIONES Y SEGURIDAD INFORMÁTICA”**

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

**Título:** *“Interfaz para acceso al servidor de correos Zimbra mediante dispositivos móviles.”*

**Autores:**

Eduardo Arañó Garcés.

Jessica Marrero Echemendia.

**Tutor:**

Ing. Yordan Ernesto Estrada Rodríguez.

**Co-tutor:**

Ing. Joan Martínez Herrera.

Ciudad de la Habana, Julio del 2010.

“Año 52 de la Revolución”

**DECLARACIÓN DE AUTORÍA**

Declaramos que Eduardo Arañó Garcés y Jessica Marrero Echemendia somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) y a la facultad 2 para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de julio del 2010.

\_\_\_\_\_  
Firma del Autor  
Eduardo Arañó Garcés

\_\_\_\_\_  
Firma del Autor  
Jessica Marrero Echemendia

\_\_\_\_\_  
Firma del Tutor  
Ing. Yordan Ernesto Estrada Rodríguez

**AGRADECIMIENTOS**

*A mis padres por enseñarme a caminar por la vida.*

*A David, que con sólo una mirada conoce mis estados de ánimos.*

*A Yanira por convertirse en mi segunda hermana*

*A mis amigas y amigos, esas personas que siempre están cuando las necesitas, los del consejo oportuno, los de la compañía imprescindible.*

*A Yordan, por su paciencia inquebrantable y por guiarnos hasta el final.*

*Agradezco a la UCI, por convertirse a lo largo de estos cinco años en nuestra casa, nuestro barrio, nuestro pueblo; a sus profes, por darnos la oportunidad de crecer como personas.*

*A todos los que de una forma u otra me ayudaron en algún momento.*

*A los que siempre estarán en mi pensamiento*

*A todos gracias*

*Jessica*

*A mis padres por guiarme por el buen camino y ser mi apoyo en todo momento.*

*A Yudelkís, por levantar mi ánimo en los momentos más difíciles.*

*A mis amigas y amigos de la universidad que nunca olvidaré.*

*A Yordan, por ser ante todo un buen amigo.*

*A todos gracias*

*Eduardo*

**DEDICATORIA**

*...a mis padres, quienes una que otra noche soñaron mis anhelos, aquí se los dedico convertidos en realidad.*

*... a mi hermana y mi sobrino, por ser parte de mi vida.*

*Jessica*

*Dedico este trabajo y todo mi esfuerzo a las personas que más amo y estimo, mis padres, por brindarme todo el apoyo incondicional desde mis primeros años de estudio, por su preocupación constante, su confianza, por comprenderme y ser una guía en todo momento, a mis dos familias, que nunca dejaron de preocuparse y siempre están cuando los necesito, mis hermanos de la universidad los cuales son para toda la vida y muy especial a mi novia por ser mi apoyo en todo momento.*

*Eduardo*

## RESUMEN

Los dispositivos móviles se han convertido en un artículo personal indispensable en la actualidad por la alta infraestructura de comunicación y cobertura que posee la telefonía celular. Los contenidos desarrollados para los mismos por ende gozan de la misma popularidad y demanda.

Actualmente en la UCI (Universidad de las Ciencias Informáticas) se encuentra en proceso de desarrollo un proyecto perteneciente a la facultad 2 en coordinación con la entidad Cubacel el cual enarbola como principal objetivo el desarrollo de la mayor cantidad de servicios posibles para la telefonía celular.

Uno de los servicios que se desea ofrecer es el acceso al servidor de correos Zimbra a través de dispositivos móviles. Sin embargo, gran parte de los dispositivos existentes poseen restricciones a la hora de manejar la información debida principalmente al tamaño de los teclados y las pantallas de los mismos. Esto impide que se visualicen de forma correcta el contenido de las páginas clientes de Zimbra, dificultando el uso del servicio de correo electrónico Zimbra para una gran parte de los modelos de teléfonos celulares existentes.

Todo lo anteriormente expuesto evidencia la necesidad de rediseñar la interfaz que ofrece Zimbra para la mayor cantidad de teléfonos posible, permitiendo el acceso de los clientes al correo electrónico independientemente de las características físicas y el navegador que estos posean.

**Palabras claves:** dispositivos móviles, telefonía celular, Zimbra, Cubacel.

**ÍNDICE**

INTRODUCCIÓN.....	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA.....	5
1.1    Introducción.....	5
1.2    Telefonía móvil.....	5
1.3    WAP (Wireless Application Protocol).....	6
1.4    Zimbra.....	9
1.4.1    ¿Qué es Zimbra?.....	9
1.4.2    Arquitectura.....	10
1.4.3    Avances.....	11
1.5    Metodología de desarrollo.....	13
1.5.1    ¿Por qué Extreme Programming?.....	14
1.6    Herramienta de Desarrollo (IDE).....	15
1.6.1    Eclipse Galileo (3.5).....	16
1.7    Lenguaje de Programación.....	16
1.7.1    Java.....	16
1.7.2    XHTML.....	17
1.8    Openwave Phone Simulator.....	18
1.9    WURFL (Wireless Universal Resource File) 2.0.17 (ultima modificación: 26-04-2010).....	18
1.10    DNS (Domain Name System).....	20
1.10.1    Bind9 (Berkeley Internet Name Domain).....	20
1.11    CSS (Cascading Style Sheets).....	21
1.12    Conclusiones del capítulo.....	21

---

CAPÍTULO 2 EXPLORACIÓN, PLANIFICACIÓN, IMPLEMENTACIÓN Y PRUEBA .....	22
2.1 Introducción.....	22
2.2 Flujo actual del proceso .....	22
2.3 Propuesta del sistema .....	22
2.4 Personas relacionadas con el sistema.....	22
2.5 Requisitos no funcionales del sistema .....	23
2.6 Fase de exploración.....	24
2.6.1 Historias de usuarios (HU) .....	24
2.7 Fase de planificación .....	26
2.7.1 Estimación de esfuerzo por historia de usuario.....	26
2.7.2 Plan iteraciones .....	26
2.7.3 Plan de duración de iteraciones .....	27
2.7.4 Plan de entregas .....	27
2.8 Implementación.....	27
2.9 Prueba .....	29
2.9.1 Pruebas Unitarias .....	29
2.9.2 Pruebas de aceptación.....	29
2.10 Conclusiones del capítulo .....	30
CAPÍTULO 3 INSTALACIÓN Y CONFIGURACIÓN DEL AMBIENTE DE TRABAJO .....	31
3.1 Introducción.....	31
3.2 Servidor de correos Zimbra .....	31
3.3 Pasos para instalar y configurar el servidor ZCS 6.0.4 y Bind9 .....	32
3.4 Conclusiones del capítulo .....	46

---

CAPÍTULO 4 REDISEÑO DE LA INTERFAZ .....	47
4.1 Introducción.....	47
4.2 Rediseño de la interfaz. ....	47
4.3 Conclusiones del capítulo .....	65
CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD .....	66
5.1 Introducción.....	66
5.2 Modelo matemático COCOMO II.....	66
5.3 Características del proyecto .....	67
5.3.1 Entradas externas (EI) .....	67
5.3.2 Salidas externas (EO).....	67
5.3.3 Ficheros internos (LIF) .....	67
5.3.4 Interfaces externas (EIF).....	67
5.3.5 Consultas externas .....	68
5.3.6 Puntos de función desajustados .....	68
5.4 Cálculos de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo....	69
5.4.1 Cálculo de instrucciones fuentes .....	69
5.4.2 Cálculo de esfuerzo nominal .....	70
5.4.3 Ajuste del esfuerzo nominal .....	71
5.4.4 Resultados.....	73
5.5 Análisis de costo .....	74
5.6 Conclusiones del capítulo .....	74
CONCLUSIONES GENERALES.....	75
RECOMENDACIONES.....	76



BIBLIOGRAFÍA .....77

## INTRODUCCIÓN

Internet permite a los usuarios que navegan a través de la Web contar con un gran número de servicios, entre los que se encuentran; Transferencia de ficheros (FTP), Grupos de Noticias, Mensajería Instantánea, Comercio Electrónico y Correo Electrónico. Estos son de gran importancia porque constituyen la base del desarrollo informático en la actualidad y posibilitan a todos los clientes de la web intercambiar información de forma más rápida y eficiente a través de un ordenador personal, un ordenador portátil o un teléfono celular, este último de gran aceptación por su portabilidad y facilidad de uso.

Uno de los servicios más importantes es el correo electrónico que permite a los usuarios enviar y recibir mensajes rápidamente. Por este medio se puede enviar, no solamente texto, sino todo tipo de documentos digitales, garantizando el aprovechamiento del tiempo al máximo, tanto en cuestiones de trabajo, como en actividades de esparcimiento.

Una de las alternativas que ha adquirido popularidad actualmente para darle soporte al servicio de correo electrónico es la Zimbra Collaboration Suite (en lo adelante ZCS); esta utiliza los protocolos estándares de comunicación disponibles y brinda los siguientes beneficios:

- Sincronización con clientes de correo, direcciones y calendario; hace posible que los usuarios accedan a la información de sus calendarios, direcciones y correos electrónicos a través de programas clientes, como Microsoft Outlook, Mozilla Thunderbird y Novell Evolution.
- Interfaz web amigable; provee características robustas y similares a un escritorio de trabajo, como por ejemplo opciones Arrastrar y Soltar y despliegue de información sobre elementos al colocar el ratón sobre ellos. Todo ello en cualquier navegador de Internet.
- Zimbra Mobile; soporte de sincronización con dispositivos móviles.
- Búsqueda, almacenamiento y organización de mensajes de manera comprensiva; permite la búsqueda avanzada, incluyendo búsqueda por objetos específicos como números telefónicos.

En Cuba se desea llevar a cabo una migración hacia el software libre como una alternativa al software privativo, ya que este provoca la dependencia tecnológica del país. Como parte de la estrategia adoptada la Universidad de las Ciencias Informáticas (UCI) desempeña un papel fundamental en el proceso de migración, siendo pioneros en la implementación de Zimbra como solución de correo electrónico en el país.

Entre los temas desarrollados por los proyectos actuales de la UCI se encuentra el de telefonía móvil. Al mismo se le desean agregar servicios entre los cuales se encuentra el acceso al correo electrónico utilizando como servidor de correo Zimbra Collaboration Suite. En este momento la mayoría de los dispositivos no pueden visualizar correctamente el contenido de las páginas clientes de Zimbra, pues el despliegue de la información es muy limitado.

Debido a esta limitación ha surgido la necesidad de implementar una interfaz estándar que permita a los usuarios el acceso mediante Internet a este servicio, independientemente de las características físicas y el navegador que posea el dispositivo móvil.

Por tanto, con el objetivo de proporcionar una solución práctica a la situación problemática planteada, se define como problema a resolver de la investigación:

¿Cómo contribuir a mejorar el acceso y la visualización mediante dispositivos móviles al servicio de correos de Zimbra independientemente de las características físicas de estos dispositivos y el navegador que posean?

El objeto de estudio que se abordará hará referencia a los clientes de correo para celulares delimitando su campo de acción al cliente del servidor de correos Zimbra para teléfonos móviles.

Como objetivo general de la investigación se definió: modificar una interfaz WAP para el servidor de correos Zimbra que permita el acceso independientemente de las características del dispositivo móvil.

Planteando como objetivos específicos:

- Analizar y seleccionar las clases y componentes ya existentes que puedan ser reutilizadas.
- Definir o seleccionar el framework más adecuado para llevar a cabo el desarrollo de este trabajo.

Con el propósito de guiar el curso de la investigación, fueron planteadas las siguientes preguntas científicas:

- ¿Cómo funciona el protocolo WAP para el envío de información?
- ¿Cómo obtener información del servidor de correos Zimbra y transmitirla mediante el protocolo WAP?
- ¿Qué herramientas de software son más flexibles para la implementación de una interfaz de correos para Zimbra?

Buscando brindar respuestas objetivas a las preguntas científicas planteadas fueron desarrolladas las siguientes tareas de investigación:

- Análisis de los clientes o interfaz WAP existente para Zimbra.
- Estudio del lenguaje de programación Java.
- Estudio del framework Spring.
- Estudio del framework MyMobileWeb.
- Investigación acerca del protocolo WAP.

Los temas a tratar durante la investigación estarán relacionados al problema de cómo llevar a cabo la implementación de una interfaz WAP para el servidor de correos Zimbra que se adapte a las características de los dispositivos móviles. Ofreciendo de forma satisfactoria el servicio de correo electrónico a los clientes de telefonía celular en la UCI. Además, la propia implementación constituye un aporte en el desarrollo de software en Cuba.

Para la realización del trabajo se tuvieron en cuenta métodos científicos como:

### **Métodos Teóricos:**

- **Método Histórico-Lógico:**  
Este método ayudará a entender el surgimiento y la evolución del tema de la investigación, así como otras temáticas estrechamente relacionadas con la misma a lo largo de la historia de la Informática. Se utiliza al realizarse un estudio del surgimiento y análisis de las plataformas de aprendizaje virtual para móviles, abordando además en su evolución.
- **Método Analítico-Sintético:**  
Este método servirá para analizar y comprender la teoría y documentación relacionada con el tema de investigación, permitiendo así, extraer los elementos más relacionados e importantes con el objeto de estudio. En nuestro trabajo se utiliza para definir los conceptos y principios de la arquitectura.
- **Método Inductivo-Deductivo:**  
Son formas de razonamiento que nos permite llegar a un grupo de conocimientos generalizadores, tanto desde el análisis de lo particular a lo general, como desde el análisis de elementos generalizadores a uno de menor nivel de generalización. Se utiliza cuando se selecciona la tecnología, después de un estudio de las existentes, a utilizar en el flujo de trabajo de Implementación con vista a obtener un producto.

### **Métodos Empíricos:**

- Método Entrevista:

Es una conversación planificada para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales del entrevistado y puede influir en determinados aspectos de la conducta humana por lo que es importante una buena comunicación. Se utiliza en nuestro trabajo para entender que es lo que el cliente desea.

El contenido de esta investigación se encuentra desglosado en 5 capítulos, las recomendaciones, referencias bibliográficas, un glosario de términos y finalmente los anexos que son de vital importancia para su entendimiento.

En el capítulo 1 “Fundamentación teórica” se realizará una descripción de las herramientas que serán utilizadas para desarrollar la aplicación. También se hará alusión a las principales características del servidor de correos Zimbra así como los avances más significativos que ha tenido en el mundo de la telefonía celular desde su surgimiento.

El capítulo 2 “Exploración, planificación, implementación y prueba” hace referencia a las fases propias de la metodología de desarrollo seleccionada. En este capítulo se encuentran los artefactos obtenidos durante el curso de este trabajo, los cuales harán más sencilla la comprensión del sistema.

En el capítulo 3 “Instalación y configuración del ambiente de trabajo” se describen los detalles acerca de la instalación y configuración de las herramientas utilizadas.

En el capítulo 4 “Rediseño de la interfaz” se describen los principales cambios realizados durante el rediseño de la interfaz con el objetivo de obtener una mejor compatibilidad con la mayor cantidad de dispositivos móviles.

El capítulo 5 “Estudio de factibilidad” se hace un estudio de la factibilidad para la ejecución del sistema propuesto. Se obtienen resultados como el esfuerzo, tiempo de desarrollo, cantidad de hombres, salario medio y costo que se necesitará para la culminación del mismo.

### CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

#### 1.1 Introducción

En este capítulo se describen los principales aspectos y conceptos que han sido objeto de análisis a lo largo de la investigación. Se hace referencia a las tecnologías y metodologías a utilizar, el lenguaje de programación elegido, así como los avances que ha tenido Zimbra en el mundo de la telefonía celular.

#### 1.2 Telefonía móvil

Las tecnologías inalámbricas han tenido mucho desarrollo en los últimos años, una de las de mayor desarrollo ha sido la telefonía celular o móvil; que es un sistema que permite al usuario establecer y recibir una comunicación de voz o datos dentro de la zona de cobertura de la antena con la que el Terminal móvil se ha conectado una vez encendido. A pesar de que fue concebida inicialmente para transmitir únicamente señales de voz debido a las limitaciones tecnológicas de la época, la tecnología celular de hoy en día es capaz de brindar otros tipos de servicios tales como almacenamiento de datos, reproducción de audio y video, navegación en Internet, realización de compras *online* y el envío de mensajes electrónicos entre otros.

El desarrollo de esta tecnología ha pasado varias etapas. Para separar una etapa de la otra se ha categorizado por generaciones, las cuales serán explicadas a continuación.

La primera generación de la telefonía móvil hizo su aparición en 1979, esta se caracterizó por el gran tamaño y peso de los teléfonos. Funcionaban de manera analógica y estrictamente para voz. La calidad de los enlaces de voz era muy baja y la seguridad no existía. Se diseñó para el uso exclusivo en el vehículo.

La segunda generación a diferencia de la primera se caracterizó por ser digital. Utiliza protocolos de codificación más sofisticados y son los sistemas de telefonía celular usados en la actualidad. Los protocolos empleados en los sistemas de esta generación soportan velocidades de información más altas para voz pero limitados en comunicaciones de datos. Se pueden ofrecer servicios auxiliares tales como datos, fax y *SMS (Short Message Service)* (1).

La segunda generación y media o generación 2.5 y 2.7 es una tecnología basada en paquetes que suponen una mejora en la transmisión de datos. Integra *WAP (Wireless Application Protocol)*, *MMS (Multimedia Messaging System)*, *SMS (Short Message Services)*. Puede llegar a la velocidad de 384 Kbps. Como tal no existe ningún estándar ni tecnología a la que se pueda llamar 2.5G o 2.75G, pero suelen ser denominados así a algunos teléfonos móviles 2G que incorporan algunas de las mejoras y tecnologías del estándar 3G.

La tercera generación soporta tecnología *IP (Internet Protocol)* y *ATM (Asynchronous Transfer Mode)* lo que hace el acceso a Internet más fiable. Los protocolos empleados en los sistemas de la tercera generación soportan más altas velocidades de información enfocados para aplicaciones más allá de la voz tales como audio (MP3), video en movimiento, video conferencia y acceso rápido a Internet (2).

En Cuba, a pesar de las difíciles condiciones económicas se han realizado grandes esfuerzos para ir a la par del desarrollo de la telefonía celular en el mundo y la diversidad de los servicios disponibles en esta tecnología. Sin embargo, en el país existe desde hace algún tiempo un proveedor de servicios conocido como Cubacel, que es una sucursal de la Empresa de Telecomunicaciones de Cuba (ETECSA); la cual brinda soporte a los teléfonos de segunda generación y realiza esfuerzos para implementar la generación 2.5 y 2.7. Actualmente en la UCI existe un proyecto en desarrollo dedicado a la implementación de servicios para celulares. Uno de los servicios que se desea ofrecer es el acceso al servidor de correo Zimbra mediante una interfaz WAP que sea accesible desde terminales móviles.

### **1.3 WAP (Wireless Application Protocol)**

El protocolo de aplicaciones inalámbricas ha sido un factor clave para la evolución de la telefonía móvil, permitiendo la implementación de un gran número de servicios. WAP es un estándar que les posibilita a los usuarios con dispositivos móviles un fácil acceso a toda la información disponible en Internet así como a las aplicaciones y servicios que la misma ofrece.

En la figura que se muestra a continuación se pueden observar los diferentes componentes en la arquitectura del protocolo WAP:

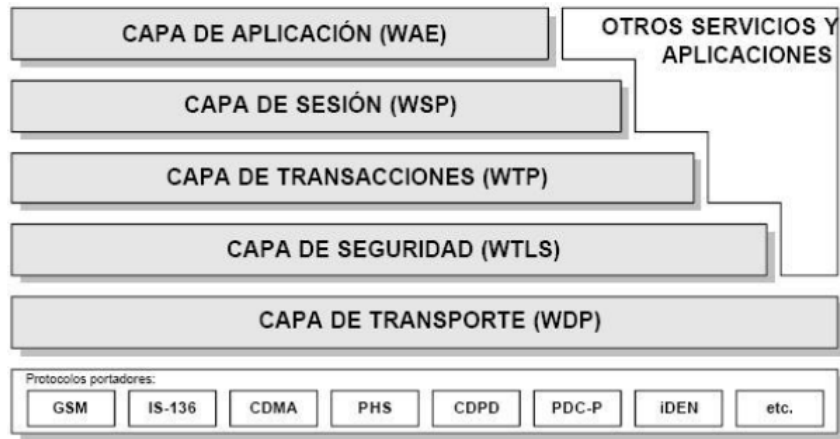


Figura 1 : Arquitectura del protocolo WAP.

## Capa de Aplicación

El entorno inalámbrico de aplicación (WAE, del inglés *Wireless Application Environment*) es un entorno de aplicación de propósito general, basado en la combinación del *World Wide Web (WWW)* y tecnologías de comunicaciones móviles.

Este entorno incluye un micro navegador, que tiene básicamente las siguientes funcionalidades:

- Un lenguaje denominado WML (Lenguaje de Marcas Inalámbrico) similar al HTML (Lenguaje de Marcas de Hipertexto), pero optimizado para su uso en terminales móviles.
- Un lenguaje denominado WMLS, que es un lenguaje de script para dotar de cierto dinamismo a sus documentos.
- Un conjunto de formatos de contenido bien definidos entre los que se encuentran imágenes, entradas en la agenda de teléfonos e información de calendario (3).

## Capa de Sesión

Actualmente, esta capa consiste en servicios adaptados a aplicaciones basadas en la navegación Web, proporcionando las siguientes funcionalidades:

- Semántica y funcionalidades del HTTP/1.1 en una codificación compacta.



- Negociación de las características del Protocolo.
- Suspensión de la Sesión y reanudación de la misma con cambio de sesión.

### Capa de Transacciones

El Protocolo Inalámbrico de Transacción (WTP) funciona por encima de un servicio de datagramas, tanto seguros como no seguros, proporcionando las siguientes funcionalidades:

- Tres clases de servicio de transacciones:
  - Peticiones inseguras de un solo camino.
  - Peticiones seguras de un solo camino.
  - Transacciones seguras de dos caminos (petición-respuesta)
- Seguridad usuario-a-usuario opcional.
- Transacciones asíncronas.

### Capa de seguridad

La Capa Inalámbrica de Seguridad de Transporte (WTLS, del inglés Wireless Transport Layer Security) es un protocolo basado en el estándar SSL, utilizado en el entorno Web para la proporción de seguridad en la realización de transferencias de datos. Para este protocolo se han definido las siguientes características:

- Integridad de los datos.
- Privacidad de los datos.
- Autenticación.

### Capa de Transporte

El Protocolo Inalámbrico de Datagramas (WDP, del inglés Wireless Datagram Protocol) proporciona un servicio fiable a los protocolos de las capas superiores de WAP y permite la comunicación de forma

transparente sobre los protocolos portadores válidos. Debido a que este protocolo proporciona un interfaz común a los protocolos de las capas superiores, las capas de Seguridad, Sesión y Aplicación pueden trabajar independientemente de la red inalámbrica que dé soporte al sistema (4).

### 1.4 Zimbra

#### 1.4.1 ¿Qué es Zimbra?

*Zimbra Collaboration Suite (ZCS)* es un cliente servidor de correo creado por Zimbra Inc.; compañía ubicada en San Mateo, California, adquirida por Yahoo! Inc. por aproximadamente 350 millones de dólares en septiembre del 2007, haciendo un acuerdo de mantener sus estándares de Software Abierto.

Zimbra ha sido desarrollado en Java y complementado con AJAX (Java script + XML). Se dispone del código fuente completo (y binaria para distribuciones Debian, Ubuntu y Red Hat Linux como Fedora y RHEL), documentación y herramientas de migración (para *Exchange*). Soporta acceso *POP (Post Office Protocol)*, acceso *IMAP (Internet Message Access Protocol)*, entre otros; e incluye protección *anti-spam* y antivirus (5).

ZCS puede funcionar con cientos de usuarios a pleno rendimiento en un solo servidor de gama media. Está compuesto por varios módulos entre los que se encuentran:

- Módulo de calendarios públicos y privados, citas con participantes y reserva de recursos. Desde este módulo los usuarios pueden gestionar su calendario y el de sus compañeros, gracias al avanzado sistema de permisos.
- Módulo de contactos y tareas. Este permite que los usuarios gestionen contactos personales y compartidos así como las listas de distribución, también posibilita mantener un conjunto de tareas personales y compartidas.
- Dispone de un módulo que permite gestionar los correos; el mismo brinda varios beneficios al usuario entre los que cabría destacar:
  - Vista por conversación: No importa en qué carpeta se encuentren los mensajes, Zimbra se encarga de buscar y mostrar por conversación todos los correos.

- Suscripción a canales corporativos de noticias o externos.
  - Opción de guardar nuestras búsquedas más habituales.
  - Compartir carpetas de correo, con otros compañeros.
  - Agregar cuentas externas POP3/IMAP.
  - Tecnología “Arrastrar y soltar”.
- Maletín de documentos: Permite gestionar los documentos, los cuales están organizados por carpetas, e indexa sus contenidos internos en la base de datos para después realizar búsquedas internas dentro de los mismos. El acceso a los documentos se realiza desde la propia interfaz o de forma nativa desde cualquier PC de escritorio.
  - Crear y compartir contenido con la Wiki, esto constituye la mejor forma de compartir contenido con otros usuarios. Ofrece una potente interfaz que permite crear espacios de textos e imágenes dentro de Zimbra y posteriormente publicarlos a otros medios.
  - Comunicarse en tiempo real con el *chat*, agilizando las comunicaciones internas. Las conversaciones quedan archivadas y se pueden dejar mensajes aún cuando el usuario no está conectado.
  - Firma y cifrado de Zimbra gracias a lo cual el usuario puede enviar y recibir correos de forma segura.
  - Incluye un potente motor de búsqueda avanzada dentro del *Webmail* el cual funciona con todos los módulos de Zimbra. Al mismo tiempo que indicamos nuestras preferencias de búsqueda, el sistema nos presenta los resultados en tiempo real. Finalmente, se pueden guardar las búsquedas avanzadas para posteriormente volver a ejecutarlas con un solo *click*.
  - Brinda el empleo de etiquetas para organizar la información de forma rápida.

### 1.4.2 Arquitectura

Cuatro son los pilares de la arquitectura del Zimbra Collaboration Suite:

- Flexibilidad: fácil personalización según las necesidades de la organización.
- Libertad: uso de cualquier navegador web y de aplicaciones de escritorio tradicionales.
- Durabilidad: servidor de correo electrónico y calendario extraordinariamente fiable y ampliable.
- Bajo coste de mantenimiento: gestión muy sencilla, tanto mediante una interfaz gráfica como desde la consola.

Se puede decir que los componentes que hay dentro de una distribución de Zimbra son:

- **Zimbra Core:** Contiene las utilidades, librerías y archivos de configuración. Este paquete se instalará por defecto en todos los servidores.
- **Zimbra LDAP:** Contiene el sistema de autenticación bajo OpenLDAP para almacenar y gestionar el almacén de usuarios, integrando de serie el soporte para la replicación. Además, permite fácilmente su configuración para el uso de directorios LDAP externos, incluyendo Active Directory de Microsoft o Directory de Novell.
- **Zimbra MTA:** (*Mail Transfer Agent*) contiene el servidor de Envío y recepción de correo en Postfix. También contiene el Spamasassin y ClamAV. El MTA puede integrarse con otras tecnologías, como Postgrey para greylisting o Spamhaus para DNSBL. Por otra parte, soporta por defecto los protocolos principales de cifrado de canal, SSL y TLS
- **Zimbra Store:** Contiene todos los buzones implementados con el Jetty, MySQL, Apache, etc. Otro componente integrado en Zimbra es Lucene, un potente motor de indexación y búsquedas que permite a los usuarios y administradores buscar mensajes a través de múltiples carpetas de correo, tanto metadatos como contenidos en el cuerpo del mensaje.

### 1.4.3 Avances

A continuación se realizará un bosquejo de los avances que ha tenido Zimbra en la implementación del servicio de correo electrónico para celulares en el mundo.

Uno de los beneficios más relevantes que brinda ZCS es *Zimbra Mobile Web (ZMW)* que no es más que un soporte de sincronización con dispositivos móviles. Zimbra Mobile Web permite que tanto los usuarios de Zimbra Collaboration Suite 6.0 Open Source como los de Network Edition puedan acceder a sus correos, sus contactos y calendarios usando el cliente web para móvil que brinda esta Suite. Todo lo que se necesita es un navegador web que soporte la tecnología *HTML (Hyper Text Markup Language)* en el cual el usuario pueda escribir la dirección de su servidor Zimbra. No requiere licencias adicionales y se puede utilizar de forma gratuita.

Zimbra Mobile Web le ofrece al cliente las siguientes características:

- Acceso en tiempo real al correo electrónico, libreta de direcciones y calendario.
- Permite ver el contenido compartido.
- Realizar búsquedas avanzadas en el buzón.
- Ver todas las carpetas, búsquedas guardadas, banderas y etiquetas.
- Vista de mensajería instantánea.

Zimbra se sincroniza con teléfonos inteligentes y los clientes de correo Outlook y Thunderbird.

Zimbra Mobile Web para teléfonos inteligentes permite la sincronización de correo entre los dispositivos móviles y el servidor Zimbra. Su principal característica es el *push mail* el cual envía mensajes en tiempo real hacia los celulares una vez que han llegado al servidor de correo.

Zimbra cuenta con un módulo para BlackBerry, la cual es una solución óptima para las organizaciones que le dan soporte a estos dispositivos y desean migrar a la próxima generación de mensajería y plataforma de colaboración con los principios del software libre.

El iPhone de Apple es un dispositivo popular para usar con Zimbra, especialmente en el mercado de la educación. El iPhone y Zimbra juntos proveen a los usuarios lo mejor de ambos mundos; una solución completa de mensajería y colaboración, además de la mejor clase de servicios y entretenimientos para el consumidor.

Hay muchas formas de usar iPhone para acceder al servidor de Zimbra:

- El módulo de Zimbra para iPhone. Obtiene la sincronización *OTA (Over The Air)* para el correo básico, libro de direcciones y el calendario en cualquier iPhone con Zimbra Network Edition y Zimbra Mobile habilitados.
- Buscador web para móviles. El cliente web de Zimbra está optimizado para Safari en el iPhone; aprovechando las capacidades únicas del buscador del iPhone para acceder al correo, libro de direcciones y calendario.
- IMAP /POP. Estándares bases para acceder al correo.

### 1.5 Metodología de desarrollo

Se entiende por metodología de desarrollo una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software (6). Una metodología es importante para el proceso de desarrollo de software porque permite obtener un producto con la calidad requerida, en el tiempo esperado y con el coste planificado.

Existen varias propuestas metodológicas entre las que se encuentran las metodologías pesadas o tradicionales y las ágiles o ligeras.

Las metodologías pesadas o tradicionales resultan convenientes para proyectos que requieran mucho tiempo y recursos, sin embargo para proyectos donde el entorno del sistema es muy cambiante y el tiempo se encuentra restringido es necesario utilizar una metodología ágil.

Las metodologías ágiles están orientadas especialmente a proyectos pequeños y permiten que los programadores se concentren solamente en aquellas funciones que se necesitan inmediatamente. Se realizan entregas al cliente frecuentemente, de las cuales se obtiene retroalimentación constante, posibilitando respuestas rápidas a los cambios en el negocio (7).

Para llevar a cabo la implementación de esta aplicación se decidió adoptar *Extreme Programming (XP)* como metodología de desarrollo a utilizar.

### 1.5.1 ¿Por qué Extreme Programming?

Básicamente se podría decir que la Programación Extrema es una “metodología ligera o ágil para el desarrollo de software eficiente” que busca clara y fundamentalmente dos objetivos: hacer software bien (con calidad) y de la forma más rápida posible (8).

XP es una de las metodologías de desarrollo de software más exitosa actualmente. Es utilizada para proyectos de corto plazo y equipo reducido. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener al cliente o usuario final como parte del equipo, pues es uno de los requisitos para llegar al éxito del proyecto. El cliente al ser participante activo del proceso de desarrollo posee algunos derechos como:

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

Esta metodología se basa en:

- Pruebas Unitarias: pruebas realizadas a los principales procesos, de tal manera que al adelantarse en algo hacia el futuro, se pueden hacer pruebas de las fallas que pudieran ocurrir. Es como si el equipo se adelantara a obtener los posibles errores.
- Refabricación: se basa en la reutilización del código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo.

Lo fundamental en Extreme Programming es:

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

Se decidió utilizar XP debido a que se adapta en gran medida al tipo de proyecto a desarrollar, a las condiciones de trabajo y a la idea de desarrollo que se tiene del sistema. Seguidamente se exponen las razones fundamentales que llevaron al uso de esta metodología:

- Posibilita la realización de reuniones informales todos los días para intercambiar experiencias del día anterior.
- El proyecto es pequeño. XP está concebida para ser utilizada dentro de proyectos pequeños.
- El cliente forma parte del equipo de desarrollo. Mediante la aplicación de XP se puede lograr una mayor retroalimentación y lograr un producto que satisfaga sus necesidades.
- Poca disponibilidad de personal. El sistema debe ser realizado por dos personas solamente, haciendo imposible la existencia de muchos roles ni la especialización de un rol en específico por parte de los miembros.
- Empieza en pequeño y añade funcionalidad con retroalimentación de forma continua. El desarrollo del sistema comienza a partir de los requerimientos básicos y a partir de ahí se van añadiendo las funcionalidades que tanto el cliente como el desarrollador consideren necesarias.

### 1.6 Herramienta de Desarrollo (IDE)

Un entorno de desarrollo integrado o *IDE (Integrated Development Environment)*, es un programa informático compuesto por un conjunto de herramientas de programación las cuales son: un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi y Visual Basic. Un IDE puede funcionar con varios lenguajes de programación, tal es el caso de Eclipse, al que se le puede añadir soporte de lenguajes adicionales mediante *plugins*.



### 1.6.1 Eclipse Galileo (3.5)

Eclipse es una plataforma de desarrollo open source basada en Java. Su principal aplicación es *JDT* (*Java Development Tool* o Herramienta de Desarrollo para Java), además de una serie de *plugins* que se encuentran acompañando a la plataforma, tales como: Ant, Compare, Core, CVS, Debug, Help, Jface, Releng, Scripting, Search, SWT, Text, UI, Update, Team y WebDAV. Eclipse presenta las siguientes características:

- Pruebas unitarias con JUnit.
- Asistentes para la creación de proyectos y clases.
- Conservación del registro de las versiones que se encarga de generar y mantener la documentación de cada etapa del proyecto.
- Es soportado por sistemas operativos como Windows, Mac OS X y Linux.

### 1.7 Lenguaje de Programación

Los lenguajes de programación son idiomas artificiales que se utilizan para hacer programas. Están formados por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura. Entre los lenguajes de programación más conocidos tenemos C++, C#, Delphi, Visual Basic, Pascal, Java, etc.

#### 1.7.1 Java

El lenguaje de programación Java, fue diseñado con el objetivo de crear un lenguaje que pudiera funcionar en redes formadas por más de un tipo de computadora y que fuera independiente de la plataforma en la que se vaya a ejecutar. Un programa de Java puede ejecutarse en cualquier máquina o plataforma. Entre sus características más significativas podemos encontrar:

- Tiene una sintaxis muy similar a lenguajes como C o C++.
- Es un lenguaje robusto porque es capaz de manejar una memoria de forma automática evitando que el usuario tenga que encargarse de eso.
- Es seguro porque contiene políticas que evitan que se puedan codificar virus.

- Es portable e independiente gracias a la máquina virtual de Java.

Es una tecnología orientada al desarrollo de software y ha adquirido importancia en el ambiente de programación web con su plataforma *JEE (Java Enterprise Edition)*. Es un lenguaje orientado a objetos y es considerado una referencia a nivel mundial para la programación destinada a terminales móviles. Es por estas razones que ha sido seleccionado para dar solución a los problemas planteados.

### 1.7.2 XHTML

Ante la llegada al mercado de un gran número de dispositivos, XHTML (acrónimo de Extensible Hypertext Markup Language, Lenguaje de Marcado de Hipertexto Extensible) surge como el lenguaje cuyo etiquetado, permite una correcta interpretación de la información independientemente del dispositivo desde el que se accede a ella.

XHTML es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos (9).

XHTML puede incluir otros lenguajes como *MathML (Mathematical Markup Language)*, *SMIL (Synchronized Multimedia Integration Language)* o *SVG (Scalable Vector Graphics)*, al contrario que HTML. Los desarrolladores de contenidos web al utilizar XHTML aseguran la compatibilidad con aplicaciones de usuario pasadas y futuras.

Al estar orientado al uso de un etiquetado correcto, exige una serie de requisitos básicos a cumplir en lo que a código se refiere. Entre estos requisitos básicos se puede mencionar una estructuración coherente dentro del documento donde se incluirían elementos correctamente anidados, etiquetas en minúsculas, elementos cerrados correctamente y atributos de valores entrecomillados.

Las interfaces mostradas por el servidor de correos Zimbra han sido creadas utilizando el lenguaje XHTML, constituyendo el principal motivo de su selección para llevar a cabo este trabajo. Además, el uso de este lenguaje aporta algunos beneficios tales como:

- Los desarrolladores de aplicaciones de usuarios y documentos descubren constantemente nuevas formas de expresar sus ideas usando nuevas etiquetas.
- Un sitio web o documento XHTML funciona con todos los navegadores web y en todos los dispositivos portátiles con soporte XHTML al mismo tiempo, por ejemplo, con un celular.
- Un mismo documento sirve para todos los soportes. Ya no hace falta desarrollar para la web por un lado y para otros dispositivos por separado.
- XHTML no deja lugar a errores. Al ser más estricto que el HTML elimina gran cantidad de errores frecuentes que resultan en la incompatibilidad entre dispositivos.

### 1.8 Openwave Phone Simulator

OpenWave Phone Simulator, es un emulador de teléfonos móviles que permite la visualización de las páginas WEB en un navegador WAP. Posee un navegador integrado donde al colocar la URL muestra el contenido dentro de una pantalla con un atractivo formato de teléfono móvil (10).

### 1.9 WURFL (Wireless Universal Resource File) 2.0.17 (ultima modificación: 26-04-2010)

Debido a la gran diversidad de dispositivos móviles que existen hoy en día y a las distintas dimensiones de pantallas, formato de imágenes y audio que soportan; surge en el mundo una gran polémica entre los desarrolladores de páginas o sitios WAP. Para dar solución a este problema se creó WURFL.

WURFL es un archivo XML de configuración que contiene información sobre las capacidades y características de todos los dispositivos móviles existentes, además, es actualizado cada cierto tiempo. Mediante su uso, se pueden construir aplicaciones que pueden ser visualizadas en diferentes entornos, adaptadas a las capacidades disponibles en cada dispositivo. De esta manera, el mismo contenido, ya sea imagen o sonido, puede utilizarse en distintos dispositivos sin la necesidad de tener varias versiones de la aplicación o codificar usando muchos bloques condicionales en función del elemento (11).

El principal objetivo de dicho fichero es recopilar toda la información posible sobre los terminales móviles que acceden a páginas WAP, de manera que los desarrolladores sean capaces de construir mejores aplicaciones y servicios para los usuarios.

### Características de WURFL

- Repositorio de dispositivos basado en fichero XML (wurfl.xml).
  - Los dispositivos se estructuran en torno a un árbol de herencia, donde cada nodo del árbol representa a un dispositivo físico o a toda una familia de dispositivos.
  - Para cada dispositivo o familia se asocia un identificador llamado *User-Agent* y un identificador de dispositivo padre (*fallback*).
  - Para cada dispositivo solo se definen las capacidades que varían con respecto a su padre.
  - Existe un dispositivo “generic” que define valores “tipo” para todas las capacidades y que es la raíz del árbol.
  - Para introducir nuevos dispositivos o rectificar valores de los existentes, sin tener que esperar a una nueva versión, se permiten introducir ficheros *wurfl\_patch.xml* que sobrescriben los valores del *wurfl.xml*.
- *API (Application Programming Interface)* de reconocimiento de dispositivos y obtención de capacidades.
- Es un proyecto Open Source.

Existen multitud de APIs escritas en diferentes lenguajes de programación para trabajar con el XML de WURFL, entre estos se encuentran:

- Java
- PHP
- Perl
- Python
- .net

Debido a las especificaciones de Zimbra, se decidió el uso de la API implementada en el lenguaje Java, por estar acorde al lenguaje de programación en el que ha sido implementada dicha aplicación.

### 1.10 DNS (Domain Name System)

En la mayoría de las redes modernas, incluyendo la Internet, los usuarios localizan otras máquinas por su nombre. Esto libera a los usuarios de la pesada tarea de recordar la dirección numérica de los recursos de red. La forma más efectiva de configurar una red para permitir tales conexiones basadas en nombres es configurando un *Domain Name Service (DNS)* o servidor de nombres, el cual resuelve los nombres de *hosts* en la red a direcciones numéricas y viceversa.

#### 1.10.1 Bind9 (Berkeley Internet Name Domain)

Una de las dependencias del Zimbra durante su instalación es la configuración de un servidor DNS para resolver nombre de dominio, garantizando el correcto envío y recepción de los correos, para esto fue seleccionado BIND9, el cual es utilizado especialmente en sistemas GNU/Linux. Entre las principales características que propiciaron su selección se encuentran:

- *Mejoras al protocolo DNS*: soporta IXFR (del inglés *Incremental Zone Transfers*, Transferencias de zona incremental), donde un servidor de nombres esclavo sólo descargará las porciones actualizadas de una zona modificada en un servidor de nombres maestro.
- *Vistas múltiples*: A través del uso de la declaración *view* en *named.conf*, BIND puede presentar información diferente dependiendo desde cuál red se esté realizando la petición (12).
- *Seguridad*: BIND soporta un número de métodos diferentes para proteger la actualización y zonas de transferencia, en los servidores de nombres maestro y esclavo:
  - *DNSSEC (DNS Security)*: Permite firmar con caracteres criptográficos zonas con una clave de zona.
  - *TSIG (Transaction SIGnatures)*: permite una transferencia desde el maestro al esclavo sólo después de verificar que una llave secreta compartida existe en ambos servidores maestro y en el esclavo.
- *IPv6*: Puede proporcionar servicios de nombres en ambientes IP versión 6.

### 1.11 CSS (Cascading Style Sheets)

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación; y es imprescindible para crear páginas web complejas (13). Dicho lenguaje se utiliza para definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, la tabulación con la que se muestran los elementos de una lista.

#### Ventajas:

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

#### Desventaja:

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

Zimbra posee un sistema de archivos que establecen una jerarquía que le permite al sistema funcionar eficientemente. Incluida en este sistema de organización, se encuentra el uso de las CSS. Teniendo en cuenta este sistema jerárquico, se decidió, en plena coordinación con el cliente, reutilizar el sistema de archivos existente, por lo que independientemente de la desventaja que incluye el uso de las CSS en el sistema se hará uso de esta tecnología.

### 1.12 Conclusiones del capítulo

En este capítulo se llevó a cabo un estudio de las herramientas que serán utilizadas durante el desarrollo de este trabajo. Se definió un entorno de desarrollo basado en Java como lenguaje de programación haciendo uso de Eclipse como editor, escogiendo como metodología de desarrollo a XP.

### CAPÍTULO 2 EXPLORACIÓN, PLANIFICACIÓN, IMPLEMENTACIÓN Y PRUEBA

#### 2.1 Introducción

En el presente capítulo se hace alusión a las fases de exploración, planificación, implementación y prueba de la metodología de desarrollo que se seleccionó para guiar la elaboración del sistema propuesto, además se exponen los artefactos generados durante este período así como los requerimientos no funcionales con los que debe contar el sistema.

#### 2.2 Flujo actual del proceso

Actualmente Zimbra brinda varias versiones para acceder al sistema, estas son:

- Predeterminada.
- Avanzada (Ajax).
- Estándar (HTML).
- Móvil.

Si el usuario desea acceder al sistema desde su móvil debe seleccionar de forma manual la opción que existe con este propósito.

#### 2.3 Propuesta del sistema

Se propone la realización de un sistema en el cual el usuario no se vea en la obligación de seleccionar la interfaz para móvil. El sistema debe ser capaz de identificar cuando una petición proviene de un teléfono móvil o de una computadora y posteriormente mostrar la interfaz correspondiente.

#### 2.4 Personas relacionadas con el sistema

Personas relacionadas con el sistema	Justificación
Usuario	Es la persona que interactúa con el sistema.

Tabla 1 Personas relacionadas con el sistema.

### 2.5 Requisitos no funcionales del sistema

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener; son las características que lo hacen atractivo, usable, rápido y confiable.

Para lograr la satisfacción del cliente y una buena calidad en el sistema se listaron los siguientes requerimientos no funcionales.

#### Apariencia o interfaz externa:

RNF1. Usar la menor cantidad de imágenes y en el formato más idóneo de acuerdo a las características del móvil, además de tener buena calidad visual y ser comprensibles.

RNF2. Cuando se usan menús o listas seleccionables con un gran número de opciones deben ser paginadas.

RNF3. La aplicación se debe visualizar de forma correcta en todos los dispositivos móviles.

#### Usabilidad:

RNF4. La información debe poder ser accedida por todos los usuarios de móviles que tengan habilitado servicio de navegación, sin poder modificar el contenido de las páginas.

RNF5. La aplicación debe ser sencilla de utilizar para cualquier cliente que necesite acceder.

#### Software:

Requerimientos mínimos para la máquina servidora:

RNF6. Disponer del Sistema Operativo Linux distribución Debian.

RNF7. Tener instalado la Máquina Virtual de Java versión 1.6.0 o superior.

RNF8. Contenedor Web que soporte la especificación Java Servlet 2.3 y JSP 1.2 o superior.

Requerimientos mínimos para los terminales móviles:

RNF9. Contar con un navegador que soporte XHTML.



### Hardware:

Requerimientos mínimos para la máquina servidora:

RNF10. CPU Intel/AMD de 32bits a 1.5 GHz o superior.

RNF11. Memoria RAM de 1 GB.

RNF12. Espacio libre en disco duro de 5GB.

### Restricciones en el diseño y la implementación:

RNF13. El análisis y el diseño de la aplicación se realizarán bajo la Metodología *Extreme Programming*.

RNF14. Se utilizarán los lenguajes de programación XHTML y Java.

## **2.6 Fase de exploración**

La metodología de desarrollo *Extreme Programming* comienza con la fase de exploración en la cual los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. A la vez el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán para dar cumplimiento al sistema propuesto.

### **2.6.1 Historias de usuarios (HU)**

El primer paso para llevar a cabo cualquier proyecto utilizando la metodología XP es definir las Historias de Usuarios (HU) con los clientes. Las HU tienen el mismo objetivo que los casos de uso pero con algunas diferencias en cuanto a la forma de descripción; constan de 3 ó 4 líneas en las cuales se describen brevemente las características que el sistema debe tener. Éstas se escriben desde la perspectiva del cliente aunque los desarrolladores pueden brindar ayuda en su identificación. El tiempo de desarrollo ideal para una historia de usuario es entre una y tres semanas (14).

Son utilizadas en la fase de pruebas para verificar si el programa cumple con lo especificado por la historia de usuario. Cada una de ellas debe ser lo más sencilla y comprensible posible para que los programadores puedan implementarla en unas semanas. Las historias de usuario solamente proporcionan los detalles sobre la estimación de riesgo y cuanto tiempo conllevará la implementación de dicha historia (15).

## Capítulo 2: Exploración, planificación, implementación y prueba

Las HU se clasifican de la siguiente manera:

### 1. Teniendo en cuenta la *Escala Nominal de Prioridad en el Negocio*:

**Alta:** Se le otorga a las Historias de Usuarios que constituyen funcionalidades de vital importancia en el desarrollo del proyecto.

**Media:** Se le otorga a las Historias de Usuarios que para el cliente constituyen funcionalidades a tener en cuenta sin que tengan una afectación directa sobre el proyecto que se está desarrollando.

**Baja:** Se le otorga a las Historias de Usuarios que constituyen funcionalidades que sirven de ayuda al control de la estructura y que no tienen nada que ver con el proyecto en desarrollo.

### 2. Teniendo en cuenta la *Escala Nominal de Riesgo en Desarrollo*:

**Alta:** Se otorga cuando para la implementación de la Historia de Usuario se considera la posible existencia de errores que lleven a la inoperatividad del código.

**Media:** Se otorga cuando pueden aparecer errores en la implementación de la Historia de Usuario que puedan retrasar la entrega de la versión.

**Baja:** Se otorga cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Seguidamente se muestran las HU que se utilizarán en el desarrollo de la aplicación:

Historia de Usuario	
Número: 1	Nombre de Historia de Usuario: Reconocer el dispositivo cliente.
Modificación de usuario número: Ninguna.	
Usuario: Eduardo Arañó Garcés. Jessica Marrero Echemendia.	Iteración asignada: 1

<b>Prioridad en negocio:</b> Alta	<b>Puntos estimados:</b> 2
<b>Riesgo en desarrollo:</b> Alta	<b>Puntos reales:</b> 2
<b>Descripción:</b> Se reconocerá si el dispositivo cliente es un móvil o una computadora para posteriormente mostrar la interfaz que corresponda.	
<b>Observaciones:</b>	

Tabla 2 Historia de Usuario “Reconocer el dispositivo cliente”.

### 2.7 Fase de planificación

Durante la fase de planificación se lleva a cabo una estimación del esfuerzo que costará implementar cada una de las Historias de Usuarios. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Como unidad de medida se utilizará el punto de estimación, considerando que una semana de trabajo sin interrupciones es equivalente a un punto.

#### 2.7.1 Estimación de esfuerzo por historia de usuario

En la siguiente tabla se muestra la estimación que se llevó a cabo para el desarrollo del sistema propuesto:

<b>Historias de Usuario.</b>	<b>Puntos de estimación.</b>
Reconocer el dispositivo cliente.	2

Tabla 3 Estimación de esfuerzo por Historia de Usuario.

#### 2.7.2 Plan iteraciones

Una vez determinadas las Historias de Usuarios del sistema y estimado el esfuerzo para realizar la implementación de cada una de ellas; se procede a la planificación de la etapa de implementación del proyecto. Para cumplir con la tarea anteriormente planteada se decidió realizar la misma en una iteración la cual se describe a continuación.

### 2.7.2.1 Iteración 1

En esta iteración se implementará la historia de usuario *Reconocer el dispositivo cliente*, de prioridad alta. Al final de esta iteración se contará con un prototipo del sistema, el cual será presentado al cliente con el objetivo de lograr una retroalimentación para el grupo de trabajo.

### 2.7.3 Plan de duración de iteraciones

Para lograr una mayor organización del trabajo se crea un plan de duración de las iteraciones; el mismo tiene como objetivo mostrar la duración de cada iteración así como el orden en que serán implementadas las historias de usuarios en cada una de ellas. En este caso se realizó un solo plan debido a que existe un único equipo de desarrolladores. A continuación se muestra como quedarán distribuidas las Historias de Usuarios según el orden en que serán abordadas en cada iteración teniendo en cuenta su prioridad y el tiempo de duración de las mismas.

Iteraciones.	Orden de las Historias de Usuarios a Implementar.	Duración de las Iteraciones.
Iteración 1	1- Reconocer el dispositivo cliente.	2

Tabla 4 Plan de duración de iteraciones.

### 2.7.4 Plan de entregas

A continuación se muestra el plan de entregas para la fase de implementación.

Sistema	3ra Semana abril	1ra Semana mayo	3ra Semana mayo
Zimbra	0.1	0.2	1.0

Tabla 5 Plan de entregas.

## 2.8 Implementación

Al inicio de cada iteración se realiza una revisión del plan de iteraciones y se modifica de ser necesario. Todo el trabajo de la iteración se expresa en tareas de programación y cada una de ellas es asignada a un programador responsable. Estas tareas, pueden escribirse utilizando un lenguaje técnico y no necesariamente deben ser entendibles para el cliente (14).

## Capítulo 2: Exploración, planificación, implementación y prueba

Seguidamente se muestran las tareas generadas en la fase de implementación para la historia de usuario “Reconocer el dispositivo cliente”.

Tarea	
<b>Número de tarea:</b> 1	<b>Número de historia:</b> 1
<b>Nombre de tarea:</b> Modificar código del fichero login.jsp.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 08/04/10	<b>Fecha fin:</b> 14/04/10
<b>Responsables:</b> Eduardo Arañó Garcés.	
<b>Descripción:</b> Se modificará el código del fichero login.jsp que se encuentra ubicado en <code>/opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/public/login.jsp</code> para eliminar el componente select que ofrece las opciones para especificar la vista que se le mostrará al usuario.	

Tabla 6 Tarea 1 de la historia de usuario “Reconocer el dispositivo cliente”.

Tarea	
<b>Número de tarea:</b> 2	<b>Número de historia:</b> 1
<b>Nombre de tarea:</b> Incluir la librería a utilizar (WURFL).	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 15/04/10	<b>Fecha fin:</b> 17/04/10
<b>Responsables:</b> Eduardo Arañó Garcés.	
<b>Descripción:</b> Se hará uso de WURFL filtrando las peticiones para determinar si la misma proviene de un celular o de una computadora.	

Tabla 7 Tarea 2 de la historia de usuario “Reconocer el dispositivo cliente”.

### 2.9 Prueba

Uno de los pilares de XP es el proceso de pruebas (14). Durante esta etapa serán probados todos los componentes del producto tanto por el cliente como por el equipo de desarrollo. Esto posibilita aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección.

XP divide las pruebas del sistema en dos grupos:

- **Pruebas unitarias**; encargadas de verificar el código diseñado por los programadores.
- **Pruebas de aceptación o pruebas funcionales**; destinadas a evaluar si al final de una iteración se consiguieron las funcionalidades requeridas diseñadas por el cliente (16).

#### 2.9.1 Pruebas Unitarias

Durante la implementación de un sistema cada desarrollador tiene que ir probando constantemente lo que va obteniendo para garantizar que las funcionalidades exigidas por el cliente estén siendo implementadas correctamente. Estas pruebas son conocidas como Pruebas Unitarias y aunque no generan artefactos y no son directamente palpables para el cliente, son de vital importancia para el desarrollo de un proyecto.

#### 2.9.2 Pruebas de aceptación

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las historias de usuario (16). Una historia de usuario puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una historia de usuario ha sido implementada correctamente. El objetivo final de estas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable (17).

Prueba de Aceptación	
<b>Código:</b> UH1_P1	<b>UH:</b> 1. Reconocer el dispositivo cliente.
<b>Nombre:</b> Reconocer el navegador del dispositivo cliente.	

<b>Descripción:</b> Prueba para comprobar que la aplicación muestra la interfaz correcta.
<b>Condiciones de ejecución:</b> El servidor debe estar ejecutándose. El usuario se debe conectar con una cuenta previamente creada.
<b>Entrada/Pasos de ejecución:</b> El usuario introduce usuario y contraseña. El sistema recibe la petición del navegador cliente. El sistema determina si la petición proviene de un móvil o de una computadora.
<b>Resultado esperado:</b> Se muestra la interfaz acorde con el navegador del dispositivo cliente que realizó la petición. No se producen errores.
<b>Evaluación de la prueba:</b> Prueba Satisfactoria.

Tabla 8 Prueba de aceptación.

### 2.10 Conclusiones del capítulo

En el presente capítulo se obtuvieron los artefactos generados por la metodología de desarrollo XP. Se obtuvo una historia de usuario y se definió el plan de entrega, así como las tareas a desarrollar en la fase de implementación. Se explicó la importancia de las pruebas en especial las de aceptación ya que miden la satisfacción del cliente con el producto desarrollado.

### **CAPÍTULO 3 INSTALACIÓN Y CONFIGURACIÓN DEL AMBIENTE DE TRABAJO**

#### **3.1 Introducción**

En el presente capítulo se abordarán detalles de la instalación y configuración de las herramientas utilizadas para el desarrollo del trabajo de curso.

#### **3.2 Servidor de correos Zimbra**

En capítulos anteriores se hizo referencia a este servidor de correos sobre el cual se desarrollará la solución propuesta. El mismo posee varias versiones con disímiles características y funcionalidades en dependencia de la versión utilizada. Luego de realizado un estudio, surgió como resultado la futura implementación de zcs-6.0.4\_GA\_2038.DEBIAN5.20091214191015 como servidor de correos en la Universidad de las Ciencias Informáticas. Esta tentativa fue en consecuencia la principal razón por la cual se decidió desarrollar sobre esta versión, además de los evidentes beneficios que aporta el mismo. Como son:

- Libreta de direcciones.
- Correo electrónico.
- Calendario de citas.
- Tareas.
- Documentos.
- Maletín.
- Mensajería instantánea.
- Preferencias.
- Etiquetado.
- Compartición.



- Cambio de contraseña.
- Elección de tema.
- Redacción de correos en formato HTML.
- Atajos de teclado.
- Acceso al Global Address List (GAL).
- Acceso externo por IMAP4/POP3.
- Creación de una dirección de reenvío automático del correo.
- Creación de una respuesta automática a la recepción de correos.
- Filtros de correo.
- Gestión de calendarios de grupos.
- Búsquedas avanzadas.
- Guardar búsquedas.

### 3.3 Pasos para instalar y configurar el servidor ZCS 6.0.4 y Bind9

1. Se instala un SO (Sistema Operativo) Debian 5 utilizando los paquetes mínimos requeridos a través de un disco de instalación. Vale destacar la necesidad de proceder, durante la instalación y configuración de las herramientas, mediante una consola con permisos de administración que permita realizar los cambios necesarios en el sistema. También se procede a instalar un servidor DNS, en este caso Bind9.

#### **Instalación del Sistema Operativo.**

Con el objetivo de llevar a cabo la instalación del servidor de correos, fue necesaria la selección e instalación de un sistema operativo compatible con Zimbra. Debian 5.0 es una opción factible ya que brinda una interfaz gráfica. Vale destacar, la necesidad de proceder durante la instalación y configuración

de las herramientas mediante una consola con permisos de administración, la cual permita realizar los cambios necesarios en el sistema facilitando la interacción con el mismo.

### **Instalación del servidor de nombres de dominio.**

Es de suma importancia tener en cuenta la necesidad que posee Zimbra de la existencia de un DNS previo a su instalación, lo cual garantiza el correcto funcionamiento del mismo. Fue seleccionado Bind9, descrito en capítulos anteriores, como alternativa al servidor DNS, debido a las ventajas que aporta durante su instalación y configuración.

### ***Pasos para la configuración del Bind9:***

- I. Luego de instalado el software, fue creado el dominio `zimbra.uci.cu`. Para lograr esto fue necesario incluir el fichero `etc/bind/db.zimbra.uci.cu`, el cual se encarga de describir y configurar el dominio que tendrá el futuro servidor de correos.

```
; BIND data file for local loopback interface
;
$TTL 604800
zimbra.uci.cu. IN SOA tesis07.zimbra.uci.cu. root.zimbra.uci.cu. (
; Serial
        604800      ; Refresh
        86400       ; Retry
        2419200    ; Expire
        604800 )   ; Negative Cache TTL
;
; DNS Server
@ IN NS tesis07.zimbra.uci.cu.
;
; Machine Names
localhost IN A 127.0.0.1
tesis07 IN A 10.31.9.6
tesis07.zimbra.uci.cu. IN A 10.31.9.6
zimbra.uci.cu IN A 10.31.9.6
client IN A 10.31.9.8
;
; MX Record
@ IN MX 10 tesis07.zimbra.uci.cu.
```

Figura 2: Código incluido en el archivo `etc/bind/db.zimbra.uci.cu`.

- II. Fue creado el fichero *etc/bind/db.inversa* que se encarga de dar solución a la necesidad de traducir de direcciones IP a nombres de dominio, lo cual posibilita la identificación de los clientes en la red.

```
; BIND reverse data file for local loopback interface
;
$TTL 604800
9.31.10.in-addr.arpa. IN SOA tesis07.zimbra.uci.cu. root.zimbra.uci.$
                        1      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@ IN NS tesis07.zimbra.uci.cu.
6 IN PTR zimbra.uci.cu.
6 IN PTR tesis07.zimbra.uci.cu.
```

Figura 3: Código incluido en el archivo *etc/bind/db.inversa*.

- III. Se hizo necesaria la modificación del fichero *etc/bind/name.conf* con el objetivo de registrar el dominio creado en el paso I.

```
zone "zimbra.uci.cu" {
    type master;
    file "/etc/bind/db.zimbra.uci.cu";
};

zone "10.in-addr.arpa" {
    type master;
    file "/etc/bind/db.inversa";
};
```

Figura 4: Código arreglado al archivo *etc/bind/name.conf*.

2. Una vez instalado el Sistema Operativo y el DNS es necesario modificar el archivo */etc/hosts*.

```
127.0.0.1    localhost.localdomain localhost
# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
fe00::0    ip6-localnet
ff00::0    ip6-mcastprefix
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
ff02::3    ip6-allhosts
10.31.9.6  tesis07.zimbra.uci.cu tesis07
```

Figura 5: Código incluido en el archivo `etc/hosts`.

Luego de este paso se debe reiniciar el servidor del `host` mediante el comando:

```
/etc/init.d/networking restart
```

Al cargar ejecutamos el comando `hostname -f` y en caso de que no ocurra ningún error en el proceso, se muestra en consola:

```
tesis07:/home/eduardo# hostname -f
tesis07.zimbra.uci.cu
tesis07:/home/eduardo#
```

Figura 6: Mensaje mostrado en consola

3. Se hace necesaria la configuración del archivo `/etc/resolv.conf` con la intención de asegurar que la configuración del IP del servidor DNS previamente instalado sea la correcta.

```
search uci.cu
search zimbra.uci.cu

nameserver 10.31.9.6
nameserver 10.0.0.3
nameserver 10.0.0.4

forward{
  10.0.0.3
  10.0.0.4
}
```

Figura 7: Código incluido en el archivo `/etc/resolv.conf`.

4. Intencionadamente y previendo futuras incompatibilidades o errores del sistema, se realiza la desinstalación de los paquetes innecesarios. A continuación se muestra el comando utilizado con este propósito:

```
apt-get remove --purge exim4 exim4-base exim4-config exim4-daemon-light
```

Finalmente, se lleva a cabo la instalación de los paquetes necesarios para la continuación del trabajo, dejando listo el sistema para la posterior instalación de ZSC (Zimbra):

```
apt-get install libc6-i686 sudo libidn11 curl fetchmail libgmp3c2 libexpat1 libgetopt-mixed-perl libxml2
libstdc++6 libpcre3 libltdl3 ssh
```

5. Una vez instaladas y configuradas las dependencias que posee Zimbra y con la intención de asegurar la integridad del software a utilizar, se obtiene el paquete de instalación de Zimbra de su sitio oficial <http://www.zimbra.com> (versión `zcs-6.0.4_GA_2038.DEBIAN5.20091214191015`)

Se descompacta el paquete comprimido:

```
tar xvfz zcs-6.0.4_GA_2038.DEBIAN5.20091214191015.tgz
```

Se redirecciona la consola al directorio:

```
cd zcs-zcs-6.0.4_GA_2038.DEBIAN5.20091214191015
```

Una vez dentro del directorio anteriormente mencionado, se da paso a la instalación del software utilizando para ello el siguiente comando:

## Capítulo 3: Instalación y configuración del ambiente de trabajo.

---

tesis07:/home/eduardo/Desktop# ./install.sh -l

Durante la instalación se necesitará la participación directa del usuario pues se deberá interactuar con la consola de configuración para introducir ciertos datos a la misma. A continuación se muestra el resultado obtenido una vez ejecutado el comando anterior:

```
Checking for existing installation...
zimbra-ldap...NOT FOUND
zimbra-logger...NOT FOUND
zimbra-mta...NOT FOUND
zimbra-snmp...NOT FOUND
zimbra-store...NOT FOUND
zimbra-apache...NOT FOUND
zimbra-spell...NOT FOUND

zimbra-proxy...NOT FOUND
zimbra-archiving...NOT FOUND
zimbra-convertd...NOT FOUND
zimbra-cluster...NOT FOUND
zimbra-core...
NOT FOUNDPLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING THE SOFTWARE.
ZIMBRA, INC. ("ZIMBRA") WILL ONLY LICENSE THIS SOFTWARE TO YOU IF YOU
FIRST ACCEPT THE TERMS OF THIS AGREEMENT. BY DOWNLOADING OR INSTALLING
THE SOFTWARE, OR USING THE PRODUCT, YOU ARE CONSENTING TO BE BOUND BY
THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS
AGREEMENT, THEN DO NOT DOWNLOAD, INSTALL OR USE THE PRODUCT.

License Terms for the Zimbra Collaboration Suite:
http://www.zimbra.com/license/zimbra_public_eula_2.1.html

Press Return to continueChecking for prerequisites...
FOUND: NPTL
FOUND: sudo-1.6.8p12-4
FOUND: libidn11-0.6.5-1
FOUND: fetchmail-6.3.6-1etch1
FOUND: libpcre3-6.7+7.4-4
FOUND: libgmp3c2-2:4.2.1+dfsg-4
FOUND: libexpat1-1.95.8-3.4
FOUND: libxml2-2.6.27.dfsg-6
FOUND: libstdc++6-4.1.1-21
FOUND: openssl-0.9.8c-4etch3
```

```
FOUND: libltdl3-1.5.22-4
Checking for suggested prerequisites...
FOUND: perl-5.8.8
Prerequisite check complete.

Checking for installable packages

Found zimbra-core
Found zimbra-ldap
Found zimbra-logger
Found zimbra-mta
Found zimbra-snmp
Found zimbra-store
Found zimbra-apache
Found zimbra-spell
Found zimbra-proxy

Select the packages to install

Install zimbra-ldap [Y] y
Install zimbra-logger [Y] y
Install zimbra-mta [Y] y
Install zimbra-snmp [Y] y
Install zimbra-store [Y] y
Install zimbra-apache [Y] y
Install zimbra-spell [Y] y
Install zimbra-proxy [N] y
Checking required space for zimbra-core
checking space for zimbra-store

Installing:
zimbra-core
zimbra-ldap
zimbra-logger
zimbra-mta
zimbra-snmp
zimbra-store
zimbra-apache
zimbra-spell
zimbra-proxy

The system will be modified. Continue? [N] y
```

```
Removing /opt/zimbra
Removing zimbra crontab entry...done.
done.
Cleaning up zimbra init scripts...done.
Cleaning up /etc/ld.so.conf...done.
Cleaning up /etc/security/limits.conf...done.

Finished removing Zimbra Collaboration Suite.

Installing packages

zimbra-core.....zimbra-core_5.0.11_GA_2695.DEBIAN4.0_i386.deb...done
zimbra-ldap.....zimbra-ldap_5.0.11_GA_2695.DEBIAN4.0_i386.deb...done
zimbra-logger.....zimbra-logger_5.0.11_GA_2695.DEBIAN4.0_i386.deb...done
zimbra-mta.....zimbra-mta_5.0.11_GA_2695.DEBIAN4.0_i386.deb...done
zimbra-snmp.....zimbra-snmp_5.0.11_GA_2695.DEBIAN4.0_i386.deb...done
zimbra-store.....zimbra-store_5.0.11_GA_2695.DEBIAN4.0_i386.deb...done
zimbra-apache.....zimbra-apache_5.0.11_GA_2695.DEBIAN4.0_i386.deb...done
zimbra-spell.....zimbra-spell_5.0.11_GA_2695.DEBIAN4.0_i386.deb...done
zimbra-proxy.....zimbra-proxy_5.0.11_GA_2695.DEBIAN4.0_i386.deb...done
Operations logged to /tmp/zmsetup.12222008-151821.log
Setting defaults...

DNS ERROR resolving MX for correo.ejemplo.com
It is suggested that the domain name have an MX record configured in DNS
Re-Enter domain name? [Yes]
Create Domain: [correo.ejemplo.com] zimbra.uci.cu
MX: correo.ejemplo.com (192.168.30.227)

Interface: 192.168.1.50
Interface: 127.0.0.1
done.
Cheking for port conflicts
```

Acá se observa un error de resolución DNS, donde Zimbra toma el nombre de la PC, para ello se tiene que corregir el nombre de dominio en este caso: zimbra.uci.cu (dominio creado previamente), una vez completado este paso se continúa con la instalación.

6.- Una vez instalado el Zimbra, en la misma consola, se muestra al usuario las opciones de configuración del servidor:



```
Main menu
1) Common Configuration:
2) zimbra-ldap: Enabled
3) zimbra-store: Enabled
+Create Admin User: yes
+Admin user to create: admin@ejemplo.comadmin@ejemplo.com
***** +Admin Password UNSET
+Enable automated spam training: yes
+Spam training user: spam.ovlhtoqr@ejemplo.comspam.ovlhtoqr@ejemplo.com
+Non-spam(Ham) training user: ham.ujyjbke@ejemplo.comham.ujyjbke@ejemplo.com
+Global Documents Account: wiki@ejemplo.comwiki@ejemplo.com
+SMTP host: correo.ejemplo.com
+Web server HTTP port: 80
+Web server HTTPS port: 443
+Web server mode: http
+IMAP server port: 7143
+IMAP server SSL port: 7993
+POP server port: 7110
+POP server SSL port: 7995
+Use spell check server: yes
+Spell server URL: http://correo.ejemplo.com:7780/aspell.php

4) zimbra-mta: Enabled
5) zimbra-snmp: Enabled
6) zimbra-logger: Enabled
7) zimbra-spell: Enabled
8) zimbra-proxy: Enabled
9) Default Class of Service Configuration:
r) Start servers after configuration yes
s) Save config to file
x) Expand menu
q) Quit
Address unconfigured (**) items (? - help)
```

En la opción 3 visualizada por consola, se puede observar que la misma no se encuentra configurada, con la intención de hacerlo, seleccionamos dicha opción con el objetivo de ingresar al siguiente submenú. Es necesario tener en cuenta la inclusión de una clave al usuario admin no menor de 6 dígitos garantizando la seguridad de la misma. Este usuario, será el único con los permisos suficientes para la administración y configuración del servidor instalado.

### Store configuration

- 1) Status: Enabled
  - 2) Create Admin User: yes
  - 3) Admin user to create: admin@ejemplo.comadmin@ejemplo.com
  - \*\* 4) Admin Password UNSET
  - 5) Enable automated spam training: yes
  - 6) Spam training user: spam.ovlhtoqr@ejemplo.comspam.ovlhtoqr@ejemplo.com
  - 7) Non-spam(Ham) training user: ham.ujyjbke@ejemplo.comham.ujyjbke@ejemplo.com
  - 8) Global Documents Account: wiki@ejemplo.comwiki@ejemplo.com
  - 9) SMTP host: correo.ejemplo.com
  - 10) Web server HTTP port: 80
  - 11) Web server HTTPS port: 443
  - 12) Web server mode: http
  - 13) IMAP server port: 7143
  - 14) IMAP server SSL port: 7993
  - 15) POP server port: 7110
  - 16) POP server SSL port: 7995
  - 17) Use spell check server: yes
  - 18) Spell server URL: http://correo.ejemplo.com:7780/aspell.php
- Select, or 'r' for previous menu [r]

En los resultados expuestos por consola es posible observar que la contraseña de administración, incluida en la opción 4, no se ha establecido. Seleccionamos esta opción e insertamos la contraseña para activarla

```
Password for admin@server1.example.com (min 6 characters): password
```

Luego se presiona enter con la intención de regresar al menú principal.

### Main menu

- 1) Common Configuration:
- 2) zimbra-ldap: Enabled
- 3) zimbra-store: Enabled
- 4) zimbra-mta: Enabled
- 5) zimbra-snmp: Enabled
- 6) zimbra-logger: Enabled
- 7) zimbra-spell: Enabled
- 8) zimbra-proxy: Enabled
- 9) Default Class of Service Configuration:

## Capítulo 3: Instalación y configuración del ambiente de trabajo.

---

```
r) Start servers after configuration yes
s) Save config to file
x) Expand menu
q) Quit
***CONFIGURATION COMPLETE --press 'a' to apply
```

Siguiendo las instrucciones dadas por la consola de instalación, se inserta “a” para aplicar los cambios realizados.

```
Select from menu, or press 'a' to apply config (? - help) a
Save configuration data to a file? [Yes] ----> Guardamos la configuracion
Save config in file: [/opt/zimbra/config.6778] ----> por default
Saving config in /opt/zimbra/config.6778...done.
The system will be modified - continue? [No] Y ----> Y para modificarlo

Operations logged to /tmp/zmsetup.12222008-151821.log
Setting local config values...done.
Setting up CA...done.
Creating SSL certificate...done.
Initializing ldap...done.
Setting replication password...done.
Setting Postfix password...done.
Setting amavis password...done.
Deploying CA to /opt/zimbra/conf/ca ...done.
Creating server entry for server1.example.com...done.
Setting spell check URL...done.
Setting service ports on server1.example.com...done.
Adding server1.example.com to zimbraMailHostPool in default COS...done.
Installing skins...
hotrod
lavender
waves
steel
sky
bones
yahoo
sand
lemongrass
beach
bare
done.
Setting zimbraFeatureIMEnabled=FALSE...done.
```

```
Setting zimbraFeatureTasksEnabled=TRUE...done.
Setting zimbraFeatureBriefcasesEnabled=TRUE...done.
Setting zimbraFeatureNotebookEnabled=TRUE...done.
Setting MTA auth host...done.
Setting TimeZone Preference...done.
Creating domain server1.example.com...done.
Creating user admin@server1.example.com...done.
Creating postmaster alias...done.
Creating user wiki@server1.example.com...done.
Creating user spam.m0bqyoayc@server1.example.com...done.
Creating user ham.ygch0qyz1@server1.example.com...done.
Setting spam training accounts...done.
Initializing store sql database...done.
Setting zimbraSmtphostname for server1.example.com...done.
Initializing logger sql database...done.
Initializing mta config...done.
Configuring SNMP...done.
Setting services on server1.example.com...done.
Setting up zimbra crontab...done.
Setting up syslog.conf...done.
```

Finalmente, dará la opción de notificar al sitio oficial de Zimbra acerca de su instalación. Seguidamente continúa configurando el sistema.

```
Starting servers...done.
Checking for deprecated zimlets...done.
Installing zimlets...
com_zimbra_date
com_zimbra_url
com_zimbra_cert_manager
com_zimbra_phone
com_zimbra_search
com_zimbra_local
com_zimbra_email
done.
Initializing Documents...done.
Restarting mailboxd...done.
Configuration complete – press return to exit
```

Por último, para verificar que todo esté bien es necesario cambiar al usuario zimbra

```
# su Zimbra
```

Después verificamos que todos los servicios estén activados de la siguiente forma:

```
$ zmcontrol status

Host correo.ejemplo.com
antispam Running
antivirus Running
imapproxy Running
ldap Running
logger Running
mailbox Running
mta Running
snmp Running
spell Running
stats Running
```

En caso de no arrojar un resultado parecido al mostrado anteriormente, se intenta ejecutar los servicios de Zimbra de la siguiente manera, logueado como usuario zimbra tecleamos el siguiente comando:

```
Zmcontrol start
```

Para entrar en el modo de administración del servidor es necesario ejecutar la consola administrativa mediante la siguiente dirección en el *host* local:

```
https://localhost:7071/zimbraAdmin/
```

Es importante mencionar que la "A" de zimbraAdmin debe estar en mayúsculas, de lo contrario no reconocerá la consola de administración. Una vez en dicha consola, se debe insertar el usuario Admin y su contraseña correspondiente (configurados durante la instalación). Esta consola permite la administración del servidor web con permisos para la gestión de usuarios y dominios.



Figura 8 : Interfaz de administración.

Para entrar en modo usuario es necesario teclear la siguiente dirección:

*http://localhost/*

Seguidamente se introduce el usuario con el siguiente formato “*usuario@midominio.com*” y la clave correspondiente, como información adicional vale destacar que antes de comenzar a instalar el Zimbra es necesario tener el DNS bien configurado con la intención de evitar un futuro error durante la instalación del servidor.



**Figura 9: Interfaz de autenticación.**

### **3.4 Conclusiones del capítulo**

En este capítulo se realizó una detallada descripción de cómo se deben instalar y configurar las herramientas para facilitar su posterior utilización.

### CAPÍTULO 4 REDISEÑO DE LA INTERFAZ

#### 4.1 Introducción

En el presente capítulo se lleva a cabo la reestructuración y modificación del código que permite la adaptación del diseño del cliente de correos Zimbra así como la inclusión y utilización de WURFL con la intención de amenizar y facilitarle al cliente su interacción con dicha interfaz desde su navegador móvil.

#### 4.2 Rediseño de la interfaz.

Una vez instaladas y configuradas las herramientas necesarias para el inicio de la implementación de este trabajo, se dio paso al cumplimiento de los principales objetivos de dicha tarea, entre los cuales destaca el rediseño de la interfaz que ofrece el ZSC para lograr una mejor compatibilidad con la mayor cantidad de dispositivos móviles posible. Para lograr este objetivo fue necesario el análisis y posterior modificación de algunos ficheros del servidor de correos Zimbra.

Primeramente se estudiaron los archivos incluidos dentro de la siguiente dirección:

*/opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/m*

Un archivo cuya adaptación fue necesaria es él:

*/opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/m/zmain*

Este archivo se encarga de la configuración de la página inicial o principal del cliente, la cual permite visualizar la bandeja de entrada del mismo. Una característica de esta página que fue necesario rediseñar es el tamaño de las imágenes incluidas en la misma, pues la imposibilidad de adaptarse a las características de los dispositivos del cliente suponía una distorsión de la interfaz. Como solución se decidió sustituir las imágenes existentes por nuevas imágenes editadas y reubicadas dentro de la carpeta *opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/img/mobile*. Además, fueron editados los estilos CSS aplicados a dichas imágenes y finalmente se incluyeron dentro de tablas XHTML que permiten una mejor organización y adaptación dentro de la pantalla del dispositivo.

Es necesario mencionar que también se decidió cambiar las opciones de fecha y cantidad de correos existentes en la bandeja de entrada, las cuales fueron incluidas también dentro de la misma tabla.



```
<table border="0" width="100%" align="center">
<tr style='width:100%'>

<td width="20%" height="22px">
<a accesskey="{requestScope.mailapp_accesskey}"
href="{aurl}?st={mailbox.prefs.groupMailBy}">

</a>
</td>

<td width="20%">
<c:if test="{mailbox.features.contacts}"><c:set var="active" value="{param.st eq 'contact' ||
param.st eq 'ab' ? '-active' : '}'"/>
<a accesskey="{requestScope.contactapp_accesskey}" href="{aurl}?st=contact">

</a></c:if>
</td>

<td width="20%">
<c:if test="{mailbox.features.calendar}"><c:set var="active" value="{param.st eq 'cal' || param.st
eq 'cals' ? '-active' : '}'"/>
<a accesskey="{requestScope.calapp_accesskey}" href="{aurl}?st=cal">

<fmt:formatDate value="{zm:getToday
(mailbox.prefs.timeZone).time}" timeZone="{mailbox.prefs.timeZone}" pattern="EEE|d|MMM"
var="today"/><c:set var="darr"
value="{fn:split(today, '|')}" />
</a></c:if>
</td>

<td width="20%">
```

```
<c:if test="{mailbox.features.briefcases || mailbox.features.notebook}">
<c:if test="{not empty param.l_view}"><c:set var="l_view"
value="{fn:escapeXml(param.l_view)}" scope="session"/></c:if>
<c:if test="{empty sessionScope.l_view}"><c:set var="l_view" value="list"
scope="session"/></c:if>
<c:set var="active" value="{param.st eq 'tasks' || param.st eq 'task' || param.st eq 'briefcase' ||
param.st eq 'wiki' || param.st eq 'briefcases' || param.st eq 'notebooks' ? '-active' : ''}/>
<a accesskey="{requestScope.fldrapp_accesskey}"
href="{aurl}?st={mailbox.features.briefcases ? 'briefcases' : 'notebooks'}">

</a></c:if>
</td>

<td width="20%">
<a accesskey="{requestScope.search_accesskey}" href="{zm:replaceAll
(currentUrl,'search=[01]', '')}&${spara}" onclick='return toggleElem("searchbar",this);'>

</a>
</td>

</tr>

</table>
```

Figura 10: Código modificado en el archivo `opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/img/mobile/zmain`.

Otro archivo que se modificó fue:

`opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/WEB-INF/tags/mobile/moConListView.tag`

Este archivo se encarga principalmente de la visualización de los correos existentes en la bandeja de entrada. La programación incluida en el mismo también suponía un problema a la hora de mostrar las listas de correos o conversaciones existentes. Por tanto, se decidió retirar la imagen que incluía por

defecto en cada mensaje así como la parte del mensaje que este incorporaba, permitiendo solamente visualizar el remitente y el asunto del correo almacenado. Además, se modificó la posición del checkbox para lograr un diseño mejor organizado.

```
<table width="100%">
<c:if test="{chit.messageCount ge 2}">
<c:set value="Conv" var="class"/>
</c:if>
<c:if test="{chit.messageCount lt 2}">
<c:set value="Msg{chit.isUnread ? ' ': 'Gray'}" var="class"/>
</c:if>

<tr><td><div align="left">
<span class="td f">
<c:set value=",{chit.id}," var="stringToCheck"/>
<input class="chk" type="checkbox" ${requestScope.selected ne 'none' &&
(fn:contains(requestScope._selectedCids,stringToCheck) || requestScope.selected eq 'all') ?
'checked="checked" : ""} name="cid" value="{chit.id}"/>
</span>

<span class="td m" onclick='return zClickLink("a{chit.id}");'>
</div></td></tr>

<tr><td>
<div class="from-span">
<c:set var="dispRec" value="{chit.displayRecipients}"/>
<c:set var="_f" value="{empty dispRec ? unknownRecipient : dispRec}"/>
<c:if test="{fn:length(_f) > 20}"><c:set var="_f" value="{fn:substring(_f, 0, 20)}..."/></c:if>
<a class="zo_m_list_from" id="a{chit.id}"
href="{fn:escapeXml(convUrl)}">{fn:escapeXml(_f)}</a></div>
<div class="sub-span">
<c:set var="_f" value="{empty chit.subject ? unknownSubject : chit.subject}"/>
```

```
<c:if test="{fn:length(_f) > 20}"><c:set var="_f" value="{fn:substring(_f, 0, 20)}..."/></c:if>
    {fn:escapeXml(_f)} </div>
</span>
<span class="td l">
<fmt:formatDate timeZone="{mailbox.prefs.timeZone}" var="on_dt" pattern="yyyyMMdd"
value="{chit.date}"/>
<a <c:if
test="{mailbox.features.calendar}">href='{context_url}?st=cal&view=month&date={o
n_dt}'</c:if>>
    {fn:escapeXml(zm:displayMsgDate(pageContext, chit.date))}
</a><br/>
<c:if test="{chit.isFlagged}">
<span class="SmllcnHldr Flag">&nbsp;</span>
</c:if>
<c:if test="{chit.hasTags}">
<mo:miniTagImage
    ids="{hit.conversationHit.tagIds}"/>
</c:if>
<c:if test="{chit.hasAttachment}">
<span class="SmllcnHldr Attachment">&nbsp;</span>
</c:if>
<c:if test="{chit.messageCount gt 1}"><span class="small-gray-
text">({chit.messageCount})</span></c:if>
</span>
</td></tr></table>
```

Figura 11: Código modificado del archivo `opt/zimbra/jetty-6.1.22.z6/weapps/zimbra/WEB-INF/tags/mobile/moContListView.tag`.

Cabe destacar que para algunas clases donde era común el uso de hojas de estilo en cascadas (CSS), fueron reeditadas dichas CSS, con el propósito de adecuar sus estilos al nuevo diseño en cuestión. Los principales ficheros CSS que fueron re implementados son:

/opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/css/iphone.css

/opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/css/iphone-cal.css

/opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/css/login.css

```
#appbar {
  background-color: #EEEEEE;
  height: 48px;
  padding: 0px;
  margin: 0px;
  white-space: nowrap;
  color: white;
  border-bottom: 0px solid gray;
  background-position: 0px -1px;
}
span.nstd {padding: 0px;margin: 0px;display: none;}
#appbar a {
  width: 55px;
  height: 48px;
  margin: 0px;
  padding: 0px;
  text-align: center;
  font-size: small;
  color: Cornsilk;
  vertical-align: bottom;
  display: inline-block;
}
.aptoday {
  color: #888888;
  display: inline !important;
```

```
font-size:smaller;
font-weight:bold;
}
.aptoday div {
margin:0px;
padding:1px;
text-align:center;
overflow:hidden;
white-space:nowrap;
text-overflow:ellipsis;
}
.aptoday div._day {
font-weight:bold;
padding-top:34px;
font-size: 10px;
left:-3px;
position:relative;
}
.aptoday div._date {
font-size:15px;
left:-2px;
position:relative;
top:-36px;
}
.aptoday div._month {
display:none;
}

.apunread {
text-align:center;
```

```
color:#888888;
display:inline !important;
font-size:10px;
font-weight:bold;
left:0;
position:relative;
top:34px;
white-space:normal;
width:55px;
text-overflow: ellipsis;
overflow: hidden;
}
.appTab,.appTab-active{
  background: url("/img/mobile/iphone-ext/AppsNav.png") no-repeat top left;
}
a.cal-active { background-position: 50% -1px !important; }
a.cal { background-position: 50% -50px !important; }
a.contact-active { background-position: 50% -99px !important; }
a.contact { background-position: 50% -148px !important; }
a.docs-active { background-position: 50% -197px !important; }
a.docs { background-position: 50% -246px !important; }
a.mail-active { background-position: 50% -295px !important; }
a.mail { background-position: 50% -344px !important; }
a.search-active { background-position: 50% -393px !important; }
a.search { background-position: 50% -442px !important; }
#searchbar,#saveseachbar {
  text-align: center;
  vertical-align: middle;
  padding:0px;
  padding-top: 2px;
```

```
padding-bottom: 2px;
margin: 0px;
background: #EEE;
border-bottom: 1px solid silver;
}
```

Figura 12: Fragmento de código del archivo re-implementado `/opt/zimbra/jetty-6.1.22.z6/wepapps/zimbra/css/iphone.css`.

```
BODY {
    background-color:#E4F6FE;
    @FullSize@
    margin: 0px;
}

FORM{
    margin: 0;
    padding: 0;
}

.ZSplashScreen {
    background-color:#E4F6FE;
    width:100%;
    height:100%;
    top:0;
    left:0;
    position:absolute;
}

#ZLoginPanel1 {
    position:relative;
    width:500;
    overflow:visible;
    background-color:#15428B;
    border-style:solid;
```



```
border-width:1px;
padding:6px;
}
#ZLoginBodyContainer1 {
background-color: #E4F6FE;
border-style:solid;
border-width:1px;
margin:0;
padding-top:20px;
}
```

Figura 13: Fragmento de código del archivo reemplazado `/opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/css/login.css`

Otro archivo imprescindible que fue necesario modificar es el:

`opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/public/login.jsp`

Este archivo se muestra inicialmente al cliente cuando este realiza una petición al servidor. En el mismo fue necesario incluir algunos cambios importantes tanto en la interfaz como en sus funcionalidades. Principalmente, se identifica el useragent, el cual no es más que una cadena de caracteres incluida en la petición proveniente del cliente, mediante la cual es posible determinar el navegador cliente que se encuentra realizando la petición al servidor.

```
<%
WURFLHolder wurflHolder = (WURFLHolder)
getServletContext().getAttribute("net.sourceforge.wurfl.core.WURFLHolder");
WURFLManager wurfl = wurflHolder.getWURFLManager();
Device device = wurfl.getDeviceForRequest(request);
if(((String)device.getCapability("is_wireless_device")).equals("true")) { %>
<c:set var="client" value="mobile"/>
<c:set var="useMobile" value="mobile"/>
<% }
else { %>
```

```
<c:set var="client" value="advanced"/>
<% } %>
```

Figura 14: Fragmento de código del archivo `opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/public/login.jsp`.

Con vistas a utilizar las clases que brinda la nueva API de Java para el desarrollo con WURFL es necesario importar las clases listadas a continuación:

```
<%@ page import="net.sourceforge.wurfl.core.Device" %>
<%@ page import="net.sourceforge.wurfl.core.WURFLManager" %>
<%@ page import="net.sourceforge.wurfl.core.WURFLHolder" %>
```

Figura 15: Código que permite el uso de la nueva API de Java.

Independientemente de la importancia que conlleva incluir este API, se hace necesario la inclusión de las clases que conforman la misma. Es importante destacar, que sin estos archivos almacenados en el directorio `opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/WEB-INF/lib` sería imposible la correcta interacción con la librería WURFL.

El listado de ficheros incluidos en dicha dirección es:

- aopalliance-1.0.jar
- backport-util-concurrent-3.0.jar
- commons-collections-3.2.jar
- commons-lang-2.3.jar
- commons-logging-1.1.jar
- ehcache-1.3.0.jar
- jsp-api-2.0.jar
- jsr107cache-1.0.jar
- servlet-api-2.4.jar
- spring-aop-2.5.6.jar
- spring-beans-2.5.6.jar
- spring-context-2.5.6.jar
- spring-context-support-2.5.6.jar

- spring-core-2.5.6.jar
- spring-web-2.5.6.jar
- standard.jar
- wurfl-1.0.1-rc3.jar

Además, fue necesario incluir los siguientes archivos:

- *opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/WEB-INF/web-spring.xml*
- *opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/WEB-INF/ehcache.xml*
- *opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/WEB-INF/web\_browsers\_patch.xml*
- *opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/WEB-INF/wurfl.zip*
- *opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/WEB-INF/jmx-ctx.xml*
- *opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/WEB-INF/web-alone.xml*
- *opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/WEB-INF/wurfl-default-ctx.xml*

Sin embargo, aún se hace necesario mapear el WURFL para poder hacer uso del mismo. Es por ello que se incluye el siguiente código en el archivo:

*opt/zimbra/jetty/etc/zimbra.web.xml.in*

```
<context-param>
<param-name>contextConfigLocation</param-name>
<param-value>/WEB-INF/wurfl-default-ctx.xml</param-value>
</context-param>

<!--<filter>
<filter-name>zimbraContextFilter</filter-name>
<filter-class>net.sourceforge.wurfl.wng.WNGContextFilter</filter-class>
</filter-->

<listener>
```

```
<listener-class>
    org.springframework.web.context.ContextLoaderListener
</listener-class>
</listener>
```

Figura 16: Código incluido en el archivo `opt/zimbra/jetty/etc/zimbra.web.xml.in`.

Una vez incluida la API anteriormente mencionada, se hace posible la identificación del navegador cliente y por consiguiente la identificación de la interfaz correspondiente a dicha petición.

```
<%
    WURFLHolder wurflHolder = (WURFLHolder)
getServletContext().getAttribute("net.sourceforge.wurfl.core.WURFLHolder");
    WURFLManager wurfl = wurflHolder.getWURFLManager();
    Device device = wurfl.getDeviceForRequest(request);
if(((String)device.getCapability("is_wireless_device")).equals("true") ) { %>
<c:set var="client" value="mobile"/>
<c:set var="useMobile" value="mobile"/>
<% }
else { %>
<c:set var="client" value="advanced"/>
<% } %>
```

Figura 17: Código de Java incluido en el archivo `opt/zimbra/jetty-6.1.22.z6/wepapps/zimbra/public/login.jsp`.

Con la identificación del cliente, queda en desuso la funcionalidad de brindarle al usuario la posibilidad de escoger la interfaz desde el cual desea acceder al sistema. De esta forma, se redirecciona automáticamente, siendo invisible completamente para el usuario.

```
<tr>
<td align="center">
<div class="ZLoginSeparator" style="margin-top:0px"></div>
<c:choose>
<c:when test="{!smallScreen}">
```



```
}  
  
function onLoad() {  
  document.loginForm.username.focus();  
  clientChange("${zm:cook(client)}");  
}  
document.write("<a href='#' onclick='showWhatsThis()'  
id='ZLoginWhatsThisAnchor'><fmt:message key='whatsThis'/><"+"/a>");  
</script>  
</td>  
</tr>  
  
<tr>  
<td align="center">  
<div id="ZLoginWhatsThis" class="ZLoginInfoMessage" style="display:none;text-  
align:left;width:90%;"><fmt:message key="clientWhatsThisMessage"/></div>  
<div id="ZLoginUnsupported" class="ZLoginInfoMessage" style="display:none"><fmt:message  
key="clientUnsupported"/></div>
```

Figura 18: Funcionalidad eliminada del archivo `opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/public/login.jsp`.

La gestión de los contactos es otra funcionalidad que necesitó rediseñarse. Principalmente por la existencia de imágenes demasiado grandes y componentes incluidos que contribuían al distorsionamiento de la interfaz en cuestión. Los cambios se realizaron específicamente en el archivo:

`opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/WEB-INF/tags/mobile/moContactListView.tag`

En este archivo se puede acceder al listado de contactos del cliente. Sin embargo, fue necesario cambiar una imagen muy grande para permitir la correcta organización de la página, auxiliándose además de la inclusión de una tabla que contribuyó con este objetivo. Esta nueva imagen fue incluida en la siguiente dirección:

`opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/img/large`

```

<table width="100%">
<c:forEach items="{context.searchResult.hits}" var="hit" varStatus="status">
<c:set var="chit" value="{hit.contactHit}"/>
<zm:currentResultUrl var="contactUrl" value="{context_url}" action="view" id="{chit.id}"
        index="{status.index}" context="{context}"/>
<c:if test="{context.isGALSearch}">
<c:url value="{contactUrl}" var="contactUrl">
<c:param name="email" value="{chit.email}"/>
</c:url>
</c:if>
<div class="tr list-row" id="cn${chit.id}">
<c:set value=",${hit.id}," var="stringToCheck"/>
<c:set var="class" value="Contact${chit.isGroup ? 'Group' : '}'"/>
<tr>
<td><div align="left">
<span class="td f">
<c:if test="{!context.isGALSearch}">
<input class="chk" type="checkbox" ${requestScope.select ne 'none' &&
(fn:contains(requestScope._selectedIds,stringToCheck) || requestScope.select eq 'all') ?
'checked="checked"' : ''}
        name="id" value="{chit.id}"/></c:if>
<%--<span class="SmllcnHldr ${class}">&nbsp;</span> yo esto lo quite porque no voy a cargar
la imagen--%>
</span>
</div></td></tr>
<tr>
<td width="100%">
<div align="left">
<span class="td m" onclick='return zClickLink("a${chit.id}");'>
<a id="a${chit.id}" href="{contactUrl}">

```

```
<div>
<strong>${zm:truncate(fn:escapeXml(empty chit.fileAsStr ? (context.isGALSearch ? chit.fullName
: '<None>') : chit.fileAsStr),23, true)}</strong>
</div>
</a>
<div class="Email">
<c:set var="nmail" value="st=newmail"/>
<c:url var="murl" value="?${nmail}&to=${chit.email}"/>
<a href="${fn:escapeXml(murl)}">${fn:escapeXml(chit.email)}</a>
</div>
</span>

<span class="td l">
<c:if test="${chit.isFlagged}">
<span class="SmllcnHdr Flag">&nbsp;</span>
</c:if>
<c:if test="${chit.hasTags}">
<mo:miniTagImage ids="${chit.tagIds}"/>
</c:if>
</span>
</div>
</c:forEach>
</div>
</td>
</tr>
</table>
```

Figura 19: Código modificado en el archivo `opt/zimbra/jetty-6.1.22.z6/wepapps/zimbra/WEB-INF/tags/mobile/moContactListView.tag`.

Además, fue necesario el cambio de los siguientes archivos con la intención de modificar los componentes que incluían errores a la hora de incluir o modificar un contacto.



*opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/WEB-INF/tags/mobile/ moContactListView.tag*

En este archivo, entre otras cosas, fue necesario incluir tablas que permitieran centrar la lista de contactos a visualizar. Además, se decidió modificar una imagen que debido a sus dimensiones demasiado grandes contribuía a la distorsión de la información durante su visualización.

Cambios similares se realizaron en el archivo:

*opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/WEB-INF/tags/mobile/ moContactEdit.tag*

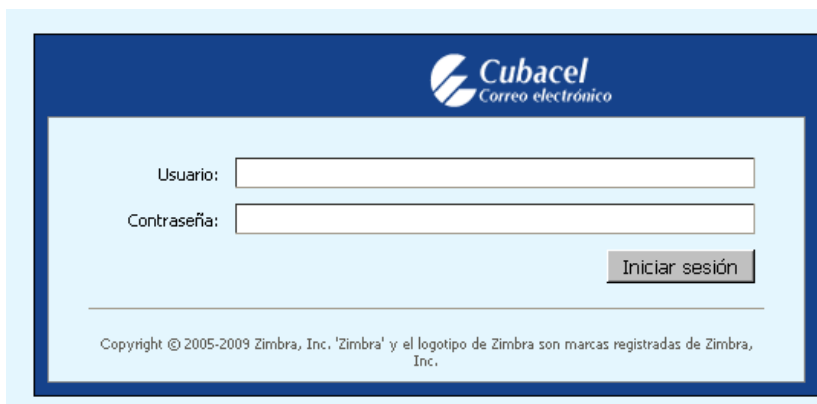
Este es el encargado de editar los campos de un contacto determinado, permitiéndole al usuario modificar la información del mismo.

Al fichero moContactEditField.tag ubicado en:

*opt/zimbra/jetty-6.1.22.z6/webapps/zimbra/WEB-INF/tags/mobile/ moContactEditField.tag*

Fue necesario especificarle un tamaño determinado al TextArea mediante el cual el usuario introduce la información que modifica los datos del contacto en cuestión.

Una vez realizadas las modificaciones pertinentes se obtuvieron como resultado diferentes interfaces acorde con las peticiones del cliente. A continuación se muestra una de ellas, el resto se encuentran en el **anexo 1**:



**Figura 20: Interfaz de logeo una vez realizados los cambios**

Se decidió seguir ciertas pautas a la hora de modificar el código del programa, con la intención de seguir una lógica que permita a los programadores adaptar el sistema a los objetivos propuestos. Estas son:

- Usar un lenguaje claro y simple.
- El contenido que solicita el usuario debe estar limitado a las restricciones del dispositivo.
- No usar imágenes que no pueden ser mostradas en el dispositivo.
- Evitar las imágenes grandes o imágenes con gran resolución.
- Lograr un buen contraste entre el color de fondo y el color del primer plano.

### 4.3 Conclusiones del capítulo

En este capítulo se realizó una descripción de los cambios realizados en el cliente de correos Zimbra y se explicó cómo se llevó a cabo la utilización de WURFL con la esperanza de que sirva de guía a todo aquel que decida realizar su propia personalización.

### CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD

#### 5.1 Introducción

La rentabilidad, estimación de esfuerzo y costo de un proyecto de software son sin lugar a dudas una de las tareas de mayor importancia para la producción de software ya que cada vez es mayor la necesidad de obtener datos que permitan evaluar, predecir y mejorar la calidad del producto. En el presente capítulo se realizará un estudio de factibilidad para la realización del sistema propuesto mediante una estimación de tamaño, esfuerzo y planificación necesaria para llevar a cabo el mismo.

#### 5.2 Modelo matemático COCOMO II

COCOMO, desarrollado por Barry Boehm, es uno de los modelos de estimación de costos mejor documentado. Este modelo permite determinar el esfuerzo y el tiempo requerido en un proyecto de software partiendo de una medida del tamaño del mismo expresada en el número de líneas de código que se estimen generar para la creación del producto.

Consiste en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin Ajustar o la cantidad de líneas de código (*SLOC, Source Lines of Code*) estimados para un proyecto. Estas ecuaciones se encuentran ponderadas por ciertos factores de costo (*cost drivers*) que influyen en el esfuerzo requerido para el desarrollo del software.

Está compuesto por tres modelos que se adaptan tanto a las necesidades de los diferentes sectores descritos, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información. Estos modelos son: Composición de aplicación, diseño temprano y post-arquitectura.

El estudio de factibilidad de este sistema se realizó mediante el modelo de diseño temprano que se utiliza en las primeras etapas del desarrollo, en las cuales se evalúan las alternativas de hardware y software de un proyecto (18).

### 5.3 Características del proyecto

#### 5.3.1 Entradas externas (EI)

Nombre de la entrada externa.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media. Compleja)
<b>Total</b>		0	

Tabla 9 Entradas externas.

#### 5.3.2 Salidas externas (EO)

Nombre de la salida externa.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media. Compleja)
Interfaz correspondiente.	1	8	Simple
<b>Total</b>		8	

Tabla 10 Salidas externas.

#### 5.3.3 Ficheros internos (LIF)

Nombre del fichero interno.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media. Compleja)
Librería WURFL.	1	6	Simple
<b>Total</b>		6	

Tabla 11 Ficheros internos.

#### 5.3.4 Interfaces externas (EIF)

Nombre de la interfaz externa.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media. Compleja)
<b>Total</b>		0	

Tabla 12 Interfaces externas.

5.3.5 Consultas externas

Nombre de la petición.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media. Compleja)
Reconocer el navegador cliente a partir de la petición.	1	1	Simple
<b>Total</b>		1	

Tabla 13 Consultas externas.

5.3.6 Puntos de función desajustados

Elementos.	Simple.		Medio.		Complejo.		Subtotal.
	No.	Peso.	No.	Peso.	No.	Peso.	
Entradas externas.	0	3	0	4	0	6	0
Salidas externas.	8	4	0	5	0	7	32
Ficheros internos.	6	7	0	10	0	15	42
Interfaces externas.	0	5	0	7	0	10	0
Consultas externas.	1	3	0	4	0	6	3
<b>Total (UFP):</b>							<b>77</b>

Tabla 14 Puntos de función desajustados.

### 5.4 Cálculos de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo

#### 5.4.1 Cálculo de instrucciones fuentes

Una vez obtenida la cantidad total de puntos de función desajustados pertenecientes al proyecto, se procede a calcular la cantidad de instrucciones fuentes del mismo, para lo cual se utiliza la ecuación que se muestra seguidamente:

$$SLOC = UFP \times ratio$$

Donde:

**SLOC:** Cantidad de instrucciones fuente. (4081)

**UFP:** Puntos de función desajustados. (77)

**Ratio:** Conversión de puntos de función desajustados a líneas de código para el lenguaje Java. (53)

Partiendo de la ecuación anterior se obtiene el siguiente resultado:

$$SLOC = 77 \times 53 = 4081$$

Las características del sistema y valores obtenidos anteriormente han sido plasmados en la siguiente tabla:

Características	Valor
Puntos de función desajustados.	77
Lenguaje (Java).	53
Instrucciones fuentes por puntos de función.	4081
Instrucciones fuentes.	4.08 ksloc

Tabla 15 Características.

### 5.4.2 Cálculo de esfuerzo nominal

$$PM_{nominal} = A \times (Size)^B$$

$$PM_{nominal} = 2.94 \times 4.08^{1.02} = 4, 19 \text{ meses/hombres}$$

Donde:

**PMnominal:** Esfuerzo nominal requerido en meses – hombres.

**Size:** Tamaño estimado del software en Puntos de Función sin Ajustar (KSLOC).

**A:** Constante que se utiliza para capturar los efectos multiplicativos en el esfuerzo requerido de acuerdo al crecimiento de tamaño del software. (2.94)

**B:** Constante denominada Factor escalar y su valor está dado por la resultante de los aspectos positivos sobre los negativos que presenta el proyecto.

$$B = 0.91 + 0.01 \times \sum (W_i)$$

$$B = 0.91 + 0.01 \times 10.71 = 0.91 + 0.1071 = 1.0171$$

Donde:

**Wi:** Variables escalares que indican las características que el proyecto presenta en lo que a su complejidad y entorno de desarrollo se refiere. La siguiente tabla muestra los valores asignados a cada una de estas variables:

Nombre.	Valor.	Justificación.
Precedentes (PREC).	1.24	Existen varios sistemas similares a nivel mundial pero no a nivel nacional.
Flexibilidad de desarrollo (FLEX).	1.01	Cuenta con una flexibilidad de desarrollo muy alta.

Cohesión del equipo (TEAM).	1.10	El equipo de desarrollo presenta una alta cohesión.
Solución de riesgos (RESL).	4.24	No se identificaron grandes riesgos.
Madurez del proceso (PMAT).	3.12	Se cuenta con la experiencia necesaria como para que el software cumpla con las funcionalidades requeridas.
<b>Total(SF)</b>	<b>10.71</b>	

Tabla 16 Factores de escala.

### 5.4.3 Ajuste del esfuerzo nominal

El esfuerzo calculado anteriormente es un valor nominal y debe ser ajustado en base a las características del proyecto. Para esto se tiene un conjunto de Multiplicadores de Esfuerzo (ME<sub>i</sub>), que representan las características del proyecto y expresan su impacto en el desarrollo total del producto de software.

Nombre.	Valor.	Justificación.
Confiabilidad y complejidad del producto (RCPX)	1.74	La complejidad del sistema es alta.
Reutilización requerida(RUSE)	1.00	El nivel de reutilización del producto es alto.
Dificultad de la plataforma(PDIF)	1.00	Uso de memoria y almacenamiento normal, plataforma estable.
Experiencia del personal(PREX)	1.12	Baja experiencia del personal en la utilización del lenguaje así como de la plataforma.
Aptitud del	1.00	La capacidad del personal es alta.



personal(PERS)		
Facilidades(FCIL)	0.82	Se utilizaron entornos de desarrollo integrados, herramientas de modelación y automatización que facilitan el trabajo.
Planificación requerida(SCED)	1.00	Se empleó el tiempo planificado para el desarrollo del sistema.
<b>Total(EM)</b>	<b>7.68</b>	

Tabla 17 Multiplicadores de esfuerzo.

$$PM_{ajustado} = PM_{nominal} \times \prod ME_i$$

$$PM_{ajustado} = 4.19 \times 7.68 = 32.17 \text{ meses/hombre}$$

#### 5.4.3.1 Calculo del tiempo de desarrollo del software

El tiempo requerido para el desarrollo del proyecto está dado por la siguiente ecuación:

$$TDEV = C \times (PM_{ajustado})^F$$

$$TDEV = 3.64 \times 32.17^{0.26} = 3.64 \times 2.46 = 8.9544$$

Donde:

**C:** Constante con valor 3.64.

$$F = D + 0.2 \times 0.01 \times \sum SF$$

$$F = 0.24 + 0.2 \times 0.01 \times 10.71 = 0.26142$$

Donde:

**D:** constante cuyo valor es 0.24.

**SF:** valor de los factores de escala.

### 5.4.3.2 Cálculo del costo total del proyecto

Para el cálculo del costo total correspondiente al proyecto en cuestión, COCOMO II propone la siguiente ecuación:

$$C = CHM \times PM$$

$$C = 446 \times 4.19 = 1868.74$$

Donde:

**C:** costo total.

**CHM:** costo teniendo en cuenta salario de todos los obreros, el cual se calcula por la siguiente ecuación.

$$CHM = CH \times sal$$

$$CHM = 4.4679 \times 100 = 446$$

Donde:

**Sal:** salario medio por cada trabajador.

**CH:** cantidad de personas destinadas al proyecto, lo que es calculado a través de la ecuación:

$$CH = PM / TDEV$$

$$CH = 4.19 / 8.9544 = 4.4679$$

### 5.4.4 Resultados

En la siguiente tabla se muestran los resultados obtenidos luego de haber efectuado todos los cálculos para determinar el costo y el esfuerzo requeridos por el proyecto.

Cálculo de:	Valor
Esfuerzo.	4.19 hombres/mes

Tiempo de desarrollo.	6 meses
Cantidad de hombres.	2 hombres
Salario medio.	100 pesos
Costo.	1868.74 pesos

**Tabla 18 Resultados.**

### **5.5 Análisis de costo**

El desarrollo de un producto siempre presenta un costo de producción, el cual se justifica en base a los beneficios reportados por el mismo. El sistema presentado no precisa de grandes gastos, debido en gran medida a la utilización de plataformas, tecnologías y herramientas libres que no requieren del pago de licencia por su uso; su costo está fundamentalmente determinado por el salario devengado por los desarrolladores que toman parte en el sistema.

### **5.6 Conclusiones del capítulo**

En este capítulo se realizó un análisis de factibilidad de la solución propuesta, obteniendo los estimados de tamaño, esfuerzo y planificación. Se llegó a la conclusión de que el desarrollo del sistema propuesto es viable, basándonos en la comparación de los costos de producción con los beneficios que reportará el sistema.

### **CONCLUSIONES GENERALES**

Con el desarrollo de este trabajo se ha obtenido la personalización del servidor de correos Zimbra para teléfonos móviles haciendo más fácil el acceso a este servicio. Se puede decir que se ha cumplido satisfactoriamente el objetivo trazado para este trabajo ya que el producto resultante está provisto de un ambiente cómodo, fácil de entender, que cumple con las especificaciones del cliente. Además, el estudio de la factibilidad determinó que el desarrollo de la aplicación es realmente factible para su puesta en funcionamiento. Por otra parte, se aplicó la metodología XP para guiar el proceso de desarrollo del software permitiendo la modelación del sistema en el tiempo estimado, posibilitando así las entregas en tiempo al cliente.

### RECOMENDACIONES

- Se recomienda hacer uso de WNG (*Wall Next Generation*) que se apoya en WURFL con la intención de lograr una portabilidad aún mayor del sistema para los disímiles modelos de dispositivos móviles

**BIBLIOGRAFÍA**

1. evdoplus.blogspot.com. *Las generaciones de la telefonía inalámbrica*. [Online] [Cited: febrero 12, 2010.] [http://evdoplus.blogspot.com/2007\\_04\\_01\\_archive.html](http://evdoplus.blogspot.com/2007_04_01_archive.html).
2. **Nava, Beatriz Ordoñez and Cuevas, Abraham Mora**. *DISEÑO DE UNA INTERFAZ GRÁFICA PARA EL MODELADO DE REDES INALÁMBRICAS DE BANDA ANCHA* disponible en [[http://profesores.fi-b.unam.mx/victor/LTesis\\_Beatriz\\_Abraham.pdf](http://profesores.fi-b.unam.mx/victor/LTesis_Beatriz_Abraham.pdf)]. México : s.n., 2006.
3. **Pérez, Pedro Navarro**. *W A P: Wireless application protocol*.
4. *Prácticas de documentación: WAP*. 1999/2000.
5. **Baeza, Oscar J**. Genbeta. *Genbeta*. [En línea] 13 de septiembre de 2005. [Citado el: 12 de febrero de 2010.] <http://www.genbeta.com/web/zimbra-correo-y-calendario-libres-al-mas-puro-estilo-gmail>.
6. **Pressman, Roger S**. *Ingeniería del Software un enfoque práctico*.
7. **José H. Canós, Patricio Letelier, Ma. Carmen Penadés**. Metodologías ágiles en el desarrollo del software. [En línea] [Citado el: 20 de febrero de 2010.] <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>.
8. **Martin, Robert C**. *Extreme Programming in Practice*. Los Angeles : Addison-Wesley, 2001. 0201709376.
9. w3c. *w3c*. [En línea] 7 de febrero de 2008. [Citado el: 4 de marzo de 2010.] <http://www.w3c.es/Divulgacion/Guiasbreves/XHTML>.
10. WebExperto. *WebExperto*. [En línea] 8 de julio de 2006. [Citado el: 4 de marzo de 2010.] <http://www.webexperto.com/downloads/262/openwave-phone-simulator-7/>.
11. **Passani, Luca**. wurfl.sourceforge.net. *WURFL Devices and WURFL Capabilities*. [En línea] Agosto de 2007. [Citado el: 17 de febrero de 2010.] [http://wurfl.sourceforge.net/help\\_doc.php](http://wurfl.sourceforge.net/help_doc.php).
12. Red Hat Enterprise Linux 4. *Manual de referencia*. [En línea] 2005. [Citado el: 13 de abril de 2010.] <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/>.
13. **Pérez, Javier Eguíluz**. *Introducción a CSS*. 2009.
14. **Beck, Kent**. *Extreme Programming Explained*. s.l. : Addison-Wesley, 1999. 0201616416.
15. **Fowler, Martin**. martinowler.com. [En línea] mayo de 2004. [Citado el: 15 de marzo de 2010.] <http://martinowler.com/articles/designDead.html>.

16. **Crispin, Lisa y House, Tip.** *Testing Extreme Programming*. s.l. : Addison-Wesley, 2002.
17. **Wells, Don.** Acceptance Tests. *extremeprogramming.org*. [En línea] 1999. [Citado el: 16 de marzo de 2010.] <http://www.extremeprogramming.org/rules/functionaltests.html>.
18. **Gómez, Adriana, y otros.** - *COCOMO - UN MODELO DE ESTIMACION DE PROYECTOS DE SOFTWARE*.
19. **Passani, Luca.** wurfl.sourceforge.net. *WURFL Java API*. [En línea] [Citado el: 17 de febrero de 2010.] <http://wurfl.sourceforge.net/java/api.php>.
20. —. wurfl.sourceforge.net. *Introducing WNG: WALL Next-Generation*. [En línea] [Citado el: 17 de febrero de 2010.] <http://wurfl.sourceforge.net/wng/tutorial.php>.
21. **Guervos, Juan Julian Merelos.** geneura.ugr.es. *Programando con JSPs*. [En línea] 22 de octubre de 2004. [Citado el: 20 de febrero de 2010.] <http://geneura.ugr.es/~jmerelo/JSP/>.
22. **Passani, Luca.** wurfl.sourceforge.net. *New WURFL Java API*. [En línea] [Citado el: 19 de febrero de 2010.] <http://wurfl.sourceforge.net/njava/>.
23. **Cardona, Alberto.** [En línea] 6 de mayo de 2001. [Citado el: 2 de marzo de 2010.] [http://www.programacion.com/articulo/servlets\\_y\\_jsp\\_82/11](http://www.programacion.com/articulo/servlets_y_jsp_82/11).
24. **Beck, K. y M.Fowler.** *Planeando en programación extrema*. s.l. : Addison-Wesley, 2000.
25. **Wake, William C.** *Extreme Programming Explored*. s.l. : Addison-Wesley, 1999. 0201733978.
26. **Letelier, P.** *Metodologías ágiles para el desarrollo de software*. 2004.
27. Center for Systems and Software Engineering. *COCOMO II*. [En línea] [Citado el: 15 de abril de 2010.] [http://csse.usc.edu/csse/research/COCOMOII/cocomo\\_main.html](http://csse.usc.edu/csse/research/COCOMOII/cocomo_main.html).
28. Tribunal de Tasaciones de la nación. *ttn.gov.ar*. [En línea] 6 de diciembre de 2007. [Citado el: 20 de abril de 2010.] [http://www.ttn.gov.ar/normas/norma\\_24\\_0.htm](http://www.ttn.gov.ar/normas/norma_24_0.htm).
29. **Ramos, Anay Carrillo.** *Herramienta Multimedia de apoyo a la Enseñanza de la Metodología RUP de Ingeniería del Software*.