

UNIVERSIDAD INTERNACIONAL SEK

FACULTAD DE SISTEMAS EN INFORMACION Y
TELECOMUNICACIONES

Trabajo de fin de carrera titulado:

SISTEMA PROCESADOR DE MENSAJES DE CORREO
ELECTRÓNICO: INVESTIGACIÓN Y CONSTRUCCIÓN
DE UNA HERRAMIENTA DE CORREO ELECTRÓNICO
UTILIZANDO API DE LINUX

Realizado por:

MARCO SANTIAGO MOLINA AGUILAR

Como requisito para la obtención del título de
INGENIERO EN SISTEMAS

QUITO, ABRIL DE 2010

DECLARACION JURAMENTADA

Yo Marco Santiago Molina Aguilar, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración mis derechos de propiedad intelectual correspondientes a este trabajo a la UNIVERSIDAD INTERNACIONAL SEK, según lo establecido por la ley de la Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

.....

Marco Santiago Molina Aguilar

AGRADECIMIENTOS

Quiero agradecer primeramente a Dios, ya que me ha dado la fuerza necesaria para realizar este trabajo, a mis profesores por la formación que me han brindado, y a mis padres que me dieron aliento hasta el final.

La experiencia adquirida con estas personas ha sido inolvidable y estoy seguro de que todo será aplicado en mi vida profesional. Por último quiero agradecer a mi amigo, Ing. Wilmer López por ser mi guía en el sistema operativo Linux y en mis estudios universitarios.

RESUMEN EJECUTIVO

El presente trabajo titulado “Sistema procesador de mensajes de correo electrónico: Investigación y Construcción de una herramienta de correo electrónico utilizando API de Linux” investiga las herramientas que componen la interfaz de programa de aplicación (API) de Linux y la manera que se realiza el flujo de mensajes.

La aplicación desarrollada, OPTIMAIL, es un prototipo de cliente de correo electrónico que proporciona los servicios básicos a los usuarios, como el envío de mensajes y tipos de documentos más usadas en una empresa. En el desarrollo de la misma se tomaran en cuenta intercambio de mensajes locales y remotos, transmisión de texto e imágenes, la utilización de formas electrónicas especiales para envío y recepción de mensajes, la creación de cuentas PostOffice para que un usuario pueda ingresar a la aplicación. La programación de la interfaz fue realizada en PHP vinculando la capacidad del Sistema Operativo Linux.

La primera parte recoge de forma organizada el diseño conceptual de la investigación. La segunda parte detalla el estudio y definición de etapas de la aplicación OPTIMAIL así como el análisis de los componentes de la misma. La tercera parte está dedicada a la instalación y configuración del sistema operativo. La cuarta parte se enfoca en la metodología de desarrollo, casos de usos, así como diagramas de secuencias, clases y de estructura de datos. La parte final está dedicada a la instalación de Optimail, su utilización general, conclusiones, así como la recapitulación de las principales líneas de trabajo, una bibliografía citada y el listado de otras fuentes de información que fueron empleadas.

ABSTRACT

The present work titled System processor of electronic mail messages: Investigation and Construction of an electronic mail tool using API of Linux investigates tools that compose the interface of the application program (API) in Linux and the way that it carries out the flow of messages.

The developed application, OPTIMAIL, is an electronic mail client's prototype that provides the basic services to all users, as the shipment of messages and types of documents more used in a company. In the development of the same one we took in bill exchange of local and remote messages, text transmission and images, the use in special electronic ways for shipment and reception of messages, the creation of bills PostOffice so that an user can enter to the application. The programming of the interface was carried out in PHP linking the capacity of the operating system Linux.

The first part picks up in an organized way the conceptual design of the investigation. The second part details the study and definition of stages of OPTIMAIL as well as the analysis of the components of the same one. The third part is dedicated to the installation and configuration of the operating system. The fourth leave it focuses in the development methodology, cases of uses, as well as diagrams of sequences, classes and of structure of data. The final part is dedicated to the installation of Optimail, its general use, conclusions, as well as the recapitulation of the main work lines. Finally a mentioned bibliography and the listing of other sources of information that were used.

CONTENIDO

CAPITULO I. DISEÑO CONCEPTUAL DE LA INVESTIGACIÓN	17
1.1. Determinación del problema	17
1.2. Definición del tema	19
1.3. Justificación e importancia	19
1.4. Objetivos	20
1.4.1. Objetivo general	20
1.4.2. Objetivos específicos	20
1.5. Delimitación del tema	21
1.6. Limitación de la Investigación	22
1.7. Marco Teórico	22
1.8. Glosario de términos	34
1.9. Diseño de la Investigación	35
1.10. Temario Tentativo	36
CAPITULO II. ESTUDIO DE ALTERNATIVAS DE SOLUCIÓN	42
2.1. Especificación de requerimientos	42
2.1.1. DNS	42
2.1.1.1. Estructura de un nombre	42
2.1.1.2. Componentes DNS	43
2.1.1.3. Aplicaciones de DNS	43
2.1.1.4. Estructura Jerárquica	44
2.1.1.5. Proceso de resolución de nombres	45
2.1.1.6. Tipos de DNS	46
2.1.1.7. Tipos de servidores DNS	47
2.1.1.8. Récords de un DNS	47

2.1.2. Agentes	48
2.1.2.1. MTA	48
2.1.2.2. MUA	49
2.1.2.3. MDA	49
2.1.3. Funcionamiento de Correo	50
2.1.4. Protocolos	50
2.1.4.1. SMTP	51
2.1.4.2. POP3	51
2.1.4.3. IMAP	53
2.1.5. Formatos de archivos de correo	54
2.1.5.1. Mail Box (MBOX)	54
2.1.5.2. Mail Dir (MDIR)	55
2.1.6. Servidor de Correo	56
2.1.6.1. Sendmail	57
2.1.6.2. Postfix	58
2.2. Alternativas de solución	58
2.2.1 Selección de la solución	63
2.3 Diseño de Optimail	64
2.3.1 Esquema global del funcionamiento	65
2.3.2 Descripción general de los componentes del servicio	65
2.4 Análisis de portabilidad	66
2.5 Análisis de Escalabilidad	66
2.6 Definición de etapas de implementación	67
CAPITULO III. CONFIGURACIONES INICIALES	68
3.1 Instalación y configuración del sistema base para el diseño de Optimail y pruebas de concurrencia	68
3.1.1 Generalidades	68

3.1.2. Configuración.....	68
3.1.3. Configuraciones Post-Instalación.....	77
3.1.3.1. Comandos importantes.....	79
3.1.3.2. Configuración /etc/hosts.....	81
3.1.3.3. Instalar Software Adicional.....	81
3.1.3.4. Instalación de Quota.....	82
3.1.3.5. Instalación de servidor DNS (BIND9).....	83
3.1.3.6. Instalación de MYSQL (5.0).....	84
3.1.3.7. Instalación de Postfix con SMTP-AUTH y TLS.....	84
3.1.3.8. Maildir.....	90
3.1.3.9. Apache2 con PHP5.....	90
3.1.3.10. Habilitar PHP Globalmente.....	91
3.1.3.11 Instalación de módulos Perl.....	92
3.2 Seguridades.....	93
3.2.1 Intercambio de información.....	93
3.2.2. Instalación de software dañino involuntariamente.....	93
3.2.3. Protección ante accesos no autorizados.....	93
3.2.3.1. Fallos de seguridad en la utilización del software.....	94
3.2.4. Software libre.....	94
3.2.4.1. Ventajas del Software Libre en el mundo de la seguridad.....	94
3.2.4.2. Desventajas del software libre.....	95
3.4.4.3. Instalación de Mailscanner y Clamav.....	98
CAPITULO IV DESARROLLO.....	101
4.1 Metodología de desarrollo.....	101
4.1.1 Metodología OO-Method.....	101
4.2 Modelo conceptual.....	102
4.2.1. Estado del arte.....	103

4.2.2. Métodos Hipermediales.....	103
4.2.3 Métodos orientados a objetos que extienden UML.....	104
4.3. Método OOWS.....	104
4.3.1 Fases del Método OOWS.....	105
4.3.2 Requisitos del Método OOWS.....	106
4.3.2.1. Diagrama Funcional.....	106
4.3.2.2. Diagrama de Agentes.....	108
4.3.2.3 Diagrama de Asociación funcional.....	109
4.3.3. Modelado Conceptual.....	110
4.3.3.1. Modelo de Objetos.....	110
4.3.3.2 Modelo Navegacional.....	110
4.4. Tecnología de desarrollo.....	112
4.5. Características del sistema.....	114
4.6. Herramientas de Desarrollo.....	115
4.7. Caso de Uso.....	117
4.7.1. Correo (Principal).....	117
4.7.2 Acceso.....	118
4.7.3. Crear Cuentas.....	119
4.7.4. Administrar Correo.....	119
4.8. Diagramas de Secuencias.....	120
4.8.1. Crear Cuentas.....	120
4.8.2. Acceso al Sistema.....	121
4.8.3 Administración de Correos.....	122
4.9 Diagrama de Clases.....	124
4.10 Diagrama de Estructura de Datos.....	126
4.11. API de Linux utilizados.....	127
4.11.1 Networking.....	128

4.12 Carpetas y archivos de Optimail	128
4.12.1. Carpeta CFGCLSES	128
4.12.2. Carpeta DATA	129
4.12.3. Carpeta EDITOR	129
4.12.4. Carpeta GRAFICOEX	129
4.12.5. Carpeta ICONOS	129
4.12.6. Carpeta IMAGES	130
4.12.7. Carpeta LIBRERIA	130
4.12.8. Carpeta MODULES	130
4.12.9. Carpeta TEMPLATEFORMS	131
4.12.10. Carpeta TEMPLATES	131
CAPITULO V. IMPLEMENTACIÓN Y PRUEBAS	134
5.1 Implementación	134
5.1.1. Compilación	134
5.1.2. Instalación de la Aplicación	138
5.1.3. Configuraciones	140
5.1.3.1. Archivo de configuración	140
5.1.3.2. Enviar email por php	140
5.1.3.3. Parámetros necesarios en todos los casos	141
5.1.3.4. Parámetros opcionales del envío de correo	141
5.1.3.5. Activando servicios y accediendo a un browser	141
5.1.3.6. Instalación y configuración de la base de datos para Optimail	142
5.2 Pruebas	144
5.2.1. Pruebas de carga	144
5.2.2. Limitaciones de pruebas	148
5.2.3. Pruebas funcionales	148
5.2.4. Pruebas de Concurrencia	150

5.3. Funcionamiento de Optimail.....	152
5.3.1. Página de inicio de sesión.....	152
5.3.2. Creación de nuevos usuarios.....	153
5.3.3. Ingreso al sistema.....	155
5.3.4. Buzones y opciones.....	156
5.3.5. Enviar correo electrónico.....	157
5.3.5.1. Adjuntar archivo.....	159
5.3.5.2. Cuadro de texto.....	159
5.3.5.3. Enviar.....	159
5.3.6. Lectura de correos.....	161
5.3.7. Responder y reenviar correo.....	162
5.3.8. Eliminar correo.....	163
CONCLUSIONES.....	164
RECOMENDACIONES.....	166
BIBLIOGRAFIA.....	168
ANEXOS.....	171

ÍNDICE IMÁGENES

CAPITULO III: CONFIGURACIONES INICIALES

3.1. Bienvenida de Fedora Core 7.....	69
3.2. Mensaje para test de verificación de medios.....	69
3.3. Idioma del proceso de instalación.....	70
3.4. Pantalla de particiones.....	71
3.5. Configuración de red.....	72
3.6. Edición de interface.....	73
3.7. Configuración de nombre del host, puerta de enlace y DNS.....	74

3.8. Tareas Soportadas.....	75
3.9. Paquetes adicionales.....	76
3.10. Empezando la instalación de Fedora.....	77
3.11. Pantalla de configuración de firewall.....	78
3.12. Deshabilitando firewall.....	78
3.13. Firewall post-instalación.....	79
3.14. Archivo hosts.....	81
3.15. Archivo de configuración de QUOTA.....	82
3.16. Archivo smtp.conf.....	85
3.17. Archivo main.cf.....	86,87,88
3.18. Código parcial del archivo httpd.conf.....	91
3.19. Archivo php.conf.....	,91,92

CAPITULO V: IMPLEMENTACION Y PRUEBAS

5.1. Creación de nuevo proyecto con InstallBuilder.....	135
5.2. Detalles del paquete.....	136
5.3. Archivos a agregar al paquete Optimail.....	136
5.4. Descripción del paquete.....	137
5.5. Plataforma y compilación del paquete.....	138
5.6. Página de Apache.....	142
5.7. Pantalla de PhpMyAdmin.....	143
5.8. Pantalla de ejecución de query de SQL en PhpMyAdmin.....	144
5.9. Inicio de sesión de Optimail.....	153
5.10. Registro de nuevo usuario.....	154
5.11. Éxito en la creación de usuario.....	155
5.12. Ingreso a Optimail.....	156
5.13. Buzones Optimail.....	156
5.14. Creación de nuevo correo.....	158

5.15. Éxito en el envío de mensaje de correo	160
5.16. Carpeta “Enviado”	160
5.17. Lista de Correos	161
5.18. Archivo adjunto en correo	161
5.19. Cuadro de diálogo para la descarga de archivo	162
5.20. Respondiendo correo electrónico	162
5.21. Reenviando correo electrónico	163
5.22. Carpeta “Borrado”	163

ÍNDICE TABLAS

CAPITULO III: CONFIGURACIONES INICIALES

3.1. Comparativa de la situación actual del software libre en el área de la seguridad	97
--	----

CAPITULO IV: DESARROLLO

4.1. Modelo de objetos	110
------------------------------	-----

CAPITULO V: IMPLEMENTACION Y PRUEBAS

5.1. Usuarios simultáneos	144
5.2. Tiempos de respuesta	145
5.3. Tiempos de respuesta con adjunto	147
5.4. Capacidad de buzones	147
5.5. Prueba de carga	148
5.6. Ingreso al sistema	149
5.7. Registro de usuarios en el sistema	149
5.8. Usuarios simultáneos	151

ÍNDICE FIGURAS

CAPITULO II: ESTUDIO DE ALTERNATIVAS DE SOLUCIÓN

2.1. Estructura jerárquica de DNS	42
2.2. Esquema de Optimail	65

CAPITULO IV: DESARROLLO

4.1. Funcionalidad mínima	107
4.2. Funcionalidad externa	107
4.3. Agrupación de funcionalidades	107
4.4. Relación entre funcionalidades	107
4.5. Nombre anónimo	108
4.6. Nombre del usuario sin permiso de acceso	108
4.7. Nombre del usuario con permiso de acceso	108
4.8. Herencia entre usuarios	109
4.9. Asociación funcional	109
4.10. Adquisición de funcionalidad	109
4.11. Modelo navegacional	111
4.12. M. Navegacional sistema externo	111
4.13. Cambio de rol	111
4.14. Contexto-Contexto	112
4.15. Servidor de correo	112
4.16. Envío de Correo	113
4.17. Correo (Principal)	117
4.18. Acceso al sistema	118
4.19. Crear Cuentas	119
4.20. Administrar correo	119

4.21. Diagrama de secuencia: crear cuentas	120
4.22. Diagrama de secuencia: Acceso al sistema	121
4.23. Diagrama de secuencia: Administración de correos.....	122
4.24. Diagrama de Clases.....	124
4.25. Diagrama de estructura de datos	126

CAPITULO V: IMPLEMENTACION Y PRUEBAS

5.1. Tiempo de respuesta (sin archivo adjunto).....	145
5.2. Tiempo de respuesta (con archivo adjunto).....	146

ÍNDICE ANEXOS

A. Raíz Optimail	171
A-1. Index.php.....	171
A-2. Formularioereg.php.....	175
B. Carpeta CFGCLASES	181
B-1. Clases.php.....	181
B-2. Clasesmod.php.....	183
B-3. Config.php.....	184
C. Carpeta LIBRERIA	185
C-1. Access.php.....	185
C-2. Aplicativo.php.....	186
C-3. Attachmentread.class.php.....	188
C-4. Class.página.Z.....	192
C-5. Dbcc.php.....	201
C-6. Func_g.php.....	203
C-7. Menú.php.....	204
C-8. Parámetros.php.....	219
C-9. Session.php.....	220
D. Carpeta IBTEMPLATEV2	223
D-1. Borrar.php.....	223
D-2. Carpetas.php.....	223
D-3. Enviar.php.....	225
D-4. Envío.php.....	227
D-5. Gotodownload.php.....	230
D-6. Inicio.php.....	233
D-7. Leer.php.....	234

D-8. Listac.php	237
D-9. Nuevacarpeta.php	240
D-10. Nuevacarpeta2.php.....	241
D-11. Nuevo.php	242
D-12. Readattachement.php.....	245
D-13. Readcell.php.....	245
D-14. Receivemail.class.php.....	253
D-15. Vaciar.php	258
E. Carpeta MENU	259
E-1. Menú.php.....	259
E. Logs de Correo	263
F-1. Maillog sin adjunto.....	263
F-2. Maillog con adjunto.....	266

CAPITULO I. DISEÑO CONCEPTUAL DE LA INVESTIGACIÓN

1.1 DETERMINACIÓN DEL PROBLEMA

Actualmente la comunicación dentro de las organizaciones y empresas es una necesidad, obteniendo de su evolución un establecimiento tecnológico importante del cual su utilización define el presente y futuro de una entidad.

En el artículo titulado: “Mejorando la competitividad mediante el uso del correo electrónico”, su autor José Manuel Alarcón indica que “...Cualquier empresa depende en gran manera de las comunicaciones para generar y sustentar su valor. Es necesario relacionarse con todo tipo de agentes, tanto internos como externos: clientes, trabajadores, distribuidores, proveedores, prensa, socios... La lista es ingente, y las necesidades son diferentes con cada uno de ellos. Una empresa jamás se podrá diferenciar de su competencia si no mejora su forma de comunicarse”¹. Por lo tanto la comunicación es un medio para conseguir los fines de las organizaciones..

Alarcón Director Técnico de Krasis, una empresa que ofrece formación y apoyo técnico a empresas en las tecnologías de información (TI) indica que...”para gestionar de forma adecuada las comunicaciones es necesario escoger las herramientas apropiadas y disponer de las pautas de comunicación precisas. Cada ámbito necesita de su propio canal, frecuencia y estilo comunicativo”.

Cuando las empresas quieren transmitir mensajes de sus productos, su actividad o necesita un acercamiento con los consumidores, los medios que se disponen son el teléfono, el correo tradicional, el correo electrónico, la página Web y la publicidad impresa, por citar los más importantes.

¹ José Manuel Alarcón, Socio-Director Técnico de Krasis
http://www.microsoft.com/spain/empresas/tecnologia/uso_correo_competitivo.msp

De acuerdo a las nuevas tecnologías y un mundo totalmente globalizado, el correo electrónico se constituye en la herramienta de comunicación más importante casi en todas las empresas. Si consideramos su efectividad, rapidez y versatilidad podemos utilizar tanto para visualizar un documento como una carta, observar un gráfico coloreado, mantener un intercambio de ideas casi instantáneas (chat) o mantener una conversación con voz y hasta con imagen (videoconferencia), por lo que con todo esto estaríamos hablando de que con toda esa tecnología al alcance de todo el mundo es posible reemplazar al teléfono, el fax y otros medios tradicionales.

Entre las ventajas de este medio tecnológico se encuentran su flexibilidad, la conveniencia de su uso, su baja interferencia, su rapidez y su bajo coste. Por eso es muy común y normal que toda persona ya sea independiente o formando parte de una empresa disponga de su dirección electrónica y por lo tanto pueda de esta manera enviar y recibir comunicaciones digitales las 24 horas del día.

Reiterando en lo antes expuesto en cuanto a la comunicación, el servicio de correo electrónico se constituye en la herramienta de mayor utilidad pero que puede ser mejor aprovechada con un poco más conocimiento de todas las ventajas posibles que ofrece este medio tecnológico.

1.2. DEFINICIÓN DEL TEMA

Sistema procesador de mensajes de correo electrónico: Investigación y Construcción de una herramienta de correo electrónico utilizando API de Linux.

1.3. JUSTIFICACIÓN E IMPORTANCIA

Actualmente existe la necesidad de contar con información a tiempo y actualizada. Es imprescindible que existan medios que permitan a los usuarios realizar esta tarea de manera eficiente y sencilla debido a la importancia del flujo de información.

El correo electrónico es uno de los servicios más usados ya que permite a los usuarios enviar y recibir mensajes rápidamente y de manera eficaz. Las plataformas más usadas son Unix y Windows de las cuales cada una de ellas presta a sus usuarios el servicio de correo electrónico.

El conocimiento de la funcionalidad de API de Linux en el desarrollo de correo electrónico permitirá solucionar necesidades específicas relacionadas con comunicaciones. La presente tesis investiga las herramientas que componen la interfaz de programa de aplicación (API) de Linux y la manera que se realiza el flujo de mensajes.

La aplicación desarrollada, OPTIMAIL, es un prototipo de cliente de correo electrónico que proporciona los servicios básicos a los usuarios, como el envío de mensajes y tipos de documentos más usadas en una empresa.

1.4. OBJETIVOS

1.4.1 Objetivo General

Construir una aplicación de correo electrónico basada en un Api de Linux que permita en el proceso conocer el potencial de las API desarrolladas para Linux.

1.4.2 Objetivos específicos

- Estudiar las diferentes librerías encontradas dentro del API de Linux que se relacionen con la transmisión de datos.
- Desarrollar una investigación que despierte el interés de los profesionales de comunicaciones y promueva la creación de software de comunicaciones, cada vez más sofisticado y completo, utilizando nuevas y existentes herramientas.
- Estudiar el Protocolo de Transferencia de Correo Simple (Simple Mail Transfer Protocol, SMTP) como un protocolo estándar para la transmisión de mensajes entre diferentes redes.
- Estudiar el Protocolo de Administración de Correo Electrónico (Post Office Protocol, POP).
- Implementar la aplicación, OPTIMAIL, como un cliente del sistema de correo electrónico de LINUX.

1.5. DELIMITACIÓN DEL TEMA

El tema de correo electrónico, por su naturaleza, será tratado en relación a los siguientes ámbitos tecnológicos:

- Estudio del protocolo de comunicación SMTP a nivel de la capa de aplicación, como herramienta de intercambio de mensajes en ambientes Linux y Windows.
- Estudio del protocolo POP a nivel de la capa de aplicación como herramienta para la administración de correo electrónico en ambiente Windows.
- Estudio de herramientas que permitan el intercambio de mensajes de correo dentro del contexto de un sistema de correo electrónico.
- Desarrollo de una aplicación cliente usando una interfase Windows para procesar correo electrónico. Esta aplicación contemplará:
 - Intercambio de mensajes locales y remotos
 - Transmisión de texto e imágenes
 - La utilización de formas electrónicas especiales para envío y recepción de mensajes
 - Creación de cuentas PostOffice para que un usuario pueda ingresar a la aplicación
 - SPAM
 - Protección de datos

1.6. LIMITACIÓN DE LA INVESTIGACIÓN

Considerando que el tema a investigar es parte de la tecnología web y utiliza la información, aplicaciones y herramientas disponibles en Internet, la investigación es completamente viable.

1.7. MARCO TEÓRICO

El Correo electrónico es un servicio de red para permitir a los usuarios enviar y recibir mensajes rápidamente mediante sistemas de comunicación electrónicos.

Principalmente se usa este nombre para denominar al sistema que provee este servicio en Internet, mediante el protocolo SMTP, aunque por extensión también puede verse aplicado a sistemas análogos que usen otras tecnologías. Su eficiencia, conveniencia y bajo costo están logrando que el correo electrónico desplace al correo normal para muchos usos habituales.

Para que una persona pueda enviar un correo a otra, ambas deben tener una dirección de correo electrónico, dirección que es facilitada por un proveedor de correo (ISP), que son las empresas que ofrecen el servicio de envío y recepción de correos electrónicos. El procedimiento se puede hacer desde un correo web o un programa de correo.

Cuando una persona desea escribir un correo electrónico, el programa de correo pedirá al menos tres cosas que son: el destinatario, asunto y el mensaje. Como opción adicional se

puede adjuntar archivos que permite compartir datos informáticos de cualquier tipo mediante el correo electrónico.

Por la demanda creciente del servicio de correo electrónico existen proveedores de servicios de correo electrónico gratuito:

- Gmail: webmail y POP3
- Hotmail: webmail
- Yahoo!: webmail y POP3 con publicidad
- Lycos: webmail

Se considera al servicio de correo electrónico una aplicación de misión crítica en el área global de negocios. No contar en una empresa con servicio de correo electrónico limita la comunicación interna, ágil y confiable. Sin embargo existe un juego de comandos que permite el manejo de la mensajería electrónica.

Ahora son posibles los comandos descriptos a continuación, no hay ni una cantidad ni una secuencia específica para la ejecución de estos comandos, a excepción del comando QUIT que es el que finaliza el estado de transacción:

STAT : no posee argumentos y la respuesta es la cantidad de mensajes actuales en el buzón que no están marcados como eliminados y su tamaño en bytes.

LIST : tiene como parámetro opcional el número de mensaje. Sin parámetro hace que el servidor responda dando la lista de los mensajes no marcados como eliminados en el buzón,

junto con el tamaño en bytes de cada uno de ellos. Si se da como parámetro un mensaje se muestra la información de ese mensaje en particular.

RETR : tiene como parámetro obligatorio el número de mensaje. Y devuelve una respuesta multi-línea donde la primera indica, si el mensaje existe, el tamaño y a continuación envía el mensaje.

DELE : tiene como parámetro el número de mensaje. Este numero de mensaje es el que se va a marcar como eliminado.

RSET : No tiene parámetros y su acción es desmarcar los mensajes que fueron marcados para su eliminación durante esta transacción.

QUIT : Pasa al próximo estado.

Estado de Actualización

El estado de actualización desaloja los mensajes marcados como eliminados en el estado de transacción, esto lo hará sólo si el cliente emite un comando QUIT en este estado en caso contrario los mensajes quedarán marcados como eliminados pero no desalojados de el almacenamiento. Además este comando libera la conexión TCP e informa el estado actual del buzón.

Comandos Adicionales

POP3: posee además un par de comandos especiales que no son necesarios para su operación básica. Estos comandos son:

TOP: requiere dos parámetros un identificador de mensaje y un número no negativo que indica la cantidad de líneas que se deben enviar de ese mensaje.

UIDL: tiene como parámetro opcional un número de mensaje, y devuelve el id único de el mensaje que se dio como parámetro, o el de todos los mensajes de el buzón. Este id único identifica unívocamente al mensaje dentro del servidor y es asignado por éste. Los mensajes marcados como eliminados no se listan.

MIME: Las extensiones MIME para el SMTP no hacen más que complementarlo para hacer más flexible su uso con tipos de datos no-ASCII, MIME especifica tres campos que se incluyen en la cabecera del mensaje, estos son MIME-Versión, que especifica que versión del MIME se uso para codificar ese mensaje; el campo Content-Type que especifica el tipo y subtipo de los datos no-ASCII; y Content-Transfer-Encoding que especifica el tipo de codificación usado para traducir los datos en ASCII.

Los valores legales para el campo Content-Type son: Text, Image, Audio, Video, Application, Multipart y Message.

Los valores definidos para Content-Transfer-Encoding son: 7bit, 8bit, binary, quoted-printable, base64, itef-token, x-token.

Los tipos del campo contenido, casi hablan por sí mismos a excepción de uno, el tipo multipart; que tiene 4 subtipos, mixed, alternative, parallel y digest. Este tipo significa que el mensaje contiene en sí mismo información de diferentes tipos o en distintos formatos. Cada una de las partes del mensaje se separa con un delimitador, que toma la forma de una cadena especificada en el campo Boudary que sigue al Content-Type. El subtipo Mixed indica que el mensaje encierra parte de distintos tipos, en cada comienzo de una nueva parte, después de la cadena delimitadora, debe especificarse el tipo y la codificación de la parte. El subtipo Alternative permite que el mismo mensaje se codifique usando distintos métodos, para asegurarse que el mensaje pueda ser leído por el programa del destinatario. El subtipo Parallel indica que las partes deben mostrarse juntas. Y por último el subtipo Digest indica que contiene un conjunto de mensaje, por ejemplo una discusión por e-mail.

Spooling

La entrega de correo electrónico es un nuevo concepto porque difiere de otros protocolos de redes para transferencia de archivo. En los otros protocolos de red se envía los paquetes directamente a sus destinos, utilizando límite de tiempo y retransmisión para los segmentos individuales si no se recibe un acuse de recibo (una confirmación). Sin embargo el caso de correo electrónico, el sistema debe proporcionar los medios cuando la maquina remota o las conexiones de red han fallado. Para eso, el sistema de correo electrónico utiliza una técnica conocida como spooling.

Cuando el usuario envía un mensaje de correo, el sistema coloca una copia en su área de almacenamiento privado spool. El sistema entonces, inicia la transferencia hacia la maquina remota como una actividad subordinada o secundaria.

El proceso subordinado de transferencia de correo se establece como un cliente, utiliza el sistema de nombre de dominio (DNS) para transformar el nombre de la maquina de destino en una dirección IP y luego trata de establecer la conexión TCP hacia el servidor de correo en la maquina destino. Si la conexión tiene éxito el PT envía una copia del mensaje al servidor remoto, el cual lo almacena en el área de spool del sistema remoto, una vez que el cliente y el servidor acuerdan que la copia ha sido copiada y almacenada, el cliente desecha la copia local. Si no se puede establecer la conexión o si la conexión falla el proceso de transferencia registra la hora y termina el proceso. El PT cada cierto tiempo realiza un barrido en el área de spool en busca de correo no enviado.

Synaptic Synaptic es un programa informático que es una interfaz gráfica GTK+ de APT, para el sistema de gestión de paquetes de Debian GNU/Linux. Generalmente se utiliza Synaptic para sistemas basados en paquetes .deb pero también puede ser usado en sistemas basados en paquetes RPM.

El desarrollo de Synaptic se inició en Conectiva, quien solicitó a Alfredo Kojima que escribiera un frontispicio para apt, continuando el trabajo iniciado con la creación del back-end apt-rpm . Finalmente, se optó por utilizarlo en el instalador de Conectiva. Gustavo Niemeyer también colaboró en el proyecto, mientras estuvo trabajando para Conectiva. Actualmente, Michael Vogt se encarga de mantener el programa.

Synaptic utiliza repositorios Debian, y permite su gestión mediante un menú interactivo. Mediante un clic, es posible actualizar la lista, de paquetes disponibles o marcar automáticamente todas las actualizaciones de los paquetes ya instalados.

Dispone de un avanzado filtro de búsqueda, es capaz de reparar las dependencias rotas de paquetes y permite deshacer y rehacer las últimas selecciones de paquetes. Su interfaz gráfica, además, es personalizable mediante el menú de configuración.

Sendmail es un MTA (agente de transporte de correo), que es el programa que se encarga de mover el correo de una máquina a otra. Sendmail lleva incorporado aliasing y *forwarding*, rutado automático hacia puertas de enlace y una configuración flexible. Una solución potente para cualquier entorno. Su configuración es complicada y durante su larga vida ha tenido numerosos problemas de seguridad. Aun así sigue siendo el más utilizado.

Postfix (en el caso de que falle sendmail como otra alternativa) . Postfix es un servidor de correo (MTA) muy potente, programado por Wietse Venema, y cuya página web es <http://www.postfix.org/>. Es la mejor elección actualmente ya que es más fácil configurarlo y no contiene bugs como sendmail. Muchos sistemas han optado por migrar a postfix teniendo que deshabilitar o desinstalar sendmail que por lo general viene por defecto en el sistema operativo.

Cada vez que se quiera comprobar si el servidor está funcionando de manera correcta, tanto para enviar como para recibir, se puede enviar un mensaje de correo a la siguiente dirección: echo@rediris.es. Cualquier mensaje que se envíe a esta dirección será devuelto.

Servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.11 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf eligió ese nombre porque quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "parcheado")

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. (Estadísticas históricas y de uso diario proporcionadas por Netcraft²).

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

Ventajas

- Modular
- Open source
- Multi-plataforma
- Extensible
- Popular (fácil conseguir ayuda/suporte)
- Gratuito

Un **servidor DNS** (Domain Name System) se utiliza para proveer a las computadoras de los usuarios (clientes) un nombre equivalente a las direcciones IP. El uso de este servidor es transparente para los usuarios cuando éste está bien configurado.

Cada LAN (Red de área local) debería contar con un servidor DNS. Estos servidores trabajan de forma jerárquica para intercambiar información y obtener las direcciones IP de otras LANs.

NIC (Network Information Center) es el organismo encargado de administrar el DNS a nivel mundial. NIC México se encarga de administrar todos los nombres (dominios) que terminen con la extensión mx. En la rectoría del sistema se encargan de administrar los nombres que terminen con itesm.mx. El Campus Ciudad de México cuenta con un Servidor DNS primario y un Servidor DNS secundario, los cuales mantienen las tablas de los nombres que terminen con ccm.itesm.mx.

Nuestro servidor DNS primario tiene la dirección 148.241.155.10 y el secundario 148.241.129.10. Para que la computadora funcione adecuadamente debes contar con estos valores. Si configuras adecuadamente el servicio de DHCP del campus automáticamente obtienes estos valores. Si cuentas con una dirección IP fija debes agregar manualmente estos valores. En el Intratec de alumnos obtiene automáticamente estos valores, pero en el Intratec de profesores hay que agregarlo a la hora de hacer configuración. Si se siguen adecuadamente las configuraciones no deberás tener problema.

Dynamic Host Configuration Protocol DHCP (sigla en inglés de Dynamic Host Configuration Protocol - Protocolo Dinámico de Configuración de Anfitrión) es un protocolo de red que permite a los nodos de una red IP obtener sus parámetros de configuración automáticamente. Se trata de un protocolo de tipo cliente/servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas van estando libres, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después.

Características

Provee los parámetros de configuración a las computadoras conectadas a la red informática con la pila de protocolos TCP/IP (Máscara de red, puerta de enlace y otros) y también incluyen mecanismo de asignación de direcciones de IP.

Este protocolo se publicó en octubre de 1993, estando documentado actualmente en la RFC 2131. Los últimos publicados como RFC 3415

Enlaces externos

- [ISC DHCP Server \(inglés\)](#)
- [Windows DHCP Server \(inglés\)](#)
- [Configurar un servidor DHCP, Configuración de un servidor DHCP en linux](#)
- [Tutorial DHCP](#)
- [Videotutorial de DHCP en Linux](#)

Servidor de correo

Un servidor de correo es una aplicación informática que nos permite enviar mensajes (correos) de unos usuarios a otros, con independencia de la red que dichos usuarios estén utilizando. Entre los más usados se encuentran sendmail y Exim este último predeterminado en la distribución GNU/Linux Debian

Para lograr la conexión se definen una serie de protocolos, cada uno con una finalidad concreta:

- **SMTP, Simple Mail Transfer Protocol:** Es el protocolo que se utiliza para que dos servidores de correo intercambien mensajes.
- **POP, Post Office Protocol:** Se utiliza para obtener los mensajes guardados en el servidor y pasárselos al usuario.

- IMAP, Internet Message Access Protocol: Su finalidad es la misma que la de POP, pero el funcionamiento y las funcionalidades que ofrecen son diferentes.

Así pues, un servidor de correo consta en realidad de dos servidores: un servidor SMTP que será el encargado de enviar y recibir mensajes, y un servidor POP/IMAP que será el que permita a los usuarios obtener sus mensajes.

Para obtener los mensajes del servidor, los usuarios se sirven de clientes, es decir, programas que implementan un protocolo POP/IMAP. En algunas ocasiones el cliente se ejecuta en la máquina del usuario (como el caso de Mozilla Thunderbird, Evolution, Microsoft Outlook). Sin embargo existe otra posibilidad: que el cliente de correo no se ejecute en la máquina del usuario; es el caso de los clientes vía web, como GMail, Hotmail, OpenWebmail, SquirrelMail o Terra. En ellos la arquitectura del servicio es más compleja:

En una máquina (A) tenemos el servidor SMTP y el servidor POP/IMAP. En otra (B) tenemos un servidor web con una aplicación cliente POP/IMAP. El usuario conecta vía WEB con (B) y entonces el cliente POP/IMAP establece una conexión POP/IMAP con el servidor de la máquina A; éste servidor le devuelve a B los mensajes del usuario, y una vez recibidos, el cliente genera una página web con los mensajes recibidos. La página web se pasa al servidor web que será el que la envíe al explorador web del usuario.

En cualquier caso, los protocolos SMTP/POP/IMAP son inseguros en cuanto a que los mensajes viajan en claro por la red, es decir, es fácil obtener nuestros mensajes y contraseñas. Para ello se suele añadir una capa SSL, es decir, un método de cifrado que puedan implementar tanto el servidor como el cliente. En el caso del correo vía web se pueden utilizar

dos capas SSL: una entre A y B y otra entre el servidor web de B y el navegador web del usuario.

1.8 GLOSARIO DE TÉRMINOS

API (Application Programming Interface): es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

DNS (Domain Name System): Es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet.

HTTP (Hypertext Transfer Protocol): Protocolo de transferencia de hipertextos. Es un protocolo que permite transferir información en archivos de texto, gráficos, video, audio y otros recursos multimedia.

ISP: Internet Service Provider (Proveedor del servicio de Internet) es una empresa que proporciona el servicio de acceso a la red Internet.

IP (Internet Protocol): Conjunto de reglas que regulan la transmisión de paquetes de datos a través de Internet.

MIME: Multipurpose Internet Mail Extensions. Extensiones Multipropósito de Correo Internet. Extensiones del protocolo de correo de Internet que permiten incluir información adicional al simple texto.

POP3 (Post Office Protocol): Protocolo estándar utilizado para recuperar correo electrónico almacenado en un servidor de correo.

PT: Realiza un barrido del área de almacenamiento privado.

SMTP (Simple Mail Transfer Protocol): Protocolo que se usa para transmitir correo electrónico entre servidores.

SPAM: Se denomina Spam o “correo basura” a todo tipo de comunicación no solicitada, realizada por vía electrónica.

SPOOLING (Simultaneous Peripheral Operations On-Line): se refiere al proceso mediante el cual la computadora introduce trabajos en un buffer (un área especial en memoria o en un disco), de manera que un dispositivo pueda acceder a ellos cuando esté listo.

TCP (Transmission Control Protocol): Forma de comunicación básica de Internet, la cual hace posible que cualquier tipo de información (mensajes, gráficos o audio) viaje en forma de paquetes sin que estos se pierdan y siguiendo cualquier ruta posible.

UNIX: Sistema operativo multiusuario y multitarea, desarrollado originalmente por Ken Thompson y Dennis Ritchie en los laboratorios Bell en 1969, para su uso en minicomputadoras.

1.9. DISEÑO DE LA INVESTIGACIÓN

El método a utilizarse en esta investigación es el Inductivo ya que al ser un tema netamente técnico se posee todo el material técnico necesario para proceder con el establecimiento de los requerimientos sustanciales en el desarrollo de la herramienta de correo electrónico.

1.10 TEMARIO TENTATIVO

CAPITULO I. DISEÑO CONCEPTUAL DE LA INVESTIGACION

- 1.1. Determinación del problema
- 1.2. Definición del tema
- 1.3. Justificación e importancia
- 1.4. Objetivos
 - 1.4.1. Objetivo general
 - 1.4.2. Objetivos específicos
- 1.5. Delimitación del tema
- 1.6. Limitación de la Investigación
- 1.7. Marco Teórico
- 1.8. Glosario de términos
- 1.9. Diseño de la Investigación
- 1.10. Temario Tentativo

CAPITULO II. ESTUDIO DE ALTERNATIVAS DE SOLUCION

- 2.1. Especificación de requerimientos
 - 2.1.1. DNS
 - 2.1.1.1. Estructura de un nombre
 - 2.1.1.2. Componentes DNS
 - 2.1.1.3. Aplicaciones de DNS
 - 2.1.1.4. Estructura Jerárquica
 - 2.1.1.5. Proceso de resolución de nombres
 - 2.1.1.6. Tipos de DNS
 - 2.1.1.7. Tipos de servidores DNS
 - 2.1.1.8. Récor ds de un DNS

2.1.2. Agentes

2.1.2.1. MTA

2.1.2.2. MUA

2.1.2.3. MDA

2.1.3. Funcionamiento de Correo

2.1.4. Protocolos

2.1.4.1. SMTP

2.1.4.2. POP3

2.1.4.3. IMAP

2.1.5. Formatos de archivos de correo

2.1.5.1. Mail Box (MBOX)

2.1.5.2. Mail Dir (MDIR)

2.1.6. Servidor de Correo

2.1.6.1. Sendmail

2.1.6.2. Postfix

2.2. Alternativas de solución

2.2.1 Selección de la solución

2.3 Diseño de Optimail

2.3.1 Esquema global del funcionamiento

2.3.2 Descripción general de los componentes del servicio

2.4 Análisis de portabilidad

2.5 Análisis de Escalabilidad

2.6 Definición de etapas de implementación

CAPITULO III. CONFIGURACIONES INICIALES

3.1 Instalación y configuración del sistema base para el diseño de Optimail y pruebas de concurrencia

3.1.1 Generalidades

3.1.2. Configuración

3.1.3. Configuraciones Post-Instalación

3.1.3.1. Comandos importantes

3.1.3.2. Configuración /etc/hosts

3.1.3.3. Instalar Software Adicional

3.1.3.4. Instalación de Quota

3.1.3.5. Instalación de servidor DNS (BIND9)

3.1.3.6. Instalación de MYSQL (5.0)

3.1.3.7. Instalación de Postfix con SMTP-AUTH y TLS

3.1.3.8. Maildir

3.1.3.9. Apache2 con PHP5.

3.1.3.10. Habilitar PHP Globalmente.

3.1.3.11 Instalación de módulos Perl

3.2 Seguridades

3.2.1 Intercambio de información.

3.2.2. Instalación de software dañino involuntariamente.

3.2.3. Protección ante accesos no autorizados

3.2.3.1. Fallos de seguridad en la utilización del software

3.2.4. Software libre

3.2.4.1. Ventajas del Software Libre en el mundo de la seguridad

3.2.4.2. Desventajas del software libre.

3.4.4.3. Instalación de Mailscanner y Clamav

CAPITULO IV DESARROLLO

4.1 Metodología de desarrollo

4.1.1 Metodología OO-Method

4.2 Modelo conceptual

4.2.1. Estado del arte

4.2.2. Métodos Hipermediales

4.2.3 Métodos orientados a objetos que extienden UML.

- 4.3. Método OOWS
 - 4.3.1 Fases del Método OOWS
 - 4.3.2 Requisitos del Método OOWS
 - 4.3.2.1. Diagrama Funcional
 - 4.3.2.2. Diagrama de Agentes
 - 4.3.2.3 Diagrama de Asociación funcional
 - 4.3.3. Modelado Conceptual.
 - 4.3.3.1. Modelo de Objetos
 - 4.3.3.2 Modelo Navegacional.
- 4.4. Tecnología de desarrollo.
- 4.5. Características del sistema.
- 4.6. Herramientas de Desarrollo
- 4.7. Caso de Uso
 - 4.7.1. Correo (Principal)
 - 4.7.2 Acceso
 - 4.7.3. Crear Cuentas
 - 4.7.4. Administrar Correo
- 4.8. Diagramas de Secuencias
 - 4.8.1. Crear Cuentas.
 - 4.8.2. Acceso al Sistema.
- 4.9 Diagrama de Clases
- 4.10 Diagrama de Estructura de Datos.
- 4.11. API de Linux utilizados
 - 4.11.1 Networking
- 4.12 Carpetas y archivos de Optimail
 - 4.12.1. Carpeta CFGCLASES
 - 4.12.2. Carpeta DATA
 - 4.12.3. Carpeta EDITOR
 - 4.12.4. Carpeta GRAFICOEX
 - 4.12.5. Carpeta ICONOS
 - 4.12.6. Carpeta IMAGES

- 4.12.7. Carpeta LIBRERÍA
- 4.12.8. Carpeta MODULES
- 4.12.9. Carpeta TEMPLATEFORMS
- 4.12.10. Carpeta TEMPLATES

CAPITULO V. IMPLEMENTACIÓN Y PRUEBAS

5.1 Implementación

- 5.1.1. Compilación
- 5.1.2. Instalación de la Aplicación
- 5.1.3. Configuraciones
 - 5.1.3.1. Archivo de configuración.
 - 5.1.3.2. Enviar email por php.
 - 5.1.3.3. Parámetros necesarios en todos los casos.
 - 5.1.3.4. Parámetros opcionales del envío de correo.
 - 5.1.3.5. Activando servicios y accediendo a un browser.
 - 5.1.3.6. Instalación y configuración de la base de datos para Optimail.

5.2 Pruebas

- 5.2.1. Pruebas de carga.
- 5.2.2. Limitaciones de pruebas.
- 5.2.3. Pruebas funcionales.
- 5.2.4. Pruebas de Concurrencia

5.3. Funcionamiento de Optimail

- 5.3.1. Página de inicio de sesión
- 5.3.2. Creación de nuevos usuarios
- 5.3.3. Ingreso al sistema.
- 5.3.4. Buzones y opciones.
- 5.3.5. Enviar correo electrónico
 - 5.3.5.1. Adjuntar archivo
 - 5.3.5.2. Cuadro de texto

5.3.5.3. Enviar

5.3.6. Lectura de correos

5.3.7. Responder y reenviar correo

5.3.8. Eliminar correo

CAPITULO II. ESTUDIO DE ALTERNATIVAS DE SOLUCIÓN

2.1. ESPECIFICACIÓN DE REQUERIMIENTOS

Para el desarrollo de la herramienta de correo OPTIMAIL se ha procedido a analizar varios conceptos, todos importantes para comprender los requerimientos necesarios para la aplicación y posteriormente la elección de una alternativa de solución adecuada.

2.1.1 DNS

DNS significa Domain Name Service. Se lo puede definir como un traductor entre nombres de dominios e IPs. Cuando un usuario entra a una página web, por ejemplo `www.mipágina.com` en realidad lo que el computador entiende es una dirección IP (ejemplo `192.168.1.9`). El DNS asigna nombres de dominio a direcciones IP y el lugar donde los servidores de correo electrónico se encuentran. Es lógico que este servicio se creó para recordar con mayor facilidad los nombres de los servidores conectados a internet ya que sería incómodo seguir accediendo a un servidor determinado recordando IPs.

2.1.1.1 Estructura de un nombre

Por ejemplo se tiene la dirección `www.cc.optimail.edu.do`, siendo la parte izquierda lo más específico mientras que la derecha es lo menos específico. Al dividir la dirección obtenemos lo siguiente:

- `www`: Nombre del computador.
- `cc`: Departamento.
- `Optimail`: Nombre de la organización
- `Edu`: Tipo de Organización
- `Do`: País

Al conjunto del nombre de la computadora y el nombre del dominio se lo conoce como FQDN (Fully Qualified Domain Name). Poniendo otro ejemplo si un computador tiene un nombre de Host servidor1 y el nombre del dominio dominio1.com entonces el FQDN será servidor1.dominio1.com.

2.1.1.2 Componentes DNS

- Clientes DNS: Son los que se ejecutan en el computador del usuario
- Servidores DNS: Son aquellos que responden las peticiones de los clientes.
- Zonas de autoridad: es una porción del espacio de nombres de dominio de la que es responsable un servidor DNS. Contienen por lo menos un dominio y pueden contener también subdominios.

2.1.1.3 Aplicaciones de DNS

Por lo general, los DNS tienen la siguiente utilidad:

- Host: Traduce obteniendo una IP a partir de un nombre de host.
- Nslookup: Se utiliza para buscar y examinar información dentro de una base de datos de un determinado servidor.
- Dig (“Domain Internet Groper“): En este caso hace un “test” de los servidores de nombres, permite reunir un volumen extenso de información y así realizar simples consultas de nombres de dominio.

2.1.1.4 Estructura Jerárquica

Se lo puede representar por medio del siguiente árbol:

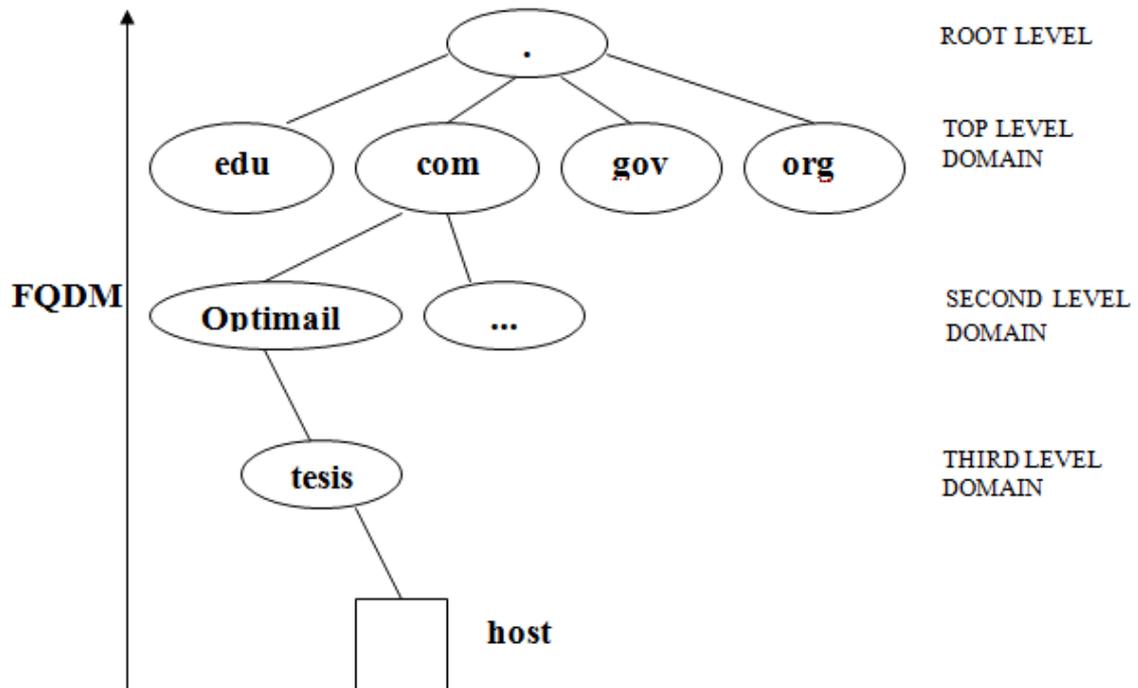


Fig 2.1 Estructura jerárquica de DNS

Siguiendo el orden del árbol, de arriba hacia abajo se tienen varios niveles. Empezando desde el punto a Root Level Domain, sigue Top Level Domain, luego Second Level Domain (SLD) y finalmente Third level Domain. Aunque existen más niveles esos son los más conocidos. El incluir todos los niveles en la dirección se conoce como Full Quality Domain Name(FQDN). Cabe recalcar que la dirección termina en un punto, aunque se lo omite comúnmente es la manera formal y completa de representar la dirección.

Root Level

Root level domain mostrado en la figura 2.1 por un punto es utilizado en la representación formal y es omitida comúnmente.

Top Level Domain

Se divide en dos categorías:

GTLD: Generic Top Level Domain: Mantenido por Internet Assigned Numbers Authority (IANA) para su uso en el internet. Se tienen los siguientes ejemplos:

- Com: comercial
- Edu: educación
- Gov: government (gobierno)
- Mil: militar
- Org: Organización

CCTLD: Country code Top Level domain: Es reservado para el país y su tamaño es de dos caracteres, por ejemplo:

- Ec: Ecuador
- Co: Colombia
- Mx: México

2.1.1.5 Proceso de resolución de nombres

1. Cliente desea resolver el FQDN de un host
2. Servidor de nombres (ISP) envía una consulta del dominio hacia los servidores root.
3. El servidor root envía respuesta al servidor de nombres sobre un grupo de servidores que manejan el siguiente nivel

4. Nuevamente el servidor de nombres envía la consulta hacia el nuevo grupo de servidores.
5. Se obtiene el nombre del servidor con autoridad sobre ese dominio
6. Resuelve el dominio.
7. El servidor de nombres comunica el cliente.

Es recomendable que se registre el dominio con registros que tengan al menos dos servidores (maestro y esclavo) y que además no posean puntos de fallo común, es decir que los dos servidores no estén ubicados en un mismo sitio, esto como un sistema redundante.

2.1.1.6 Tipos de DNS

Se divide en las siguientes categorías:

- DNS de caché (Forward)
- DNS de zona.

DNS de caché

Son aquellos que no contienen récords o zonas que componen un dominio. Almacenan información.

DNS de zona

Los DNS de zona se dividen en:

- Master (Primario)
- Slave (Secundario)

2.1.1.7 Tipos de servidores DNS

- Primarios: un espacio de nombres es guardado en sus ficheros.
- Secundarios: Por medio de una transmisión de zona los servidores DNS secundarios obtienen los datos de los servidores primarios.
- Locales o Caché: Estos no contienen una base de datos para la resolución de nombres aunque tiene otro uso. Consultan a los servidores secundarios cuando se les realiza una consulta. Una vez hecho eso almacenan la respuesta dentro de su base de datos ayudando en la velocidad si existieren repeticiones en las peticiones.

2.1.1.8 Réconds de un DNS

Un récord de una zona se lo puede representar de la siguiente manera:



Se tienen varios tipos de registros de recursos que se los mencionan a continuación:

- 1.- SOA: Start of Authority o inicio de Autoridad Indica el inicio de los datos para una zona además de definir parámetros que afectan a todos los registros para la zona.
- 2.- A: Representa la dirección. Transforma un nombre a una IP
- 3.- NS: Servidor de nombres. Sirve para identificar o declarar DNS de un dominio.
- 4.- PTR: Puntero que realiza lo inverso a A.
- 5.- MX: Oficina de Correos o Mail Exchange identifica el lugar donde se debe enviar el correo electrónico para el dominio.
- 6.- CNAME: Define Alias para la estación por ejemplo de www2.dominio.com a www.dominio.com.
- 7.- HINFO: Muestra la información de la estación.

8.- WKS: Servicios de redes ofertados.

9.- TXT: Texto arbitrario.

Para intercambio de correo se usa el récord MX y la configuración se la debe hacer con un Registrar autorizado por INTERNIC.

2.1.2 Agentes

Importantes para el funcionamiento de una aplicación de correo para la transferencia y recepción de mensajes. Entre los más importantes se pueden mencionar:

- MTA.
- MDA.
- MUA.

Los agentes serán explicados con más detalle a continuación.

2.1.2.1 MTA

Agente de Transferencia de Correo (del inglés Mail Transport Agent o MTA; también Message Transport Agent, Agente de Transporte de Mensajes) es un programa que transfiere correo electrónico de una computadora a otra. En otras palabras, es el servidor de correo (SMTP) en sí y no la parte que usa el usuario para recuperar los mensajes que éste recibió.

El MTA, recibe los mensajes desde otro MTA (relaying), un MSA (Mail submission Agent) que toma por sí mismo el mensaje electrónico desde un MUA (Mail user agent), o recibe directamente el correo desde un MUA, actuando como un MSA. El MTA trabaja en segundo plano, mientras el usuario usualmente interactúa con el MUA.

Algunos de los más conocidos son Sendmail, Postfix, QMail, Exim y Microsoft Exchange Server. En el estudio se analizó dos diferentes MTA para el proyecto, en este caso se eligió Sendmail y Postfix para realizar las pruebas preliminares.

2.1.2.2 MUA

Un MUA es el agente de usuario de correo que se encarga de enviar y recibir correos electrónicos. Para ellos procede con el uso de algunos protocolos de envío y recepción. Para hacer el envío MUA usa el protocolo SMTP y para la recepción usa POP3 o IMAP. Tiene como opción permitir a un usuario leer y crear mensajes además de ayudar al agente de transferencia de correo MTA en el envío del mensaje asegurando de que este llegue a su destino final. Se lo puede instalar en el sistema de usuario, en ese caso se lo conocería como cliente de correo electrónico pero, si se usa una interfaz web para que exista interacción con el servidor de correo que entra, entonces se lo conoce simplemente como correo electrónico. Los MUAs Pueden ser muy gráficos como es el caso de Mozilla Mail o tener una interfaz primitiva a base de texto como lo es Mutt o Pine. Como otros ejemplos de MUA tenemos:

- Kmail
- Sylpheed
- Squirrelmail (los webmails).

Cabe recalcar que la actual herramienta de correo Optimail desarrollada en Linux se basa en MUA por lo que se considera uno de los ejemplos anteriormente mencionados.

2.1.2.3 MDA

Es el agente que se encarga de entregar un mensaje de correo electrónico al buzón de un usuario en particular. Para ello parte de un MTA y de un MUA para enviar y recibir correo pero por esa razón no puede utilizarse directamente sin ellos. Sin embargo, algunos de los MDAs son usados para ordenar los mensajes antes de que sean leídos por el usuario lo que

sería de una gran ayuda si este recibiera mucho correo. No es necesario que los destinatarios estén conectados para poder enviarles un correo electrónico. Como ejemplo se puede obtener los siguientes:

- Qmail
- Exim: Generalmente usado para Debian
- Cyrus
- Courier
- Postfix
- Procmail: Se lo conoce también como LDA (Local Delivery Agent).

2.1.3 Funcionamiento de Correo

Si tendríamos que relacionar lo virtual con lo real, se compararían los MTAs con las oficinas de correo ya que se encarga del transporte de los mensajes y también de la clasificación de ellos. Los MDA serían casilleros, tales casilleros tendrían los mensajes de los usuarios además de que los casilleros tendrían una dimensión y volumen.

Al enviar un correo electrónico se envía al servidor de correo MTA que a su vez tiene que enviar el mensaje al destinatario. La manera en que se comunican es por medio del protocolo SMTP y por esa razón se los conoce como servidores de correo saliente o servidores SMTP. Después de eso el mensaje llega al MDA que lo único que hace es almacenar el correo esperando que el usuario recupere. Para la recuperación se usan dos protocolos que son el POP3 e IMAP a través del MUA.

2.1.4 Protocolos

Se ha mencionado anteriormente que para enviar y recibir correos se usan los protocolos SMTP, IMAP y POP3. La idea general es que SMTP se usa para la transferencia mientras que IMAP y POP3 para su recepción. La mensajería de servidores de correo consiste en la utilización de software que permite enviar mensajes de un usuario a otro

independientemente de la red. A continuación, se verá con más detalle la función de cada uno.

2.1.4.1 SMTP

Cuando se desea realizar una transferencia punto a punto entre servidores se lo usa. El puerto que SMTP usa por defecto es el 25. Este protocolo, encapsulado en una trama TCP/IP, y siendo además un MTA, usa comandos de texto. Esos comandos son recibidos y entendidos por el servidor SMTP que además responderá con un número y mensaje. Se tienen comandos típicos para la sesión SMTP. La manera más común de iniciar una sesión SMTP es usando el comando telnet en la consola de comandos. Una vez iniciada la sesión se procede con los comandos que sólo el servidor SMTP podrá entender. Esos comandos son los siguientes:

- **HELO, EHLO:** Cuando se desea identificar el equipo remitente. Normalmente va seguido por el número de IP (o el nombre del equipo).
- **MAIL FROM:** Como en un correo electrónico común se indica la dirección del remitente como por ejemplo `midirección@midominio.com`
- **DATA:** Cuando se pone DATA se escribe a continuación el mensaje. Constituye el cuerpo del correo electrónico.
- **QUIT:** Finaliza la sesión SMTP.
- **HELP:** Es muy útil cuando uno no tiene conocimiento sobre los comandos que acepta el servidor SMTP. Se desplegará una lista de ellos.

Cuando un comando se ha ingresado satisfactoriamente se verá una respuesta del servidor SMTP que consiste por lo general en "250 OK".

2.1.4.2 POP3

A este protocolo se lo podría considerar como una oficina de correos. La razón es porque recoge el correo electrónico en un servidor POP (servidor remoto). No es necesario que los

individuos estén permanentemente conectados a internet para poder revisar sus correos electrónicos. Hay varias versiones de este protocolo como lo son el POP2 (puerto 109) y POP3 (puerto 110). Tiene un parecido al SMTP en lo que se refiere a los comandos de texto enviados al servidor, pero en este caso es un servidor POP. Este servidor recibe comandos compuestos por una palabra clave que puede tener argumentos en algunos casos y responderá con un número y un mensaje descriptivo.

La versión de POP2 tiene los siguientes comandos:

- HELLO: Comando que sirve para identificarse. Debe contener la dirección del equipo remitente (IP).
- FOLDER: Es importante en la consulta ya que con este comando se direcciona a la bandeja de entrada que se desea consultar. Se debe escribir este comando más el nombre de la bandeja.
- READ: El correo está organizado en varios mensajes y con este comando se indica cuál es el que se desea leer. Se escribe este comando y a continuación el número del mensaje.
- RETRIEVE: Usado de la misma manera que READ pero con la diferencia del que el mensaje se recogerá.
- SAVE: Un mensaje escogido se guardará. Precisararlo agregando el número correspondiente.
- DELETE: Al ingresar este comando más el número de mensaje se lo borrará de la bandeja.
- QUIT: Finaliza la sesión POP2

La versión actual POP3 posee algunas modificaciones. Administra autenticación de usuario con contraseña, pero lamentablemente no tiene cifrado alguno. No es usado por muchos servidores ya que tiene la desventaja de bloquear la bandeja de entrada evitando que los usuarios puedan accederla simultáneamente. Los comandos son los siguientes.

- USER: autenticación. Se debe precisar el usuario (cadena de caracteres)
- PASS: autenticación. Se debe precisar la contraseña del usuario ingresado anteriormente con USER.

- STAT: Indica información de los mensaje del servidor.
- RETR: Mensaje que se recogerá. Para ello agregar el número.
- DELE: Mensaje que se eliminará. Para ello agregar el número.
- LIST: Mensaje que mostrará en la lista. Para ello agregar el número.
- NOOP: En caso de inactividad no cerrará sesión y mantendrá la conexión abierta.
- TOP <n>: Mensaje que muestra *n* líneas de ese mensaje. Si el comando fue ingresado satisfactoriamente, el servidor responderá con los encabezados del mensaje, una línea en blanco y las primeras *n* líneas del mensaje.
- QUIT: finaliza sesión.

Al igual que STMP se puede iniciar una sesión POP2 o POP3 usando el comando telnet.

Procedimiento POP3

1. El usuario se autentica al servidor.
2. El servidor indica cuántos mensajes tiene en el buzón del usuario.
3. El cliente POP baja los mensajes.
4. Una vez que se bajan los mensajes, el servidor procede a borrarlos.
5. El usuario sale del sistema.

2.1.4.3 IMAP

Este protocolo es una versión mejorada de POP3 que usa su funcionalidad además de agregar algunas características adicionales:

- Acceso simultáneo.
- Varias bandejas de entrada.
- Posee más criterios para la organización de correos electrónicos.

El puerto que usa IMAP es el 143.

Procedimiento IMAP

1. El usuario se autentica al servidor.
2. El servidor indica cuántos mensajes tiene en su buzón.
3. El cliente baja parcialmente los mensajes. Contiene la siguiente información:
 - Subject
 - Date
 - From
 - To
 - Prioridad
 - Tamaño
 - Comprobación de lectura (Si fue o no leído)

Optimail es una herramienta de correo que usa STMP (puerto 25) para el envío de correo electrónico e IMAP (puerto 143) para la recepción de correo. La elección de IMAP se debe al acceso simultáneo de usuarios (algo que no posee POP3) y a una mejor administración que este posee. Sólo los subscriptores de Optimail podrán enviarse correo electrónico.

2.1.5 Formatos de archivos de correo

Entre los formatos de archivos de correo se pueden mencionar dos importantes:

- Mailbox
- Maildir

2.1.5.1 Mail Box (MBOX)

En un MBOX se puede almacenar un conjunto de correos electrónicos en un único fichero.

En Linux MBOX se encuentra en el siguiente path:

`/var/spool/mail/archivo_del_usuario`

Dentro del archivo tengo todos los mensajes recibidos. Se convierte en un archivo muy pesado causando problemas de velocidad del sistema ya que se está administrando un archivo de gran volumen de información. Aunque fue usado mucho en el pasado posee una grave desventaja: El acceso simultáneo. Cuando existe acceso concurrente entre dos o más procesos causa anomalías ya que si un programa de reparto de correo quiere escribir un mensaje nuevo en el fichero y un cliente de correo electrónico borra un mensaje al mismo tiempo provoca corrupción y no garantiza la integridad del mensaje. Este inconveniente muy común en MBOX se lo conoce como el problema de bloqueo.

2.1.5.2 Mail Dir (MDIR)

Entendiendo el problema del bloqueo en MBOX, se procedió con la solución que creó a MDIR. No sólo resuelve el problema del bloqueo sino que también posee un directorio con varios subdirectorios. El directorio principal se llama Maildir que contiene tres subdirectorios:

- Tmp: Aquí se almacena el mensaje temporalmente el correo.
- New: Sirve de enlace con la carpeta tmp
- Cur: Se encarga de añadirle sufijos informativos al mensaje guardado en la carpeta new para que se lo pueda leer más adelante. Generalmente representa en poner dos puntos, un '2' (versión de la información actual), una coma y algunos indicadores.

El correo recorre las carpetas tmp, new y cur en el orden respectivo haciendo un proceso de enlace y desenlace al mover de una carpeta a otra. MDIR usa esa estrategia para evitar que un cliente de correo electrónico lea algo del mensaje mientras se está en movimiento.

Como ejemplo en Linux se obtiene la siguiente estructura:

/mail/dir/mensaje1

/mail/dir/mensaje2

/mail/dir/mensaje3

Como se puede apreciar se tiene un path único para cada mensaje incrementando también la velocidad en el acceso

Optimail usa MDIR, además de que se soluciona el problema de bloqueo de MBOX se asegura la integridad de la información y el acceso concurrente. MDIR ofrece también más velocidad a la aplicación.

2.1.6 Servidor de Correo

Linux utiliza los programas Sendmail, QMail y PostFix como servidor de correo. Estos programas son utilizados por muchos proveedores de servicio. Entre otros, utiliza los protocolos SMTP, POP3 e IMAP. Esto quiere decir que como clientes acepta programas de correos como el Exchange Netscape Communicator, Outlook, Eudora, Pegassus mail, etc. El SendMail es el servidor de correos mas usado a nivel mundial, pero no recomendado para un servidor con más de 500 cuentas. El Postfix es un servidor de correos mucho más estable seguro y de capacidad ilimitada de cuentas de correo, utilizado en las principales compañías proveedoras de servicios de Internet en el mundo..

Para ambos servidores existen módulos adicionales para soporte de Listas de Correo, Scan Antivirus Anti Spam, Interface Webmail y Otros.

El servidor de correo Postfix es un agente de transferencia de correo (MTA) orientado a la seguridad, velocidad, y facilidad de configuración, cosas en las que Sendmail suele fallar por lo general. La única parte de Postfix que se ejecuta como root es un programa de control maestro, llamado "master", que llama a otros programas para procesar el correo a la cola ("pickup"), un programa para gestionar la cola, espera conexiones entrantes, repartos de correo retrasados, etc. ("qmgr"), un programa que en realidad envía y recibe el correo ("smtpd") etc. Cada parte de Postfix está muy bien pensada, y generalmente hace una o dos tareas, muy bien. Por ejemplo, en lugar del modelo de sendmail, donde el correo simplemente se volcaba a /var/spool/mqueue, en Postfix existe un directorio accesible por

el mundo llamado "maildrop" el cual se comprueba mediante "pickup", el cual alimenta los datos a "cleanup", el cual mueve el correo (si está correctamente formateado, etc.) a un directorio seguro de cola para el procesado real.

Una empresa encontrará conveniente el poder contar con un Servidor de Correo electrónico con capacidad de recuperar mensajes por medio de los protocolos IMAP, POP3 e interfaz Web. Sus usuarios podrán acceder fácilmente a sus mensajes ya sea desde su cliente de correo electrónico favorito o bien desde el navegador Web de su elección.

2.1.6.1 Sendmail

Sendmail es un popular "agente de transporte de correo" (MTA - Mail Transport Agent) en Internet, cuya tarea consiste en "encaminar" los mensajes correos de forma que estos lleguen a su destino. Se afirma que es el más popular MTA, corriendo sobre sistemas Unix y el responsable de la mayoría de envío del correo de internet, aunque se le critica su alto número de alertas de seguridad (la mayoría de ellas parchadas a las pocas horas), además de no ser sencillo de configurar.

Servidor de Correo Sendmail

SendMail funciona con el protocolo SMTP (Simple Mail Transfer Protocol), el cual es utilizado para comunicarse con otros servidores SendMail, manteniéndose a la escucha de posibles comunicaciones por el socket 25 (Se puede comprobar si un MTA está activo haciendo un Telnet al puerto 25 de una máquina). Este programa abre una conexión contra el mail server remoto. Lo que hace es enviar su nombre de máquina local, así como el nombre del emisor, el buzón de destino y un comando diciendo que empieza el texto del mensaje. en este punto el servidor finaliza el tratamiento de lo que ha asumido como comandos y comienza a aceptar el mensaje hasta que recibe una marca especial (sencillamente, un punto como principio de línea). Después de esto, ambos programas entienden que el envío de comandos ha sido retomado.

2.1.6.2 Postfix

Postfix es un Agente de Transporte de Correo (MTA) de software libre / código abierto, un programa informático para el enrutamiento y envío de correo electrónico, creado con la intención de que sea una alternativa más rápida, fácil de administrar y segura al ampliamente utilizado Sendmail. Formalmente conocido como VMailer e IBM Secure Mailer, fue originalmente escrito por Wietse Venema durante su estancia en el Thomas J. Watson Research Center de IBM, y continúa siendo desarrollado activamente.

Postfix es el agente de transporte por omisión en diversas distribuciones de Linux y en las dos últimas versiones del Mac OS X (Panther y Tiger).

2.2 ALTERNATIVAS DE SOLUCIÓN

Durante las Instalaciones se llegó a la conclusión de que existen varias alternativas de solución, estas alternativas tienen los siguientes detalles:

Sendmail.

¿Qué es sendmail?

- Es un agente de transferencia de correo (MTA)
- Fue creado por Eric Allman en 1983
- Es uno de los sistemas más usados en Internet
- Funciona en la mayoría de los sistemas *nix

Ventajas

- Licencia Open Source

- Existen paquetes para la mayor parte de los *nix
- Funciona bajo alta demanda
- Manejo de dominios múltiples
- Soporta filtros externos (antivirus, antispam, etc.)
- Control de acceso
- Listas de correo, alias y usuarios virtuales
- Amplia documentación disponible

Desventajas

- Sintaxis de configuración compleja
- Su mantenimiento requiere experiencia
- Historial de vulnerabilidades
- Ejecuta como root en versiones previas a la 8.12

Postfix.-

Postfix es un Agente de Transporte de Correo (MTA) de código abierto, o lo que es lo mismo, un programa informático para el enrutamiento y envío de correo electrónico.

Postfix fue creado como alternativa a Sendmail, buscando un servidor que fuera más rápido, fácil de administrar y seguro.

Postfix es de hecho, el MTA que se usa por defecto en muchos sistemas operativos derivados de UNIX, entre ellos, GNU/Linux.

Se distribuye bajo Licencia Pública IBM v.1.0, que a pesar de ser una licencia de software libre, resulta incompatible con la Licencia GPL (aunque sí lo sea con GPL v1.0).

Fue desarrollado por Wietse Venema durante una estancia en el Centro de Investigación Thomas J. Watson de IBM y fue conocido con el nombre de VMailer e IBM Secure Mailer, siendo distribuido al gran público por primera vez a mediados de 1999. Desde entonces sigue en proceso de desarrollo y mejora de una forma activa.

Las principales características de Postfix son:

- Soporte para TLS (Transport Layer Security).
- Soporte para distintas bases de datos LDAP, MySQL, PostgreSQL.
- Soporte para mbox, maildir y dominios virtuales.
- SMTP-AUTH, SASL y reescritura de dirección.
- Soporte para Militer
- Capacidad de manejar altos volúmenes de correo.

Ventajas:

- Diseño modular (no es un único programa monolítico): Postfix está compuesto de varios procesos que se comunican entre sí, con las ventajas e inconvenientes (salvables) que ello conlleva. Esta característica repercute positivamente en otras como veremos.
- Gran Seguridad: Desde el comienzo de su diseño ésta ha sido una de sus premisas fundamentales. De este modo, nos provee de:
 - Seguridad frente a ataques contra el servidor.
 - Seguridad contra el uso inadecuado (spam, relay, etc.), debido a que soporta directamente (sin modificaciones suplementarias) listas negras y que es complicado configurarlo como relay abierto. Además, se puede instalar Postfix de forma que corra en modo chroot, lo que le confiere a su operativa más seguridad.

- Filtrado de cabeceras y cuerpos de mensajes por expresiones regulares.
- Cada proceso corre con los mínimos permisos necesarios para realizar su tarea (ventaja dada por su modularidad).
- Gran Rendimiento: Postfix puede procesar cientos de miles o millones de mensajes al día sin problemas (algo que le desmarca de Sendmail). De hecho, según la documentación, un "PC normal" puede recibir y entregar hasta un millón de mensajes distintos al día con Postfix. Esta potencia se debe en parte a su modularidad, que además viene con el añadido de que se pueden definir ciertos parámetros para cada uno de los procesos, como el número máximo de procesos simultáneos de un tipo, activar o desactivar un proceso (funcionalidad) innecesario, etc., que permiten optimizar aún más su funcionamiento. Además, el sistema de gestión de colas de mensajes es también modular, consistiendo en 4 colas distintas que están procesadas muy eficientemente.
- Soporte para las tecnologías más actuales: Al estar muy activamente actualizado, emplea técnicas desarrolladas para aprovechar mejor y dar soporte a los servidores Web más modernos. Postfix soporta: LDAP, Bases de datos (MySQL), autenticación mediante SASL, LMTP, etc.
- Estricto cumplimiento de los estándares de correo-e: Detrás de unos límites, claro está, dado que estos estándares no están tan difundidos como se desearía.
- Muy buen soporte para administrar dominios virtuales
- Facilidad de configuración:
- Con los dos ficheros de configuración: main.cf y master.cf se controla todo, y además están muy bien explicados.

- Uso sencillo de listas negras.
- Compatibilidad hacia/desde fuera con Sendmail
- Abundante documentación, y de calidad: Aunque en su mayoría, en inglés.
- Fácil integración con programas antivirus: Debido a que se pueden insertar procesos externos entre ciertas partes del sistema Postfix, lo cual es muy útil para integrar un antivirus. Un ejemplo de ello es ClamAV (véase artículo configuración de un servidor incluyendo Postfixadmin, Mysql, Spamassassin y ClamAv).
- Soporta de forma nativa el formato de buzones Maildir original de qmail.
- Facilidad para detectar errores: Postfix tiene múltiples formas de obtener información de los errores ocurridos y los logs generados son muy claros y explicativos. Además, gracias a su modularidad es más sencillo saber qué proceso es el que falla, además de que se puede activar la emisión de más información de depuración de forma independiente para cada programa.
- Posibilidad de lanzar varias instancias de Postfix en la misma máquina con distintas configuraciones: Podemos usar cada una de estas instancias con distintas direcciones IP, distintos puertos, etc. De esta forma podemos tener más de un servidor para cada necesidad.
- Utilidades: Incluye utilidades para varias tareas, como por ejemplo, para gestionarlas colas de mensajes. Otra ventaja conferida en parte por su modularidad.
- Código Fuente abierto y bien estructurado: Se considera el código fuente de Postfix como un ejemplo de diseño, claridad y documentación, por lo que facilita su

mantenimiento por parte de desarrolladores así como la incorporación de nuevas capacidades, corrección de errores, adaptaciones, etc.

Desventajas:

En la práctica se instalaron los dos MTA, y al comparar el Postfix al parecer no tiene desventajas salvo por algunas configuraciones en las que se tuvo problemas, pero fueron pocas y de fácil solución. Por las características y ventajas mencionadas se utilizará como MTA servidor de correo POSTFIX.

2.2.1 Selección de la Solución

Luego del análisis entre los diferentes MTA como servidores de correo se escoge Postfix como servidor de correo para las diferentes pruebas que se harán con Optimail como cliente de correo.

Requerimientos Técnicos.

Para las diferentes pruebas de carga de Optimail se utilizará un equipo de características reales de carga por donde cursarán los correos ya que el MTA escogido es totalmente configurable de acuerdo a las características físicas del equipo servidor.

Hardware.-

- Computador Hewlett Packard.
- Modelo Hp Pavilion dv6000
- Intel (R) Core(TM)2 CPU T4300 1.73 Ghz
- Memoria Ram 2 GB
- Disco duro. 140 GB.

Software.-

Postfix como MTA correo en software basado en RPM así como debían, para las pruebas de carga, se establece como sistema base Fedora Core 7, aunque Optimail puede ser cliente de cualquier sistema. Adicionalmente se necesitan algunos paquetes con los cuales puede trabajar Optimail sin problemas:

- Servidor Apache
- Base de Datos Mysql
- ATM Postfix. Con dovecot, Imap, pop3,cyrus.
- Soporte para PHP y Ajax.

2.3 DISEÑO DE OPTIMAIL

En la implementación de Optimal se utilizo php y ajax como lenguaje base enlazado a Mysql, para esto se utilizaron herramientas de diseño y un editor de php, como los son las herramientas Macromedia.

2.3.1 Esquema Global de Funcionamiento

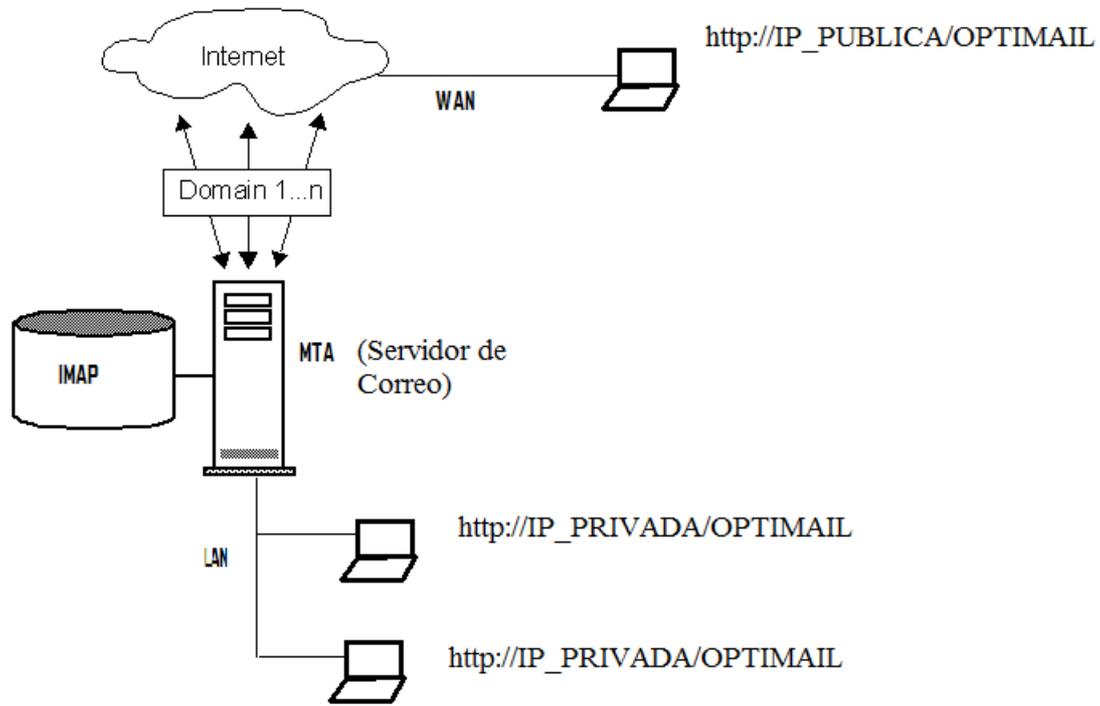


Fig. 2.2. Esquema de Optimail

Optimail puede ser accedido por medio de una IP privada en el caso de una red interna y por medio de una IP en el caso de una red WAN.

2.3.2 Descripción general de los componentes del servicio

El sistema Optimal tiene los siguientes componentes:

- Lector de correos, es un modulo de enlace entre el sistema y el MTA mediante IMAP.

- Modulo de envío de correos. Este modulo utiliza la conexión mediante IMAP par enviar los correos además PHP tiene la función mail para este proceso pero por seguridad se utiliza las funciones IMAP de PHP.
- Modulo de usuarios. En este componente se manejan tanto los usuarios de correo como los usuarios del sistema Optimail, para el ingreso al sistema debe estar registrado tanto en el servidor de correo MTA Postfix como en el sistema Optimail.
- Componente dovecot, que sirve para el enlace mediante imap, ssl al MTA Postfix.

2.4 ANÁLISIS DE PORTABILIDAD

El sistema Optimail, se lo realizo enteramente en software libre por lo tanto al manejar estándares de conexión será capaz de enlazarse con cualquier MTA tanto en Windows como el Linux, tan solo con la configuración de un módulo de enlace en el sistema.

2.5 ANÁLISIS DE ESCALABILIDAD

El sistema Optimail al manejar el lenguaje PHP como base lo hace muy versátil, ya que su programación modular permitirá realizar constantes actualizaciones, además su base de datos MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en un red corporativa. Para agregar, acceder a y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones. Estas características lo hacen versátil.

2.6 DEFINICIÓN DE ETAPAS DE IMPLEMENTACIÓN PARA PRUEBAS DE CARGA DE OPTIMAIL

En esta sección se detallarán todas las etapas para la instalación del servidor con MTA Postfix.

- Instalacion de Fedora 7.
- Configuración de tarjetas de red y dominios.
- Instalación de Web server apache 2.2.4
- Instalacion de PHP 5.2.2
- Instalación de Mysql 5.0.37
- Instalacion de Mail server: Postfix
- Instalación y Configuración de servidor DNS BIND9
- Instalación de servidor FTP, proftpd
- Instalación y configuración de : POP3 / IMAP / Dovecot
- Diseño y programación de Optimail.
- Pruebas de funcionamiento.

CAPITULO III. CONFIGURACIONES INICIALES

3.1 INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA BASE PARA EL DISEÑO DE OPTIMAIL Y PRUEBAS DE CONCURRENCIA

3.1.1 Generalidades

Ya que el sistema operativo base de Optimail es Linux se procederá con la instalación del mismo. La distribuidora a usar será Fedora Core 7. Tiene muchas ventajas, por ejemplo el ahorro de tiempo en instalaciones de paquetes (ya que están incluidos la mayoría), sin embargo, entre las más importantes se puede mencionar:

- Presto. Se reduce el consumo de ancho de banda para la actualización de paquetes. Debido a que los paquetes necesarios para que Optimail son numerosos, esta ventaja ayuda a resolver drásticamente el tiempo que se pierde en ello.
- RPM y Yum. Mejor rendimiento y menor consumo de memoria. Yum y RPM son muy usados en la práctica y caería muy bien una mejoría de esos comandos.

3.1.2 Configuración

Para Instalar el sistema base se procede según los pasos detallados a continuación:

1. Se arranca desde el live CD (CD #1) de la distribuidora de Fedora core 7. Luego se escoge la opción “Install or upgrade an existing system”:



Imagen 3.1: Bienvenida de Fedora Core 7

2. Una de las opciones del sistema Linux es un test de los medios de instalación que permite verificar la integridad de los medios, es recomendable hacer un test antes de proceder con la instalación. Se escoge “OK” para empezar el test.



Imagen 3.2: Mensaje para test de verificación de medios

3. Una vez finalizado el test de verificación de medios se despliega una pantalla de bienvenida, se presiona “next” para continuar.
4. La siguiente opción permite escoger el lenguaje que se usará durante el proceso de instalación y el que se tendrá en el sistema base. Se escogió “English” debido a que la traducción de algunas distribuidoras no siempre es precisa.



Imagen 3.3: Idioma del proceso de instalación

5. Luego se debe definir el idioma que manejará el teclado, se escoge a gusto del usuario. En este caso se escogió “english”
6. A continuación se particionará. Seleccionar el modo de particionamiento que se dará al disco duro. En este caso se va a dejar una configuración por defecto con Administración de Volúmenes lógicos.



Imagen 3.4: Pantalla de particiones

Se quiere eliminar todas las particiones de Linux (recordemos que es un sistema vacío), entonces se responderá “Yes” a la siguiente pregunta:

7. En “network settings” normalmente se configura las interfaces con DHCP pero como se está instalando un servidor entonces se usarán IPs estáticas. Se hace click en “Edit”.



Imagen 3.5: Configuración de red

En la siguiente ventana se selecciona “Enable IPv4 support > Manual configuration” y luego la IP estática que en este caso es: 192.168.1.9 y máscara 255.255.255.0, deshabilitar “Enable IPv6 support” ya que no se va a usar IPv6:



Imagen 3.6: Edición de Interface

8. Configurar el hostname manualmente, (optmail.com). Se debe definir un nombre y un dominio para el servidor. En este caso: localhost.optmail. Recordar que se debe ingresar el FQDN explicado en el capítulo anterior en la casilla marcada por la opción “manually”



Imagen 3.7: Configuración de nombre del host, puerta de enlace y DNS

9. Se procede en escoger la zona horaria. Buscar en el mapa Guayaquil.
10. Se configura una contraseña para el administrador del sistema. Por defecto se lo conoce como “root”. Aparecerán dos cuadros de texto:
 - Root Password
 - Confirm
11. Ahora se seleccionan los paquetes necesarios para el correcto funcionamiento de Optimail. Quitar el visto en “Office and Productivity” y poner visto en “Software Development and Web server”, “Customize now”, y luego click on “Next”:



Imagen 3.8: Tareas Soportadas

Se personaliza la selección de paquetes específicos para Optimail escogiendo los siguientes:

- Editors
- Text-based Internet
- Development Libraries
- Development Tools
- DNS Name Server
- FTP Server
- Mail Server
- MySQL Database
- Server Configuration Tools
- Web Server
- Administration Tools Base
- Hardware Support
- Java

- System Tools

Verificar que está seleccionado “Customize now”. Finalmente hacer click en “Next”:

12. El instalador chequea las dependencias de los paquetes seleccionados



Imagen 3.9: Paquetes adicionales

13. Click en Next para comenzar con la instalación:



Imagen 3.10: Empezando la instalación de Fedora

12. Al finalizar la instalación se debe reiniciar:

3.1.3 Configuraciones Post-Instalación

Luego de reiniciar escoger “Firewall configuration” y seleccionar “Run Tool”:



Imagen 3.11: Pantalla de configuración de Firewall

SELinux es una extensión de seguridad de Fedora que provee más seguridad.

Usualmente causa más problemas que ventajas en algunos servicios por lo que es mejor deshabilitarlo (se instalará ISPCConfig que contiene su propio firewall). Seleccionar "OK"



Imagen 3.12: Deshabilitando Firewall

Luego salir seleccionando “Exit”:



Imagen 3.13: Firewall post-instalación

Entrar como “root” y reiniciar el sistema para que los cambios se apliquen:

```
# shutdown -r now
```

3.1.3.1 Comandos importantes

Se procederá con la explicación de algunos comandos que son necesarios para la configuración del sistema operativo. Estos son ingresados en la terminal de Linux.

YUM

Yellow dog Updater o YUM es un comando muy útil para instalar, desinstalar paquetería RPM y actualizar el sistema.

Para instalar un paquete que no esté instalado, se usa el siguiente comando:

```
# yum install [nombre del paquete]
```

Para desinstalar se usa “remove”:

```
# yum remove [nombre del paquete]
```

Para instalar las actualizaciones se usa “update”:

```
# yum update
```

Finalmente otro uso de YUM es la búsqueda de paquetes, para ello se usa “search”

```
# yum search [cualquier nombre]
```

VI

Es un editor Linux. Se usa generalmente para modificar archivos. La sintaxis es la siguiente:

```
# vi [nombre del archivo]
```

CHMOD

Cada archivo y directorio de Linux tiene su propio set de permisos que controlan el acceso a ellos. Cada uno tiene tres tipos de permisos.

- R: Lectura (read)
- W: Escritura (write)
- X: Ejecución (execute)
- Ellos pueden ser accedidos por tres tipos de usuario:
 - Dueño del archivo.
 - Grupo de usuario.
 - Todos los demás usuarios.

La sumatoria del permiso absoluto es 7, representando 4 para lectura, 2 para escritura y 1 para ejecución. Como ejemplo se tiene el siguiente comando:

```
# chmod 777
```

El primer número representa el dueño del archivo, el segundo el grupo de usuarios y finalmente el tercero resto de usuarios. En este caso se está dando un permiso total a todos los usuarios.

3.1.3.2 Configuración /etc/hosts

A continuación se edita /etc/hosts con el siguiente comando:

```
# vi /etc/hosts
```

Recordar que las facilidades de un DNS es permitir asignar nombres a direcciones IP para memorizar más fácilmente. En Linux ese archivo se encuentra en /etc/hosts. Una vez abierto el archivo, hacer la siguiente modificación:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          localhost.localdomain localhost
192.168.1.109     localhost.optimail localhost

::1               localhost6.localdomain6 localhost6
```

Imagen 3.14: archivo hosts

3.1.3.3 Instalar Software Adicional

Para instalar el software que describiremos primero se importan los GPG keys para paquetes de software con el siguiente comando:

```
# rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY*
```

- Luego se actualizan los paquetes existentes:

```
# yum update
```

- Se instala software extra que se necesitará después:

```
# yum install fetchmail wget bzip2 unzip zip nmap openssl lynx fileutils ncftp gcc gcc-
c++
```

3.1.3.4 Instalación de Quota

Las cuotas permiten especificar límites en aspectos de almacenamiento de disco. Se debe poner un límite a los usuarios para que no tengan la habilidad de consumir espacio ilimitado de disco en un sistema. Es lo ideal para la configuración del sistema operativo base para que no exista un desorden en los usuarios ni grupos.

1. Para instalar quota se ejecuta el siguiente comando:

```
# yum install quota
```

2. Editar `/etc/fstab` y añadir `,usrquota,grpquota` en la partición `/` (`/dev/VolGroup00/LogVol00`):

```
# vi /etc/fstab
```

```
/dev/VolGroup00/LogVol00 /                ext3 defaults,usrquota,grpquota
1 1
LABEL=/boot          /boot                ext3 defaults 1 2
tmpfs                /dev/shm             tmpfs defaults 0 0
devpts               /dev/pts             devpts gid=5,mode=620 0 0
sysfs                /sys                 sysfs defaults 0 0
proc                 /proc                proc defaults 0 0
/dev/VolGroup00/LogVol01 swap                swap defaults 0 0
```

Imagen 3.15: Archivo de configuración de QUOTA

3. Luego se va a crear los archivos necesarios para que se aplique la cuota dentro del directorio `/`. Estos archivos son:

- `Aquota.user` (usuarios)
- `Aquota.group` (grupos)

Para crear archivos se usa el comando `touch`:

```
# touch laquota.user
```

```
# touch laquota.group
```

```
# touch /aquota.user /aquota.group
```

Se le cambiará de permisos a los archivos creados. Sólo se permitirá la lectura y escritura de estos sólo por el usuario local:

```
# chmod 600 /aquota.*
```

4. Luego se procede con el reinicio del equipo para que se apliquen los cambios que se ha hecho a /etc/fstab o simplemente remontar la partición con el siguiente comando:

```
# mount -o remount /
```

5. Se hace un chequeo completo por archivos, directorios en la partición que se ha configurado para cuotas:

```
# quotacheck -avugm
```

6. Finalmente se activa la cuota con el comando:

```
# quotaon -avug
```

Si tuviésemos muchos usuarios a los que quisiéramos editar las cuotas, se puede usar este pequeño script que por defecto cambiará las cuotas de todos los usuarios por las cuotas que tenga asignado un usuario determinado. En este caso el usuario es “Marco”:

```
# edquota -p Marco 'awk -F : '$3 > 499 {print $1}' /etc/passwd'
```

3.1.3.5 Instalación de servidor DNS (BIND9)

Se procede con la instalación:

```
# yum install bind-chroot
```

Después, se cambian los permisos y se inicia BIND:

```
# chmod 755 /var/named/
```

```
# chmod 775 /var/named/chroot/
```

```
# chmod 775 /var/named/chroot/var/
```

```
# chmod 775 /var/named/chroot/var/named/
```

```
# chmod 775 /var/named/chroot/var/run/
```

Definimos rwx (Read, Write, Execution) para todos los usuarios:

```
# chmod 777 /var/named/chroot/var/run/named/
```

```
# cd /var/named/chroot/var/named/
```

```
# ln -s ../../ chroot
```

```
# chkconfig --levels 235 named on
```

```
# /etc/init.d/named start
```

3.1.3.6 Instalación de MySQL (5.0)

Para instalar MySQL, se ejecuta el siguiente comando:

```
# yum install mysql mysql-devel mysql-server
```

Con este comando se ha instalado el servidor Mysql y algunas dependencias.

Por razones de seguridad se requiere que se le asigne una contraseña de root caso contrario cualquiera podrá acceder a la base de datos.

Para evitar este inconveniente se ejecutan los siguientes comandos:

```
# /usr/bin/mysqladmin -u root password 'nueva-contraseña'
```

```
# /usr/bin/mysqladmin -u root -h localhost.optimail 'nueva-contraseña'
```

Cambiar 'nueva contraseña' por la contraseña escogida.

3.1.3.7 Instalación de Postfix con SMTP-AUTH y TLS

Ahora se instala Postfix y Dovecot (Dovecot será nuestro servidor POP3/IMAP):

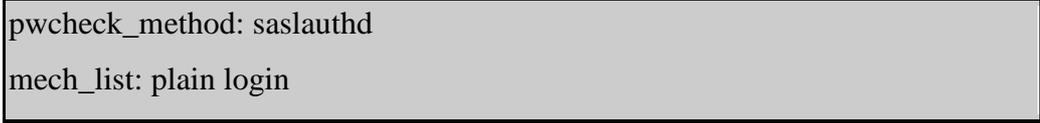
```
# yum install cyrus-sasl cyrus-sasl-devel cyrus-sasl-gssapi cyrus-sasl-md5 cyrus-sasl-plain postfix dovecot
```

Se configura SMTP-AUTH y TLS (no es necesario cambiar de path), para ello se usa el comando postconf que sirve para imprimir o cambiar los valores de los parámetros de configuración. En este caso se hará de la siguiente manera:

```
# postconf -e 'smtpd_sasl_local_domain ='
# postconf -e 'smtpd_sasl_auth_enable = yes'
# postconf -e 'smtpd_sasl_security_options = noanonymous'
# postconf -e 'broken_sasl_auth_clients = yes'
# postconf -e 'smtpd_recipient_restrictions =
permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination'
# postconf -e 'inet_interfaces = all'
# postconf -e 'mynetworks = 127.0.0.0/8'
```

Se edita /usr/lib/sasl2/smtpd.conf por lo que Postfix permite PLAIN y LOGIN de accesos. Al final se verá como esto:

```
# vi /usr/lib/sasl2/smtpd.conf
```



```
pwcheck_method: saslauthd
mech_list: plain login
```

Imagen 3.16: archivo smtp.conf

Después se crean certificados para TLS, para esto requiere la creación de una nueva carpeta (comando mkdir):

```
# mkdir /etc/postfix/ssl
```

Se cambia al directorio ssl:

```
# cd /etc/postfix/ssl/
# openssl genrsa -des3 -rand /etc/hosts -out smtpd.key 1024
# chmod 600 smtpd.key
# openssl req -new -key smtpd.key -out smtpd.csr
```

```
# openssl x509 -req -days 3650 -in smtpd.csr -signkey smtpd.key -out smtpd.crt
# openssl rsa -in smtpd.key -out smtpd.key.unencrypted mv -f smtpd.key.unencrypted
smtpd.key
# openssl req -new -x509 -extensions v3_ca -keyout cakey.pem -out cacert.pem -days
3650
```

Luego se configura Postfix para TLS con los siguientes comandos:

```
# postconf -e 'smtpd_tls_auth_only = no'
# postconf -e 'smtp_use_tls = yes'
# postconf -e 'smtpd_use_tls = yes'
# postconf -e 'smtp_tls_note_starttls_offer = yes'
# postconf -e 'smtpd_tls_key_file = /etc/postfix/ssl/smtpd.key'
# postconf -e 'smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.crt'
# postconf -e 'smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem'
# postconf -e 'smtpd_tls_loglevel = 1'
# postconf -e 'smtpd_tls_received_header = yes'
# postconf -e 'smtpd_tls_session_cache_timeout = 3600s'
# postconf -e 'tls_random_source = dev:/dev/urandom'
# postconf -e 'smtpd_sasl_authenticated_header = yes'
```

Luego se asigna un hostname en nuestra instalación de Postfix:

```
# postconf -e 'myhostname = server1.example.com'
```

Después de configurar se verá /etc/postfix/main.cf de esta manera:

```
# cat /etc/postfix/main.cf
```

```
queue_directory = /var/spool/postfix
command_directory = /usr/sbin
```

```
daemon_directory = /usr/libexec/postfix
mail_owner = postfix
inet_interfaces = all
mydestination = $myhostname, localhost.$mydomain, localhost
unknown_local_recipient_reject_code = 550
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
debug_peer_level = 2

debugger_command =
    PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin
    xxgdb $daemon_directory/$process_name $process_id & sleep 5

sendmail_path = /usr/sbin/sendmail.postfix
newaliases_path = /usr/bin/newaliases.postfix
mailq_path = /usr/bin/mailq.postfix
setgid_group = postdrop
html_directory = no
manpage_directory = /usr/share/man
sample_directory = /usr/share/doc/postfix-2.3.6/samples
readme_directory = /usr/share/doc/postfix-2.3.6/README_FILES
smtpd_sasl_local_domain =
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions=
    permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination
mynetworks = 127.0.0.0/8
smtpd_tls_auth_only = no
smtp_use_tls = yes
smtpd_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtpd_tls_key_file = /etc/postfix/ssl/smtpd.key
```

```
smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.crt
smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom
smtpd_sasl_authenticated_header = yes
myhostname = server1.example.com
```

Imagen 3.17: Archivo main.cf

Se inicia Postfix, saslauthd, y Dovecot:

Se procederá a apagar el servicio de sendmail ya que no es compatible con Postfix:

```
# chkconfig --levels 235 sendmail off
```

El servicio ha sido bajado en los niveles de ejecución (run levels) 2,3 y 5.

Requiere levantar el servicio de Postfix en los niveles de ejecución 2,3 y 5:

```
# chkconfig --levels 235 postfix on
```

Igual para saslauthd:

```
# chkconfig --levels 235 saslauthd on
```

Igual para Dovecot:

```
# chkconfig --levels 235 dovecot on
```

Con esto los servicios estarán activados permanentemente evitando que al reiniciar el servidor, los servicios no suban.

Para saber si SMTP-AUTH and TLS funciona correctamente se debe ejecutar el siguiente comando:

```
# telnet localhost 25
```

Después de establecer la conexión se digita “ehlo localhost”:

```
ehlo localhost
```

Debe aparecer

```
250-STARTTLS y
```

```
250-AUTH LOGIN PLAIN
```

Significando que todo está bien.

```
[root@server1 ssl]# telnet localhost 25
```

```
Trying 127.0.0.1...
```

```
Connected to localhost.
```

```
Escape character is '^]'.  
220 server1.example.com ESMTP Postfix
```

```
ehlo localhost
```

```
250-server1.example.com
```

```
250-PIPELINING
```

```
250-SIZE 10240000
```

```
250-VRIFY
```

```
250-ETRN
```

```
250-STARTTLS
```

```
250-AUTH PLAIN LOGIN
```

```
250-AUTH=PLAIN LOGIN
```

```
250-ENHANCEDSTATUSCODES
```

```
250-8BITMIME
```

```
250 DSN
```

```
Quit
```

```
221 2.0.0 Bye
```

Connection closed by foreign host.

```
[root@server1 ssl]#
```

Escribir:

Quit

Para regresar a la consola.

3.1.3.8 Maildir

Para configurar Maildir ejecutar los siguientes comandos:

```
# postconf -e 'home_mailbox = Maildir/'
```

```
# postconf -e 'mailbox_command ='
```

Finalmente se reinicia el servicio Postfix:

```
# /etc/init.d/postfix restart
```

3.1.3.9 Apache2 con PHP5

Ahora hay que instalar Apache con PHP5 (PHP 5.2.2):

```
# yum install php php-devel php-gd php-imap php-ldap php-mysql php-odbc php-pear  
php-xml php-xmlrpc curl curl-devel perl-libwww-perl ImageMagick libxml2 libxml2-  
devel
```

Se procede a editar dentro de /etc/httpd/conf/httpd.conf:

```
# vi /etc/httpd/conf/httpd.conf
```

Cambiar DirectoryIndex a

```
[...]  
DirectoryIndex index.html index.htm index.shtml index.cgi index.php  
index.php3 index.pl  
[...]
```

Imagen 3.18: Código parcial de archivo httpd.conf

Luego configurar para que el servicio inicie por defecto:

```
# chkconfig --levels 235 httpd on
```

Por ultimo iniciar Apache con el siguiente comando:

```
# /etc/init.d/httpd start
```

3.1.3.10 Habilitar PHP Globalmente

ISPConfig especifica cuál website ejecuta PHP scripts y cuál no. Esto funciona solo si PHP está deshabilitado caso contrario todos los websites ejecutarán scripts PHP.

Optimail necesita de los módulos de PHP por lo que requiere editar /etc/httpd/conf.d/php.conf y descomenta las líneas AddHandler y AddType en caso de que estén:

```
# vi /etc/httpd/conf.d/php.conf
```

```
#  
# PHP is an HTML-embedded scripting language which attempts to make it  
# easy for developers to write dynamically generated webpages.  
#
```

```
LoadModule php5_module modules/libphp5.so

#
# Cause the PHP interpreter to handle files with a .php extension.
#
AddHandler php5-script .php
AddType text/html .php

#
# Add index.php to the list of files that will be served as directory
# indexes.
#
DirectoryIndex index.php

#
# Uncomment the following line to allow PHP to pretty-print .phps
# files as PHP source code:
#
#AddType application/x-httpd-php-source .phps
```

Imagen 3.19: archivo php.conf

Después se reinicia Apache:

```
/etc/init.d/httpd restart
```

3.1.3.11 Instalación de módulos Perl

ISPConfig viene con SpamAssassin que necesita algunos módulos de Perl para trabajar.

Se instala con el siguiente comando:

```
# yum install perl-HTML-Parser perl-DBI perl-Net-DNS perl-Digest-SHA1
```

3.2 SEGURIDADES

Los mecanismos de seguridad son necesarios en cualquier herramienta de correo electrónico. Los ataques de computadores desconocidos, troyanos, virus y spam son comunes en la web. Afortunadamente en Linux existen varias herramientas que servirán de solución a este problema. Optimail usa Mailscanner y Clamav para la seguridad del sistema. La configuración e instalación se lo explicará más adelante.

A continuación se describen los casos más comunes en la seguridad web.

3.2.1 Intercambio de información

Cuando se intercambia información con un ordenador remoto, esa información circula por una serie de sistemas intermedios que son desconocidos a priori (excepto en ámbitos muy específicos). Además, no sólo no se sabe cuales serán estos sistemas intermedios, sino que además no se dispone de ningún control sobre ellos o sobre lo que puedan hacer con nuestros datos al pasar por ellos.

3.2.2 Instalación de software dañino involuntariamente

Otra posibilidad que no se debe descartar es que se instale software en un ordenador sin conocimiento del usuario o administrador.

3.2.3 Protección ante accesos no autorizados

Cuando se ofrecen servicios o información en una red para sus usuarios legítimos, al mismo tiempo se abre la puerta a posibles intrusos en estos sistemas.

3.2.3.1 Fallos de seguridad en la utilización del software

Se puede hacer un análisis agrupando los fallos de seguridad que se pueden dar en el software para ello primero daremos un concepto de lo que significa esta facilidad informática.

3.2.4 Software Libre

En general se entiende como software libre aquel programa o conjunto de ellos de los que el usuario puede disponer del código fuente, sin restricciones, y el cual puede modificar y redistribuir también sin restricciones.

3.2.4.1 Ventajas del Software Libre en el mundo de la seguridad

Si se analiza la descripción realizada previamente de la definición de software libre se derivan una serie de ventajas principales de este tipo de software sobre el software propietario, algunas de las cuales son muy adecuadas para el mundo de la seguridad. A saber:

Seguridad informática y software libre

- Al disponer del código fuente de los programas en su totalidad, éste puede ser analizado por terceras personas ajenas a sus autores en busca de fallos de diseño o de implementación.
- La posibilidad de realizar modificaciones libremente al código fuente y distribuir las permite que cualquiera pueda ofrecer mejoras sobre éste.
- Las características del software libre hacen que no sea lógico cargar costes sobre el software en sí (dado que se ha de distribuir sin cargo), lo que permite que este tipo de software pueda ser utilizado por organizaciones y personas con menos recursos económicos. Esto se presenta como una ventaja cuando se compara con los precios de lo

que cuesta el software de seguridad propietario hoy en día (licencias de cortafuegos, vpns, sistemas de detección de intrusos, etc

- De igual forma, la posibilidad de modificar libremente el software permite a las organizaciones que lo adapten a sus propias necesidades, pudiendo eliminar funcionalidades que no le sean de interés

Frente al análisis de fallos que puede sobrevenir en la realización del software (presentado anteriormente), el software libre protege a sus usuarios con una serie de mecanismos determinados. Entre estos:

- La posibilidad de una auditoria de código en las herramientas software reduce los riesgos de seguridad debido a la aparición de fallos desconocidos, a la introducción de funcionalidades no deseadas en el código o la incorrecta implementación de algoritmos públicos.
- La posibilidad de corregir los programas y distribuir dichas correcciones permite que los programas evolucionen de una forma más abierta. En el mundo de la seguridad, un fallo en el sistema significa exponer a éste a una “ventana de vulnerabilidad” que tiene lugar desde la detección del fallo (por parte de sus usuarios legítimos o de terceras partes, hostiles incluso)
- El hecho de que exista una cierta independencia entre el software y su fabricante, o distribuidor original, permite que los usuarios de este software, en caso de pérdida de soporte, puedan realizar el mantenimiento de éste ellos mismos o subcontratarlo a una tercera empresa.

3.2.4.2 Desventajas del software libre

Sin embargo, el uso de software libre no está exento de desventajas. Así se podrían enumerar las siguientes:

- La posibilidad de una generación más fácil de troyanos, dado que el código fuente también puede ser modificado con intenciones maliciosas. Si el troyano logra confundirse con la versión original puede haber problemas graves.
- Al no tener un respaldo directo, la evolución futura de los componentes software no está asegurada o se hace demasiado despacio.

Es posible hacer un análisis de los distintos productos y tecnologías de seguridad disponibles actualmente. Análisis realizados respecto a “El sistema operativo GNU/Linux y sus herramientas libres en el mundo de la seguridad: estudio, se muestra en la tabla adjunta. En ésta se muestra, de forma resumida, las distintas áreas, y la valoración que pueden recibir las soluciones dividiendo en software libre y software propietario.

La calificación está hecha de una forma, en gran medida, subjetiva, en base a diferentes apreciaciones. Para esta calificación se ha utilizado una nota expresada de la A (mejor) a C (peor). Una A significa que un área está muy desarrollada, una B que implementa la funcionalidad suficiente para ser operativa (pero no capacidades que la puedan convertir en una tecnología plenamente desarrollada) y una C que aún está en desarrollo. Para tener una mayor flexibilidad en la calificación se han añadido '+' y '-' indicando una mejora, o degradación, dentro de una misma calificación.

	Área Situación sw libre	Situación sw propietario
Sistema operativos	A	A
Aplicaciones finales	A	A
Cortafuegos personales	B+	A
Cortafuegos de filtrado	A	A+
Cortafuegos de aplicación	C	A
Herramientas de Auditoría Externa	A+	A
Herramientas de Auditoría Interna	B	B
Detección de intrusos	A	A
Sistemas de autenticación	A	A
Firma digital	A	A
Autoridades de certificación	C+	A
Comunicaciones cifradas	B+	A
Alta disponibilidad	C+	A+

Tabla 3.1: Comparativa de la situación actual del software libre en el área de la seguridad

En base a estudios realizados al utilizar software libre existe seguridad al hacerlo, pues depende. Que, aunque pueda parecer una respuesta ambigua, está en realidad suficientemente fundamentada.

En la tabla, sin embargo, se puede ver que, en determinadas áreas, hoy por hoy, no es viable basar una solución de seguridad en software libre y va a ser necesario acudir a soluciones propietarias por estar el primero en una etapa aún inmadura de desarrollo.

Sin embargo, sí que es posible adivinar que el software libre está, en determinadas áreas, compitiendo codo con codo con las soluciones propietarias existentes. La situación ha ido cambiando a medida que las distintas soluciones desarrolladas se han demostrado competitivas y han ido siendo aceptadas por el público general. Esta aceptación ha dado lugar a un desarrollo exponencial en el que se pueda esperar que, en aquellas áreas en las que el software libre aún no alcanza al software propietario, la situación llegue a igualarse (e incluso invertirse) pasado un cierto tiempo.

Por otro lado, independientemente del ritmo de crecimiento del software, del lado de la seguridad, las ventajas ofrecidas por el software libre son evidentes frente a las alternativas propietarias. Máxime en determinados entornos en los que una persona no se puede “fiar” de aquella compañía que le vende la solución o no puede depender de la seguridad “garantizada” por un determinado producto que no tiene forma de demostrar.

Seguridad informática y software libre.

Por tanto, si bien el software libre en la actualidad tiene una cobertura desigual de las distintas necesidades de seguridad de una empresa o corporación, éste es, definitivamente, una apuesta de futuro provechosa en aquellas áreas aún no desarrolladas y una oportunidad real e inmediata en las demás áreas para utilizar soluciones equivalentes a las propietarias con:

- Un menor costo
- Unas mayores garantías de seguridad, debido a la posibilidad de auditar el código en uso
- Una mayor flexibilidad en la adaptación e integración, gracias a la posibilidad de modificar dicho código.
- La posibilidad del mantenimiento asegurado de una solución de seguridad con independencia del origen del producto en sí.

3.4.4.3 Instalación de Mailscanner y Clamav

Se obtiene primero el archivo de la página web descargándolo del browser o usando el siguiente comando:

```
# wget http://www.mailscanner.info/files/4/rpm/MailScanner-4.69.9-3.rpm.tar.gz
```

Luego se lo descomprime:

```
# tar zxvf MailScanner-4.69.9-3.rpm.tar.gz
```

Ahora hay que entrar a la carpeta descomprimida:

```
# cd MailScanner-4.69.9-3
```

Finalmente se procede con la instalación, en este caso poner:

```
# sh install.sh
```

El siguiente es Clamav. Afortunadamente se lo puede instalar usando simplemente el comando yum:

```
# yum -y install clamav clamav-update clamav-data-empty
```

Ahora se procede con la configuración editando el archivo MailScanner.conf:

```
# vi /etc/MailScanner/MailScanner.conf
```

En la línea:

```
%report-dir% = /etc/MailScanner/reports/en ##
```

Se cambia por "es" para recibir reportes en Español.

Ahora se configura el Anti-Virus que se va a utilizar:

```
Virus Scanners = none ##
```

Se lo cambia por "clamav" (si se tiene más de un antivirus se puede poner la opción "auto".)

La configuración del Spam se hace en las siguientes líneas:

Spam Checks = yes

Spam List = ##

Se añade en Spam List la siguiente cadena;

"ORDB-RBL SBL+XBL spamcop.net NJABL SORBS"

Max Spam Check Size = 200k ###

Con este valor se tuvieron muchos problemas, se lo cambió por 100mb y se solucionó.

CAPITULO IV. DESARROLLO

4.1 METODOLOGÍA DE DESARROLLO

4.1.1 Metodología OO-Method

La metodología aplicada para el desarrollo de este proyecto es la “metodología OO-METHOD”¹ basado en el “Método OOWS”².

OO-Method es una metodología de producción de software basada en el paradigma de la programación automática y en los principios del modelo Orientado a Objetos al igual que permite describir aplicaciones de forma abstracta mediante el uso de Modelos Conceptuales orientados a objetos.

La metodología como el lenguaje ha sido desarrollada por el grupo de investigación de programación lógica e ingeniería del software del departamento de sistemas informáticos y computación de la Universidad Politécnica de Valencia.

La característica principal de esta metodología consiste en que los esfuerzos de los desarrolladores están centrados en los pasos del modelado conceptual, donde los analistas capturan los requerimientos del sistema. Posteriormente, podemos obtener una implementación completa de forma automática siguiendo un modelo de ejecución preciso (que incluye estructura y comportamiento).

La definición de un problema puede suceder con independencia de cualquier reificación concreta. OO-Method es una metodología de desarrollo de software basada en una clara separación del Espacio del Problema (el 'qué') del Espacio de la Solución (el 'cómo'). La definición de un problema (la descripción abstracta de una aplicación, representada en

¹ PASTOR, O (2004) MDA, OO-Method

² VALDERAS, P. (2002). *Análisis, Diseño e Implementación de un Portal Web para un Departamento Universitario*. Valencia.

el correspondiente Modelo Conceptual), puede suceder con independencia de cualquier reificación concreta (implementación concreta de una solución). Esto posiciona a OO-Method como una metodología para implementar herramientas que sigan las directrices MDA de separación de la lógica de las aplicaciones de las posibles implementaciones de las mismas.

El formalismo que subyace a OO-Method es OASIS, un lenguaje formal y orientado a objetos para la especificación de sistemas de información. Este marco formal proporciona una caracterización de los elementos conceptuales requeridos para especificar un sistema de información.

Sus dos componentes principales son: Modelo Conceptual y el Modelo de Ejecución.

4.2 EL MODELO CONCEPTUAL

El **Modelo Conceptual** está dividido en cuatro vistas complementarias:

- Modelo de objetos
- Modelo dinámico
- Modelo funcional
- Modelo navegacional

Estos modelos pueden ser usados conjuntamente como entrada para entornos de generación de código. Como resultado se obtiene el método de producción automática de aplicaciones software para ambientes web llamado OOWS.

Estas cuatro vistas permiten que todos los aspectos funcionales de una aplicación puedan ser descritos de forma abstracta por medio de una serie de elementos conceptuales (a los que también nos referimos como primitivas conceptuales o patrones conceptuales) que tienen una semántica precisa.

El Modelo de Ejecución establece las reglas de transformación de un Modelo

Conceptual (sistema especificado) a su representación software correspondiente en una plataforma tecnológica concreta, de manera que se preserve la semántica de los elementos del Modelo Conceptual y se asegure que la aplicación resultante es funcionalmente equivalente a dicho modelo.

Un Compilador de Modelos puede implementar el conjunto de transformaciones que relacionan los patrones conceptuales con representaciones software, asumiendo que cualquier decisión de diseño e implementación tiene su correspondiente contrapartida que puede ser obtenida y adecuadamente representada a un mayor nivel de abstracción (Modelo Conceptual).

4.2.1 Estado del arte

“En el área de desarrollo de aplicaciones web se han introducido un gran número de iniciativas para la creación de soluciones web al diseño clásico de software. Han sido presentadas docenas de métodos y aproximaciones para abordar la construcción de estas aplicaciones pero no existen todavía métodos establecidos que permitan modelar los nuevos requisitos impuestos por las tecnologías emergentes”³.

Los métodos pueden ser clasificados en dos enfoques principales:

- Métodos Hipermediales de Desarrollo.
- Métodos Orientados a Objetos que extienden UML.

4.2.2 Métodos Hipermediales

De los métodos hipermediales se puede destacar:
HDM, RMM, EORM, OOHDM, Araneus y WebML.

³ VALDERAS, P. (2002). *Análisis, Diseño e Implementación de un Portal Web para un Departamento Universitario. Valencia.*

Los métodos hipermediales generalmente parten del modelo Entidad-Relación (ER) para describir un conjunto de tipos entidad y sus relaciones.

También extienden métodos hipermediales de desarrollo añadiendo índices más elaborados. Su diseño puede empezar a partir de entidades finalizando con el diseño de la interfaz de usuario e implementación.

4.2.3 Métodos orientados a objetos que extienden UML

Estos métodos son similares a otros métodos de diseño de aplicaciones web. Se propone una extensión de UML para modelar navegación e interfaces de usuario.

Según autores consideran el diseño de tres modelos:

- Modelo conceptual
- Navegacional
- Presentación

El modelo conceptual consiste en un diagrama de clase que identifica el objeto de dominio del problema y sus relaciones.

El modelo navegacional describe la estructura de la navegación por el diagrama de clase especificando qué nodos navegacionales se definen.

El modelo de presentación describe la interfaz abstracta del usuario a través de objetos compuestos y su comportamiento dinámico por diagramas de estado. Se construye cada modelo usando las notaciones proporcionadas por UML.

4.3 MÉTODO OOWS

Cada vez es más importante la necesidad de construir sistemas fiables, utilizables y adaptables.

La Ingeniería Web hoy en día aplica un enfoque genérico que se suaviza con estrategias, tácticas y métodos especializados al igual ofrece varios métodos para la construcción de aplicaciones web de calidad, de fácil mantenimiento y con el menor esfuerzo posible por consiguiente se ha desarrollado el método OOWS el que se lo define como un método orientado a objetos para la construcción de aplicaciones web en un ambiente de generación automática de código.

OOWS esta basado en OO-Method y siguiendo una serie de etapas pretende obtener aplicaciones web de forma automática a partir del modelado conceptual realizado de dicha aplicación. En la primera etapa se realiza una búsqueda de requisitos, tanto de funcionalidad como de usuarios, en la segunda etapa se realiza el modelado conceptual del sistema y por último, en una tercera etapa se realiza la implementación, que en este caso, tratándose de un método para la generación automática de código, bastará con pasarle el modelado conceptual a un generador para que nos construya la aplicación web.

4.3.1 Fases del Método OOWS

Las fases del método orientado a objetos OOWS son:

- * Búsqueda de requisitos
- * Modelado conceptual
- * Obtención del sistema.

La búsqueda de requisitos es la primera fase del método orientado a objetos OOWS, la que se refiere a qué tipos de usuarios podrán interactuar y cual será la funcionalidad correspondiente con el sistema.

La búsqueda de requisitos presenta tres modelos:

- Diagrama funcional
- Diagrama de agentes

- Diagrama de asociación funcional.

El modelado conceptual es la segunda fase en la que se refiere a la construcción de dos modelos:

- Modelo de objetos
- Modelo navegacional

Los que satisfacen los requisitos especificados.

Por último se tiene la obtención final del sistema.

4.3.2 Requisitos del Método OOWS

Los requisitos agrupan en tres diagramas con el objetivo de de asociar a los usuarios con la funcionalidad correspondiente:

- Diagrama Funcional
- Diagrama de Agentes
- Diagrama de Asociación Funcional

4.3.2.1 Diagrama Funcional⁴

El Diagrama Funcional es un modelo que se representa únicamente la información referente a la funcionalidad del sistema, se utiliza la notación siguiente:

* Funcionalidad de granularidad mínima: representa aquellos servicios básicos que nos ofrece el sistema y que no se los puede dividir en otros subservicios.

⁴ VALDERAS, P. (2002). *Análisis, Diseño e Implementación de un Portal Web para un Departamento Universitario*. Valencia

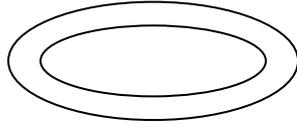


Fig 4.1 Funcionalidad mínima

* Funcionalidad externa: representa los servicios que no pertenecen al sistema y que son enviados desde otros sistemas.



Fig 4.2 Funcionalidad

- Agrupación de funcionalidades: Es la que permite agrupar aquellos servicios básicos, servicios externos u otras agrupaciones que están relacionadas funcionalmente.

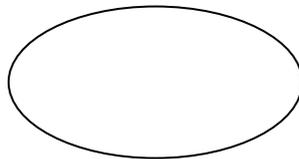


Fig 4.3 Agrupación de funcionalidades

- Descomposición Funcional: Permite establecer una relación entre la agrupación funcional y sus funcionalidades.



Fig 4.4 Relación entre

4.3.2.2 Diagrama de Agentes⁵

En este modelo se capta la información referente a qué actores (o usuarios) son relevantes para el sistema, qué relaciones existen entre ellos y cuales pueden interactuar directamente con el sistema. Para ello vamos a utilizar la siguiente notación:

*Agente anónimo. No necesita identificación para entrar al sistema:

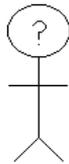


Fig 4.5 Nombre anónimo

- Agente sin permiso de acceso:



Fig 4.6 Nombre del usuario sin permiso de acceso

- Agente con permiso de acceso. Necesita identificarse al sistema:



Fig 4.7 Nombre del usuario con permiso de acceso

⁵ VALDERAS, P. (2002). *Análisis, Diseño e Implementación de un Portal Web para un Departamento Universitario*. Valencia

- Relación de herencia entre agentes:



Fig 4.8 Herencia entre usuarios

Con esta primitiva agente hereda la funcionalidad de su usuario “padre”. Esta herencia implica que toda funcionalidad ala que un usuario puede acceder, su especializado también.

4.3.2.3 Diagrama de Asociación funcional

En este modelo se asocia la funcionalidad detectada en el modelo funcional a los actores especificados en el modelo de agentes. Representa la intersección de los dos diagramas anteriores.

Mediante la primitiva siguiente se relaciona dichos actores con sus funcionalidades correspondientes.



Fig 4.9 Asociación funcional

La adquisición de funcionalidad tiene como origen a un actor y como destino a otro actor o a cualquier representación de funcionalidad.

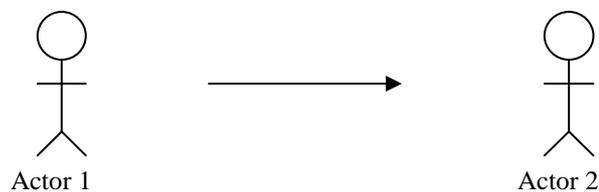


Fig 4.10 Adquisición de funcionalidad

4.3.3 Modelado Conceptual⁶

En esta fase se construyen dos modelos:

- Modelo de Objetos
- Modelo Navegacional

4.3.3.1 Modelo de Objetos

Esta clase nos permite acceder a objetos (clases) y nos permite utilizar en el modelo de navegación que propone OOWS. Este modelo presenta la notación que se describe a continuación (Tabla 2.1) será utilizada para representar la información del modelo de objetos:

Nombre clase
Atributos
Operaciones

Tabla 4.1 Modelo de Obietos

4.3.3.2 Modelo Navegacional

El modelo navegacional presenta las siguientes tipos de notación para captar toda la información que ha de ofrecernos:

- Subsistema de exploración o de secuencia: Ya sea de exploración o de secuencia, se utiliza para definir una agrupación de contextos u otros subsistemas, cuyos contenidos se encuentran estrechamente relacionados.

⁶ VALDERAS, P. (2002). *Análisis, Diseño e Implementación de un Portal Web para un Departamento Universitario*. Valencia

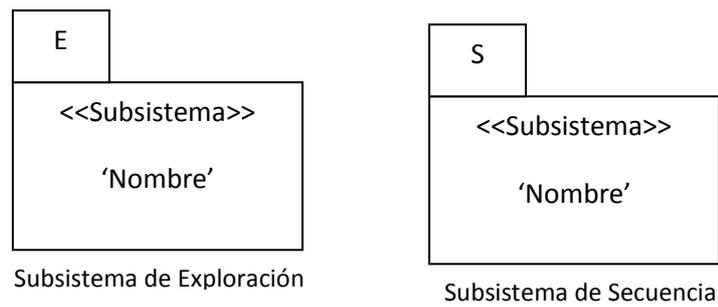


Fig.4.11 Modelo Navegacional

- Sistema externo de exploración o de secuencia: Se utiliza para representar el acceso a un sistema totalmente independiente al sistema que se está modelando ya sea de exploración o de secuencia.

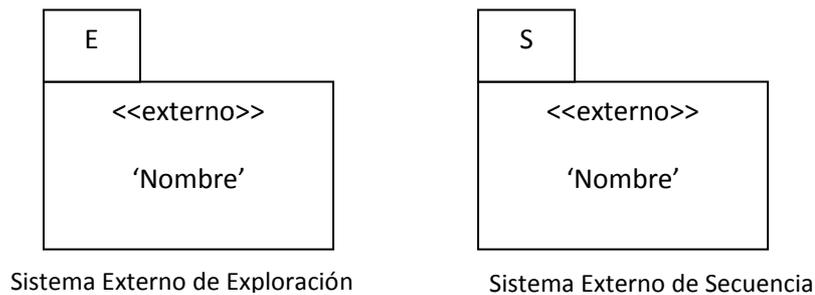


Fig.4.12 M.Navegacional Sistema Externo

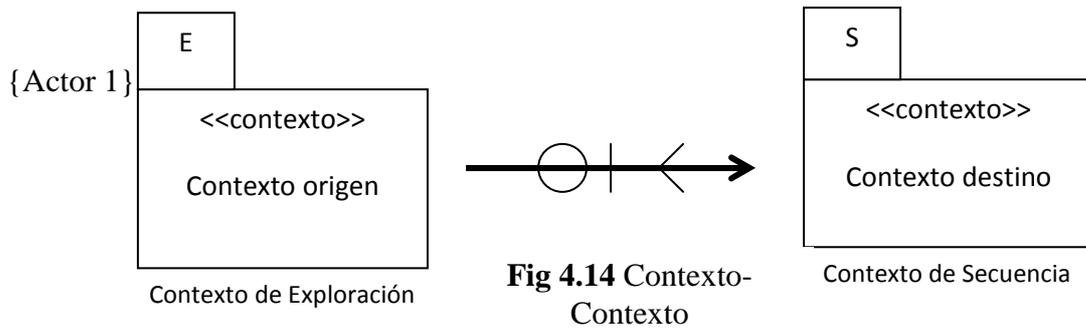
- Cambio de rol: Mediante esta notación se representa el hecho de que un usuario pueda convertirse en otro, previa identificación.

{Actor 1}



Fig.4.13 Cambio de Rol

- Contexto – Contexto: En este caso, se esta modelando el hecho de que en el contexto ‘Contexto origen’ se nos permite acceder al contexto ‘Contexto destino’ del mapa navegacional del actor ‘Actor 1’, siempre y cuando nos identifiquemos como este.



4.4 TECNOLOGÍA DE DESARROLLO

Postfix es un programa de correo electrónico al cual configuraremos como MTA o Mail Transfer Agent. Fue escrito por Wietse Venema, y desarrollado bajo la Licencia Pública de Software de IBM.

Como dijimos antes, Postfix no actúa de POP3, ni WebMail server, ni posee usuarios/contraseñas en él, sino que actúa de intermediario entre Internet y el Servidor de Correo IMail. Creemos posible utilizar cualquier otro Servidor de Correo, ya que lo que se hace es modificar en dicho Servidor, quién será el 'Gateway' que entregará el correo a Internet.



Fig 4.15. Servidor de correo

Un servidor de correo permite:

- Enviar correo al exterior. Servidor de correo saliente. (SMTP)
- Recibir correo del exterior. Servidor de correo entrante (IMAP, POP, Local)

Postfix permite el recibir correo y enviarlo al usuario de linux.

Postfix es nuestro sustituto preferido para Sendmail. Postfix es más rápido que Sendmail, tiene una arquitectura más segura y modular y ofrece muchas funciones requeridas por un proveedor de grandes volúmenes de correo. Postfix no soporta protocolos obsoletos, sino que usa el Protocolo Simple de Transporte de Correo (SMTP), y es el que tiene el menor número de incidencia en la lista CVE. Por todas estas razones, usaremos Postfix en lugar de Sendmail como MTA.

Copiar mails enviados y recibidos con Postfix

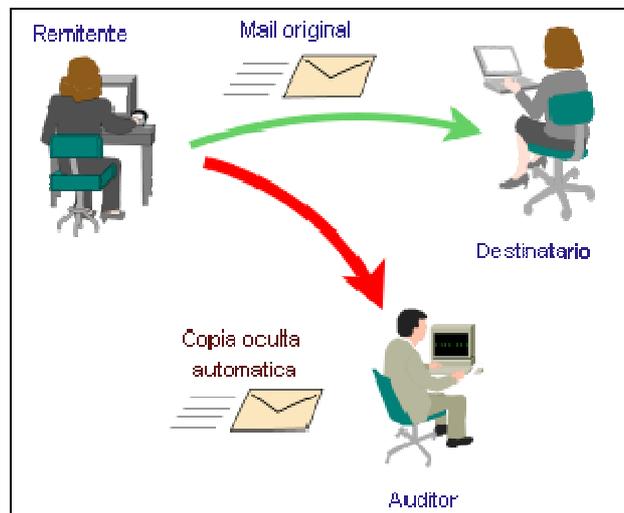


Fig 4.16 Envío de Correo

4.5 CARACTERÍSTICAS DEL SISTEMA

Para el desarrollo de la tesis o el tema propuesto se procedió a la instalación previa de su ambiente de funcionamiento o plataforma en la que estará soportado el sistema, para esto se procedió la instalación del siguiente software:

Fedora 7.0 .- Sistema operativo de software libre que nos proveerá todos los servicios para el funcionamiento de nuestro servidor de correo.

Postfix .- Servidor de correo con funcionalidad estándar que nos proveerá del servicio de correo electrónico.

Dovecot .- Es una fuente de servidor IMAP y POP3 para Linux/UNIX, escrita principalmente pensando en la seguridad.

Apache .- Servidor http para la publicación de nuestro aplicativo de correo electrónico en internet.

PHP.- Lenguaje de programación que nos permitirá crear nuestro aplicativo cliente de correo.

MYSQL .- Base de datos que se utilizará para el registro de usuarios, perfiles y todos los datos que intervienen en nuestro aplicativo de correo.

Características Principales

Una vez instalado el ambiente del servidor de correo se procede a la programación y puesta en marcha del sistema cliente para la administración de los correos con las siguientes características:

- 1.- El sistema manejará usuarios y perfiles de acceso para administrar los correos individuales.
- 2.- Optimail proveerá de varios módulos con los que se puede crear, borrar, reenviar correos.
- 3.- El sistema provee de un ambiente gráfico amigable y de fácil uso.
- 4.- Intercambio de mensajes locales y remotos.
- 5.- Transmisión de texto con estilos.
- 6.- La utilización de formas electrónicas especiales para envío y recepción de mensajes.
- 7.- Creación de cuentas PostOffice para que un usuario pueda ingresar a la aplicación.
- 8.- Posibilidad de adjuntar un archivo.

El sistema está completamente diseñado en software libre con módulos nativos de Linux mediante funciones IMAP controlados por PHP.

4.6 HERRAMIENTAS DE DESARROLLO

En el desarrollo de esta aplicación se utilizaron varias herramientas tanto en el desarrollo como en la instalación del sistema cliente.

Herramientas de instalación

Linux permite la instalación fácil obteniendo los archivos de un servidor mediante internet, para esto se utilizó el comando YUM

YUM.-

Es una herramienta de software libre de gestión de paquetes para sistemas Linux basados en RPM. Fue desarrollada principalmente para actualizar y controlar los sistemas Red Hat. Ha sido adoptada por Fedora, CentroOS, y otras distribuciones de

GNU/Linux basadas en RPM. Con los paquetes "yum-updatesd" o "yum-updateonboot" se puede hacer una actualización de software automática.

Herramientas de desarrollo

Para el diseño de nuestro sistema se utilizó un editor gráfico y un editor PHP de Macromedia. Fireworks , automatiza el proceso de producción de gráficos web y crea múltiples gráficos a partir de una sola fuente de datos y gana tiempo para lograr diseños integrales y racionalizados.

Macromedia Fireworks

Diseña, optimiza e integra gráficos web. Permite crear, optimizar y exportar gráficos interactivos en un entorno único y centrado en el web de una manera fácil. Ofrece capacidades de edición de mapas de bit de calidad profesional en un solo entorno centrado en el Web.

Con esta herramienta se creó la parte gráfica de nuestro sistema Optimail lo que incluye el gráfico y los colores corporativos.

Dreamweaver

Macromedia Dreamweaver facilita tarea de realizar sitios webs complejos y bien estructurados el cual utilizamos principalmente como un editor PHP para el desarrollo de los módulos y subaplicaciones de Optimail en un entorno gráfico.

4.7 CASO DE USO

4.7.1. Correo (Principal)

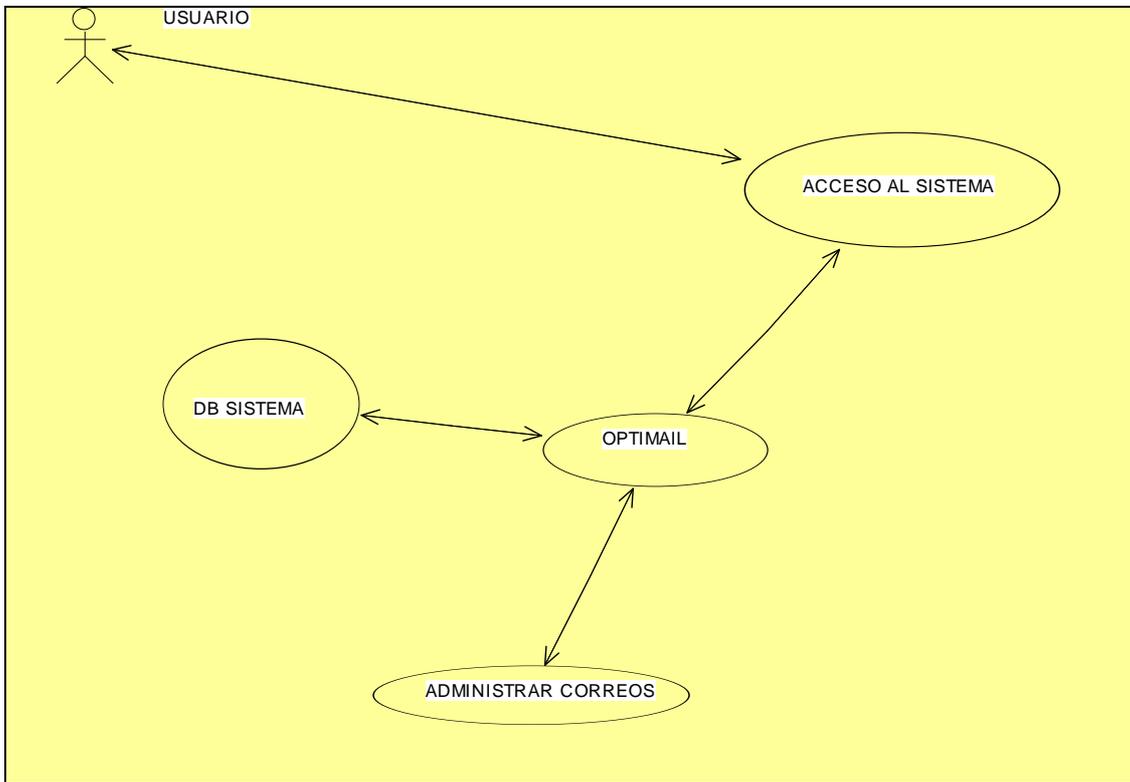


Fig.4.17 Correo (Principal)

El usuario ingresa al sistema (PHP/Postfix) por medio de una pantalla de inicio de sesión. Al autenticarse con éxito Optimail empieza su funcionamiento interactuando con la base de datos de MYSQL recuperando información de los usuarios registrados y los buzones creados en Postfix para empezar la administración de mensajes de correo electrónico.

4.7.2. Acceso

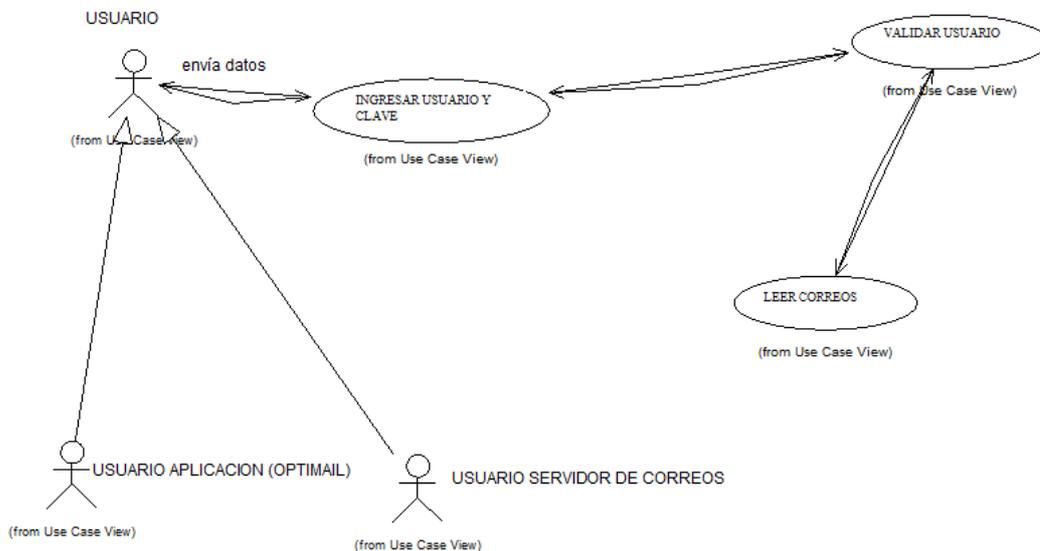


Fig 4.18. Acceso al sistema

Al usuario se lo puede representar por dos subcategorías. Optimail funciona enlazando estos dos usuarios mostrados en la imagen. El usuario de aplicación se encuentra en la base de datos de Optimaildb en phpmyadmin mientras que el usuario de servidor de correos es un usuario creado en Linux que crea en /home/nombredeusuario usando Dovecot y Postfix para su funcionamiento. Enlazando cada uno se obtiene un usuario común que accede a nuestro sistema Optimail. Los datos son validados y finalmente se desplegará una pantalla al usuario en la que se podrán leer los correos y demás (módulo de administración).

4.7.3 Crear Cuentas

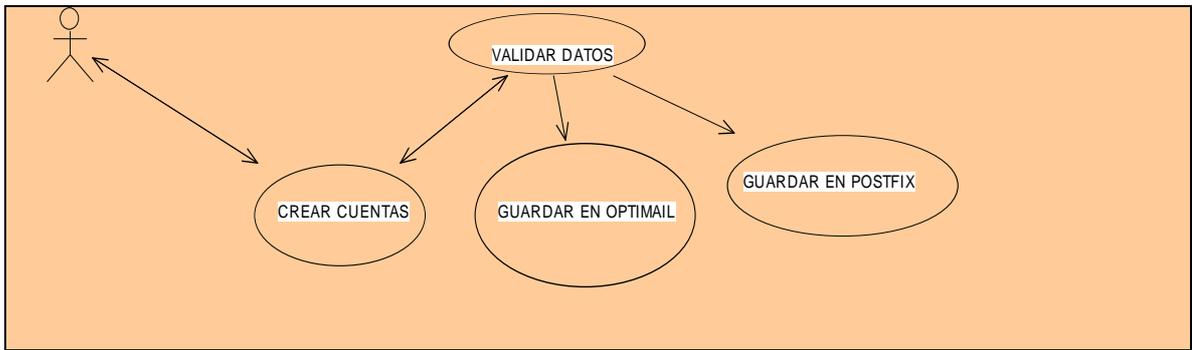


Fig 4.19. Crear Cuentas

Al crear cuentas de correo el usuario ingresa los datos y después el sistema validará los datos. El almacenamiento se hará de dos formas:

- Base de datos de Optimail. (Optimaildb)
- En Servidor Postfix

4.7.4. Administrar Correo

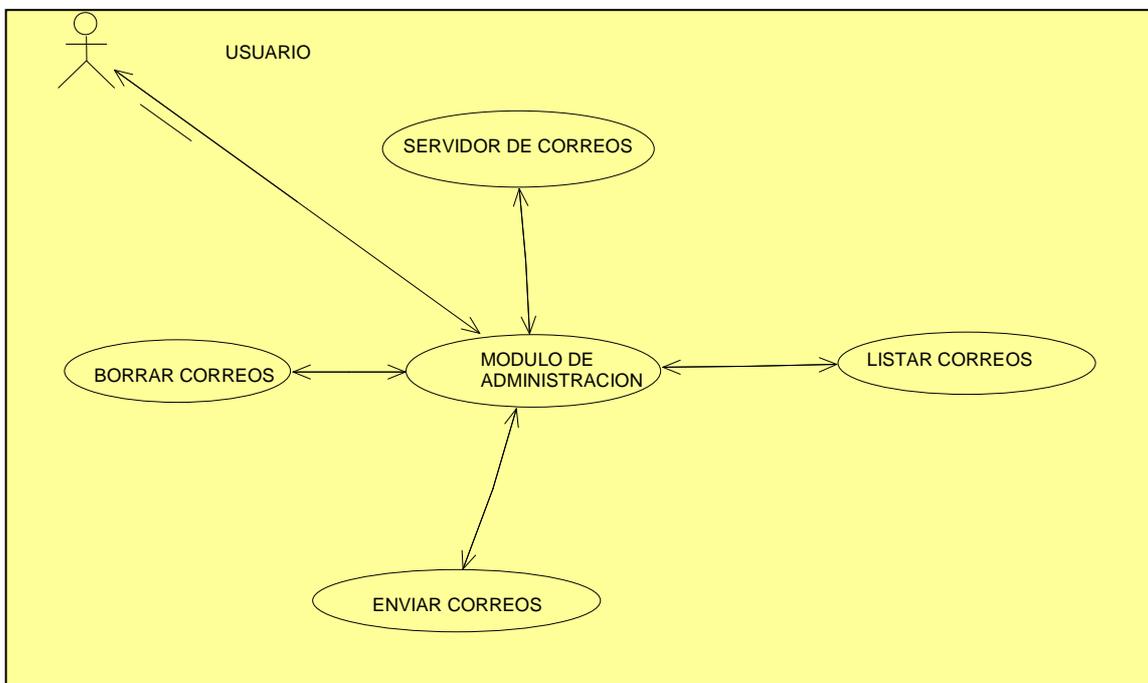


Fig 4.20 Administrar correo

El usuario interactúa con el módulo de administración que es clave para el funcionamiento del listado, borrado y el envío de correo. Todos los módulos son realizados en php. El módulo de administración interactúa lógicamente con el servidor de correos en este caso se usa Dovecot junto con Postfix.

4.8 DIAGRAMA DE SECUENCIAS

4.8.1 Crear Cuentas

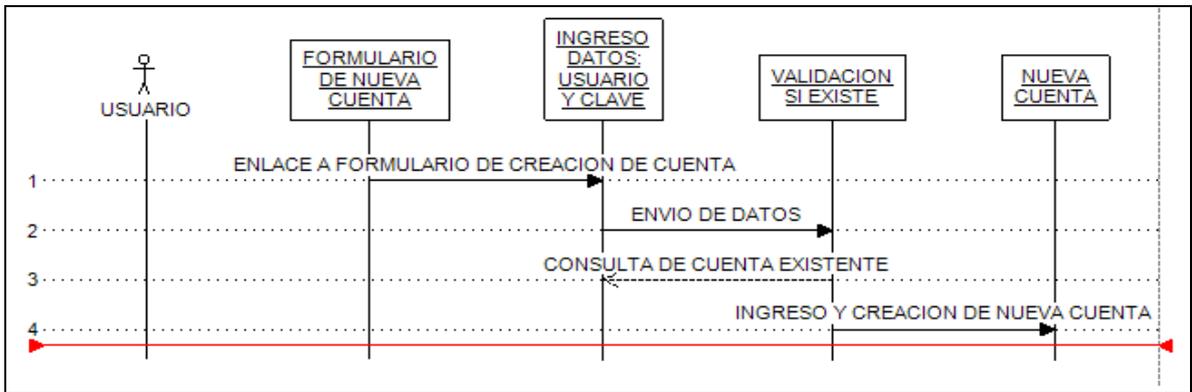


Fig 4.21 Diagrama de secuencia: crear cuentas

En la pantalla de inicio se puede crear un nuevo usuario. Al hacer click en el link, un formulario pequeño aparecerá. Tal formulario deberá ser llenado por el usuario, sin embargo, el sistema validará los datos y buscará en la base de datos si el determinado usuario existe. En el caso de que ocurra, el sistema desplegará un mensaje indicando que el usuario existe y se tendrá que ingresar un nombre de usuario diferente.

4.8.2. Acceso al Sistema

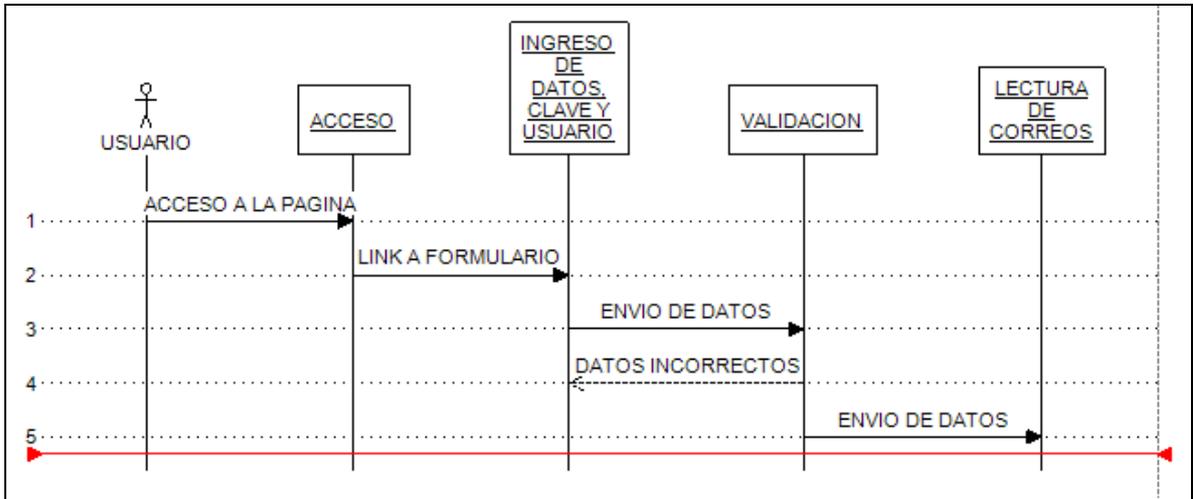


Fig 4.22. Diagrama de secuencia: Acceso al sistema

Al acceder a Optimail el usuario obtiene acceso a la página de inicio del cual debe ingresar el nombre de usuario y contraseña. Luego el sistema valida y autentica los datos. En caso de ingresar los datos correctos, el sistema procederá a la siguiente pantalla la cual corresponde a la lectura de correos, caso contrario si falla la autenticación, volverá a pedir los datos.

4.8.3 Administración de Correos

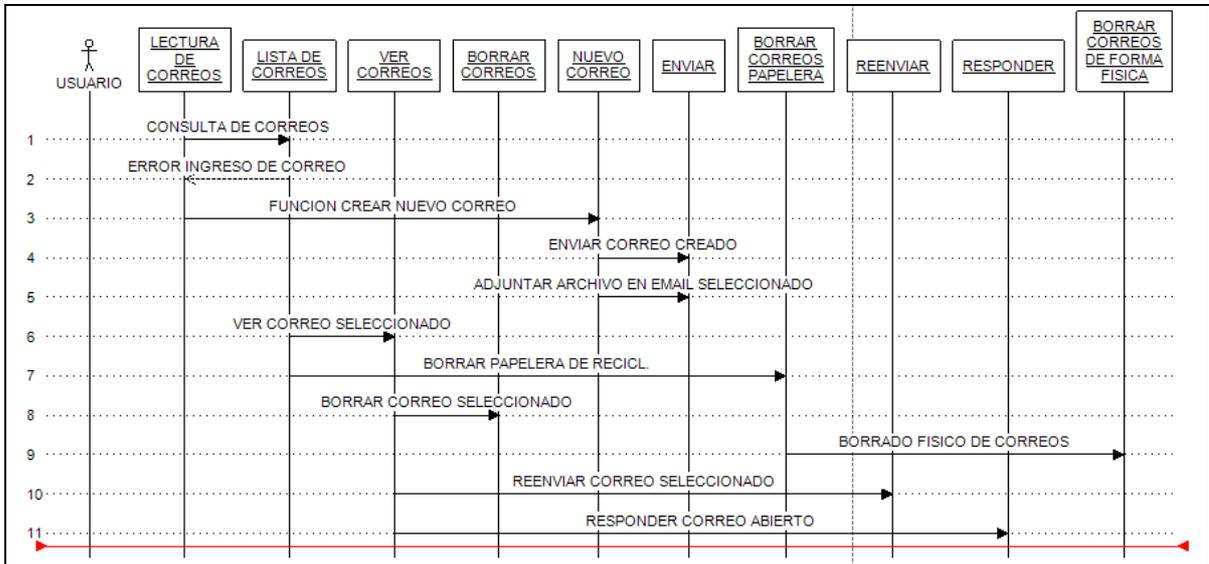


Fig 4.23. Diagrama de secuencia: Administración de correos

Una vez ingresado al sistema se desplegará una pantalla donde se podrá administrar el correo electrónico. La administración de correo electrónico consiste en:

- Lectura de Correos
- Lista de Correos
- Leer Correos
- Borrar Correos
- Crear Correos
- Enviar Correos
- Borrar correos (de la papelera)
- Reenviar Correos
- Responder Correos

El usuario accede a la página de administración de correos y desde ahí puede ver la lista de correos. Optimail enviará un mensaje de error en el caso de que no se pueda acceder al buzón eso ocurrirá en el caso de existir problemas en la red interna local.

En la lista de correos que hay en cada buzón el usuario puede leer el mensaje de correo electrónico y también se lo puede eliminar directamente a la papelera (la bandeja se llama “BORRADO”). Tal como se muestra en la imagen un correo puede eliminarse sin la necesidad de leerlo. Finalmente las últimas opciones de OptiMail en la lectura de correo son responder y reenviar.

Un correo enviado a la papelera puede ser borrado definitivamente (físicamente), así mismo si existe un conjunto de mensajes electrónicos en esa carpeta pueden eliminarse vaciando el buzón o bandeja. Cuando el usuario accede a nuevo correo podrá enviar el mensaje redactado y ese mensaje podrá tener un archivo adjunto (máximo de 1mb, cualquier extensión).

4.9 DIAGRAMA DE CLASES

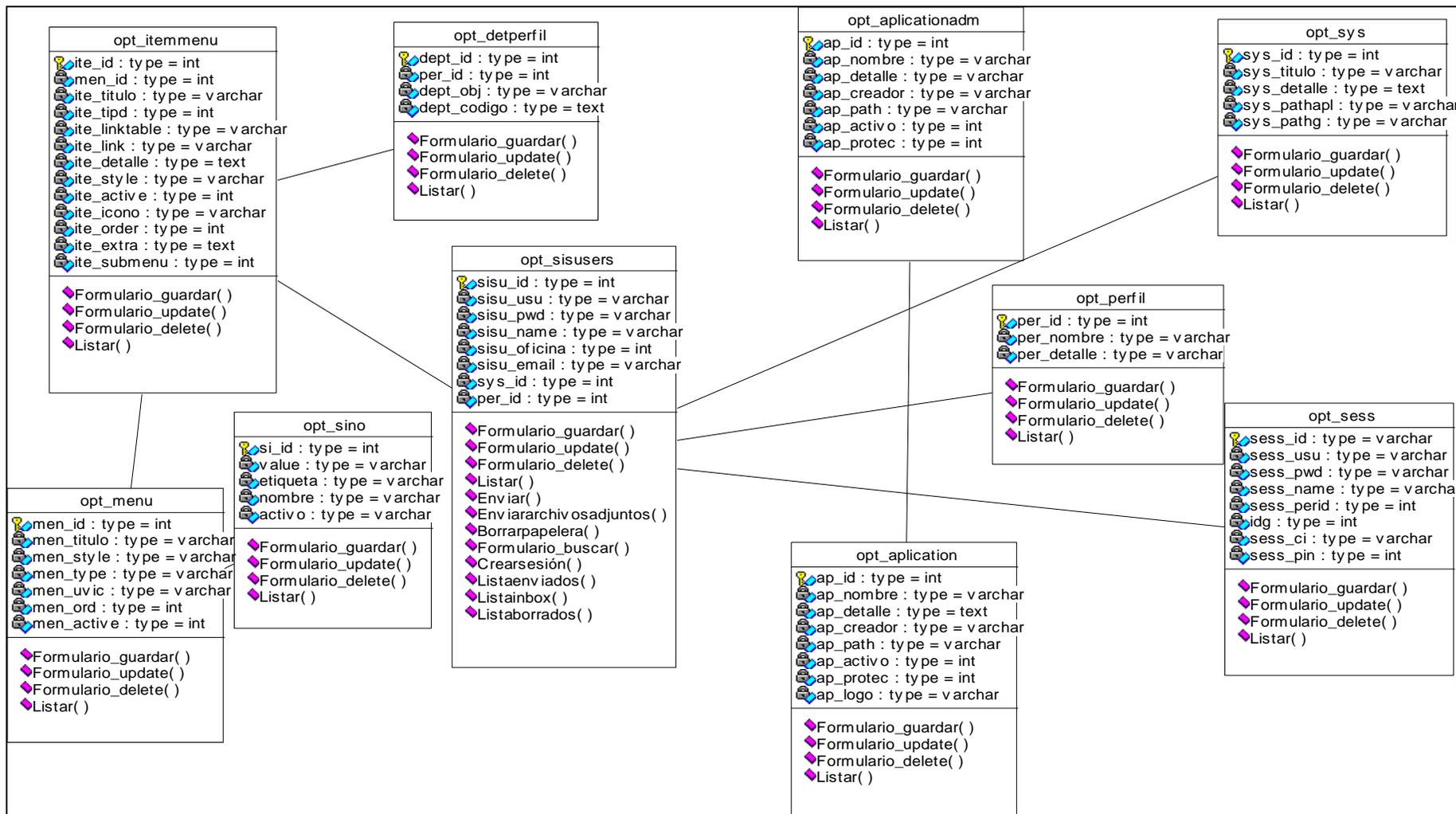


Fig 4.24 Diagrama de Clases

Las clases utilizadas en Optimail son:

- OPT_ITEMMENU
- OPT_MENU
- OPT_DETPERFIL
- OPT_SINO
- OPT_SISUSERS
- OPT_APLICACIONADM
- OPT_APLICACION
- OPT_PERFIL
- OPT_SYS
- OPT_SESS

Cada clase contiene funciones que se enlazan mutuamente y se relacionan con cada clase. Las funciones utilizadas son:

- Formulario_guardar(): Esta función guarda la información ingresada en un formulario dentro de la base de datos.
- Formulario_update(): Actualiza la información ingresada desde un formulario dentro de la base de datos
- Formulario_delete(): Elimina la información dentro de la base de datos.
- Formulario_buscar(): Función que realiza una consulta dentro de la base de datos
- Listar(): Función de listado de buzones o bandejas.
- Enviar(): Función que permite el envío de mensaje de correo electrónico.
- EnviarArchivosAdjuntos(): Función que permite el envío de archivos adjuntos.
- BorrarPapelera(): Función que borra los correos físicamente (de la papelera)
- CrearSesión(): Función de creación de sesión.
- ListaEnviados(): Función de listado de correo del buzón “Enviados”
- ListaInbox(): Función de listado de correo del buzón “Bandeja de Entrada”
- ListaBorrados(): Función de listado de correo del buzón “Borrados”

4.10 DIAGRAMA DE ESTRUCTURA DE DATOS

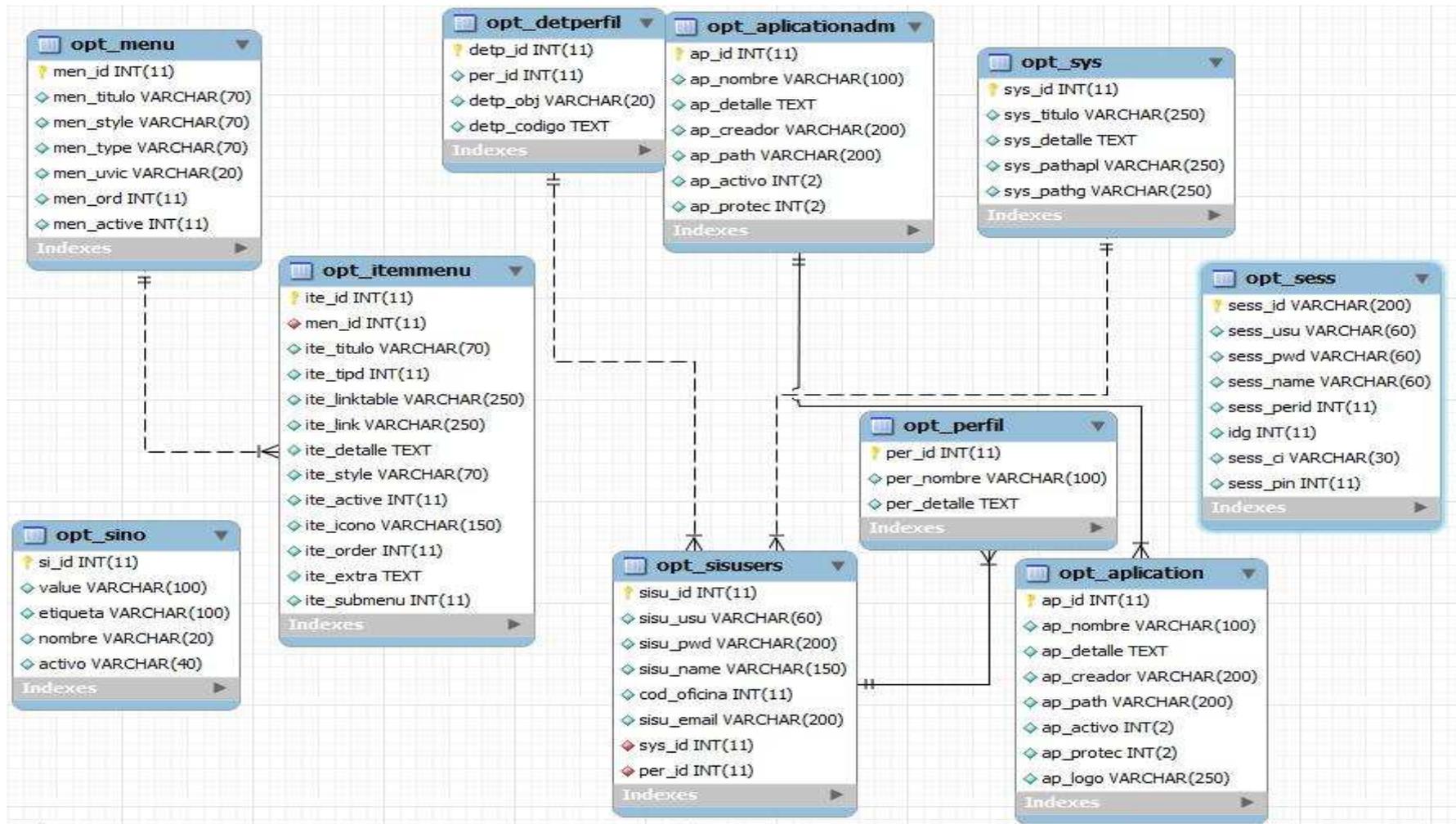


Fig 4.25 Diagrama de Estructura de datos

Las tablas utilizadas para Optimail son las siguientes:

- OPT_MENU: Contiene el manejo de menús.
- OPT_SINO: Contiene la información de los combobox (SI/NO)
- OPT_DETPERFIL: Se manejan los perfiles de ingreso al sistema como es para cambiar clave.
- OPT_SYS: Registra los datos del sistema.
- OPT_SESS: Se guarda un id de la sesión para seguridad y validación.
- OPT_PERFIL: Guarda el detalle del perfil.
- OPT_SISUSERS: Guarda los datos del usuario.
- OPT_APPLICATIONADM: Datos aplicativos internos del sistema. En este caso son los submódulos.
- OPT_APPLICATION: Contiene las aplicaciones externas. Optimail no posee aplicaciones externas pero es indispensable para la flexibilidad de la aplicación.

4.11 API DE LINUX UTILIZADOS

Dovecot y Postfix son servicios que utilizan el API de Linux. El API utilizado entonces es Linux Kernel API. Se conoce que el API de Linux de Kernel se divide en:

- Sistema
- Procesamiento
- Memoria
- Almacenamiento
- Networking
- Interfaz de usuario

Nuestra herramienta de correo Optimail no usaría entonces todos los puntos mencionados anteriormente ya que el interés viene en la transferencia y recepción de mensajes de correo electrónico. Se procede entonces a escoger la característica de Networking.

4.11.1 Networking

Existe un driver para cada tipo de NIC (Network Interface Card). Por dentro, Linux siempre llama a un estándar de nivel alto de enrutamiento: “netif_rx [net/core/dev.c] que controlará a quién pertenece la trama de nivel 3 del protocolo.

4.12 Carpetas y archivos de Optimail

Optimail contiene archivos y carpetas que son necesarios para su funcionamiento. La descripción de cada uno de ellos se detalla en los anexos. Aunque el código esté a la disposición, no se recomienda alterar ni modificar su contenido, a excepción del archivo de configuración config.php que es necesario para el funcionamiento de la aplicación y que se explicará más adelante en el capítulo V.

En la carpeta raíz de Optimail existen dos archivos importantes. Los archivos son los siguientes:

Index.php: Es la página principal de Optimail. Al ingresar el URL en el navegador no es necesario incluir la página de “index.php” (La explicación se detalla en el capítulo 5). (Ver Anexo (A-1)

Formularioereg.php: Contiene el formulario utilizado en la creación de un nuevo usuario de Optimail (Anexo A-2)

4.12.1 Carpeta CFGCLASES

Contiene clases utilizadas en Optimail. Esta carpeta contiene los siguientes archivos

- Clases.php: Archivo de enlace con todas las clases que se usan en el sistema (Ver Anexo B-1)
- Clasesmod.php: Clases para módulos. (Anexo B-2)

- Config.php: Archivo para configuración de la conexión a la base de datos. (Anexo B-3)

4.12.2 Carpeta DATA

Contiene el archivo tablas.txt. Es un script muy importante para la configuración de la base de datos de Optimail. Ese script debe ser pegado al entrar a phpmyadmin. (Se explicará su uso en el capítulo V).

4.12.3 Carpeta EDITOR

Contiene clases y frameworks para el despliegue de los editores. Contiene la barra de herramientas que se despliega en Optimail al crear un nuevo correo (ver Capítulo V).

4.12.4 Carpeta GRAFICOEX

Contiene gráficos adicionales de Optimail que normalmente no se utilizan pero pueden ser reemplazadas en la carpeta original

4.12.5 Carpeta ICONOS

Esta carpeta contiene los íconos utilizados en Optimail y otros adicionales. Estos íconos pueden ser personalizados siempre y cuando se conserve el nombre y tamaño.

4.12.6 Carpeta IMAGES

Imágenes principales de Optimail en extensión .jpg. Pueden ser personalizadas siempre y cuando se mantenga el mismo nombre, extensión y tamaño

4.12.7 Carpeta LIBRERIA

Carpeta indispensable para el funcionamiento de Optimail. Contiene las librerías

- Access.php: Clase de acceso al sistema (Anexo C-1)
- Aplicativo.php: Clase para dar funcionalidad a los sub aplicativos. (Anexo C-2)
- Attachmentread.class.php: Clase para realizar attachments de archivos. (Anexo C-3)
- Class.página.Z: Clase para despliegue de páginas (Anexo C-4)
- Dbcc.php: Clase para la conexión a la base de datos. (Anexo C-5)
- Func_g.php: contiene funciones de: actualizar, borrar, guardar y borrar en las tablas. (Anexo C-6)
- Menú.php: Contiene la clase del menú. (Anexo C-7)
- Parámetros.php: Variables generales del sistema. (Anexo C-8)
- Sesión.php: Usado para la creación de sesión. (Anexo C-9)

4.12.8 Carpeta MODULES

Esta carpeta despliega todas las funciones de las aplicaciones del sistema como por ejemplo la lista de archivos.

4.12.9 Carpeta TEMPLATEFORMS

Esta carpeta contiene todos los diseños de las tablas que se ven como por ejemplo usuarios. Contiene su diseño estándar. (Para pruebas y testing)

4.12.10 Carpeta TEMPLATES

Dentro de Templates está la subcarpeta IBTEMPLATEV2 contiene el núcleo de Optimail lo que es:

- lectura de emails
- borrado de emails
- crear emails y envío.

Los archivos creados en Optimail son los siguientes:

- **Borrar.php**: Contiene código de la eliminación de un correo electrónico. (Ver Anexo D-1)
- **Carpeta.php**: Lectura de buzones (Bandeja de Entrada, Enviados, Borrados) (Anexo D-2)
- **Enviar.php**: Envío de Correo electrónico. (Anexo D-3)
- **Envío.php**: Correo electrónico almacenado en buzón “Enviado”. (Anexo D-4)
- **Gotodownload.php**: Descarga el archivo adjunto de un mensaje. (Anexo D-5)
- **Inicio.php**: Pantalla de inicio de sesión. (Anexo D-6)
- **Leer.php**: Lectura de correo electrónico. (Anexo D-7)
- **Listac.php**: Contiene la lista de correos que se desplegará en cada buzón. (Ver Anexo D-8)
- **Nuevacarpeta.php**: Creación de nuevo buzón (Borrados) (Anexo D-9)
- **Nuevacarpeta2.php**: Creación de nuevo buzón (Enviados) (Anexo D-10)

- Nuevo.php: Creación de un nuevo mensaje de correo electrónico. (Anexo D-11)
- Readattachment.php: Contiene el código que sirve para leer un archivo adjunto. (Anexo D-12)
- Readcell.php: Enlaza las clases de lectura de correo. Por ejemplo cuando un usuario desee leer un nuevo correo de la bandeja de entrada entonces Readcell abre el archivo Carpetas y busca la carpeta correspondiente que es bandeja de entrada o inbox.(Anexo D-13)
- Receivemail.class.php: Clase que contiene la administración de lectura de correo recibido. (ejemplo el número de mensajes de correo recibidos). (Anexo D-14)
- Vaciar.php: Contiene código que sirve para vaciar la papelera de correos electrónicos. (Anexo D-15)

Dentro de IBTEMPLATEV2 existen más subcarpetas, de las cuales:

Images

Contiene íconos usados en Optimail tales como:

- Ícono de borrado
- Icono de nuevo correo
- Icono de adjuntos
- Icono de Papelera
- Icono de Vaciar papelera.
- Icono de Enviar

Menú

Plantilla o patrón del menú de inicio. Contiene el único archivo llamado menuinicio.php. (Anexo E-1)

Styles

Contiene el estilo de letra (por defecto Times New Roman). El archivo tiene como nombre `formato.css`

CAPITULO V. IMPLEMENTACIÓN Y PRUEBAS

5.1 IMPLEMENTACIÓN

En el capítulo anterior se mencionó la metodología de desarrollo de nuestra herramienta de correo electrónico de igual manera se han mencionado las características que Optimail posee. Ahora es el momento de instalar el software y para ello requiere realizar algunas configuraciones adicionales. Se explicará igualmente cómo se procedió en obtener el paquete final usando una herramienta adicional de nombre Bitrock InstallBuidler Enterprise. Se mostrarán resultados de pruebas de concurrencia y chequeos con usuarios simultáneos. Por último se verá el manejo de Optimail en funcionamiento.

5.1.1 Compilación

Para obtener el paquete RPM para una arquitectura específica se utiliza la aplicación Bitrock installBuilder Enterprise V6.0.

BitRock InstallBuilder y su aplicación se convierte en un despliegue rápido, fácil. La herramienta de desarrollo permite crear de forma rápida fácil de utilizar, multiplataforma instaladores nativos que tienen un aspecto y estilo a través de Windows, Mac OS X, Linux, HP-UX, AIX, FreeBSD, Mac OS X, Solaris, Windows y construir a partir de un único entorno. GUI, texto, y sin atender los modos de permitir que los instaladores que se ejecuta en servidores, estaciones de trabajo, y de escritorio.

InstallBuilder ofrece varias funciones avanzadas, incluyendo la integración de la funcionalidad RPM. Esta herramienta tiene incorporadas las facilidades necesarios para la instalación de funciones, tales como: autodetecta de Java (tm) de tiempo de ejecución, la modificación de los permisos de archivos y la propiedad, sustituyendo el texto en un archivo, añadiendo las variables de entorno, agregando los directorios a la ruta, la creación de enlaces simbólicos, cambiando el registro de Windows, y el lanzamiento exterior de los scripts.

La forma fácil de utilizar la nueva interfaz gráfica permite a los usuarios aprender rápidamente a usar InstallBuilder. Para usuarios avanzados, está disponible un amigable formato XML, pudiendo ejecutar proyectos de control de código fuente de integración, colaboración y personalización de proyectos de desarrollo tanto a mano y usando scripts externos. Una interfaz de línea de comandos permite automatizar e integrar el proceso de construcción.



Imagen 5.1: Creación de nuevo proyecto con InstallBuilder

Al abrir BitRock InstallBuilder se creó un nuevo proyecto. En “Product name” el nombre del producto y en “Product name” el nombre de la aplicación compilada más adelante. Como se trata de la primera versión de Optimail se deja la opción de “Version Number” en 1.0.

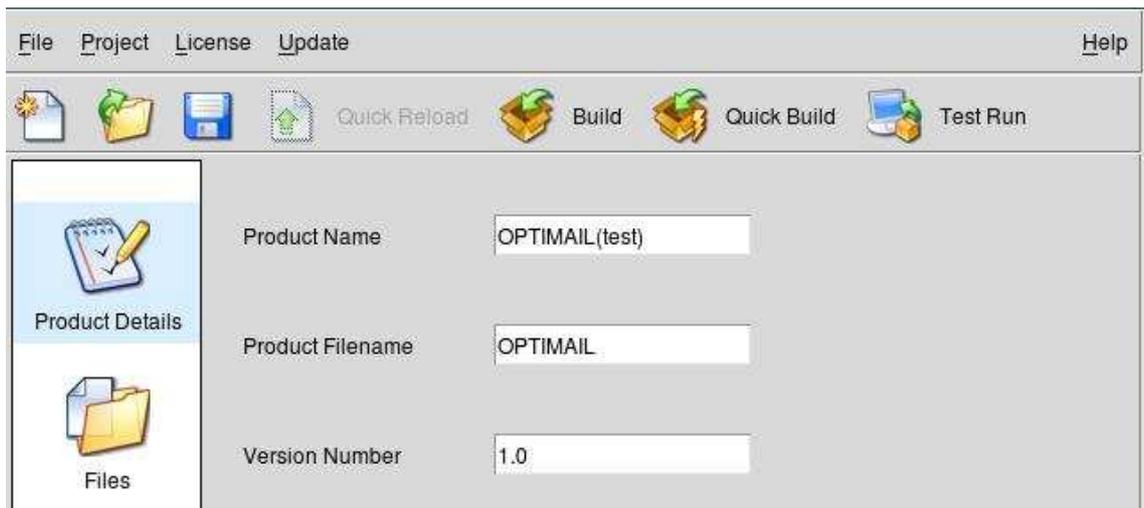


Imagen 5.2: Detalles del paquete

Una vez creado un nuevo proyecto se comprobará en la opción “Product Details” que los datos estén ingresados correctamente.

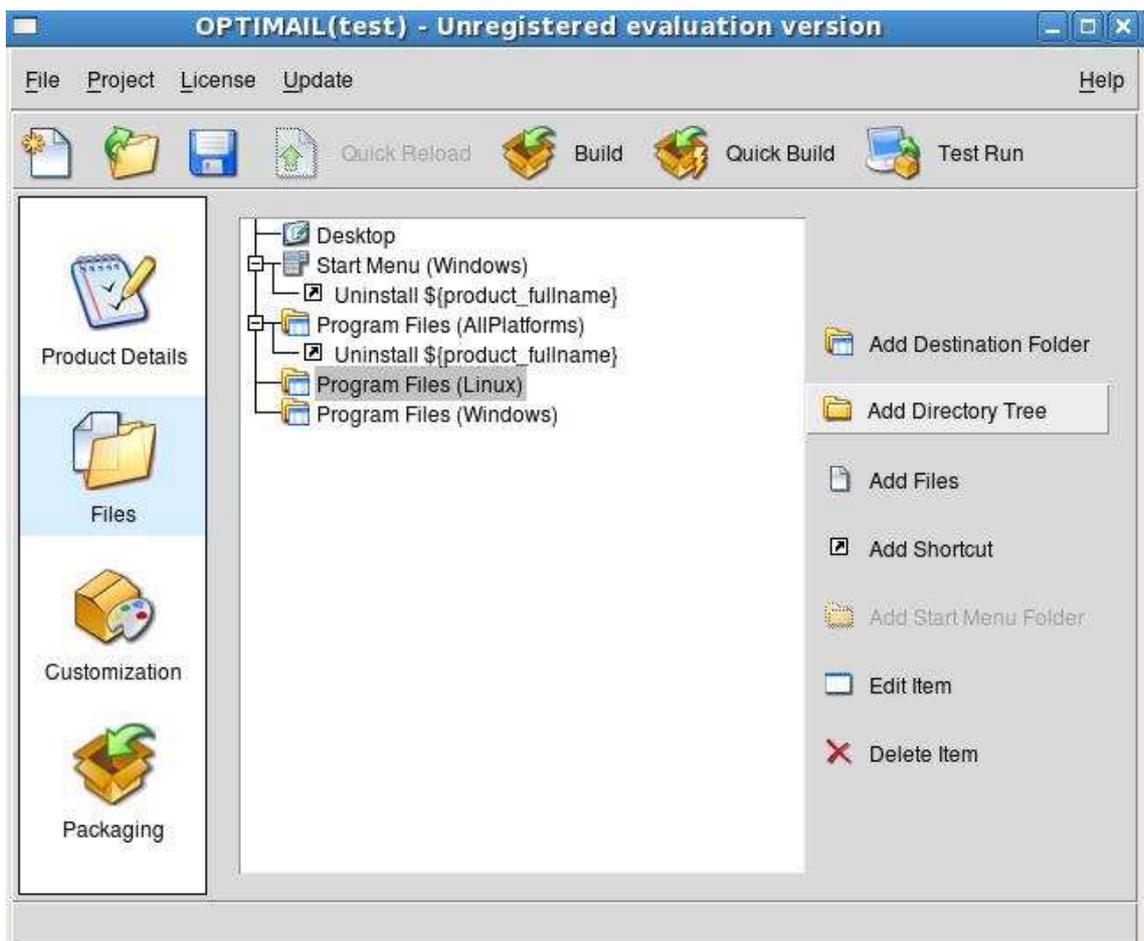


Imagen 5.3 Archivos a agregar al paquete Optimail

Ahora se debe escoger los archivos a compilar para ello se debe escoger la opción “Files” en la barra de herramientas de la izquierda. Optimail tendrá un instalador en Linux por lo que se selecciona la opción “Program Files Linux) y a continuación “Add Directory Tree”. Una vez encontrada la aplicación se le agrega a la lista.



Imagen 5.4 Descripción del Paquete

Se debe indicar dónde se instalará dicha aplicación. Optimail debe trabajar en la siguiente carpeta en Linux: /var/www/html/optimail por lo que en la opción “Add destination folder” se agrega el path de destino y se escoge además en “Platforms” Linux. En el caso de que no exista ese path InstallBuilder lo creará automáticamente.

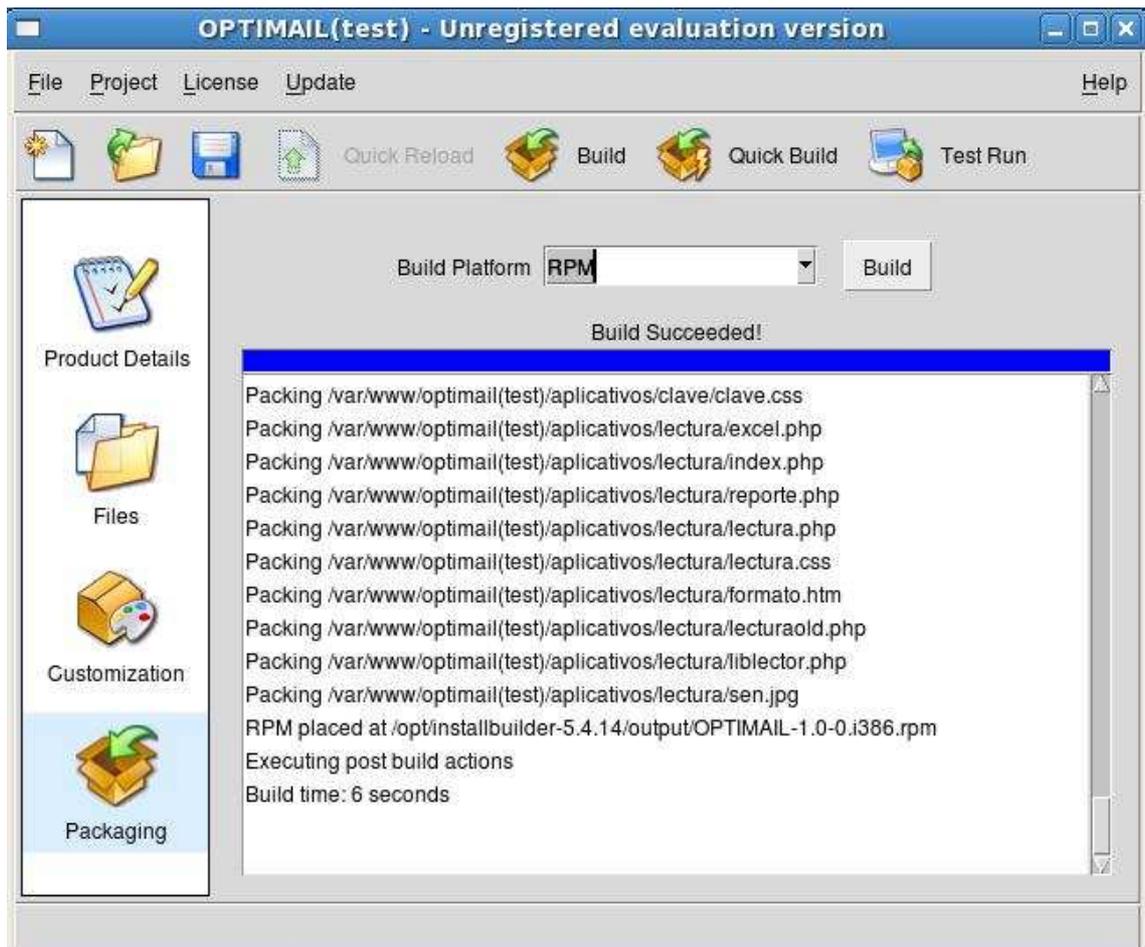


Imagen 5.5: Plataforma y compilación del paquete

Por último se procede con la compilación. En la opción “Packaging” en la barra de herramientas se escoge la plataforma en este caso “RPM”. Hecho esto se hace click en “Build” y luego se obtiene un archivo de extensión .rpm .El archivo normalmente se encuentra en el path /opt/installbuilder/output del compilador. InstallBuilder ha terminado de realizar su compilación.

5.1.2 Instalación de la Aplicación

Una vez instalado el sistema operativo Fedora 7 indispensable para el funcionamiento de Optimail explicado anteriormente en el capítulo 3: “Configuraciones iniciales” se procede con la instalación del software. Optimail se lo puede encontrar en la siguiente dirección electrónica: <http://rapidshare.com/files/230782321/OPTIMAILMM-1.0->

0.i386.rpm.html Todos los archivos de Optimail se encuentran en ese compilado y no requiere descarga adicional para su funcionamiento.

Para instalar el paquete se debe ejecutar el siguiente comando en la consola (terminal) de fedora: `Rpm -ivh optimail`.

El proceso de Instalación requiere las siguientes dependencias: MySQL , Apache y Dovecot de acuerdo a las siguientes características:

MYSQL .- Base de datos que se utilizará para el registro de usuarios, perfiles y todos los datos que intervienen en nuestro aplicativo de correo.

Apache .- Servidor http para la publicación de nuestro aplicativo de correo electrónico en internet.

Dovecot .- Es una fuente de servidor IMAP y POP3 para Linux/UNIX, escrita principalmente pensando en la seguridad.

MYSQL, Apache y Dovecot se instalaron anteriormente en el capítulo 3 junto con el software base.

Este sistema no es un paquete *rpm*, sino un sistema que tiene una similitud a phpmyadmin y joomla pero se le puede acoplar como rpm. A este sistema se lo compiló para obtener el rpm final, utilizando Bitrock installBuilder Enterprise V6.0.

Recordar que las dependencias de cada uno ya fueron instaladas en el capítulo 3 por lo que no requiere cambios o instalaciones adicionales.

5.1.3 Configuraciones

5.1.3.1 Archivo de configuración.

Dentro de la carpeta optimail sólo se tiene un archivo de configuración. Existe en la carpeta cfgclases el archivo config.php el cual hay que modificar para su funcionamiento. Al abrir el archivo se tiene la siguiente estructura:

```
<?php
$basededatos="optimaildb" //Base de datos de Optimail creada desde Mysql
$host="localhost"; //Host del cual está creada la base de datos
$userdb="root"; //Usuario creador de la base de datos
$password="13272713"; //Password del usuario creador de la base de datos
$ip='192.168.1.9'; //Dirección IP del servidor de base de datos
//Path editor
$ptaeditor="/optimail/"; //indica el path de Optimail y nombre de carpeta
?>
```

Al usar por primera vez optimail requiere modificar el ingreso a la cuenta de root del actual computador por lo que se modificará el valor de \$password por la contraseña indicada. En cuanto a \$ip se modificará igualmente el valor a la dirección actual del computador. Se guardan los cambios y se habrán modificado correctamente los cambios necesarios para el funcionamiento de OPTIMAIL.

5.1.3.2 Enviar e mail por php

Para el envío de correos electrónicos utilizando PHP disponemos de una función bastante potente llamada mail(), incluida en todas las versiones de PHP, sin necesidad de instalar ningún añadido, en contra de lo que ocurría con ASP. La función se llama mail() y recibe tres parámetros de manera obligada y otros dos parámetros que podemos colocar opcionalmente. Devuelve true si se envió el mensaje correctamente y false en caso contrario.

5.1.3.3 Parámetros necesarios en todos los casos

Destinatario: la dirección de correo o direcciones de correo que han de recibir el mensaje. Si incluimos varias direcciones debemos separarlas por una coma.

Asunto: para indicar una cadena de caracteres que queremos que sea el asunto del correo electrónico a enviar.

Cuerpo: el cuerpo del mensaje, lo que queremos que tenga escrito el correo.

Ejemplo de envío de un mail sencillo

```
<?
mail("pepito@desarrolloweb.com,maria@guiartemultimedia.com","asuntillo","Este
es el cuerpo del mensaje")
?>
```

5.1.3.4 Parámetros opcionales del envío de correo

Headers: Cabeceras del correo. Datos como la dirección de respuesta, las posibles direcciones que recibirán copia del mensaje, las direcciones que recibirán copia oculta, si el correo está en formato HTML, etc.

Additional_parameters: esta opción no suele utilizarse y, además, sólo está disponible a partir de la versión PHP 4.0.5 y desde PHP 4.2.3 está deshabilitado en modo seguro. Puede usarse para pasar parámetros adicionales al programa configurado para enviar el correo, cuando se manda el mail usando la opción de configuración `sendmail_path`. Podemos obtener más información en la documentación de PHP para la función `mail()`.

5.1.3.5 Activando servicios y accediendo a un browser

Para el funcionamiento de Optimail requiere activar por terminal algunos servicios tales como Dovecot, Apache, Postfix y mysql. Para ello ejecutar los siguientes comandos:

```
# service dovecot start
```

```
# service httpd start
```

```
# service postfix start
```

```
# service mysql start
```

Para verificar el correcto funcionamiento el usuario ingresará a un browser (preferible Mozilla Firefox) e ingresará la dirección IP local que fue ingresada también en el archivo de configuración.

El resultado mostrará una pantalla como la siguiente indicando que todo está bien:

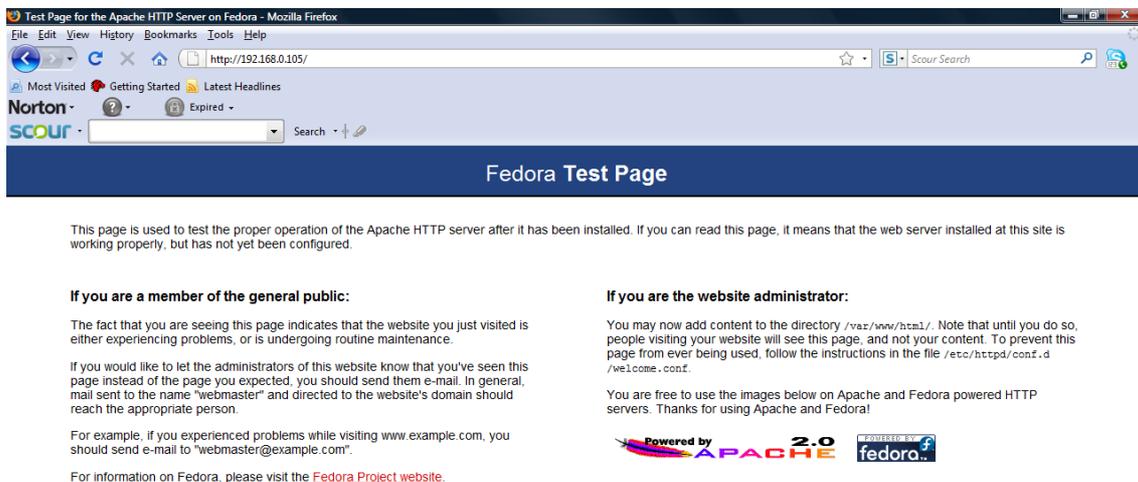


Imagen 5.6: Página de Apache

5.1.3.6 Instalación y configuración de la base de datos para Optimail

Para instalar la base de datos de Optimail (script), nos vamos a la carpeta DATA dentro de nuestra aplicación. Luego abrimos en el browser nuestro administrador de la base de datos en el siguiente link:

http://192.168.1.9/phpmyadmin. (el ip del link varía de acuerdo a la configuración del archivo config.php)

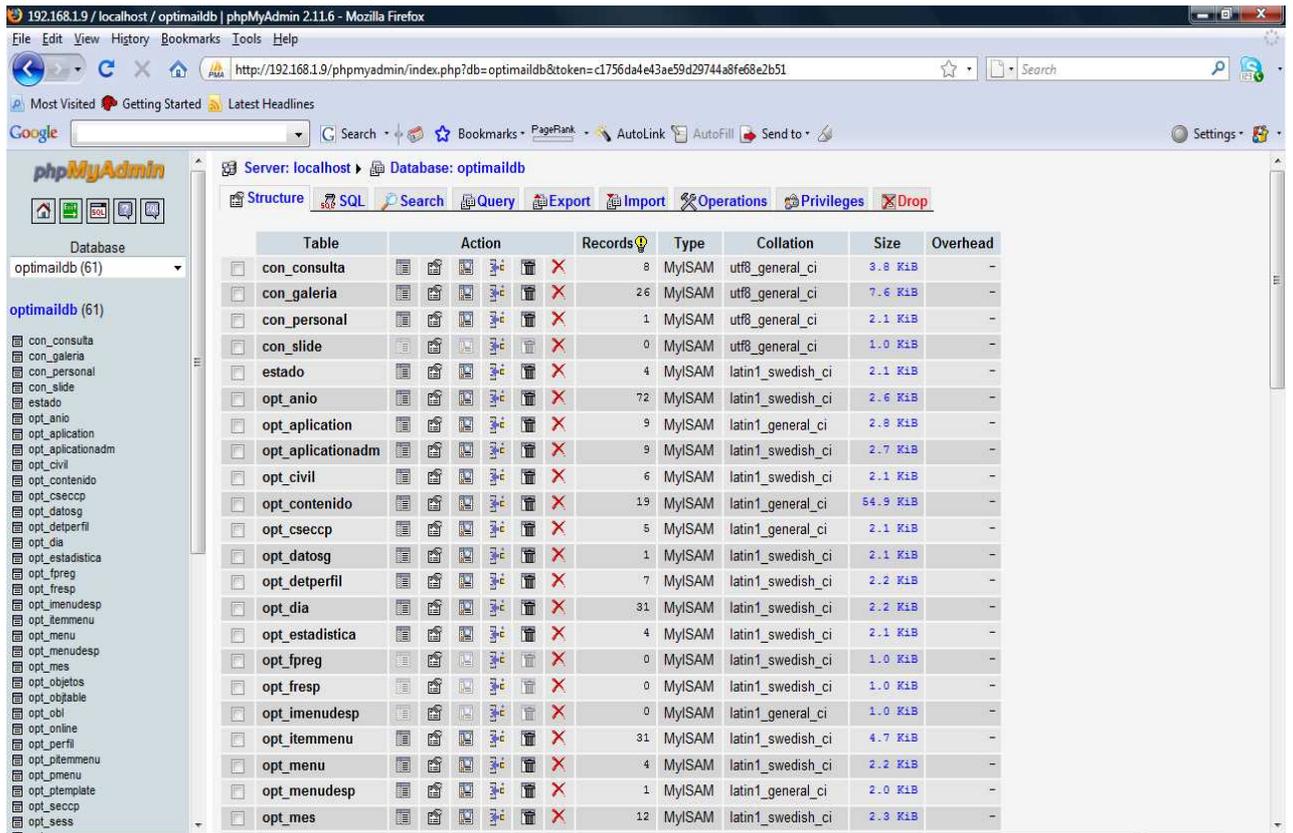


Imagen 5.7: Pantalla de PhpMyAdmin

Seleccionar la opción SQL en la barra superior

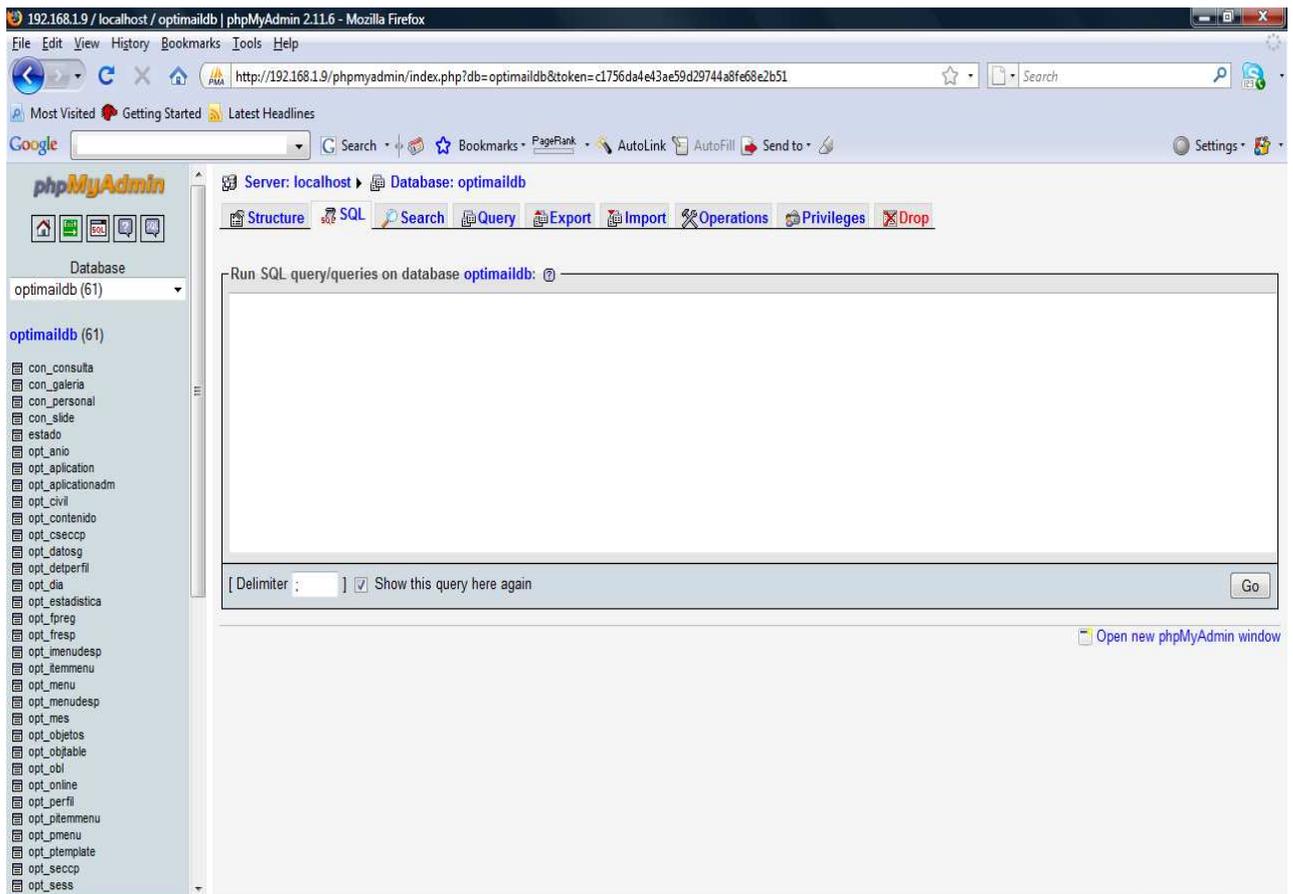


Imagen 5.8: Pantalla de ejecución de queries de SQL en PhpMyAdmin

Pegamos el script del archivo tablas.txt y ejecutamos ese script haciendo click en GO.

5.2 PRUEBAS

5.2.1 Pruebas de carga

- Usuarios simultáneos
- Tiempos de respuesta
- Capacidad de buzones (bandeja de entrada) 1000 kb

CASO DE PRUEBA	ENTRADA	RESULTADO
Usuarios simultáneos	20 usuarios entran simultáneamente al sistema	Los usuarios ingresan satisfactoriamente al sistema.

Tabla 5.1: Usuarios Simultáneos

Según el archivo Maillog (Ver Anexo F-1) se puede obtener el siguiente cuadro estadístico con 20 usuarios:



Figura 5.1: Tiempo de respuesta sin adjunto

El tiempo de respuesta se encuentra en el rango de 1.5 segundos a 1.9 dando un promedio de 1.7 segundos por envío.

PROCEDIMIENTO

1. El sistema desplegará la pantalla correspondiente al ingreso al sistema OPTIMAIL
2. El usuario debe ingresar su nombre y clave.
3. Seleccionar la opción de “ACEPTAR”
4. Los usuarios son verificados y acceden al sistema.

CASO DE PRUEBA	ENTRADA	RESULTADO
Tiempos de respuesta	Usuario envía un correo sin adjunto	El correo se envía exitosamente a los destinatarios
	Se envía mail con tamaño 3kb	El tiempo de envío de los correos es de promedio 1.7 segundos
	Se envía mail con tamaño 5kb	
	Se envía mail con tamaño 1kb	

Tabla 5.2: Tiempos de respuesta

Para el envío de correo con archivo adjunto se obtiene el siguiente cuadro estadístico según el archivo Maillog (Ver Anexo F-2):



Figura 5.2: Tiempo de respuesta con adjunto

El archivo adjuntado tiene 800 kb de tamaño y se realizó la prueba para 20 usuarios distintos de Optimail.

PROCEDIMIENTO

1. El usuario selecciona “NUEVO EMAIL”
2. El usuario ingresa correo electrónico en el campo “PARA”
3. El usuario ingresa asunto en el campo “ASUNTO”
4. El usuario ingresa un mensaje en el cuadro de texto.
5. Hacer click en el botón “ENVIAR”
6. El sistema desplegará el siguiente mensaje: “MENSAJE ENVIADO”

CASO DE PRUEBA	ENTRADA	RESULTADO
Tiempos de respuesta	Usuario envía un correo con archivo adjunto	El correo se envía exitosamente a los destinatarios
	Se envía mail con tamaño 500kb	El tiempo de envío de los correos es de promedio 3.5 segundos
	Se envía mail con tamaño 800kb	
	Se envía mail con tamaño 1mb	

Elaborado por: Marco Molina

Fuente: De Campo

Tabla 5.3 Tiempos de respuesta

PROCEDIMIENTO

1. El usuario selecciona “NUEVO EMAIL”
2. El usuario ingresa correo electrónico en el campo “PARA”
3. El usuario ingresa asunto en el campo “ASUNTO”.
4. Hacer click en el botón “ADJUNTAR”.
5. El usuario selecciona el archivo deseado a adjuntar.
6. El usuario ingresa un mensaje en el cuadro de texto.
7. Hacer click en el botón “ENVIAR”.
8. El sistema desplegará el siguiente mensaje: “MENSAJE ENVIADO”

CASO DE PRUEBA	ENTRADA	RESULTADO
Capacidad de buzones	Todos los mensajes de correo de BANDEJA DE ENTRADA, ENVIADOS y BORRADOS.	El sistema funciona correctamente con la capacidad de cada buzón como se describe a continuación: Bandeja de entrada: 10mb Enviados: 10 mb Borrados: 10mb

Elaborado por: Marco Molina

Fuente: De Campo

Tabla 5.4: Capacidad de Buzones

CASO DE USO	LIMITE	DESCRIPCION
Acceso al sistema	más de 1000 usuarios simultáneos	Por falta de recursos ya que el sistema es un prototipo no tenemos recursos para muchos usuarios
Crear cuentas	saturación de la base de datos	Por falta de recursos para realizar las pruebas suficientes de carga
Administrar correo	Prueba de la administración de Postfix en una arquitectura servidor en cuanto a Hardware	Por ser un prototipo se realizaron las pruebas en el equipos disponibles en este caso utilizamos una computadora portátil HP
Los límites de prueba nos permiten ver hasta qué nivel podemos probar el sistema y esto viene dado por los siguientes factores que son: recursos disponibles tanto en hardware, software y recurso económico		

Tabla 5.5 Prueba de carga

5.2.2 Limitaciones de pruebas

En algunos usuarios cuando se configura Outlook 2003 para recibir los correos de nuestro servidor, al darle clic en el botón <Probar configuración de la cuenta...> se realizan pruebas de configuración y sale una marca de OK al lado de cada tarea.

Pero en la tarea "Iniciar sesión en el servidor de correo entrante (POP3)", la prueba de configuración falla. A pesar que se reciben y envían los correos, sale ese fallo.

Otra de la limitaciones que viene desde dentro en ocasiones, es cuando los usuarios realizan envíos masivos y se cuelga el sistema.

5.2.3 Pruebas Funcionales

Una forma de probar los módulos del sistema es mediante los casos de prueba mas significativos, incluyendo las entradas y resultados esperados.

CASO DE PRUEBA	ENTRADA	RESULTADO
Ingresar al sistema	Se obtendrá el valor correspondiente al nombre de usuario y clave respectivos.	El sistema debe dar acceso al usuario si este se encuentra registrado en la base de datos
	El valor son los siguientes: Usuario: jperez Clave : xxxxxxxx	Se desplegara la pantalla con las opciones a las cuales el usuario tiene acceso.

Elaborado por: Marco Molina

Fuente : De Campo

Tabla 5.6: Ingreso al sistema

PROCEDIMIENTO

1. El sistema desplegara la pantalla correspondiente al ingreso al sistema
2. El usuario debe ingresar su nombre y clave
3. Seleccionar la opción de Aceptar o Ingresar(Botón)
4. El usuario es verificado y accede al sistema
 - a. Caso contrario se despliega una pantalla de error al ingreso “acceso negado”

CASO DE PRUEBA	ENTRADA	RESULTADO
Registro de usuarios en el sistema	Se ingresará la información correspondiente en los campos: usuario, password, confirmación password, nombres y apellidos, dirección, teléfono, email perfil.	El sistema debe crear el usuario correspondiente en la base de datos.

Elaborado por: Marco Molina

Fuente : De Campo

Tabla 5.7: Registro de usuarios en el sistema

PROCEDIMIENTO

1. El sistema desplegará la pantalla correspondiente al ingreso de información
2. El usuario debe ingresar el nombre usuario correspondiente en el campo “USUARIO”

- a. Caso contrario el sistema desplegará la siguiente pantalla de error:
“DEBE LLENAR EL CAMPO USUARIO”
3. El usuario debe ingresar el password correspondiente en el campo
“PASSWORD”
 - a. Caso contrario el sistema desplegará la siguiente pantalla de error:
“DEBE LLENAR EL CAMPO PASSWORD”
4. El usuario debe ingresar la confirmación de password en el campo
“CONFIRMACION PASSWORD”.
 - a. El sistema desplegará la siguiente pantalla de error en caso de no ser iguales las contraseñas: “EL PASSWORD NO CONCUERDA CON SU CONFIRMACION: PASSWORD”
5. El usuario debe ingresar sus nombres y apellidos en el campo “NOMBRES Y APELLIDOS”
6. El usuario debe ingresar su dirección de domicilio en el campo
“DIRECCION”
 - a. Caso contrario el sistema desplegará la siguiente pantalla de error:
“DEBE LLENAR EL CAMPO DIRECCION”
7. El usuario debe ingresar su teléfono en el campo “TELEFONO”
8. El usuario debe ingresar su e-mail en el campo “EMAIL ALTERNO”
 - a. Caso contrario el sistema desplegará la siguiente pantalla de error
“VERIFIQUE LA SINTAXIS DEL EMAIL EN EL CAMPO: EMAIL ALTERNO”
9. Seleccionar la opción “GUARDAR”
10. El sistema desplegará una pantalla para confirmación de creación.
11. Seleccionar la opción “ACEPTAR”

5.2.4 Pruebas de Concurrencia

Es el tipo de prueba que se enfoca en certificar la capacidad del sistema de atender múltiples solicitudes de parte de los actores que acceden a un mismo recurso (un dato que esté almacenado en memoria, un conjunto de registros en base de datos o una

interfaz con un dispositivo de hardware o un sistema externo). Este tipo de pruebas también reciben el nombre de pruebas de contención¹.

CASO DE PRUEBA	ENTRADA	RESULTADO
Usuarios simultáneos	10 usuarios ingresan simultáneamente al sistema.	Los usuarios ingresan satisfactoriamente al sistema con un promedio de 7.5 segundos
	10 usuarios ingresan a cualquiera de las carpetas de optimail simultáneamente (Bandeja de Entrada, Enviados, Borrados)	Los usuarios ingresan satisfactoriamente al sistema con un promedio de 3 segundos
	10 usuarios envían correo simultáneamente (no adjunto)	Los correos se envían satisfactoriamente en un promedio de 5 segundos
	10 usuarios envían correo simultáneamente con archivos adjuntos de tamaño variable.	Los correos se envían satisfactoriamente en un promedio de 9.7 segundos

Elaborado por: Marco Molina

Fuente : De Campo

Tabla 5.8: Usuarios simultáneos

Las pruebas de “conurrencia” se simularon conectándose desde 10 computadoras de la Universidad al sitio desarrollado (<http://140.148.4.38/bcvwm/index.jsp>), para ello se accedió al link de capítulos y posteriormente al link de capítulo 2, esto fue realizado 10 veces por cada usuario. Con ello se aseguro que el sistema responde oportunamente a las peticiones de los usuarios y aunque en algunas ocasiones tarda un poco más de tiempo en contestar, pero no se pierde la comunicación entre el cliente y servidor.

¹ (<http://www.avatar-global.com/website/pcomolohac.htm>)

Max Residents:	<input type="text" value="1024"/>
Max Sessions:	<input type="text" value="1024"/>
Swap Directory:	<input type="text" value="sessionSwap"/>
Swap Interval:	<input type="text" value="30"/> Seconds
Invalidation Interval:	<input type="text" value="10"/> Seconds
Invalidation Time:	<input type="text" value="30"/> Minutes

Fig 5.3: Pantalla de configuración de Servlet Exec ISAPI para aumentar el tiempo de swap

Para mejorar el tiempo de respuesta dentro de las pruebas de concurrencia se aumento el tiempo de Intercambio de Datos del ServletExec ISAPI (Engine que ejecuta los Servlets), el cual tenia un tiempo por default de 10 segundos y fue aumentada a 30 segundos, lo cual mejoro el tiempo de respuesta en las peticiones de los clientes.

Actualmente el sistema funciona sin ningún error de programación, los problemas posteriores que pueden presentarse son respecto a agregar nuevos módulos para controlar información. Seguramente los errores más comunes al construir un sitio serán por falta de experiencia en esta herramienta, pero para ello se cuenta con un manual de usuario.

5.3 FUNCIONAMIENTO DE OPTIMAIL

5.3.1 Página de inicio de sesión

Al ejecutar el browser (Mozilla Firefox o Internet Explorer) se ingresa a Optimail de la siguiente manera:

[Ip]/optimail

Siendo *[Ip]* la dirección IP local del computador o una dirección IP privada dentro de una red LAN. Una vez realizado se desplegará la siguiente pantalla:

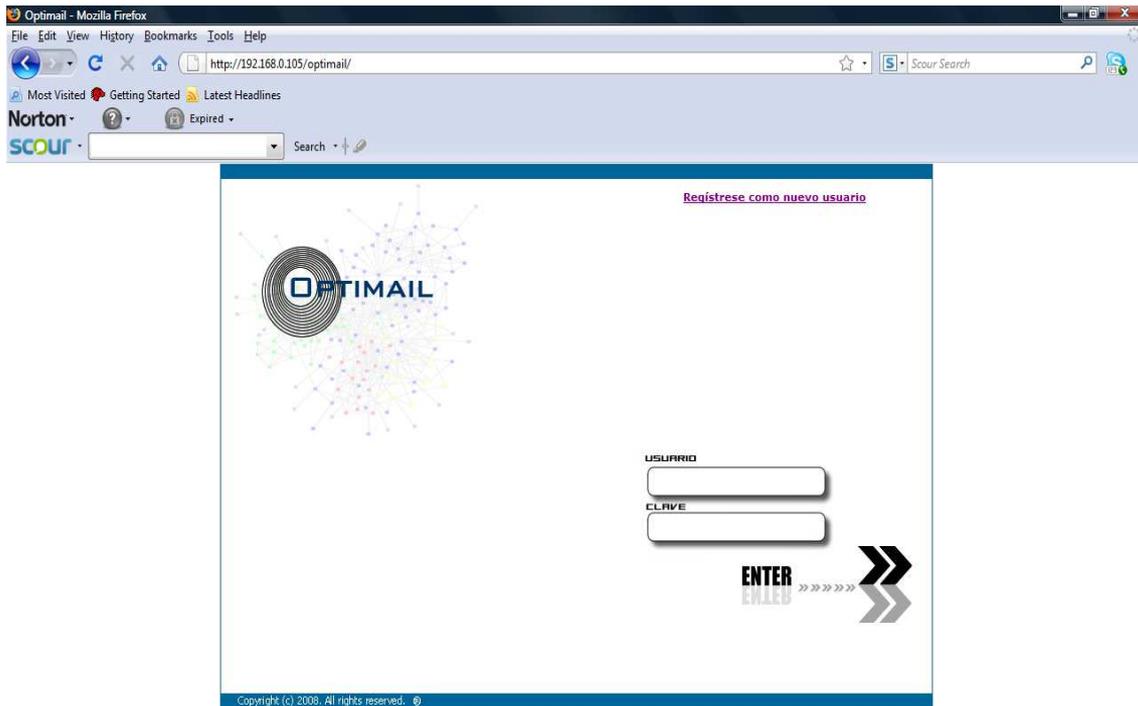


Imagen 5.9: Inicio de sesión de Optimail

Al conectar a Optimail asegurarse que los servicios están levantados (Mysql, Apache, Postfix y Dovecot cada uno de ellos descritos en el capítulo 3) en el computador instalado caso contrario se desplegará el siguiente mensaje de error: “Error al enlazar el servidor!”. Al acceder a Optimail desde un cliente no requiere configuración.

5.3.2 Creación de nuevos usuarios

Optimail permite la creación de nuevos usuarios de correo. Siendo la primera vez en que se utiliza la aplicación se procederá con la creación del primer usuario, para ello requiere hacer click en “Regístrese como nuevo usuario”. En la misma pantalla se desplegará un formulario corto del cual requiere llenar los campos especificados. Se mostrará la siguiente información:

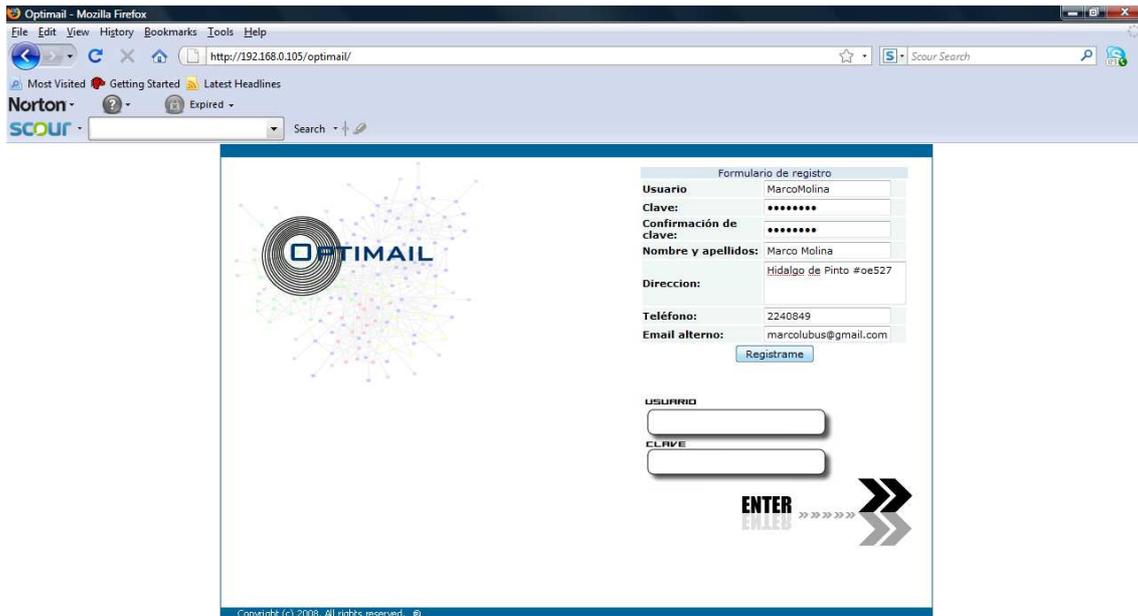


Imagen 5.10 Registro de nuevo usuario

Los campos a ingresar son:

- Usuario: apodo o nickname que se usará para ingresar al sistema. El resultado de la creación de un usuario dará el formato de correo de la manera *Usuario@localhost.optimail*.
- Clave: Se ingresa la clave a utilizar para dicho usuario.
- Confirmación de clave: Requiere confirmar la clave.
- Nombre y Apellidos: Nombre y Apellido del usuario.
- Dirección: Se ingresa la dirección de domicilio. El ingreso de información en ese campo es opcional.
- Teléfono: Número de teléfono. Es un campo opcional
- Email alternativo: Email alternativo del usuario. Es un campo opcional

Una vez ingresada la información se procederá en hacer click en el botón “Regístrate”

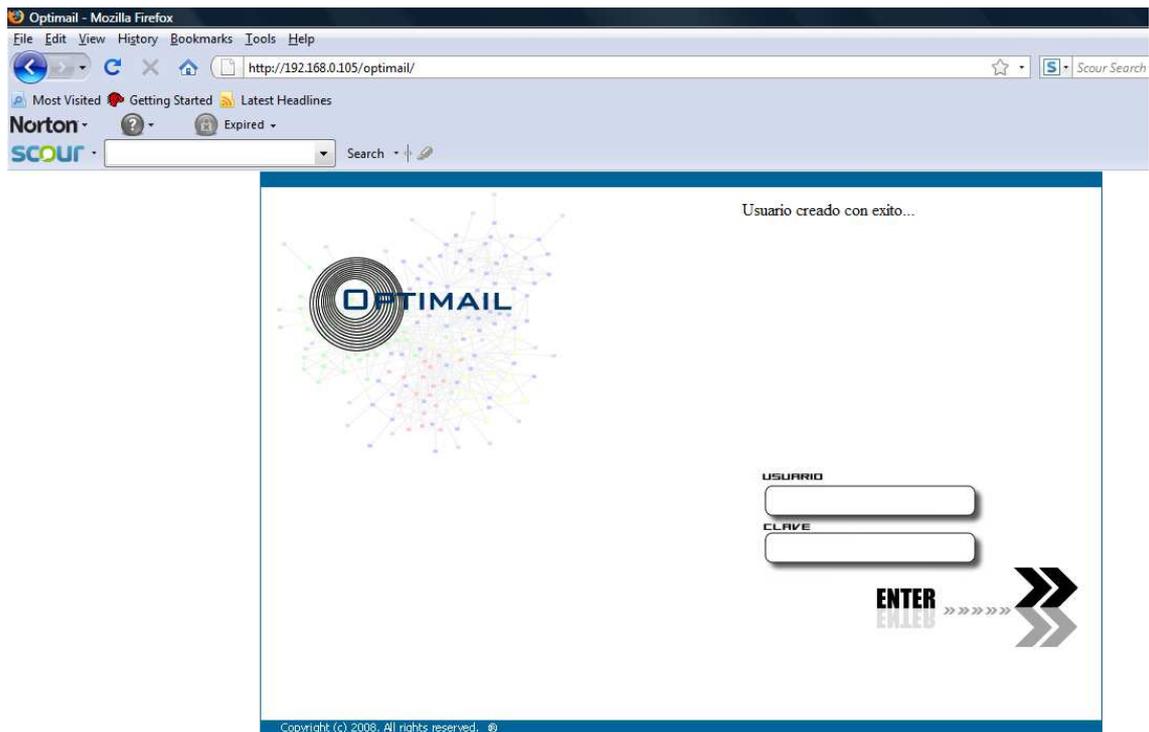


Imagen 5.11: Éxito en la creación de usuario

Se desplegará el mensaje “Usuario creado con éxito” indicando que se ha creado satisfactoriamente el primer usuario de Optimail.

5.3.3 Ingreso al sistema

Para ingresar al sistema Optimail se llenan los campos mostrados en la imagen y luego hacer click en “ENTER”. Si el usuario es correspondido aparecerá el mensaje “Acceso permitido”, caso contrario aparecerá “Acceso negado”:

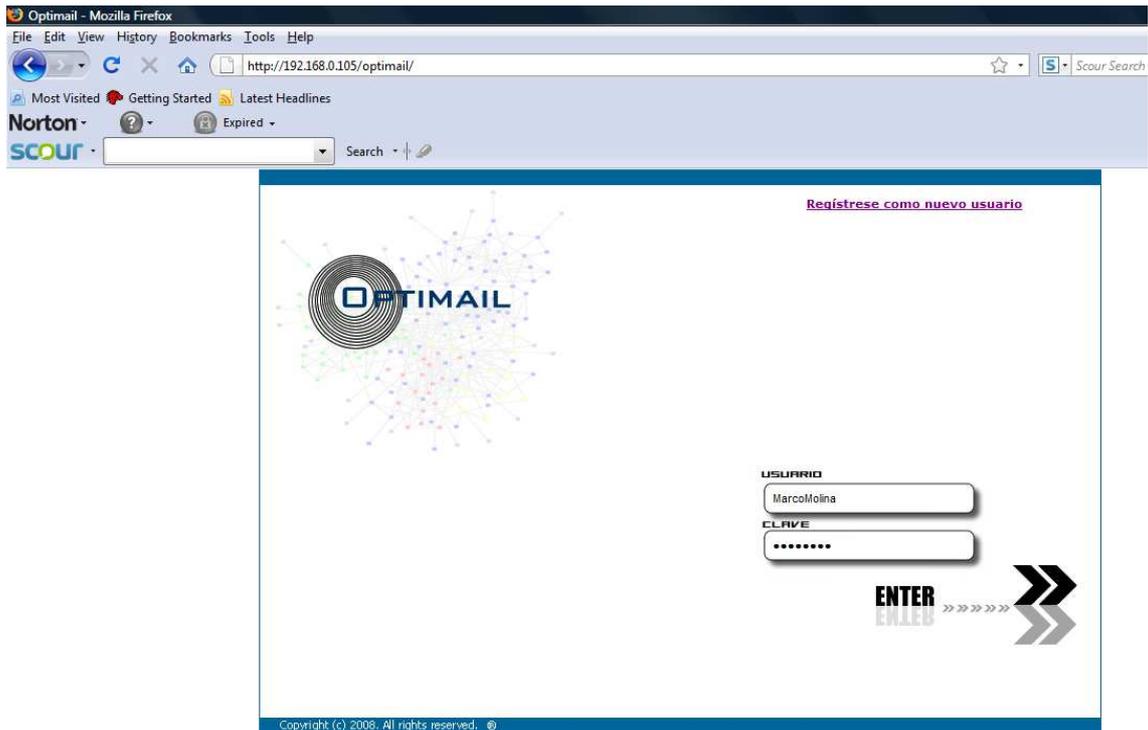


Imagen 5.12: Ingreso a Optimail

5.3.4 Buzones y opciones

Una vez ingresado a Optimail se podrá observar la siguiente pantalla:



Imagen 5.13: Buzones Optimail

Se pueden observar varios buzones y opciones de los cuales:



- **NUEVO EMAIL** : Es la opción encargada de la redacción de nuevos correos electrónicos



- **BANDEJA DE ENTRADA** : Contiene los mensajes de correo electrónico recibidos, diferenciando entre correos nuevos y correos leídos. (En negrita se marcan los nuevos)



- **BORRADO** : Los correos eliminados irán a este buzón.



- **ENVIADO** : Los correos enviados se almacenarán en este buzón.



- **VACIAR PAPELERA** : Limpia el buzón de “Borrado”

- **SALIR** : Cierra el sistema y volverá a la página de inicio de sesión.

5.3.5 Enviar correo electrónico

Para enviar un correo nuevo se debe hacer click en la opción “Nuevo Email” y se desplegará la siguiente pantalla de la cual se ingresará información:

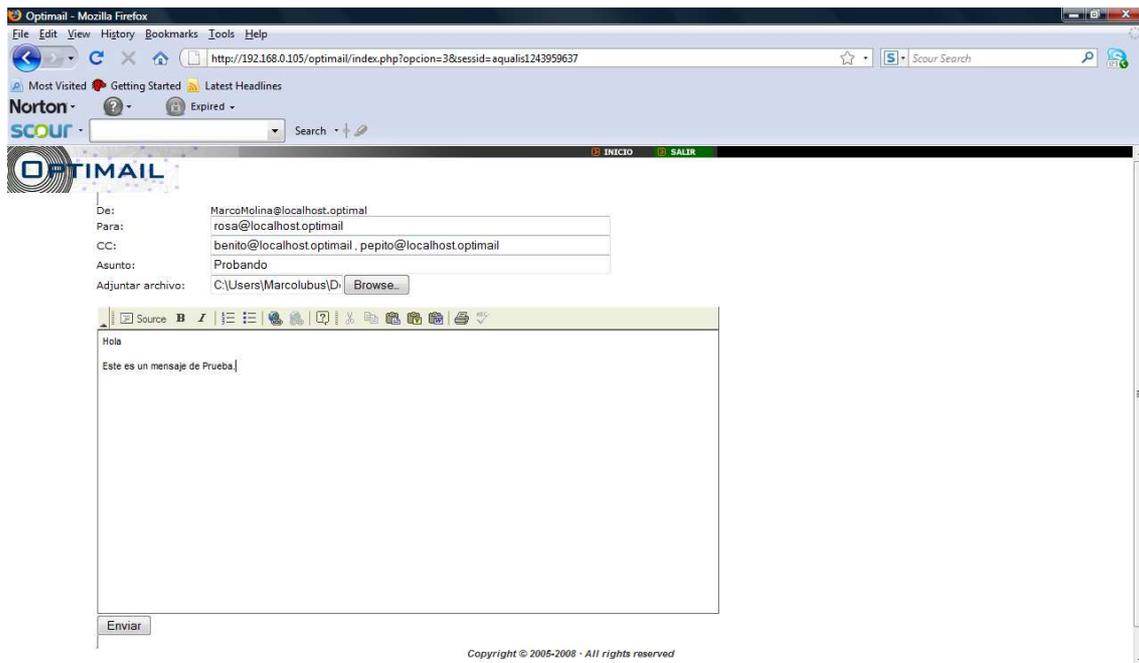


Imagen 5.14: Creación de nuevo correo

A continuación se describe el uso de cada campo:

De

Indica la dirección de correo del remitente. Este campo no es modificable

Para

Se ingresará la dirección de correo electrónico del destinatario siendo el formato de Optimail *usuario@localhost.optimail*.

CC

Se enviará copia de este mensaje de correo electrónico a más usuarios de Optimail. Requiere separar por una coma en el caso de querer enviar una copia a más de un usuario.

Asunto

El usuario ingresará el título o asunto del mensaje de correo electrónico.

5.3.5.1 Adjuntar archivo

Optimail tiene la opción de adjuntar un archivo al enviar un correo electrónico. Al hacer click en el botón “Browse” se debe seleccionar el archivo a adjuntar en el correo. El archivo puede ser de cualquier extensión, aunque hay que mencionar que el límite de tamaño de este no puede superar a 1 MB ni tampoco se puede adjuntar más de un archivo en el mismo mensaje.

5.3.5.2 Cuadro de texto

En el cuadro de texto se procede con la redacción del mensaje. Existe una barra de herramienta que permite darle estilo al mensaje, estos estilos incluyen:

- Contraer, expandir barra de herramienta.
- Fuente HTML
- Negrita
- Cursiva
- Numeración
- Viñetas
- Insertar o editar hipervínculo
- Eliminar Hipervínculo
- Cortar, copiar, pegar texto
- Imprimir

5.3.5.3 Enviar

Al hacer click en el botón “enviar” el mensaje viajará al destinatario apareciendo la siguiente pantalla



Imagen 5.15 Éxito en el envío de mensaje de correo

El correo enviado será almacenado también en el buzón “ENVIADO” para verlo hacer click en esa opción:

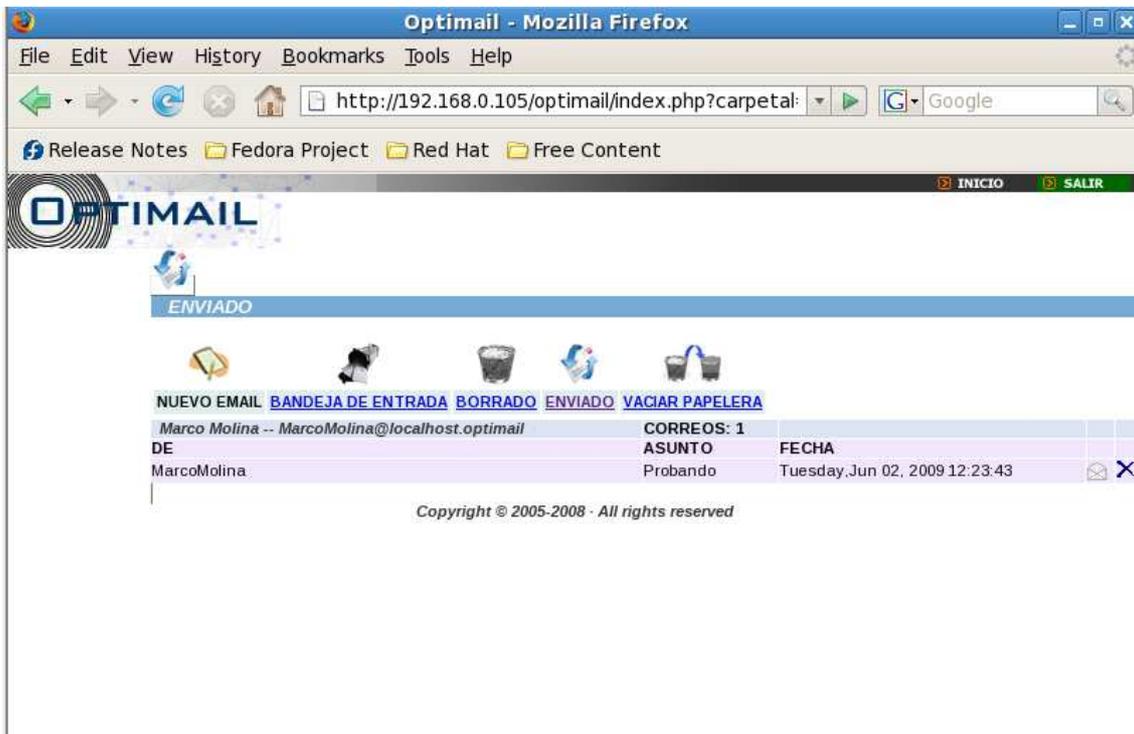


Imagen 5.16 Carpeta “Enviado”

5.3.6 Lectura de Correos

Cuando se han recibido correos electrónicos se verá la lista con el formato siguiente:

DE	ASUNTO	FECHA		
Rosa	hola	Tue, 2 Jun 2009 16:02:04 -0500	✉	✕
MarcoMolina	Probando	Tue, 2 Jun 2009 12:23:27 -0500	✉	✕

Imagen 5.17: Lista de Correos

Al hacer click en bandeja de Entrada se obtiene la lista de correos recibidos. Los nuevos están marcados en negrita y con ícono de sobre amarillo (✉) mientras que los leídos son marcados por un ícono de un sobre abierto (✉). Para abrir un correo específico hacer click en el ícono de buzón amarillo. El ícono de clip (📎) indica que el mensaje contiene un archivo adjunto. Una vez abierto se verá una imagen similar:

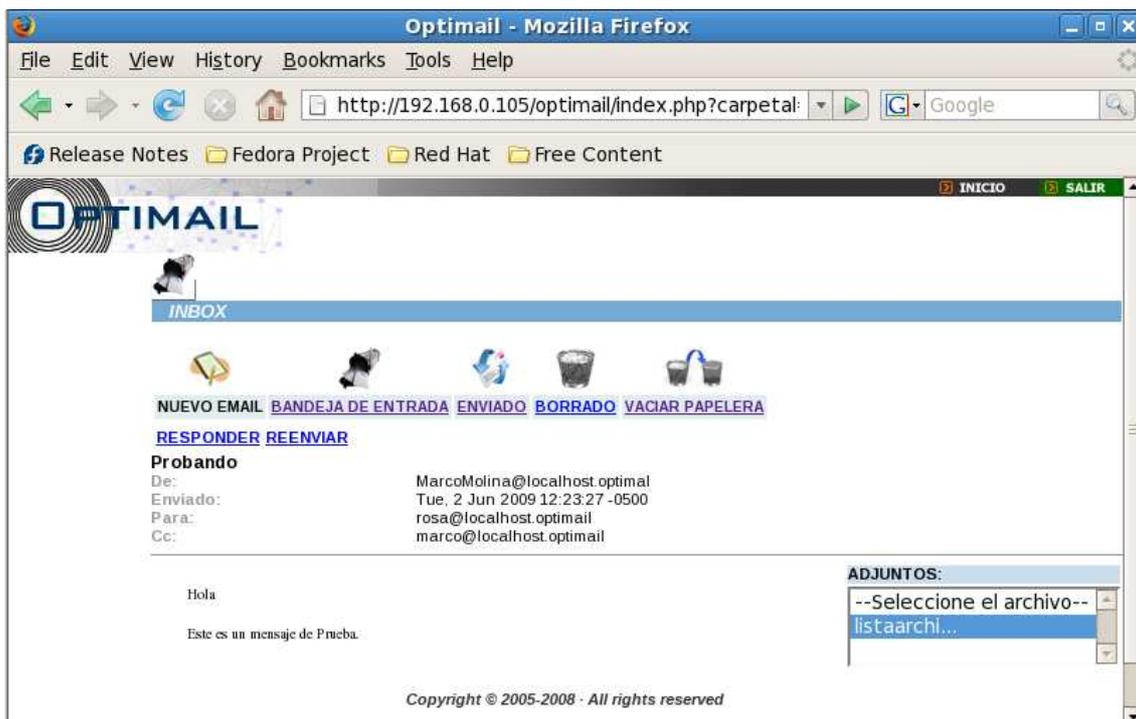


Imagen 5.18: Archivo adjunto en correo

Hay que mencionar que, en caso de existir un archivo adjunto en el mensaje electrónico aparecerá una opción a la derecha la cual contiene el archivo. Al hacer click aparecerá el siguiente cuadro de confirmación:



Imagen 5.19: Cuadro de diálogo para la descarga de archivo

Se descargará el archivo por defecto en el escritorio del computador local.

5.3.7 Responder y Reenviar correo

Cuando se abre un correo electrónico aparecerán dos opciones adicionales que son Responder y Reenviar. Tales opciones facilitan el envío de correo sin necesidad de volver a escribir información. Cuando se responde un correo aparecerá la siguiente pantalla similar a “Nuevo Email”:

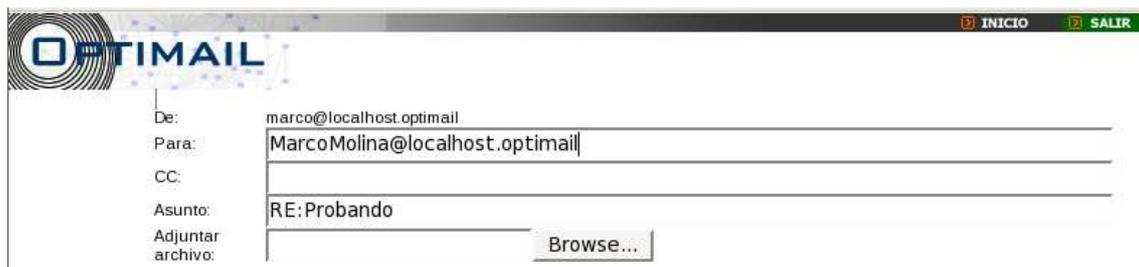


Imagen 5.20: Respondiendo correo electrónico

Se procede igual que al crear un mensaje de correo electrónico nuevo. En el caso de la opción “Para” no es necesario volver a escribir la dirección de correo mientras que en asunto el sistema agrega “RE:” para indicar respuesta.

El mismo caso ocurre con la opción “REENVIAR” con excepción de que se conserva además el mensaje redactado:

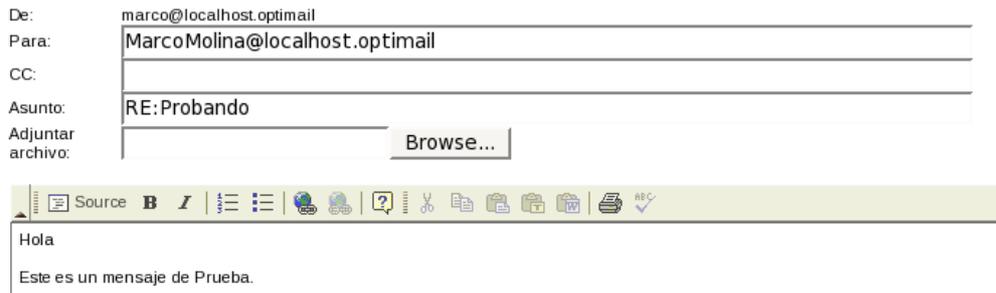


Imagen 5.21: Reenviando correo electrónico

5.3.8. Eliminar Correo

Para eliminar un correo electrónico de los buzones de enviados y bandeja de entrada se procede a hacer click en el ícono de eliminar (✕). El mensaje irá a la bandeja de borrados:

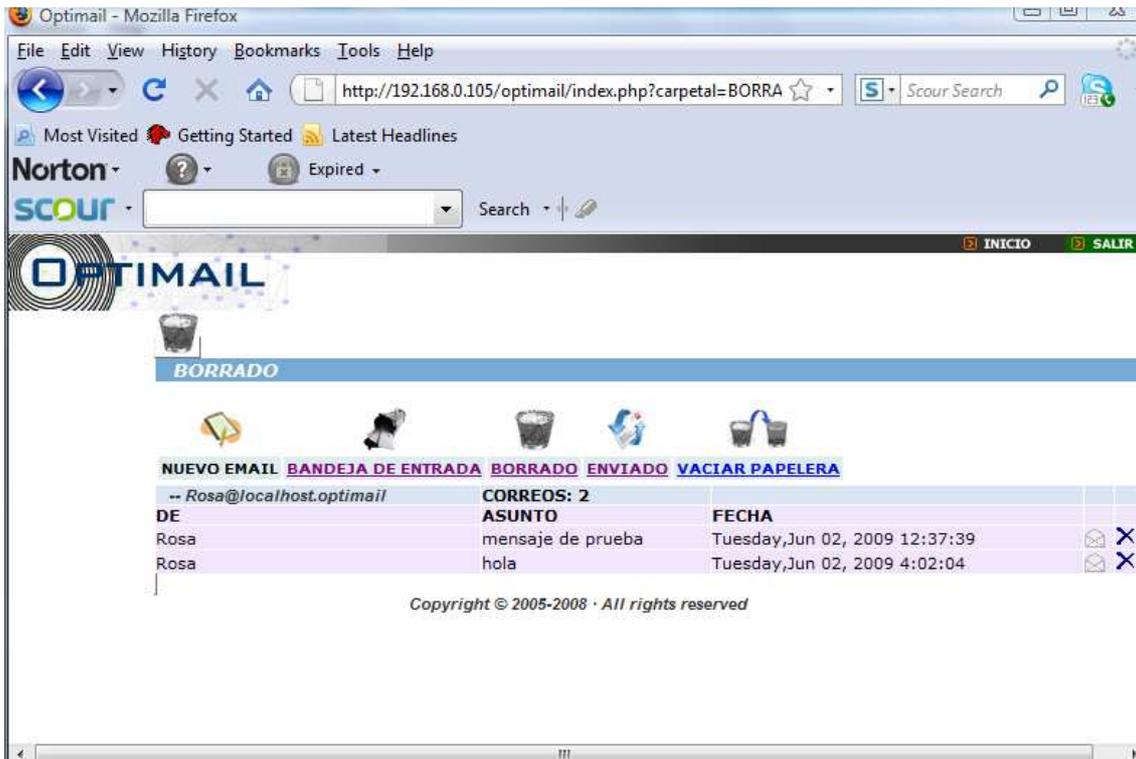


Imagen 5.22 Carpeta “Borrado”

Para vaciar la bandeja de borrados se hace click en la opción “VACIAR PAPELERA”. (Nota: Borrar un correo de la bandeja de “BORRADO” lo eliminará definitivamente).

CONCLUSIONES

- Se implementó la aplicación, OPTIMAIL, como un cliente del sistema de correo electrónico de LINUX utilizando los APIs de Linux de Dovecot y Postfix para el envío y recepción de correo electrónico.
- Se realizó el estudio correspondiente del Protocolo de Transferencia de Correo Simple (Simple Mail Transfer Protocol, SMTP) como un protocolo estándar para la transmisión de mensajes entre diferentes redes.
- Se configuró óptimamente el sistema operativo Linux de Fedora Core 7 siendo la base para la instalación del sistema de cliente de correo electrónico Optimail.
- El Protocolo de Administración de Correo Electrónico (*Post Office Protocol, POP*) utilizado para la recepción de mensajes de correo electrónico ha sido estudiado comprobando que los mensajes de correo electrónico se almacenan en el cliente.
- El estudio de IMAP (*Internet Message Access Protocol*) demuestra que es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor, por lo tanto, es utilizado para evitar la dispersión de mensajes en los clientes.
- Se utilizó el protocolo IMAP para la recepción de correo en lugar de POP ya que el servidor de correo es el que contiene todos los mensajes de correo electrónico y no el cliente además de que no causaría la pérdida de correo si el cliente fallara.
- Optimail es un cliente del sistema de correo electrónico, por lo tanto, puede ser accedido desde cualquier sistema operativo por medio de un browser.
- Se compiló el cliente de correo electrónico Optimail a un RPM utilizando BitRock Install Builder y subido a una página de correo electrónico de Rapidshare.

- El desarrollo de esta investigación despertará el interés de los profesionales de comunicaciones y promoverá la creación de software de comunicaciones, cada vez más sofisticado y completo, utilizando nuevas y existentes herramientas.

RECOMENDACIONES

- Proteger con contraseña fuerte al usuario root de la base de datos en mysql para su seguridad, caso contrario cualquiera podrá acceder a la Base de datos.
- Cambiar permisos al archivo Config.php después de realizar su configuración. Este contiene la contraseña del usuario root que sólo el administrador del sistema debe manipular.
- Optimizar el rendimiento de Optimail y del sistema operativo Linux en general actualizando los paquetes del sistema a su última y más reciente versión.
- No manipular ni modificar el código de programación de Optimail a menos que sepa exactamente lo que está haciendo.
- De manera general se recomienda utilizar como browser a Mozilla Firefox en lugar de Internet Explorer debido a su navegación que es mucho más rápida.
- Preferir Postfix a Sendmail ya que soporta Maildir para la administración de correo además de que es mucho más fácil la configuración y mantenimiento.
- Revisar la configuración del Firewall en Linux ya que es necesario que los puertos de envío y de recepción de mensajes de correo electrónico estén habilitados.
- Configurar los servicios de Mysql, Postfix, Dovecot y Apache con el comando chkconfig para que inicien por defecto sin necesidad de reactivarlos cada vez que se ingrese a Linux.
- Crear copias de los archivos originales que hayan sido modificados durante la configuración del sistema Operativo Linux para su utilización en caso de respaldo o equivocación.

- El servidor de correo electrónico debe tener características óptimas de Hardware para la administración de un número elevado de usuarios de Optimail.
- De manera general se recomienda actualizar los paquetes de Linux con Yum update para estar al día con el sistema operativo Linux.

BIBLIOGRAFÍA

- SÁNCHEZ, S.- GARCÍA, Ó. Linux. Guía Práctica, Editorial Ra-ma, 2008, 400 p.
- BAUER, MICHAEL D, O'REILLY VLG, Linux Server Security, 2a edición, GmbH & Co, 2005, 526p.
- PHILLIPS, JON A.; DAVIS, MICHELE E. Php Y Mysql, ANAYA MULTIMEDIA, 2008, 464 p.
- PÉREZ, C, Dreamweaver 8. Desarrollo De Páginas Web Dinámicas Con Php Y Mysql, Editorial Ra-ma, 2007, 464p.
- HESTER, N, Creating A Web Page In Dreamweaver 8: Visual Quickproject Guide, PEACHPIT PRESS, 2006, 160p.
- IÑAKI ALEGRÍA LOINAZ, ROBERTO CORTIÑAS RODRÍGUEZ, AITZOL EZEIZA RAMOS, Linux. Administración Del Sistema y La Red, Pearson Educación, 2005 ,336p.
- LUBANOVIC, BILL; ADELSTEIN, TOM, Administración De Sistemas Linux, ANAYA MULTIMEDIA, 2007, 336p.
- ANDREW TANENBAUM, redes de computadoras, 1990, 250p
- CRAIG LARMAN, UML y Patrones, Prentice Hall, 1999, 350p
- SCHRODER, CARLA, Redes En Linux. Guía De Referencia, ANAYA MULTIMEDIA, 2008, 720p.
- GARCÍA JIMÉNEZ, FCO. Guía de Campo de Linux, Editorial Ra-ma, 176p.
- MCCARTY, BILL O"REILLY VLG. GmbH & Co. 238p.
<http://www.linuxparatodos.net/portal/staticpages/index.php?page=04-disk-quota>
- <http://www.unixmexico.org/modules.php?name=News&file=comments&op=showreply&tid=194&sid=329&pid=189&mode=&order=&thold=#194>

- http://www.ecualug.org/?q=2007/oct/22/comos/c_mo_activar_quotas_en_tu_linux_centos
- <http://www.howtoforge.org/forums/showthread.php?p=195166>
- <http://www.ccm.itesm.mx/dinf/redes/sdns.html>
- <http://sendmail.softonic.com/linux>
- <http://notinet.blogspot.com/2007/11/el-futuro-es-el-correo-electronico.html>
- http://www.microsoft.com/spain/empresas/tecnologia/uso_correo_competitivo.msp
- <http://www.speccy.org/websromero/articulos/lxinet/lxinet3.html>
- <http://www.malavida.com/mvimag/download/webmin-464-1>
- http://www.howtoforge.com/perfect_server_fedora7
- <http://www.ole-web.net/spamassassin.asp>
- <http://es.tldp.org/Tutoriales/doc-guia-sendmail/doc-guia-sendmail-html/>
- <http://www.postfix.org/>
- <http://www.enetchile.cl/productos/productoslinux/sercorreo.php>
- http://www.howtoforge.com/perfect_server_fedora7_p1
- http://www.howtoforge.com/perfect_server_fedora7_p2
- http://www.howtoforge.com/perfect_server_fedora7_p3
- http://www.howtoforge.com/perfect_server_fedora7_p4
- http://www.howtoforge.com/perfect_server_fedora7_p5
- http://help.adobe.com/en_US/Dreamweaver/10.0_Using/index.html
- <http://communities.vmware.com/help.jspa>
- <https://help.ubuntu.com/community/VMware>
- <http://www.linuxhelp.net/guides/vmware/>
- <http://www.vmware.com/help/>
- <http://www.servitux.org/view.php/page/postfix>

- http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/martinez_v_lm/capitulo5.pdf

ANEXOS

ANEXO A: DIRECTORIO RAÍZ

A-1: Index.php

Página principal de Optimail en la que valida el inicio de sesión al sistema

```
<?php
//Encabezado
if (!(session_is_registered ('nov_ususer')==true))
{
session_cache_limiter('nocache,private');
session_name('sisnovus');
session_start();
session_register('nov_ususer');
session_register('nov_pwd');
session_register('nov_name');
session_register('verif');
session_register('buscardb');
session_register('oficinausu');
session_register('iduser');
}
?>
<?
/**VARIABLES POR GET ***/
```

```

$numero = count($_GET);
$tags = array_keys($_GET);// obtiene los nombres de las variables
$valores = array_values($_GET);// obtiene los valores de las variables

// crea las variables y les asigna el valor
for($i=0;$i<$numero;$i++){
    $$tags[$i]=$valores[$i];
}

/**VARIABLES POR POST */

$numero2 = count($_POST);
$tags2 = array_keys($_POST); // obtiene los nombres de las variables
$valores2 = array_values($_POST);// obtiene los valores de las variables

// crea las variables y les asigna el valor
for($i=0;$i<$numero2;$i++){
    $$tags2[$i]=$valores2[$i];
}

?>
<?php
//Llamando objetos
include("cfgclases/config.php");
include("cfgclases/clases.php");

$link = $objBd->enlace;

if ($close==1)
{

```

```

$_SESSION['nov_ususer']="";
$_SESSION['nov_pwd']="";
$_SESSION['nov_name']="";
$_SESSION['oficinausu']="";
$_SESSION['iduser']="";
}
$objacceso_session->select_session($sessid);
$objacceso_session->select_perfil($objacceso_session->sess_perid);
if ($objacceso_session->sess_user and $_SESSION['nov_ususer'])
{
$permisotabla=strchr($objacceso_session->sess_tabla,strval("rx".$table));
if ($permisotabla)
{
$table="";
}
$objformulario->geamv =$geamv;
if (!($objformulario->geamv))
{

include($objtemplate->path_template."index.php");
}
else
{
$objtemplate->path_template="templates/base/";
include($objtemplate->path_template."index.php");
}
}
else
{
//-----
if ($ocacceso==1)

```

```

{
$objjacceso_system->accept_session($sisusu,$pwdusu);
$ssid=$objjacceso_session->create_session($objjacceso_system-
>user,$objjacceso_system->pass,$objjacceso_system->name,$objjacceso_system->perid);

if ($objjacceso_system->user)
{
$_SESSION['nov_ususer']=$objjacceso_system->user;
$_SESSION['nov_pwd']=$objjacceso_system->pass;
$_SESSION['nov_name']=$objjacceso_system->name;
$_SESSION['oficinausu']=$objjacceso_system->oficina;
$_SESSION['iduser']=$objjacceso_system->iduser;
include("acces.php");

}
else
{
$mensajeacc="Acceso negado...";
include("modules/acces.php");
}
}
else
{
include("modules/acces.php");
}
//-----
}
?>

```

A-2 Formularioereg.php

Contiene el formulario que despliega al crear un nuevo usuario de Optimail

```
<?
/**VARIABLES POR GET ***/

$numero = count($_GET);
$tags = array_keys($_GET);// obtiene los nombres de las variables
$valores = array_values($_GET);// obtiene los valores de las variables

// crea las variables y les asigna el valor
for($i=0;$i<$numero;$i++){
    $$tags[$i]=$valores[$i];
}

/**VARIABLES POR POST ***/

$numero2 = count($_POST);
$tags2 = array_keys($_POST); // obtiene los nombres de las variables
$valores2 = array_values($_POST);// obtiene los valores de las variables

// crea las variables y les asigna el valor
for($i=0;$i<$numero2;$i++){
    $$tags2[$i]=$valores2[$i];
}

?>
<?php
```

```

include("libreria/dbcc.php");
include("libreria/formulario.php");
include("cfgclases/config.php");
//Conexion a la base de datos
$objBd = new conecc();
$objformulario = new formulario();
$objBd->conectardb($basededatos,$host,$userdb,$passwdb);
$link = $objBd->enlace;

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
.style1 {      font-size: 11px;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-weight: bold;
color: #006666;
}
.style2 {
font-size: 11px;
font-family: Verdana, Arial, Helvetica, sans-serif;
color: #000033;
}
.style7 {font-size: 11px; font-family: Verdana, Arial, Helvetica, sans-serif; font-weight:
bold; }

```

```

.style8 {
font-size: 11px;
font-family: Verdana, Arial, Helvetica, sans-serif;
}
-->
</style>
</head>

<body>

<?php
if (!($vle))
{
echo '<div align="center" class="style1"><a
href="formularioeg.php?vle=1">Reg&iacute;strese como nuevo usuario </a></div>';
}
?>
<?php
if ($vle==1)
{
?>
<form id="form1" name="form1" method="post" action="formularioeg.php?vle=2">
<table width="300" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td bgcolor="#DBEAF0"><div align="center" class="style2">Formulario de registro
</div></td>
</tr>
<tr>
<td><table width="100%" border="0" cellpadding="0" cellspacing="2">
<tr>
<td width="55%" bgcolor="#F0F7F5"><span class="style7">Usuario</span></td>

```

```

<td width="45%" bgcolor="#F0F7F5"><input name="usuario" type="text" class="style8"
id="usuario" /></td>
</tr>
<tr>
<td bgcolor="#F0F7F5"><span class="style7">Clave:</span></td>
<td bgcolor="#F0F7F5"><input name="clave" type="password" class="style8" id="clave"
/></td>
</tr>
<tr>
<td bgcolor="#F0F7F5"><span class="style7">Confirmaci&oacute;n de clave:
</span></td>
<td bgcolor="#F0F7F5"><input name="clave1" type="password" class="style8"
id="clave1" /></td>
</tr>
<tr>
<td bgcolor="#F0F7F5"><span class="style7">Nombre y apellidos:</span></td>
<td bgcolor="#F0F7F5"><input name="nombre" type="text" class="style8" id="nombre"
/></td>
</tr>
<tr>
<td bgcolor="#F0F7F5"><span class="style7">Direccion:</span></td>
<td bgcolor="#F0F7F5"><textarea name="direccion" class="style8"
id="direccion"></textarea></td>
</tr>
<tr>
<td bgcolor="#F0F7F5"><span class="style7">Tel&eacute;fono:</span></td>
<td bgcolor="#F0F7F5"><input name="telefono" type="text" class="style8" id="telefono"
/></td>
</tr>
<tr>
<td bgcolor="#F0F7F5"><span class="style7">Email alterno: </span></td>

```

```

<td bgcolor="#F0F7F5"><input name="email" type="text" class="style8" id="email"
/></td>
</tr>
</table>
<label>
<div align="center">
<input name="Submit" type="submit" class="style8" value="Registrame" />
</div>
</label></td>
</tr>
</table>
</form>
<?php
}
?>

<?php
if ($vle==2)
{
$selecTablaing="insert into opt_sisusers
(sisu_usu,sisu_pwd,sisu_name,sisu_direccion,sisu_telefono,sisu_email,per_id) values
('$usuario','$clave','$nombre','$direccion','$telefono','$email',11)";

$resultadoing = mysql_query($selecTablaing);

$connection = ssh2_connect($ip, 22);
//ssh2_auth_password($connection, 'root','13272713');
ssh2_auth_password($connection, $userdb,$passwdb);

$password="\$1\$KtO9pODg\$embAldqf7x\.t6eHj2Y72R0";
$stream = ssh2_exec($connection , 'useradd -m -p '.$password.' '.$usuario);

```

```
echo 'Usuario creado con exito...';  
}  
?>  
</body>  
</html>
```

ANEXO B: CFGCLASES

Contiene todas las clases creadas para el funcionamiento de Optimail.

B-1 Clases.php

Archivo de enlace con todas las clases que se usan en el sistema.

```
<?php

if ($director)
{
$director=$director."/";
}
else
{
$director="";
}

//Clases para el sistema
include($director."libreria/dbcc.php");
include($director."libreria/formulario.php");
include($director."libreria/menu.php");
include($director."libreria/acces.php");
include($director."libreria/session.php");
include($director."libreria/varsend.php");
include($director."libreria/template.php");
include($director."libreria/templateform.php");
include($director."libreria/estadisticas.php");
include($director."libreria/class.paginaZ.php");
```

```
include($director."libreria/botones.php");
include($director."libreria/parametros.php");
include($director."libreria/ayuda.php");
include($director."libreria/receivemail.class.php");
include($director."libreria/attachmentread.class.php");
include($director."fckeditor.php") ;
```

```
//Utilizando Librerias
```

```
//Base de datos
```

```
$objBd = new conec();
```

```
//Formulario
```

```
$objformulario = new formulario();
```

```
$objformulario->ptaeditor=$ptaeditor;
```

```
//Menu General
```

```
$objmenu = new menu();
```

```
$objmenu->submenuact=$smenu;
```

```
//Conectando a la base de datos
```

```
$objBd->conectardb($basededatos,$host,$userdb,$passwdb);
```

```
//Sistemas
```

```
$objacceso_system = new acceso_system();
```

```
//Sesiones
```

```
$objacceso_session = new session_system();
```

```
//Template General
```

```
$objtemplate = new template();
```

```
$objtemplate->select_template();
```

```
//Templates Formularios de tablas
```

```
$objtableform= new templateform();
```

```
if ($stable)
```

```
{
```

```

$objtableform->select_templateform($stable);
}
//Botones
$objbotones= new boton_aqualis();
//Parametros generales
$objparametros= new parametros_generales();
$objparametros->parametros();

//Ayuda
$objayuda=new ayuda_aqualis();
$objayuda->desplegar_ayuda($stable);
?>

```

B-2 Clasesmod.php

Clases para módulos

```

<?php
$director="../";

//Clases para el sistema
include($director."libreria/dbcc.php");
include($director."libreria/formulariog.php");
include($director."libreria/menu.php");
include($director."libreria/acces.php");
include($director."libreria/session.php");
include($director."libreria/varsend.php");
include($director."libreria/template.php");
include($director."libreria/templateform.php");
include($director."fckeditor.php") ;

```

```
// include($director."fckeditor.php") ;  
?>
```

B-3 Config.php

Archivo que contiene el código de conexión a la base de datos y al sistema. Sus parámetros deben ser modificados para su correcto funcionamiento. Path editor debe quedar configurado por defecto de esa manera para que no se ingrese todo el path en el browser para ingresar a la aplicación.

```
<?php  
$basededatos="optimaildb"; // Nombre de la base de datos de Optimail  
$host="localhost";        // Nombre del host  
$userdb="root";          // Usuario que se conectará a la base de datos  
$passwdb="13272713";     // contraseña del usuario  
$ip='192.168.1.9';       // IP local utilizado para la conexión  
//Path editor  
$ptaeditor="/optimail/";  //ubicación de Optimail (var/www/html)  
?>
```

ANEXO C: LIBRERÍA

C-1 Access.php

Clase de acceso al sistema.

```
<?php
class acceso_system{
var $resultado;
var $user;
var $pass;
var $name;
var $perid;
var $oficina;
var $iduser;

function acept_session($sisusu,$pwdusu)
{
$sisusu = addslashes(trim($sisusu));
$pwdusu = addslashes(trim($pwdusu));
$selecmenu="select * from opt_sisusers,opt_perfil where
opt_sisusers.per_id=opt_perfil.per_id and sisu_usu like '$sisusu' and sisu_pwd like
'$pwdusu'";
$resultado = mysql_query($selecmenu);
while($row = mysql_fetch_array($resultado))
{
```

```

$this->user=$row[sisu_usu];
$this->iduser=$row[sisu_id];
$this->pass=$row[sisu_pwd];
$this->name=$row[sisu_name];
$this->perid=$row[per_id];
$this->oficina=$row[cod_oficina];
}
}

} ?>

```

C-2 Aplicativo.php

Clase para dar funcionalidad a los sub-aplicativos.

```

<?php
//panel de manejo para aplicaciones
if ($apl)
{
$sqlnom = "select * from opt_applicationadm where ap_id=$apl";
$resultnom = mysql_query($sqlnom);
if ($resultnom)
{
while($rownom = mysql_fetch_array($resultnom))
{
$ap_path=$rownom[ap_path];
$ap_activo=$rownom[ap_activo];
$ap_protec=$rownom[ap_protec];
}
}
}

```

```

mysql_free_result($resultnom);
}

if ($ap_activo==1)
{
if ($ap_protec==1)
{
if ($sessid)
{
include($ap_path."index.php");
}
else
{
echo "<br><br><br><br><center><div class=titulo>Para ingresar a esta seccion debe estar
registrado</div>";
printf("<a href='indexr.php?system=%s' target='_top'><img
src=aqualisv3/logotipo/logoaqualis.png border=0><img src=aqualisv3/logotipo/cube.gif
border=0></a></center><br><br><br><br>", $system);
}

}
else
{
include($ap_path."index.php");
}

}
else
{
include("msg/inactive.php");
}

```

```
}  
  
}  
else  
{  
include("msg/inactive.php");  
  
} ?>
```

C-3 attachmentread.class.php

Clase para realizar proceso de adjunto de archivos.

```
class readattachment  
{  
  
function getdecodevalue($message,$coding)  
{  
if ($coding == 0)  
{  
$message = imap_8bit($message);  
}  
elseif ($coding == 1)  
{  
$message = imap_8bit($message);  
}  
elseif ($coding == 2)  
{  
$message = imap_binary($message);  
}  
}
```

```

elseif ($coding == 3)
{
$message=imap_base64($message);
}
elseif ($coding == 4)
{
$message = imap_qprint($message);
}
elseif ($coding == 5)
{
$message = imap_base64($message);
}
return $message;
}

```

```

function getdata($host,$login,$password,$savedirpath)
{
$mbox = imap_open ($host, $login, $password) or die("can't connect: " .
imap_last_error());
$message = array();
$message["attachment"]["type"][0] = "text";
$message["attachment"]["type"][1] = "multipart";
$message["attachment"]["type"][2] = "message";
$message["attachment"]["type"][3] = "application";
$message["attachment"]["type"][4] = "audio";
$message["attachment"]["type"][5] = "image";
$message["attachment"]["type"][6] = "video";
$message["attachment"]["type"][7] = "other";

for ($j = 1; $j <= imap_num_msg($mbox); $j++)
{

```

```

$structure = imap_fetchstructure($mbox, $jk , FT_UID);
$parts = $structure->parts;
$fpos=2;
for($i = 1; $i < count($parts); $i++)
{
$message["pid"][$i] = ($i);
$part = $parts[$i];

if($part->disposition == "ATTACHMENT")
{

$message["type"][$i] = $message["attachment"]["type"][$part->type] . "/" .
strtolower($part->subtype);
$message["subtype"][$i] = strtolower($part->subtype);
$ext=$part->subtype;
$params = $part->dparameters;
$filename=$part->dparameters[0]->value;

$mege="";
$data="";
$message = imap_fetchbody($mbox,$jk,$fpos);
$filename="$filename";
$fp=fopen($filename,w);
$data=$this->getdecodevalue($mege,$part->type);
fputs($fp,$data);
fclose($fp);
$fpos+=1;
}
}

//imap_delete tags a message for deletion
//imap_delete($mbox,$jk);

```

```
}  
// imap_expunge deletes all tagged messages  
//imap_expunge($mbox);  
imap_close($mbox);  
}}  
?>
```

C-4 Class.página.Z.php

Colores, apariencia, formato y estética de Optimail.

```

/*****
**/
class sistema_paginacion{
/*****

DATOS A CONFIGURAR
/*****/

/* Colores - Configurable a gusto del consumidor :)* */

var $color_link_inactivo="ffffff"; /* Color del enlace cuando el ratón no esta por encima
*/
var $color_link_activo="ffffff"; /* Color del enlace al pasar el ratón por encima */
var $color_texto="ffffff"; /* Color del texto normal (Que no es enlace) */
var $color_tablas="#000000"; /* Color de fondo de la tabla que contiene los enlaces */

/*****
*/
/* MUY IMPORTANTE!!!: NO MODIFICAR NADA A PARTIR DE ESTA LINEA
*/
/*****/
**/
/* Datos para la creación de las paginas ¡NO MODIFICAR! */

var $numero_por_pagina; /* Número de registros a mostrar por página */
var $numero_paginas;
var $total;
```

```

var $condiciones;      /* Condiciones para realizar el query ...p.e. where visible=1.. */
var $id_inicio=0;
/* Datos para la conexion a la base de datos (Deben ser modificados) */
var $nombre_tabla;    /* Esta variable se establece al crear la clase */
var $campo_ordenacion; /* Campo de la tabla por el que se ordenan los resultados */

/* Error */

var $error=0;
var $estilos_creados=0;
var $url;

/***** METODOS DE LA CLASE *****/

// Constructor -> establece el nombre de tabla donde consultar
function sistema_paginación($tabla)
{
    global $id_inicio;

    $this->nombre_tabla=$tabla;
    // $this->conectar();

    if(isset($id_inicio))
    $this->id_inicio=$id_inicio;
    else
    $this->id_inicio=0;

}
function crear_estilos()
{

```

```

echo "<style type='text/css'>
a.paginaZ
{
color: $this->color_link_inactivo;
font-size: 11px;
font-family: Verdana, Arial, Helvetica, sans-serif;
text-decoration: none;
}
a.paginaZ:hover
{
color: $this->color_link_activo;
font-size: 11px;
font-family: Verdana, Arial, Helvetica, sans-serif;
text-decoration: none;
}
</style>";
}
// Constructor -> establece el nombre de tabla y las condiciones para la consulta
function set_condicion($cond)
{
$this->condiciones=$cond;
}

// Obtiene la pagina en la que estamos a partir del id_inicio
function obtener_pagina_actual()
{
$pagina_actual=($this->id_inicio/$this->numero_por_pagina)+1;
return($pagina_actual);
}
/* Establece el campo por el que se realizara la ordenacion de registros */
function ordenar_por($campo)

```

```

{
$this->campo_ordenacion=$campo;
}
/* Obtiene el numero total de registros a paginar (No modificar!) */
function obtener_total()
{
$query="SELECT count(*) as total from $this->nombre_tabla ";

if ($this->condiciones!="")
{
$query.=" $this->condiciones";
}
mysql_select_db($this->sql_db);
$result=mysql_query($query);
$row=mysql_fetch_object($result);
$this->total=$row->total;
}
// Obtiene el numero de paginas a crear
function obtener_numero_paginas()
{
$this->obtener_total();

$this->numero_paginas=$this->total/$this->numero_por_pagina;

// Si hay alguna página con menos del maximo tb se añade
if (($this->total%$this->numero_por_pagina)>0)
$this->numero_paginas++;

$this->numero_paginas=floor($this->numero_paginas);

}

```

```

//Establece un número máximo de elementos por pagina
function set_limite_pagina($num)
{
$this->numero_por_pagina=$num;
$this->obtener_numero_paginas();
}

/* Obtiene la url donde enlazar (NO MODIFICAR!!!!) */
function obtener_url()
{
global $_GET;
global $_POST;

while (list ($clave, $val) = each ($_GET)) {
if($clave!="id_inicio")
$variables .= $clave."=".$val."&";
if($clave=="table")
$bandet=1;
if($clave=="sessid")
$bandes=1;
}

if(strpos($variables, "&")) $pag_devuelta = $HTTP_REFERER."?". $variables;
else $pag_devuelta = $HTTP_REFERER."?";

if (!$bandet)
$vari1="table=".$_POST['table'];
if (!$bandes)
$vari1.="&sessid=".$_POST['sessid']."&";

$this->url=$pag_devuelta.$vari1;

```

```

}

// Devuelve una variable $result con los resultados de la consulta
function obtener_consulta()
{

mysql_select_db($this->sql_db);
$query="SELECT * from $this->nombre_tabla ";
if ($this->condiciones!="") $query.=" $this->condiciones";
if($this->campo_ordenacion!="") $query.=" order by $this->campo_ordenacion desc";
$query.=" limit $this->id_inicio,$this->numero_por_pagina";
$result=mysql_query($query);
//echo $query;
return($result);
}

function error($num)
{
$this->error=$num;
switch($num)
{
case 1:echo "Error al conectar a la BD";break;
case 2:break;
case 3:break;
case 4:break;
}
}

function set_color_tabla($color)
{
$this->color_tablas=$color;
}

```



```

echo "

&nbsp;&nbsp;&nbsp;<b>Páginas
";

for($i=0;$i<=$this->numero_paginas-1;$i++)
{
$numero_inicial=$i*$this->numero_por_pagina;
$numero_final=$numero_inicial+$numero_por_pagina;
$enlace_num=$this->url."id_inicio=$numero_inicial";
$pagina=$i+1;
echo " - ";

if($pagina_actual!=$pagina)
echo " <a class='csslista' href='$enlace_num' >$pagina</a> ";
else
echo "<u>$pagina</u>";
}
/* Mostramos el enlace de >>Siguiete si es necesario */

$numero_siguiete=$this->id_inicio+$this->numero_por_pagina;
$enlace_sig=$this->url."id_inicio=$numero_siguiete";

if(isset($this->id_inicio)&&($this->id_inicio+$this->numero_por_pagina<$this->total))
echo "&nbsp;&nbsp;&nbsp;<a class='csslista' href='$enlace_sig' >&gt;&gt; Siguiete </a>";

echo "&nbsp;&nbsp;&nbsp;</td>
</table>";
}} ?>

```

C-5 Dbcc.php

Esta clase contiene el código de conexión a la base de datos MYSQL. Despliega mensajes de error en caso de no realizarse la conexión correctamente. Es muy importante que exista el enlace hacia la base de datos para que Optimail funcione desde el inicio de sesión.

```
<?php
/*
#####
#    Conexión a la base de datos
#
#####
*/

//Objeto conexión, Conecta a la base de datos Mysql
class conecc{
//Atributos Básicos de la clase
var $servidor; //host
var $nombredb; //Nombre de la Base de Datos
var $nombreu; //Nombre del usuario autorizado para entrar a la Base de Datos

//Atributos Modificados
var $enlace; //Almacena el enlace con la Base de Datos una vez establecido
var $resultado; //Almacena el resultado obtenido por la consulta a la BD
var $consulta; //Almacena la consulta realizada con el método consultaBD();

//Usuarios
var $usuario;
var $clave;
var $privilegios;
var $descuento;
```

```

//constructor
Function constructor($servidor,$nombredb,$nombreu)
{
$this->servidor=$servidor;
$this->nombredb=$nombredb;
$this->nombreu=$nombreu;
}
//Función de conexión
function conectardb($basededatos,$host,$userdb,$passwd)
{
if($this->enlace=mysql_connect($host,$userdb,$passwd))
//if($this->enlace=mysql_connect("localhost","sistema_unlugar_com","sistemadb"))
{
if(mysql_select_db($basededatos,$this->enlace)) //seleccionabd
{

}
else
{
echo "Error al seleccionar la base de datos!";
exit();
}
}
else
{
echo "Error al enlazar al Servidor!";
exit();
}}}
?>

```

C-6 Func_g.php

Contiene funciones de actualizar, borrar, guardar y buscar en las tablas

```
<?php
switch ($opcion)
{
case "guardar":
{
$objformulario->tableant=$tableant;
$objformulario->tableant1=$tableant1;
$objformulario->campoant=$campoant;
$objformulario->opcp=$opcp;
$objformulario-
>formulario_guardar($table,$_POST,$typesql,$varsend,$listab,$campo,$obp);

}
break;
case "actualizar":
{
$objformulario->tableant=$tableant;
$objformulario->tableant1=$tableant1;
$objformulario->campoant=$campoant;
$objformulario->opcp=$opcp;
$objformulario-
>formulario_update($table,$_POST,$typesql,$idab,$varsend,$listab,$campo,$obp);
}
case "buscar":
{
$objformulario->vatajo=$csearch;
```

```

$objformulario->formulario_buscar($table,$csearch,$varsend,$listab,$campo,$obp);

}
break;
case "borrar":
{
$objformulario->tableant=$tableant;
$objformulario->tableant1=$tableant1;
$objformulario->campoant=$campoant;
$objformulario->opcp=$opcp;
$objformulario->formulario_delete($table,$idab,$varsend,$listab,$campo,$obp);
}
break;
default:
}
?>

```

C-7 Menú.php

Contiene la clase del menú

```

<?php
class menu{
var $resultado;
var $submenuact;

function display_menu($menu,$varsend,$menuperfil,$imenuperfil)
{

$permiso=strchr($menuperfil,strval("-".$menu."-"));

```

```

if (!($permiso))
{
$selecmenu1="select * from opt_menu where men_id=$menu";
$resultado1 = mysql_query($selecmenu1);
while($row1 = mysql_fetch_array($resultado1))
{
$titulo=$row1[men_titulo];
$stylot=$row1[men_style];
$activem=$row1[men_active];
$tipom=      $row1[men_type];
}
if ($activem==1)
{
if ($tipom==1)
{
//Horizontal
$selecmenu="select * from opt_menu,opt_itemmenu where
opt_menu.men_id=opt_itemmenu.men_id and opt_menu.men_id=$menu order by ite_order
asc";
$resultado = mysql_query($selecmenu);
$por="% ";
printf("<table border='0' cellspacing='0' cellpadding='1' width='100%'><tr><td><div
id='%s'><ul>", $por, $stylot);
while($row = mysql_fetch_array($resultado))
{
$ipermiso=strchr($imenuperfil, strval("-".$row[ite_id]."-"));
if (!($ipermiso))
{
if ($row[ite_active]==1)
{
if ($row[ite_icono])

```

```

{

if ($row[ite_link])
{
printf("<li><a %s href='opcp=%s%s' target='_top' id='%s'><img src='%s'
border=0>%s</a></li>", $row[ite_extra], $row[ite_link], $varsend, $row[ite_style], $row[ite_i
cono], $row[ite_titulo]);
}
else
{
if ($row[ite_tipd]==1)
{
printf("<li><a %s href='index.php?table=%s%s' target='_top' id='%s'><img src='%s'
border=0>%s</a></li>", $row[ite_extra], $row[ite_linktable], $varsend, $row[ite_style], $row
[ite_icono], $row[ite_titulo]);
}
else
{
printf("<li><a %s href='index.php?tablelista=%s%s' target='_top' id='%s'><img src='%s'
border=0>%s</a></li>", $row[ite_extra], $row[ite_linktable], $varsend, $row[ite_style], $row
[ite_icono], $row[ite_titulo]);
}
}
}
else
{

if ($row[ite_link])
{

```

```

printf("<li><a %s href='opcp=%s%s' target='_top'
id='%s'>%s</a></li>", $row[ite_extra], $row[ite_link], $varsend, $row[ite_style], $row[ite_titulo]);
}
else
{

if ($row[ite_tipd]==1)
{
printf("<li><a %s href='index.php?table=%s%s' target='_top'
id='%s'>%s</a></li>", $row[ite_extra], $row[ite_linktable], $varsend, $row[ite_style], $row[ite_titulo]);
}
else
{
printf("<li><a %s href='index.php?tablelista=%s%s' target='_top'
id='%s'>%s</a></li>", $row[ite_extra], $row[ite_linktable], $varsend, $row[ite_style], $row[ite_titulo]);
}}}}}}
printf("</ul></div></td></tr></table>");
}
else
{
//Vertical

$selectmenu="select * from opt_menu,opt_itemmenu where
opt_menu.men_id=opt_itemmenu.men_id and opt_menu.men_id=$menu order by ite_order
asc";
$resultado = mysql_query($selectmenu);
printf("<table border='0' cellspacing='1' cellpadding='1' width='100%'>", $por);
printf("<tr><td><div align='center'></div></td></tr><tr><td><ul id='%s'>", $stytot);

```



```

}
else
{
printf("<li><a %s href='index.php?opcp=7&apl=%s%s' target='_top' class='%s'><img
src='%s'
border=0>%s</a></li>", $row[ite_extra], $row[ite_link], $varsend, $row[ite_style], $row[ite_i
cono], $row[ite_titulo]);
}
//Fin submenus
}
}
else
{
if ($row[ite_tipd]==1)
{
printf("<li><a %s href='index.php?table=%s%s' target='_top' class='%s'><img src='%s'
border=0>%s</a></li>", $row[ite_extra], $row[ite_linktable], $varsend, $row[ite_style], $row
[ite_icono], $row[ite_titulo]);
}
else
{
printf("<li><a %s href='index.php?tablelista=%s%s' target='_top' class='%s'><img src='%s'
border=0>%s</a></li>", $row[ite_extra], $row[ite_linktable], $varsend, $row[ite_style], $row
[ite_icono], $row[ite_titulo]);
}}}
else
{
if ($row[ite_link])
{

```

```

if ($row[ite_link]=="#")
{
printf("<li><a %s href='#' target='_top'
class='%s'>%s</a></li>", $row[ite_extra], $row[ite_style], $row[ite_titulo]);
}
else
{

//Para submenus

if ($row[ite_link]=="##")
{

printf("<li><a %s href='index.php?smenu=%s%s' target='_top'
class='%s'>%s</a></li>", $row[ite_extra], $row[ite_submenu], $varsend, $row[ite_style], $row[ite_titulo]);
if ($row[ite_submenu]==$this->submenuact)
{
$this->display_submenu($row[ite_submenu], $varsend, $menuperfil, $imenuperfil);
}
}
else
{
printf("<li><a %s href='index.php?opcp=7&apl=%s%s' target='_top'
class='%s'>%s</a></li>", $row[ite_extra], $row[ite_link], $varsend, $row[ite_style], $row[ite_titulo]);
}
//Fin submenus
}
}
else

```

```

{
if ($row[ite_tipd]==1)
{
printf("<li><a %s href='index.php?table=%s%s' target='_top'
class='%s'>%s</a></li>", $row[ite_extra], $row[ite_linktable], $varsend, $row[ite_style], $ro
w[ite_titulo]);
}
else
{
printf("<li><a %s href='index.php?tablelista=%s%s' target='_top'
class='%s'>%s</a></li>", $row[ite_extra], $row[ite_linktable], $varsend, $row[ite_style], $ro
w[ite_titulo]);
}}}}
}
printf("</ul></td></tr></table>");
}}}}
//Funcion duplicada para submenus

```

```

function display_submenu($menu, $varsend, $menuperfil, $imenuperfil)
{
$permiso=strchr($menuperfil, strval("-".$menu."-"));
if (!$permiso)
{
$selecmenu1="select * from opt_menu where men_id=$menu";
$resultado1 = mysql_query($selecmenu1);
while($row1 = mysql_fetch_array($resultado1))
{
$titulo=$row1[men_titulo];
$stylot=$row1[men_style];
$activem=$row1[men_active];
$tipom=      $row1[men_type];

```

```

}

if ($activem==0)
{
if ($tipom==1)
{
//Horizontal
$selecmenu="select * from opt_menu,opt_itemmenu where
opt_menu.men_id=opt_itemmenu.men_id and opt_menu.men_id=$menu order by ite_order
asc";
$resultado = mysql_query($selecmenu);
$por="%";
printf("<table border='0' cellspacing='0' cellpadding='1' width='100%'><tr><td><div
id='%s'><ul>", $por, $stylot);
while($row = mysql_fetch_array($resultado))
{
$permiso=strchr($menuperfil, strval("-".$row[ite_id]."-"));
if (!$permiso)
{
if ($row[ite_active]==1)
{
if ($row[ite_icono])
{

if ($row[ite_link])
{
printf("<li><a %s href='opcp=%s%s' target='_top' id='%s'><img src='%s'
border=0>%s</a></li>", $row[ite_extra], $row[ite_link], $varsend, $row[ite_style], $row[ite_i
cono], $row[ite_titulo]);
}
}
}
}
}
}
}
}
}

```

```

else
{
if ($row[ite_tipd]==1)
{
printf("<li><a %s href='index.php?table=%s%s' target='_top' id='%s'><img src='%s'
border=0>%s</a></li>", $row[ite_extra], $row[ite_linktable], $varend, $row[ite_style], $row
[ite_icono], $row[ite_titulo]);
}
else
{
printf("<li><a %s href='index.php?tablelista=%s%s' target='_top' id='%s'><img src='%s'
border=0>%s</a></li>", $row[ite_extra], $row[ite_linktable], $varend, $row[ite_style], $row
[ite_icono], $row[ite_titulo]);
}}}

else
{

if ($row[ite_link])
{
printf("<li><a %s href='opcp=%s%s' target='_top'
id='%s'>%s</a></li>", $row[ite_extra], $row[ite_link], $varend, $row[ite_style], $row[ite_tit
ulo]);
}
else
{

if ($row[ite_tipd]==1)
{

```

```

printf("<li><a %s href='index.php?table=%s%s' target='_top'
id='%s'>%s</a></li>", $row[ite_extra], $row[ite_linktable], $varsend, $row[ite_style], $row[it
e_titulo]);
}
else
{
printf("<li><a %s href='index.php?tablelista=%s%s' target='_top'
id='%s'>%s</a></li>", $row[ite_extra], $row[ite_linktable], $varsend, $row[ite_style], $row[it
e_titulo]);
}}}}}}

printf("</ul></div></td></tr></table>");
}
else
{
//Vertical

$selecmenu="select * from opt_menu,opt_itemmenu where
opt_menu.men_id=opt_itemmenu.men_id and opt_menu.men_id=$menu order by ite_order
asc";
$resultado = mysql_query($selecmenu);
printf("<table border='0' cellspacing='1' cellpadding='1' width='100%'>", $por);
printf("<tr><td><div align='center'></div></td></tr><tr><td><ul id='%s'>", $stylot);
while($row = mysql_fetch_array($resultado))
{

$permiso=strchr($menuperfil, strval("-".$row[ite_id]."-"));
if (!$permiso)
{
if ($row[ite_active]==1)
{

```

```

if ($row[ite_icono])
{
if ($row[ite_link])
{
if ($row[ite_link]=="#")
{
printf("<li><a %s href='#' target='_top' class='%s'><img src='%s'
border=0>%s</a></li>", $row[ite_extra], $row[ite_style], $row[ite_icono], $row[ite_titulo]);
}
else
{
//Para submenuas
if ($row[ite_link]=="##")
{
printf("<li><a %s href='index.php?smenu=%s&opcp=7&apl=%s%s' target='_top'
class='%s'><img src='%s'
border=0>%s</a></li>", $row[ite_extra], $row[ite_submenu], $row[ite_link], $varsend, $row[
ite_style], $row[ite_icono], $row[ite_titulo]);
if ($row[ite_submenu]==$this->submenuact)
{
echo $row[ite_submenu];
}
}
else
{
printf("<li><a %s href='index.php?opcp=7&apl=%s%s' target='_top' class='%s'><img
src='%s'
border=0>%s</a></li>", $row[ite_extra], $row[ite_link], $varsend, $row[ite_style], $row[ite_i
cono], $row[ite_titulo]);

```

```

}
//Fin submenus
}
}
else
{
if ($row[ite_tipd]==1)
{
printf("<li><a %s href='index.php?smenu=%s&table=%s%s' target='_top' class='%s'><img
src='%s' border=0>%s</a></li>", $row[ite_extra], $this-
>submenuact, $row[ite_linktable], $varsend, $row[ite_style], $row[ite_icono], $row[ite_titulo]
);
}
else
{
printf("<li><a %s href='index.php?smenu=%s&tablelista=%s%s' target='_top'
class='%s'><img src='%s' border=0>%s</a></li>", $row[ite_extra], $this-
>submenuact, $row[ite_linktable], $varsend, $row[ite_style], $row[ite_icono], $row[ite_titulo]
);
}}
else
{

if ($row[ite_link])
{
if ($row[ite_link]=="#")
{
printf("<li><a %s href='#' target='_top'
class='%s'>%s</a></li>", $row[ite_extra], $row[ite_style], $row[ite_titulo]);
}
else

```

```

{

//Para submenus

if ($row[ite_link]=="##")
{

printf("<li><a %s href='index.php?smenu=%s%s' target='_top'
class='%s'>%s</a></li>", $row[ite_extra], $row[ite_submenu], $varsend, $row[ite_style], $ro
w[ite_titulo]);
if ($row[ite_submenu]==$this->submenuact)
{
echo $row[ite_submenu];
}
}
else
{
printf("<li><a %s href='index.php?opcp=7&apl=%s%s' target='_top'
class='%s'>%s</a></li>", $row[ite_extra], $row[ite_link], $varsend, $row[ite_style], $row[ite
_titulo]);
}
//Fin submenus
}
}
else
{
if ($row[ite_tipd]==1)
{
printf("<li><a %s href='index.php?smenu=%s&table=%s%s' target='_top'
class='%s'>%s</a></li>", $row[ite_extra], $this-
>submenuact, $row[ite_linktable], $varsend, $row[ite_style], $row[ite_titulo]);

```

```

}
else
{
printf("<li><a %s href='index.php?smenu=%s&tablelista=%s%s' target='_top'
class='%s'>%s</a></li>", $row[ite_extra], $this-
>submenuact, $row[ite_linktable], $varsend, $row[ite_style], $row[ite_titulo]);
}}}}}}
printf("</ul></td></tr></table>");
}}}}

//Despliegue de menus en la sección
function menu_posicion($uvic,$varsend,$menuperfil,$imenuperfil)
{
$sql = "SELECT * FROM opt_menu where men_uvic like '$uvic' order by men_ord";
$result = mysql_query($sql);
if ($result)
{
while($row = mysql_fetch_array($result))
{
$this->display_menu($row["men_id"], $varsend, $menuperfil, $imenuperfil);
//echo "<br>";
}
mysql_free_result ($result);
}
}
}
?>

```

C-8 Parámetros.php

Se definen las variables generales del sistema

```
<?php
//Variables generales del sistema
class parametros_generales{
var $titulo;
var $timp;
var $pimp;

function parámetros()
{

$selectTabla="select * from opt_datosg";
$resultado = mysql_query($selectTabla);
while($row = mysql_fetch_array($resultado))
{
$this->titulo=$row["dat_titulo"];
$this->timp=$row["dat_timp"];
$this->pimp=$row["dat_pimp"];
}
}}
?>
```

C-9 Session.php

Esta clase contiene la creación de sesión Optimail

```
<?php
class session_system{
var $resultado;
var $sess_id;
var $sess_user;
var $sess_pwd;
var $sess_name;
var $sess_perid;
var $sess_menu;
var $sess_imenu;
var $sess_boton;
var $sess_tabla;
var $sess_oficina;

function create_session($sess_usu,$sess_pwd,$sess_name,$perid)
{
$sesst_id="aqualis".time();
$sql= "insert into opt_sess (sess_id,sess_usu,sess_pwd,sess_name,sess_perid) values
('$sesst_id','$sess_usu','$sess_pwd','$sess_name',$perid)";
$resultado=mysql_query($sql);
return $sesst_id;
}
```

```

function select_session($sessid)
{
$selecmenu="select * from opt_sess where sess_id like '$sessid'";
$resultado = mysql_query($selecmenu);
while($row = mysql_fetch_array($resultado))
{
$this->sess_user=$row[sess_usu];
$this->sess_pwd=$row[sess_pwd];
$this->sess_name=$row[sess_name];
$this->sess_perid=$row[sess_perid];
}
}

function select_perfil($sess_perid)
{
$selecmenu="select * from opt_perfil,opt_detperfil where
opt_perfil.per_id=opt_detperfil.per_id and opt_perfil.per_id = $sess_perid";
$resultado = mysql_query($selecmenu);
if ($resultado)
{
while($row = mysql_fetch_array($resultado))
{

switch ($row[detp_obj])
{
case "menu":
{
$this->sess_menu=$row[detp_codigo];

}
}
break;

```

```
case "imenu":
{
$this->sess_imenu=$row[detp_codigo];

}
break;
case "boton":
{
$this->sess_boton=$row[detp_codigo];
}
break;
case "tabla":
{
$this->sess_tabla=$row[detp_codigo];
}
break;
}
} } }
?>
```

ANEXO D: CARPETA IBTEMPLATEV2

D-1 Borrar.php

Contiene el código de eliminación de un correo electrónico.

```
<?php
imap_delete($mbox,$del);
imap_mail_move ( $mbox , $del ,"BORRADO" );
imap_expunge ($mbox);
?>
```

D-2 Carpetas.php

Lectura de buzones (Bandeja de Entrada, Enviados, Borrados)

```
<?php
$name1 = "{".$mail_servr.":143/imap/notls}" . "ENVIADO";
$name2 = "{".$mail_servr.":143/imap/notls}" . "BORRADO";
$bandera1=0;
$bandera1=0;
$listacarpetas=0;
$ij=0;
$i=0;
```

```

$list = imap_list($mbox,$ServerNamer, "*");
if (is_array($list)) {
foreach ($list as $val) {
$carpeta[$i]=imap_utf7_decode($val);
$i++;
}
} else {
echo "imap_list failed: " . imap_last_error() . "\n";
}
if ($i==1)
{
imap_createmailbox($mbox, $name1);
imap_createmailbox($mbox, $name2);
}
$i=0;
$list1 = imap_list($mbox, $ServerNamer, "*");
if (is_array($list1)) {
foreach ($list1 as $val1) {
$carpetaex[$i]=imap_utf7_decode($val1);
$ncarpeta[$i]=trim(str_replace($ServerNamer, "", $carpetaex[$i]));
$i++;
}
}
//print_r ($list1);
$listacarpetas=$i;
//imap_deletemailbox ( $mbox , $ServerName."INBOXENVIADO" );
//imap_deletemailbox ( $mbox , $ServerName."INBOXBORRADO" );
?>

```

D-3 Enviar.php

Envío de correo electrónico siendo uno de los archivos más importantes de Optimail. No modificar.

```
<?php
function form_mail($sPara, $sAsunto, $sTexto, $sDe)
{
    $bHayFicheros = 0;
    $sCabeceraTexto = "";
    $sAdjuntos = "";

    if ($sDe)$sCabeceras = "From:". $sDe. "\n";
    else $sCabeceras = "";
    $sCabeceras .= "MIME-version: 1.0\n";
    foreach ($_POST as $sNombre => $sValor)
    $sTexto = $sTexto. "\n". $sNombre. " = ". $sValor;
    print_r($_FILES);
    foreach ($_FILES as $vAdjunto)
    {
        if ($bHayFicheros == 0)
        {
            $bHayFicheros = 1;
            $sCabeceras .= "Content-type: multipart/mixed;";
            $sCabeceras .= "boundary=\"--_Separador-de-mensajes_--\"\n";

            $sCabeceraTexto = "----_Separador-de-mensajes_--\n";
            $sCabeceraTexto .= "Content-type: text/plain;charset=iso-8859-1\n";
```

```

$sCabeceraTexto .= "Content-transfer-encoding: 7BIT\n";

$sTexto = $sCabeceraTexto.$sTexto;
}
if ($vAdjunto["size"] > 0)
{
$sAdjuntos .= "\n\n----_Separador-de-mensajes_--\n";
$sAdjuntos .= "Content-type: ".$vAdjunto["type"].";name=\"".$vAdjunto["name"]."\"";
$sAdjuntos .= "Content-Transfer-Encoding: BASE64\n";
$sAdjuntos .= "Content-disposition: attachment;filename=\"".$vAdjunto["name"]."\"";

$fichero = fopen($vAdjunto["tmp_name"], 'r');
$sContenido = fread($fichero, filesize($vAdjunto["tmp_name"]));
$sAdjuntos .= chunk_split(base64_encode($sContenido));
fclose($fichero);
}
}

if ($bHayFicheros)
$sTexto .= $sAdjuntos."\n\n----_Separador-de-mensajes_----\n";
return(mail($para, $asunto, $mensaje, $deq));
}

//cambiar aqui el email
if (form_mail($deq, $_POST[asunto],
"Los datos introducidos en el formulario son:\n\n", $_POST[email]))
echo "Su formulario ha sido enviado con exito";
?>

```

D-4 Envío.php

Almacena los mensajes de correo electrónico enviados al buzón “ENVIADO”

```
<style type="text/css">
<!--
.style1 {
font-size: 11px;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-weight: bold;
font-style: italic;
}
-->
</style>
<p class="style1">&nbsp;</p>
<p align="center" class="style1"> Mensaje enviado...</p>
<?php
if ($banderaenv==1)
{

?>

<?php
$mensajearm="";
function form_mail($sPara,$sCc, $sAsunto, $sTexto, $sDe)
{
global $mensajearm;
$bHayFicheros = 0;
$sCabeceraTexto = "";
$sAdjuntos = "";
```

```

if ($sDe)$sCabeceras = "From:". $sDe. "\n";
else $sCabeceras = "";
$sCabeceras .= "MIME-version: 1.0\n";
foreach ($_POST as $sNombre => $sValor)
$sTexto = $sTexto. "\n". $sNombre. " = ". $sValor;
//print_r($_FILES);
foreach ($_FILES as $vAdjunto)
{
if ($bHayFicheros == 0)
{
$bHayFicheros = 1;
$sCabeceras .= "Content-type: multipart/mixed;";
$sCabeceras .= "boundary=\"--_Separador-de-mensajes_--\"\n";
$sCabeceras .= "Cc:". $sCc. "\n";

$sCabeceraTexto = "----_Separador-de-mensajes_--\n";
$sCabeceraTexto .= "Content-type: text/plain;charset=iso-8859-1\n";
$sCabeceraTexto .= "Content-transfer-encoding: 7BIT\n";

$sTexto = $sCabeceraTexto. $sTexto;
}
if ($vAdjunto["size"] > 0)
{
$sAdjuntos .= "\n\n----_Separador-de-mensajes_--\n";
$sAdjuntos .= "Content-type: ".$vAdjunto["type"].";name=\"".$vAdjunto["name"]."\"\n";
$sAdjuntos .= "Content-Transfer-Encoding: BASE64\n";
$sAdjuntos .= "Content-disposition: attachment;filename=\"".$vAdjunto["name"]."\"\n\n";

$oFichero = fopen($vAdjunto["tmp_name"], 'r');

```

```

$sContenido = fread($oFichero, filesize($vAdjunto["tmp_name"]));
$sAdjuntos .= chunk_split(base64_encode($sContenido));
fclose($oFichero);
}}
if ($bHayFicheros)
$sTexto .= $sAdjuntos."\n\n----_Separador-de-mensajes_----\n";
$mensajearm= $sTexto;
return(mail($sPara, $sAsunto, $sTexto, $sCabeceras));
}

```

```
//cambiar aqui el email
```

```

if (form_mail($_POST[para],$_POST[cc], $_POST[asunto],
"Los datos introducidos en el formulario son:\n\n", $_POST[deq]))

```

```
//echo "Su formulario ha sido enviado con exito";
```

```
?>
```

```
<?php
```

```
$mensajearm1 = str_replace("----_Separador-de-mensajes_----", "", $mensajearm);
```

```
$now = date("l,M d, Y g:i:s");
```

```
imap_append($mbox, $ServerName."ENVIADO"
```

```
, "From: ".$_POST[deq]."\r\n"
```

```
. "To: ".$_POST[para]."\r\n"
```

```
. "Subject: ".$_POST[asunto]."\r\n"
```

```
. "Cc: ".$_POST[cc]."\r\n"
```

```
. "Date: ".$now."\r\n"
```

```
. "\r\n"
```

```
. $mensajearm1."\r\n"
```

```
);
```

```
}
```

```
?>
```

D-5 Gotodownload.php

Contiene el código que permite la descarga de un archivo adjunto dentro de un mensaje.

```
<?
/**VARIABLES POR GET */

$numero = count($_GET);
$tags = array_keys($_GET);// obtiene los nombres de las variables
$valores = array_values($_GET);// obtiene los valores de las variables

// crea las variables y les asigna el valor
for($i=0;$i<$numero;$i++){
    $$tags[$i]=$valores[$i];
}

/**VARIABLES POR POST */

$numero2 = count($_POST);
$tags2 = array_keys($_POST); // obtiene los nombres de las variables
$valores2 = array_values($_POST);// obtiene los valores de las variables

// crea las variables y les asigna el valor
for($i=0;$i<$numero2;$i++){
    $$tags2[$i]=$valores2[$i];
}

?>
<?
```

```

$imap =
imap_open("{localhost.optimail:143/imap/notls}INBOX",$mail_user,$mail_passw);
$mbox=$imap;
//$msgno=;

$struct = imap_fetchstructure($mbox,$msgno);
$contentParts = count($struct->parts);

if ($contentParts >= 2) {
for ($i=2;$i<=$contentParts;$i++) {
$att[$i-2] = imap_bodystruct($mbox,$msgno,$i);
}
}

if ($download == "1") {
$strFileName = $att[$file]->parameters[0]->value;
$strFileType = strrev(substr(strrev($strFileName),0,4));
$fileContent = imap_fetchbody($mbox,$msgno,$file+2);
downloadFile($strFileType,$strFileName,$fileContent);
}

function downloadFile($strFileType,$strFileName,$fileContent) {
$contentType = "application/octet-stream";

if ($strFileType == ".asf")
$contentType = "video/x-ms-asf";
if ($strFileType == ".avi")
$contentType = "video/avi";
if ($strFileType == ".doc")
$contentType = "application/msword";
if ($strFileType == ".zip")

```

```
$ContentType = "application/zip";
if ($strFileType == ".xls")
$ContentType = "application/vnd.ms-excel";
if ($strFileType == ".gif")
$ContentType = "image/gif";
if ($strFileType == ".jpg" || $strFileType == "jpeg")
$ContentType = "image/jpeg";
if ($strFileType == ".wav")
$ContentType = "audio/wav";
if ($strFileType == ".mp3")
$ContentType = "audio/mpeg3";
if ($strFileType == ".mpg" || $strFileType == "mpeg")
$ContentType = "video/mpeg";
if ($strFileType == ".rtf")
$ContentType = "application/rtf";
if ($strFileType == ".htm" || $strFileType == "html")
$ContentType = "text/html";
if ($strFileType == ".xml")
$ContentType = "text/xml";
if ($strFileType == ".xsl")
$ContentType = "text/xsl";
if ($strFileType == ".css")
$ContentType = "text/css";
if ($strFileType == ".php")
$ContentType = "text/php";
if ($strFileType == ".asp")
$ContentType = "text/asp";
if ($strFileType == ".pdf")
$ContentType = "application/pdf";

header ("Content-Type: $ContentType");
```

```
header ("Content-Disposition: attachment; filename=$strFileName; size=$fileSize;");
```

```
// Updated oktober 29. 2005
```

```
if (substr($ContentType,0,4) == "text") {
```

```
echo imap_qprint($fileContent);
```

```
} else {
```

```
echo imap_base64($fileContent);
```

```
}}
```

```
?>
```

D-6 Inicio.php

Pantalla inicial de Optimail

```
<div align='center'>
```

```
<table width="100%" height="100%" border=0>
```

```
<tr>
```

```
<td><div align="center">
```

```
<p><img src='<?php echo $objtemplate->path_template ?>images/centro.png'></p>
```

```
<p>&nbsp;</p>
```

```
<p>&nbsp;</p>
```

```
</div></td></table>
```

```
<br>
```

```
</div>
```

D-7 Leer.php

Permite la lectura de correo electrónico.

```
<style type="text/css">
<!--
.bodyst1 {
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 12px;
font-weight: bold;
}
.bodyst2 {
font-size: 11px;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-weight: bold;
color: #999999;
}
.bodyst3 {
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 11px;
}
.Estilo2 { font-family: Verdana, Arial, Helvetica, sans-serif}
.Estilo4 { font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>

<table border="0" cellpadding="0" cellspacing="4">
<tr>
```

```

<td><span class="Estilo2"><a href="index.php?bus=<?php echo $bus
?>&responder=1&opcion=3&sessid=<?php echo $sessid;
?>">RESPONDER</a></span></td>

<td><span class="Estilo2"><a href="index.php?bus=<?php echo $bus
?>&reenviar=1&opcion=3&sessid=<?php echo $sessid; ?>">REENVIAR
</a></span></td>

</tr>

</table> //Dibuja en pantalla una tabla con datos
<p class="bodyst1"><?php echo $subject[$bus]; ?> </p>
<table width="512" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="181"><span class="bodyst2">De:</span></td>
<td width="315"> <span class="bodyst3"> <?php echo $from[$bus]; ?> </span></td>
</tr>
<tr>
<td><span class="bodyst2">Enviado:</span></td>
<td> <span class="bodyst3"> <?php echo $fromdate[$bus]; ?> </span> </td>
</tr>
<tr>
<td><span class="bodyst2">Para:</span></td>
<td> <span class="bodyst3"> <?php echo $to[$bus]; ?> </span> </td>
</tr>
<tr>
<td><span class="bodyst2">Cc:</span></td>
<td><span class="bodyst3"><?php echo $contenidoarr[ccaddress] ?></span></td>
</tr>
</table>

<hr size="1" />

<table width="100%" border="0" cellpadding="0" cellspacing="2">
<tr>
<td width="96%" rowspan="2" valign="top"><br />

```

```

<table width="90%" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td><div align="left"><span class="bodyst3">
<?php
echo $contenidodpurado;
?>
<?php
if(ereg("Content-disposition: attachment",$contenido))
{
?>
</span></div> </td>
</table>
<td width="4%" bgcolor="#CBDEED"><span class="Estilo4">ADJUNTOS:</span></td>
</tr>
<tr>
<td><?php
include("templates/ibtemplatev2/prueba1.php");
//include("templates/ibtemplatev2/gotodownload.php");
}
?>
</td>
</tr>
</table>
<p>&nbsp;</p>

```

D-8 ListaC.php

Contiene la lista de correos que se desplegará en cada buzón o bandeja.

```
<table width="100%" border="0" cellpadding="0" cellspacing="1">
<tr>
<td bordercolor="#" bgcolor="#DAE4F3"><span class="Estilo1"> &nbsp;&nbsp;&nbsp;<?php
echo $nombreu ?> -- <?php echo $mailu ?> </span></td>
<td bordercolor="#" bgcolor="#DAE4F3"><span class="Estilo2">CORREOS: <?php echo
imap_num_msg($mbox); ?> </span></td> <td bgcolor="#DAE4F3"> </td>
<td bgcolor="#DAE4F3"> </td>
<td bgcolor="#DAE4F3"> </td>
</tr>
<tr>
<td bordercolor="#696969" bgcolor="#F2E6FF"><span class="Estilo2">DE</span></td>
<td bordercolor="#696969" bgcolor="#F2E6FF"><span
class="Estilo2">ASUNTO</span></td>
<td bordercolor="#696969" bgcolor="#F2E6FF"><span
class="Estilo2">FECHA</span></td>
<td bordercolor="#696969" bgcolor="#F2E6FF"><span class="Estilo2"> </span></td><td
bordercolor="#696969" bgcolor="#F2E6FF"><span class="Estilo2"> </span></td>
</tr>
<?php
for ($i = imap_num_msg($mbox); $i >= 1; $i--)
{
$header = imap_headerinfo($mbox, $i, 80, 80);
$fromaddress[$i] = $header->from[0]->host;
$fromname[$i] = $header->from[0]->mailbox;
$fromdate[$i] = $header->Date;
```

```

$from[$i]= $fromname[$i]."@".$fromaddress[$i];

$subject[$i] = $header->fetchsubject;

imap_setflag_full($mbox, imap_uid($mbox, $i), "\\Recent \\Unseen \\Seen \\Flagged",
ST_UID);
imap_setflag_full($mbox, imap_uid($mbox, $i), "\\Recent \\Unseen \\Seen \\Flagged",
ST_UID);
$informativo= imap_headerinfo($mbox, $i);
//print_r($informativo);

$info = imap_fetchstructure($mbox,$i);

// Saber cuántas partes tiene el objeto
$numparts = count($info->parts);

if ($informativo->Unseen=="U")
{
echo ' <tr>
<td bordercolor="#696969" bgcolor="#F2E6FF"><b> '.$fromname[$i].</b></span> </td>
<td bordercolor="#696969" bgcolor="#F2E6FF"> <b>'.$subject[$i].</b></span> </td>
<td bordercolor="#696969" bgcolor="#F2E6FF"><b> '.$fromdate[$i].</b></span></td>
<td bordercolor="#696969" bgcolor="#F2E6FF"> <a
href="index.php?carpetal='.$carpetal.'&opcion=1&bus='.$i.'&sessid='.$sessid.'"> ;

if ($numparts>1)
{
echo ' ;
}

```

```

echo '</a> </span></td>
<td bordercolor="#696969" bgcolor="#F2E6FF"> <a
href="index.php?carpetal='.$carpetal.'&opcion=2&del='.$i.'&sessid='.$sessid.'">  </a> </span></td>
<td bordercolor="#696969" bgcolor="#F2E6FF"> </span></td>
</tr>';
}
else
{ //dibuja en pantalla
echo ' <tr>
<td bordercolor="#696969" bgcolor="#F2E6FF"> '.$fromname[$i].</span> </td>
<td bordercolor="#696969" bgcolor="#F2E6FF"> '.$subject[$i].</span> </td>
<td bordercolor="#696969" bgcolor="#F2E6FF"> '.$fromdate[$i].</span></td>
<td bordercolor="#696969" bgcolor="#F2E6FF"> <a
href="index.php?carpetal='.$carpetal.'&opcion=1&bus='.$i.'&sessid='.$sessid.'"> ';
if ($numparts>1)
{ //muestra una imagen
echo '';
}
echo '</a> </span></td>
<td bordercolor="#696969" bgcolor="#F2E6FF"> <a
href="index.php?carpetal='.$carpetal.'&opcion=2&del='.$i.'&sessid='.$sessid.'">  </a> </span></td>
<td bordercolor="#696969" bgcolor="#F2E6FF"> </span></td>

</tr>';
}
}
?>
</table>

```

D-9 Nuevacarpeta.php

Creación de Nuevo buzón (INBOX)

```
<?php
```

```
$ServerName = "{localhost.optimail:143/imap/notls}";
```

```
$ServerName1 = "{localhost.optimail:143/imap/notls}";
```

```
// $mbox = imap_open($ServerName, $username,$password) or die("No puedo abrir  
Mailbox.");
```

```
// Crear Buzón
```

```
$name3 = $ServerName . "INBOX1";
```

```
//imap_createmailbox($mbox, $name3);
```

```
// echo imap_last_error();
```

```
imap_deletemailbox ( $mbox , $name3 );
```

```
$list = imap_list($mbox, $ServerName1, "*");
```

```
print_r($list);
```

```
?>
```

D-10 Nuevacarpeta2.php

Crea un Nuevo buzón (BORRADOS)

```
<?php
$imap = imap_open("{localhost.optimail/pop3/notls}INBOX",$mail_usern,$mail_passw);
echo $imap;
$mailbox = "BORRADOS";
$new_mailbox = "test_newmailbox";

if
(!imap_createmailbox($imap,imap_utf7_encode("{localhost.optimail/pop3/notls}$mailbox
"))) {
echo "Unable to create new mailbox!\n";
echo imap_last_error();
}

// lista de buzones para mostrar está aqui
$boxes = imap_listmailbox($imap, "{localhost.optimail/pop3/notls}", "BO*");
echo "<b>Mailboxes:</b><BR>\n";
for ($i=0; $i<count($boxes); $i++) {
echo "Mailbox: $boxes[$i]<BR>\n";
}
?>
```

D-11 Nuevo.php

Contiene el código que sirve para que el usuario pueda redactar un nuevo mensaje de correo electrónico.

```
<?php
if ($reenviar==1 or $responder==1)
{
echo $reenviar;
//leer correo
$header = imap_headerinfo($mbox, $bus, 80, 80);
//print_r($header);
$fromaddress[$bus] = $header->from[0]->host;
$fromname[$bus] = $header->from[0]->mailbox;
$fromdate[$bus] = $header->Date;
$from[$bus]= $fromname[$bus]."@".$fromaddress[$bus];
//to[3]=$header->from[0]->to;
// $to[$bus]=$emailu;
$to[$bus]=$header->toaddress;

$subject[$bus] = $header->fetchsubject;
$contenido = imap_body ($mbox,$bus);

if ($responder==1)
{
$contenido="";
}
}
?>
```

```
<!-- <form name="form1" method="post" action=""> -->
```

```
<form name='formulario' id='formulario' method='post' action="" target='_self'  
enctype="multipart/form-data">
```

```
<table width="100%" border="0" cellpadding="0" cellspacing="0">
```

```
<tr>
```

```
<td width="11%">De:</td>
```

```
<td width="89%"><input name="deq" type="hidden" id="deq" value="<?php echo  
$mailu ?>">
```

```
<?php echo $mailu?></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Para:</td>
```

```
<td><label>
```

```
<input name="para" type="text" id="para" value="<?php  
echo $from[$bus];
```

```
?>" size="70">
```

```
</label></td>
```

```
</tr>
```

```
<tr>
```

```
<td>CC:</td>
```

```
<td><label>
```

```
<input name="cc" type="text" id="cc" size="70">
```

```
</label></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Asunto:</td>
```

```
<td><label>
```

```
<input name="asunto" type="text" id="asunto" value="<?php
```

```

if ($bus)
{
echo "RE:". $subject[$bus];
}

?>"
size="70">
<input name="banderaenv" type="hidden" id="banderaenv" value="1" />
<input name="opcion" type="hidden" id="opcion" value="4" />
<input name="sessid" type="hidden" id="sessid" value="<?php echo $sessid ?>" />
</label></td>
</tr>
<tr>
<td>Adjuntar archivo: </td>
<td><input type='file' name='archivo1' id='archivo1' /></td>
</tr>
</table>
<p>&nbsp;</p>
<p>
<?php
$oFCKeditor = new FCKeditor("mensaje");
$oFCKeditor->BasePath = "/optimail/";
$oFCKeditor->Value = $contenido;
$oFCKeditor->Width = 700 ;
$oFCKeditor->Height = 350 ;
$oFCKeditor->ToolbarSet='Basic';
$oFCKeditor->tabindex= $tabindx;
$oFCKeditor->Create();
?>
</label>

```

```
<input type="submit" value="Enviar" >
</label>
</p>
</form>
```

D-12 Readattachment.php

Contiene el código que sirve para leer un archivo adjunto.

```
<?
require_once("attachmentread.class.php");
$host="{my.mailserver.net:110/pop3}"; // pop3host
$login="user1"; //pop3 login
$password="passwd1"; //pop3 password
$savedirpath="" ; // adjuntos se guardan en el mismo directorio donde ejecuta el script
$jkl=new readattachment(); // Creando instancia de la clase#####
$jkl->getdata($host,$login,$password,$savedirpath); // Llamando a la función miembro
?>
```

D-13 Readcell.php

Célula o núcleo de Optimail que enlaza las clases de lectura de correo. Por ejemplo cuando el usuario desee leer un nuevo correo en la bandeja de entrada entonces Readcell abrirá el archivo de lectura de carpetas y buscará su respectiva función.

```
<?php

$consultau="select * from opt_sisusers where sisu_id="._SESSION['iduser'];
$resultadou = mysql_query($consultau);
```

```

while($rowu = mysql_fetch_array($resultadou))
{
$nombremail=$rowu["sisu_usu"];
$nombreu=$rowu["sisu_name"];
$emailu=$nombremail."@localhost.optimail"; //formato

}
//echo $emailu;
?>
<?php

#verbinden
$mail_servr=$ip;
$mail_usern=$_SESSION['nov_ususer'];

if (!($mail_usern=='marco' or $mail_usern=='pepito'))
{

$mail_passw="benito31";
}
else
{
$mail_passw=$_SESSION['nov_pwd'];
}

$ServerName = "{".$mail_servr.":143/imap/notls}".$scarpetal;
$ServerNamer = "{".$mail_servr.":143/imap/notls}"; //conexión

$mbox = imap_open ($ServerName, "$mail_usern", "$mail_passw");

```

```

$inbox = imap_open ($ServerName, "$mail_usern", "$mail_passw");
?>
<style type="text/css">
<!--
.Estilo1 {
font-size: 11px;
font-family: Geneva, Arial, Helvetica, sans-serif;
color: #333333;
font-weight: bold;
font-style: italic;
}
.Estilo2 {
font-size: 11px;
font-weight: bold;
font-family: Verdana, Arial, Helvetica, sans-serif;
}
.Estilo6 { font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 11px; }
-->
</style>
<?php

if ($inbox)
{

$headers = imap_headers ($inbox);
$totalmsg=imap_num_msg ($inbox);

?>

<?php
//imap_setflag_full($inbox, imap_uid($inbox, 0), "\\Seen \\Flagged", ST_UID);

```

```

//print_r( imap_headerinfo($mbox, 0) );

switch($opcion)
{
case 1:
{

include("templates/ibtemplatev2/carpetas.php");
include ("templates/ibtemplatev2/menu/menuinicio.php");

//leer correo
$header = imap_headerinfo($mbox, $bus, 80, 80);

$fromaddress[$bus] = $header->from[0]->host;
$fromname[$bus] = $header->from[0]->mailbox;
$fromdate[$bus] = $header->Date;
$from[$bus]= $fromname[$bus]."@".$fromaddress[$bus];
//$to[3]=$header->from[0]->to; if (!$partsemail) { /* Simple mensaje, sólo 1pieza */
$attachment = array(); /* No attachments */
$contenido = imap_body($mbox, $bus);
// $to[$bus]=$emailu;
$to[$bus]=$header->toaddress;

$subject[$bus] = $header->fetchsubject;

// print_r($header);
$struct = imap_fetchstructure($mbox,$bus);
$partsemail = count($struct->parts);

$i = 0;

```

```

if (!$partsemail) { /* Simple message, only 1 piece */
$attachment = array(); /* No attachments */
$contenido = imap_body($mbox, $bus);
}

```

```

$contenido = imap_body($mbox, $bus);

```

```

////////////////////////////////////

```

```

function retrieve_message($mbox, $messageid)
{
$message = array();

```

```

$header = imap_header($mbox, $messageid);
$structure = imap_fetchstructure($mbox, $messageid);

```

```

$message['subject'] = $header->subject;
$message['fromaddress'] = $header->fromaddress;
$message['toaddress'] = $header->toaddress;
$message['ccaddress'] = $header->ccaddress;
$message['date'] = $header->date;

```

```

if (check_type($structure))
{
$message['body'] = imap_fetchbody($mbox,$messageid,"1"); ## Obtiene el cuerpo de
mensaje multi parte
if (!$message['body']) {$message['body'] = '[No hay texto en el mensaje]\n\n';}
}
else

```

```

{
$message['body'] = imap_body($mbox, $messageid);
if(!$message['body']) {$message['body'] = '[No hay texto en el mensaje]\n\n';}
}

return $message;
}

function check_type($structure) ## Chequea el tipo
{
if($structure->type == 1)
{
return(true); ## si, es un mensaje multi parte
}
else
{
return(false); ## no es un mensaje multi parte
}
}

$contenidoarr=retrieve_message($mbox, $bus);

$pos1 = stripos($contenidoarr[body], "mensaje =");
$pos1=$pos1+9;
$pos2 = $pos1* -1;
$contenidodpurado= substr($contenidoarr[body],$pos1);

////////////////////////////////////

```

```
include("templates/ibtemplatev2/leer.php");

//fin leer correo

}
break;

case 2:
//Borrar correo
include("templates/ibtemplatev2/carpetas.php");
include ("templates/ibtemplatev2/borrar.php");
include ("templates/ibtemplatev2/menu/menuinicio.php");

include("templates/ibtemplatev2/listac.php");
// Fin borrar correo
break;
case 3:
//Nuevo correo
include ("templates/ibtemplatev2/nuevo.php");
//Fin nuevo correo
break;

case 4:
//Envío de correo
include("templates/ibtemplatev2/carpetas.php");
include ("templates/ibtemplatev2/menu/menuinicio.php");
include ("templates/ibtemplatev2/envio.php");
//Fin nuevo correo
break;
```

case 5:

//Nueva carpeta

include ("templates/ibtemplatev2/nuevacarpeta.php");

//Fin nuevo correo

break;

case 6:

include ("templates/ibtemplatev2/vaciar.php");

include("templates/ibtemplatev2/carpetas.php");

include ("templates/ibtemplatev2/menu/menuinicio.php");

include("templates/ibtemplatev2/listac.php");

break;

default:

include("templates/ibtemplatev2/carpetas.php");

include ("templates/ibtemplatev2/menu/menuinicio.php");

include("templates/ibtemplatev2/listac.php");

break;

}

?>

<p>

<?php

//imap_expunge (\$mbox);

imap_close (\$mbox);

}

else

{

echo "Error clave o usuario";

}

```
?>
</p>
<p> </p>
```

D-14 Receivemail.class.php

Clase que contiene la administración de lectura de correo.

```
<?php
class receiveMail
{
var $server="";
var $username="";
var $password="";

var $marubox="";

var $email="";

function
receiveMail($username,$password,$EmailAddress,$mailserver='localhost',$servertype='po
p',$port='110') //Constructor
{
if($servertype=='imap')
{
if($port=="") $port='143';
$strConnect='{!.$mailserver.!.$port. }INBOX!';
}
else
{
```

```

$strConnect='{'. $mailserver.'!:$port. /pop3}INBOX';
}
$this->server          =      $strConnect;
$this->username        =      $username;
$this->password        =      $password;
$this->email           =      $EmailAddress;
}
function connect() //Connect To the Mail Box
{
$this->marubox=imap_open($this->server,$this->username,$this->password);
}
function getHeaders($mid) // Obtiene información del Header
{
$mail_header=imap_header($this->marubox,$mid);
$sender=$mail_header->from[0];
$sender_replyto=$mail_header->reply_to[0];
if(strtolower($sender->mailbox)!='mailer-daemon' && strtolower($sender-
>mailbox)!='postmaster')
{
$mail_details=array(
'from'=>strtolower($sender->mailbox).'@'.$sender->host,
'fromName'=>$sender->personal,
'toOth'=>strtolower($sender_replyto->mailbox).'@'.$sender_replyto->host,
'toNameOth'=>$sender_replyto->personal,
'subject'=>$mail_header->subject,
'to'=>strtolower($mail_header->toaddress)
);
}
return $mail_details;
}
function get_mime_type(&$structure) //Obtiene tipo Mime

```

```

{
$primary_mime_type = array("TEXT", "MULTIPART", "MESSAGE", "APPLICATION",
"AUDIO", "IMAGE", "VIDEO", "OTHER");

if($structure->subtype) {
return $primary_mime_type[(int) $structure->type] . '/' . $structure->subtype;
}
return "TEXT/PLAIN";
}

function get_part($stream, $msg_number, $mime_type, $structure = false, $part_number =
false) //Obtiene parte del mensaje (uso privado)
{
if(!$structure) {
$structure = imap_fetchstructure($stream, $msg_number);
}
if($structure) {
if($mime_type == $this->get_mime_type($structure))
{
if(!$part_number)
{
$part_number = "1";
}
$text = imap_fetchbody($stream, $msg_number, $part_number);
if($structure->encoding == 3)
{
return imap_base64($text);
}
else if($structure->encoding == 4)
{
return imap_qprint($text);
}
}
}
}

```

```

else
{
return $text;
}
}
if($structure->type == 1) /* multipart */
{
while(list($index, $sub_structure) = each($structure->parts))
{
if($part_number)
{
$prefix = $part_number . '.';
}
$data = $this->get_part($stream, $msg_number, $mime_type, $sub_structure, $prefix .
($index + 1));
if($data)
{
return $data;
}
}
}
return false;
}
function getTotalMails() //Obtiene el número total de mensajes no leídos.
{
$headers=imap_headers($this->marubox);
return count($headers);
}
function GetAttech($mid,$path) // Obtiene Archivo del email
{

```

```

$structure = imap_fetchstructure($this->marubox,$mid);
$ar="";
foreach($structure->parts as $key => $value)
{
$enc=$structure->parts[$key]->encoding;
if($structure->parts[$key]->ifdparameters)
{
$name=$structure->parts[$key]->dparameters[0]->value;
$message = imap_fetchbody($this->marubox,$mid,$key+1);
if ($enc == 0)
$message = imap_8bit($message);
if ($enc == 1)
$message = imap_8bit ($message);
if ($enc == 2)
$message = imap_binary ($message);
if ($enc == 3)
$message = imap_base64 ($message);
if ($enc == 4)
$message = quoted_printable_decode($message);
if ($enc == 5)
$message = $message;
$fp=fopen($path.$name,"w");
fwrite($fp,$message);
fclose($fp);
$ar=$ar.$name." ";
} //imap puede ser binario, de 8 bits o base 64. Se compara.
}
$ar=substr($ar,0,(strlen($ar)-1));
return $ar;
}
function getBody($mid) // Retorna el cuerpo del Mensaje

```

```

{
$body = $this->get_part($this->marubox, $mid, "TEXT/HTML");
if ($body == "")
$body = $this->get_part($this->marubox, $mid, "TEXT/PLAIN");
if ($body == "") {
return "";
}
return $body;
}
function deleteMails($mid) // Borra el correo
{
imap_delete($this->marubox,$mid);
}
function close_mailbox() // Cierra el correo
{
imap_close($this->marubox,CL_EXPUNGE);
}
}
?>

```

D-15 Vaciar.php

Vacía la papelera de reciclaje (Borra el correo físicamente).

```

<?php
imap_expunge ($mbox);
?>

```

ANEXO E: CARPETA MENU

E-1 Menuinicio.php

Página del menú de inicio que recoge las imágenes a desplegarse en la pantalla

```
<style type="text/css">
<!--
.style5 {font-size: 10px; font-family: Verdana, Arial, Helvetica, sans-serif; font-weight:
bold; }
.style6 {
font-size: 13px;
font-weight: bold;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-style: italic;
color: #FFFFFF;
}
-->
</style>
<?php
$pathimg="";
if ($carpetal=="ENVIADO")
{
$pathimg="images/enviado.png";
```

```

}
if ($carpetal=="BORRADO")
{
$pathimg="images/papelera.png";
}
if ($carpetal=="INBOX")
{
$pathimg="images/inbox.png";
}
if (!$pathimg=="")
{
echo '';
}
?>
<table width="100%" border="0" cellpadding="0" cellspacing="1">
<tr>
<td colspan="5" bgcolor="#75AAD5">

<span class="style6">&nbsp;&nbsp;&nbsp;&nbsp;<?php echo $carpetal; ?></span></td>
</tr>
<tr>
<td colspan="5" bgcolor="#FFFFFF">
&nbsp;<?php

$x=0;
?>

```

```

<table border="0" cellpadding="2" cellspacing="2">
<tr>
<td><div align="center"><a href="index.php?opcion=3&sessid=<?php echo $sessid;
?>"></a></div> </td>

<?php

for ($x = $listacarpetas-1; $x >=0 ; $x--)
{
$pathimg="";
if ($ncarpeta[$x]=="ENVIADO")
{
$pathimg="images/enviado.png";
}
if ($ncarpeta[$x]=="BORRADO")
{
$pathimg="images/papelera.png";
}
if ($ncarpeta[$x]=="INBOX")
{
$pathimg="images/inbox.png";
}

echo '<td ><div align="center"><span class="style5"><a
href="index.php?carpetal='.$ncarpeta[$x].'&sessid='.$sessid.'"></a></span></div></td>';

}

?>

<td ><div align="center"><span class="style5"><a
href="index.php?carpetal=BORRADO&opcion=6&sessid=<?php echo $sessid ?>"></a></span></div></td>

```

```

</tr> <tr>

<td bgcolor="#E1EEF0"><div align="center"><span class="style5">NUEVO EMAIL
</span></div></td>

<?php

for ($x = $listacarpetas-1; $x >=0 ; $x--)
{
if ($ncarpeta[$x]=="INBOX")
{
$titulos="BANDEJA DE ENTRADA";
}
else
{
$titulos=$ncarpeta[$x];
}
echo '<td bgcolor="#E1EEF0"><div align="center"><span class="style5"><a
href="index.php?carpetal='.$ncarpeta[$x].'&sessid='.$sessid.'">'.$titulos.'</a></span></div
></td>';
}
?>

<td bgcolor="#E1EEF0"><div align="center"><span class="style5"><a
href="index.php?carpetal=BORRADO&opcion=6&sessid=<?php echo $sessid
?>">VACIAR PAPELERA </a></span></div></td>

</tr>

</table>

</td>

</tr>

</table>

```

ANEXO F: LOGS DE CORREO

F-1 Maillog (sin adjuntos)

```
Jul 12 00:34:36 localhost dovecot: Dovecot v1.0.13 starting up
Jul 12 00:34:38 localhost sendmail[2107]: alias database /etc/aliases
rebuilt by root
Jul 12 00:34:38 localhost sendmail[2107]: /etc/aliases: 76 aliases,
longest 10 bytes, 765 bytes total
Jul 12 00:34:40 localhost postfix/postfix-script[2158]: starting the
Postfix mail system
Jul 12 00:34:40 localhost postfix/master[2159]: daemon started --
version 2.4.5, configuration /etc/postfix
Jul 12 00:51:16 localhost dovecot: imap-login: Login: user=<usuario3>,
method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105,
secured
Jul 12 00:51:17 localhost dovecot: imap-login: Login: user=<usuario3>,
method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105,
secured
Jul 12 00:51:17 localhost dovecot: IMAP(usuario3): Disconnected:
Logged out
Jul 12 00:51:17 localhost dovecot: IMAP(usuario3): Disconnected:
Logged out
Jul 12 00:57:54 localhost dovecot: imap-login: Login: user=<rosal>,
method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105,
secured
Jul 12 00:57:54 localhost dovecot: imap-login: Login: user=<rosal>,
method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105,
secured
Jul 12 00:57:54 localhost dovecot: IMAP(rosal): Disconnected: Logged
out
Jul 12 00:57:55 localhost dovecot: IMAP(rosal): Disconnected: Logged
out
Jul 12 00:57:58 localhost dovecot: imap-login: Login: user=<rosal>,
method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105,
secured
Jul 12 00:57:58 localhost dovecot: imap-login: Login: user=<rosal>,
method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105,
secured
Jul 12 00:57:58 localhost dovecot: IMAP(rosal): Disconnected: Logged
out
Jul 12 00:57:58 localhost dovecot: IMAP(rosal): Disconnected: Logged
out
Jul 12 01:00:38 localhost dovecot: imap-login: Login: user=<rosal>,
method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105,
secured
Jul 12 01:00:38 localhost dovecot: IMAP(rosal): Disconnected: Logged
out
```

Jul 12 01:00:38 localhost dovecot: imap-login: Login: user=<rosal>, method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105, secured

Jul 12 01:00:39 localhost sendmail[3626]: n6C60dxe003626: from=apache, size=1597, class=0, nrcpts=20, msgid=<200907120600.n6C60dxe003626@localhost.optimail>, relay=apache@localhost

Jul 12 01:00:41 localhost postfix/smtpd[3627]: connect from localhost.localdomain[127.0.0.1]

Jul 12 01:00:41 localhost postfix/smtpd[3627]: setting up TLS connection from localhost.localdomain[127.0.0.1]

Jul 12 01:00:41 localhost postfix/smtpd[3627]: TLS connection established from localhost.localdomain[127.0.0.1]: TLSv1 with cipher DHE-RSA-AES256-SHA (256/256 bits)

Jul 12 01:00:41 localhost sendmail[3626]: STARTTLS=client, relay=[127.0.0.1], version=TLSv1/SSLv3, verify=FAIL, cipher=DHE-RSA-AES256-SHA, bits=256/256

Jul 12 01:00:41 localhost postfix/smtpd[3627]: BA317AFACC: client=localhost.localdomain[127.0.0.1], sasl_sender=apache@localhost.optimail

Jul 12 01:00:41 localhost postfix/cleanup[3631]: BA317AFACC: message-id=<200907120600.n6C60dxe003626@localhost.optimail>

Jul 12 01:00:41 localhost postfix/qmgr[2163]: BA317AFACC: from=<apache@localhost.optimail>, size=2183, nrcpt=20 (queue active)

Jul 12 01:00:42 localhost sendmail[3626]: n6C60dxe003626: to=usuario2@localhost.optimail,usuario3@localhost.optimail,usuario4@localhost.optimail,usuario5@localhost.optimail,usuario6@localhost.optimail,usuario7@localhost.optimail,usuario8@localhost.optimail,usuario9@localhost.optimail,usuario10@localhost.optimail,usuario11@localhost.optimail,usuario12@localhost.optimail,usuario13@localhost.optimail,usuario14@localhost.optimail,usuario15@localhost.optimail,usuario16@localhost.optimail,usuario17@localhost.optimail,usuario18@localhost.optimail,usuario19@localhost.optimail,usuario20@localhost.optimail,usuario1@localhost.optimail, ctladdr=apache (48/48), delay=00:00:03, xdelay=00:00:03, mailer=relay, pri=601597, relay=[127.0.0.1] [127.0.0.1], dsn=2.0.0, stat=Sent (Ok: queued as BA317AFACC)

Jul 12 01:00:42 localhost postfix/smtpd[3627]: disconnect from localhost.localdomain[127.0.0.1]

Jul 12 01:00:43 localhost postfix/local[3635]: BA317AFACC: to=<usuario12@localhost.optimail>, relay=local, delay=1.5, delays=0.44/1/0/0.03, dsn=2.0.0, status=sent (delivered to maildir)

Jul 12 01:00:43 localhost postfix/local[3635]: BA317AFACC: to=<usuario14@localhost.optimail>, relay=local, delay=1.5, delays=0.44/1/0/0.03, dsn=2.0.0, status=sent (delivered to maildir)

Jul 12 01:00:43 localhost postfix/local[3633]: BA317AFACC: to=<usuario10@localhost.optimail>, relay=local, delay=1.5, delays=0.44/0.62/0/0.48, dsn=2.0.0, status=sent (delivered to maildir)

Jul 12 01:00:43 localhost postfix/local[3635]: BA317AFACC: to=<usuario15@localhost.optimail>, relay=local, delay=1.6, delays=0.44/1.1/0/0.03, dsn=2.0.0, status=sent (delivered to maildir)

Jul 12 01:00:43 localhost postfix/local[3632]: BA317AFACC: to=<usuario13@localhost.optimail>, relay=local, delay=1.6, delays=0.44/0.97/0/0.15, dsn=2.0.0, status=sent (delivered to maildir)

Jul 12 01:00:43 localhost postfix/local[3634]: BA317AFACC:
to=<usuario11@localhost.optimail>, relay=local, delay=1.6,
delays=0.44/0.94/0/0.19, dsn=2.0.0, status=sent (delivered to maildir)
Jul 12 01:00:43 localhost postfix/local[3635]: BA317AFACC:
to=<usuario16@localhost.optimail>, relay=local, delay=1.7,
delays=0.44/1.1/0/0.11, dsn=2.0.0, status=sent (delivered to maildir)
Jul 12 01:00:43 localhost postfix/local[3635]: BA317AFACC:
to=<usuario20@localhost.optimail>, relay=local, delay=1.7,
delays=0.44/1.2/0/0.05, dsn=2.0.0, status=sent (delivered to maildir)
Jul 12 01:00:43 localhost postfix/local[3634]: BA317AFACC:
to=<usuario19@localhost.optimail>, relay=local, delay=1.7,
delays=0.44/1.1/0/0.15, dsn=2.0.0, status=sent (delivered to maildir)
Jul 12 01:00:43 localhost postfix/local[3635]: BA317AFACC:
to=<usuario2@localhost.optimail>, relay=local, delay=1.7,
delays=0.44/1.3/0/0.01, dsn=2.0.0, status=sent (delivered to maildir)
Jul 12 01:00:43 localhost postfix/local[3635]: BA317AFACC:
to=<usuario4@localhost.optimail>, relay=local, delay=1.7,
delays=0.44/1.3/0/0.01, dsn=2.0.0, status=sent (delivered to maildir)
Jul 12 01:00:43 localhost postfix/local[3635]: BA317AFACC:
to=<usuario5@localhost.optimail>, relay=local, delay=1.7,
delays=0.44/1.3/0/0.01, dsn=2.0.0, status=sent (delivered to maildir)
Jul 12 01:00:43 localhost postfix/local[3632]: BA317AFACC:
to=<usuario18@localhost.optimail>, relay=local, delay=1.8,
delays=0.44/1.1/0/0.19, dsn=2.0.0, status=sent (delivered to maildir)
Jul 12 01:00:43 localhost postfix/local[3635]: BA317AFACC:
to=<usuario6@localhost.optimail>, relay=local, delay=1.8,
delays=0.44/1.3/0/0.01, dsn=2.0.0, status=sent (delivered to maildir)
Jul 12 01:00:43 localhost postfix/local[3635]: BA317AFACC:
to=<usuario7@localhost.optimail>, relay=local, delay=1.8,
delays=0.44/1.3/0/0.02, dsn=2.0.0, status=sent (delivered to maildir)
Jul 12 01:00:43 localhost postfix/local[3633]: BA317AFACC:
to=<usuario17@localhost.optimail>, relay=local, delay=1.8,
delays=0.44/1.1/0/0.23, dsn=2.0.0, status=sent (delivered to maildir)
Jul 12 01:00:43 localhost postfix/local[3636]: BA317AFACC:
to=<usuario1@localhost.optimail>, relay=local, delay=1.8,
delays=0.44/1.2/0/0.22, dsn=2.0.0, status=sent (delivered to maildir)
Jul 12 01:00:43 localhost postfix/local[3635]: BA317AFACC:
to=<usuario9@localhost.optimail>, relay=local, delay=1.9,
delays=0.44/1.3/0/0.09, dsn=2.0.0, status=sent (delivered to maildir)
Jul 12 01:00:43 localhost postfix/local[3634]: BA317AFACC:
to=<usuario3@localhost.optimail>, relay=local, delay=1.9,
delays=0.44/1.3/0/0.15, dsn=2.0.0, status=sent (delivered to maildir)
Jul 12 01:00:43 localhost postfix/local[3632]: BA317AFACC:
to=<usuario8@localhost.optimail>, relay=local, delay=1.9,
delays=0.44/1.3/0/0.12, dsn=2.0.0, status=sent (delivered to maildir)
Jul 12 01:00:43 localhost postfix/qmgr[2163]: BA317AFACC: removed
Jul 12 01:00:43 localhost dovecot: IMAP(rosal): Disconnected: Logged
out
Jul 12 01:01:33 localhost dovecot: imap-login: Login:
user=<usuario10>, method=PLAIN, rip=::ffff:192.168.0.105,
lip=::ffff:192.168.0.105, secured

```
Jul 12 01:01:34 localhost dovecot: imap-login: Login:
user=<usuario10>, method=PLAIN, rip=::ffff:192.168.0.105,
lip=::ffff:192.168.0.105, secured
Jul 12 01:01:34 localhost dovecot: IMAP(usuario10): Disconnected:
Logged out
```

F-2 Maillog (con archivo adjunto)

```
Jul 15 22:38:00 localhost dovecot: Dovecot v1.0.13 starting up
Jul 15 22:38:03 localhost sendmail[2121]: alias database /etc/aliases
rebuilt by root
Jul 15 22:38:03 localhost sendmail[2121]: /etc/aliases: 76 aliases,
longest 10 bytes, 765 bytes total
Jul 15 22:38:08 localhost postfix/postfix-script[2172]: starting the
Postfix mail system
Jul 15 22:38:08 localhost postfix/master[2173]: daemon started --
version 2.4.5, configuration /etc/postfix
Jul 15 22:41:35 localhost dovecot: imap-login: Login: user=<rosal>,
method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105,
secured
Jul 15 22:41:36 localhost dovecot: imap-login: Login: user=<rosal>,
method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105,
secured
Jul 15 22:41:36 localhost dovecot: IMAP(rosal): Disconnected: Logged
out
Jul 15 22:41:36 localhost dovecot: IMAP(rosal): Disconnected: Logged
out
Jul 15 22:41:59 localhost dovecot: imap-login: Login: user=<rosal>,
method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105,
secured
Jul 15 22:41:59 localhost dovecot: imap-login: Login: user=<rosal>,
method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105,
secured
Jul 15 22:41:59 localhost dovecot: IMAP(rosal): Disconnected: Logged
out
Jul 15 22:41:59 localhost dovecot: IMAP(rosal): Disconnected: Logged
out
Jul 15 22:43:22 localhost dovecot: imap-login: Time just moved
backwards by 1 seconds. I'll sleep now until we're back in present.
http://wiki.dovecot.org/TimeMovedBackwards
Jul 15 22:43:22 localhost dovecot: imap-login: Time just moved
backwards by 1 seconds. I'll sleep now until we're back in present.
http://wiki.dovecot.org/TimeMovedBackwards
Jul 15 22:43:22 localhost dovecot: pop3-login: Time just moved
backwards by 1 seconds. I'll sleep now until we're back in present.
http://wiki.dovecot.org/TimeMovedBackwards
Jul 15 22:43:22 localhost dovecot: pop3-login: Time just moved
backwards by 1 seconds. I'll sleep now until we're back in present.
http://wiki.dovecot.org/TimeMovedBackwards
```

```

Jul 15 22:46:05 localhost dovecot: imap-login: Login: user=<rosal>,
method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105,
secured
Jul 15 22:46:06 localhost dovecot: imap-login: Login: user=<rosal>,
method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105,
secured
Jul 15 22:46:06 localhost dovecot: IMAP(rosal): Disconnected: Logged
out
Jul 15 22:46:06 localhost sendmail[2964]: n6G3k6kg002964: from=apache,
size=3605, class=0, nrcpts=20,
msgid=<200907160346.n6G3k6kg002964@localhost.optimail>,
relay=apache@localhost
Jul 15 22:46:10 localhost postfix/smtpd[2965]: connect from
localhost.localdomain[127.0.0.1]
Jul 15 22:46:10 localhost postfix/smtpd[2965]: setting up TLS
connection from localhost.localdomain[127.0.0.1]
Jul 15 22:46:10 localhost postfix/smtpd[2965]: TLS connection
established from localhost.localdomain[127.0.0.1]: TLSv1 with cipher
DHE-RSA-AES256-SHA (256/256 bits)
Jul 15 22:46:10 localhost sendmail[2964]: STARTTLS=client,
relay=[127.0.0.1], version=TLSv1/SSLv3, verify=FAIL, cipher=DHE-RSA-
AES256-SHA, bits=256/256
Jul 15 22:46:11 localhost postfix/smtpd[2965]: 8EAB3AFACC:
client=localhost.localdomain[127.0.0.1],
sasl_sender=apache@localhost.optimail
Jul 15 22:46:11 localhost postfix/cleanup[2969]: 8EAB3AFACC: message-
id=<200907160346.n6G3k6kg002964@localhost.optimail>
Jul 15 22:46:11 localhost postfix/qmgr[2177]: 8EAB3AFACC:
from=<apache@localhost.optimail>, size=4222, nrcpt=20 (queue active)
Jul 15 22:46:11 localhost sendmail[2964]: n6G3k6kg002964:
to=usuario2@localhost.optimail,usuario3@localhost.optimail,usuario4@lo
calhost.optimail,usuario5@localhost.optimail,usuario6@localhost.optima
il,usuario7@localhost.optimail,usuario8@localhost.optimail,usuario9@lo
calhost.optimail,usuario10@localhost.optimail,usuario11@localhost.opti
mail,usuario12@localhost.optimail,usuario13@localhost.optimail,usuario
14@localhost.optimail,usuario15@localhost.optimail,usuario16@localhost
.optimail,usuario17@localhost.optimail,usuario18@localhost.optimail,us
uario19@localhost.optimail,usuario20@localhost.optimail,usuario1@local
host.optimail, ctladdr=apache (48/48), delay=00:00:05,
xdelay=00:00:05, mailer=relay, pri=603605, relay=[127.0.0.1]
[127.0.0.1], dsn=2.0.0, stat=Sent (Ok: queued as 8EAB3AFACC)
Jul 15 22:46:11 localhost postfix/smtpd[2965]: disconnect from
localhost.localdomain[127.0.0.1]
Jul 15 22:46:12 localhost postfix/local[2970]: 8EAB3AFACC:
to=<usuario11@localhost.optimail>, relay=local, delay=1.9,
delays=1.3/0.26/0/0.33, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:12 localhost postfix/local[2971]: 8EAB3AFACC:
to=<usuario10@localhost.optimail>, relay=local, delay=1.9,
delays=1.3/0.26/0/0.33, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:12 localhost postfix/local[2973]: 8EAB3AFACC:
to=<usuario12@localhost.optimail>, relay=local, delay=2.4,
delays=1.3/0.55/0/0.57, dsn=2.0.0, status=sent (delivered to maildir)

```

Jul 15 22:46:12 localhost postfix/local[2972]: 8EAB3AFACC:
to=<usuario13@localhost.optimail>, relay=local, delay=2.4,
delays=1.3/0.59/0/0.57, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:13 localhost postfix/local[2973]: 8EAB3AFACC:
to=<usuario16@localhost.optimail>, relay=local, delay=3,
delays=1.3/1.6/0/0.12, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:13 localhost postfix/local[2974]: 8EAB3AFACC:
to=<usuario15@localhost.optimail>, relay=local, delay=3,
delays=1.3/1.6/0/0.13, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:13 localhost postfix/local[2972]: 8EAB3AFACC:
to=<usuario17@localhost.optimail>, relay=local, delay=3,
delays=1.3/1.7/0/0.11, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:13 localhost postfix/local[2970]: 8EAB3AFACC:
to=<usuario18@localhost.optimail>, relay=local, delay=3,
delays=1.3/1.7/0/0.11, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:13 localhost postfix/local[2971]: 8EAB3AFACC:
to=<usuario19@localhost.optimail>, relay=local, delay=3.1,
delays=1.3/1.7/0/0.11, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:13 localhost postfix/local[2975]: 8EAB3AFACC:
to=<usuario14@localhost.optimail>, relay=local, delay=3.1,
delays=1.3/1.6/0/0.18, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:13 localhost postfix/local[2976]: 8EAB3AFACC:
to=<usuario20@localhost.optimail>, relay=local, delay=3.3,
delays=1.3/2/0/0.06, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:13 localhost postfix/local[2977]: 8EAB3AFACC:
to=<usuario1@localhost.optimail>, relay=local, delay=3.3,
delays=1.3/1.9/0/0.09, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:14 localhost postfix/local[2972]: 8EAB3AFACC:
to=<usuario4@localhost.optimail>, relay=local, delay=3.5,
delays=1.3/2/0/0.17, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:14 localhost postfix/local[2970]: 8EAB3AFACC:
to=<usuario5@localhost.optimail>, relay=local, delay=3.5,
delays=1.3/2/0/0.19, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:14 localhost postfix/local[2975]: 8EAB3AFACC:
to=<usuario3@localhost.optimail>, relay=local, delay=3.5,
delays=1.3/2/0/0.21, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:14 localhost postfix/local[2971]: 8EAB3AFACC:
to=<usuario2@localhost.optimail>, relay=local, delay=3.5,
delays=1.3/2/0/0.25, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:14 localhost postfix/local[2976]: 8EAB3AFACC:
to=<usuario9@localhost.optimail>, relay=local, delay=3.6,
delays=1.3/2.1/0/0.22, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:14 localhost postfix/local[2973]: 8EAB3AFACC:
to=<usuario6@localhost.optimail>, relay=local, delay=3.6,
delays=1.3/2/0/0.26, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:14 localhost postfix/local[2974]: 8EAB3AFACC:
to=<usuario7@localhost.optimail>, relay=local, delay=3.6,
delays=1.3/2.1/0/0.24, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:14 localhost postfix/local[2977]: 8EAB3AFACC:
to=<usuario8@localhost.optimail>, relay=local, delay=3.7,
delays=1.3/2.1/0/0.35, dsn=2.0.0, status=sent (delivered to maildir)
Jul 15 22:46:14 localhost postfix/qmgr[2177]: 8EAB3AFACC: removed

```
Jul 15 22:46:15 localhost dovecot: IMAP(rosal): Disconnected: Logged out
Jul 15 22:46:26 localhost dovecot: imap-login: Login: user=<rosal>, method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105, secured
Jul 15 22:46:26 localhost dovecot: imap-login: Login: user=<rosal>, method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105, secured
Jul 15 22:46:26 localhost dovecot: IMAP(rosal): Disconnected: Logged out
Jul 15 22:46:26 localhost dovecot: IMAP(rosal): Disconnected: Logged out
Jul 15 22:47:18 localhost dovecot: imap-login: Login: user=<usuariol>, method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105, secured
Jul 15 22:47:18 localhost dovecot: imap-login: Login: user=<usuariol>, method=PLAIN, rip=::ffff:192.168.0.105, lip=::ffff:192.168.0.105, secured
Jul 15 22:47:18 localhost dovecot: IMAP(usuariol): Disconnected: Logged out
Jul 15 22:47:19 localhost dovecot: IMAP(usuariol): Disconnected: Logged out
Jul 15 22:59:21 localhost dovecot: Killed with signal 15
Jul 15 23:34:26 localhost dovecot: Dovecot v1.0.13 starting up
Jul 15 23:34:27 localhost sendmail[2103]: alias database /etc/aliases rebuilt by root
Jul 15 23:34:27 localhost sendmail[2103]: /etc/aliases: 76 aliases, longest 10 bytes, 765 bytes total
Jul 15 23:34:30 localhost postfix/postfix-script[2154]: starting the Postfix mail system
Jul 15 23:34:30 localhost postfix/master[2155]: daemon started -- version 2.4.5, configuration /etc/postfix
```