Universidad de las Ciencias Informáticas

Facultad 2

Software para la interconexión de redes aisladas. Módulo correo electrónico V2.0

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores:

Bárbara Daysi Bedoya López

Yosbel Morales Vazquez

Tutor(es):

Ing. Carlos Manuel Hernández Vega

Ing. Adrian Hernández Yeja

La Habana

"Año 54 de la Revolución"

Junio,2012

DECLARACIÓN DE AUTORÍA

Declaramos que _____ y _____somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) y a la Facultad (2) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de junio del 2012.

_____ _____

Firma del Autor Firma del Autor

Bárbara Daysi Bedoya López Yosbel Morales Vázquez

_____ _____

Firma del Tutor Firma del Tutor

Ing. Carlos Manuel Hernández Vega Ing. Adrian Hernández Yeja

DEDICATORIA

De Yosbel:

A mi familia, por estar todo el tiempo pendientes de mi, y sobre todo a mis dos grandes tesoros, mi mamá y mi papá, por apoyarme todo el tiempo, por sacrificarse todos los días y por entregarlo todo para verme hoy aquí, por hacer de mi el hombre que soy hoy y por el amor que solo ustedes saben darme, porque son lo más grande que tengo, por ser mi motivación y mis guías.

Para ustedes y para todos los que creyeron en mi es todo lo que he logrado hasta hoy.

De Baby:

Le dedico mi trabajo de diploma a mi mamita querida, por estar siempre a mi lado y apoyarme en todo momento, y sobre todo por el maravilloso ejemplo que siempre me ha dado. A mi hermano, que es mi vida, y mi compañero de juegos, de peleas, de todo, desde que nació, junto a él y mi madre somos los 3 mosqueteros. A mi bisabuela por ser nuestro DaÇtañán, te adoro ia, a ti también va dedicado este triunfo.

RESUMEN

La capacidad para transportar correos de una red externa hacia una red interna y viceversa, ha constituido siempre una funcionalidad clave en centros de datos que prestan servicios de correo institucional. Esta comunicación es realizada en toda su trayectoria por conexiones de red TCP, lo que trae consigo la apertura de las brechas de seguridad propias del protocolo. En el Ministerio de Informática y Comunicacionesse implementa una pasarela de correos que aísla físicamente las redes interna y externa. La pasarela requiere de personas que manualmente mueven los mensajes de una subred a otra almacenando los archivos de los mensajes en memorias externas tipo Flash con puerto USB.

La presente solución contribuye con la automatizacióndel proceso y a su vez seguirmanteniéndolo transparente al usuario. La solución se compone de un servidor de transporte de correosadaptado para prestar servicios en cada red aislada; un dispositivo de hardware con memoria interna para trasportar los mensajes y una aplicación de gobierno para controlar el dispositivo y gestionar las copias desde y hacia los servidores de transporte de correo.

En el presente trabajo se describen los entornos de aplicación y la eliminación de las desventajas del sistema anterior del MIC con la nueva versión de la solución.

PALABRAS CLAVE

Red privada, servidor de correos, transporte de correos, MTA, agente de transferencia de correo.

ÍNDICE

ÍNDICE DEFIGURAS

INTRODUCCIÓN

Las amenazas originadas desde internet se han incrementado en los últimos años y están lejos de considerarse en el término bajo control.Esto es el resultado del crecimiento acelerado del intercambio de información entre empresas oficinas de gobierno, organizaciones académicas y otros usos que extienden la expansión de la transferencia de datos.

En otras palabras existe una gran cantidad de información disponible para empleados, especialistas y personas en sentido general, que crean peligrosas oportunidades de ataques por los llamados hackers. Existen múltiples riesgos a los que están expuestas las empresas como virus, gusanos, accesos no autorizados, etc.; lo que hace que la protección de la información se convierta en un problema primariode las organizaciones.

Hasta el momento,un gran número de tecnologías de seguridad, como cortafuegos, herramientas anti-virus, o los sistemas de detección de intrusos, se ofrecen para proteger los intercambios de datos y comunicaciones electrónicas. Sin embargo, a pesar de la ubicuidad y el desarrollo constante de este tipo de soluciones, las redes y sus recursos siguen siendo muy delicadas y vulnerables. Hasta ahora, todos estos métodos no son suficientes para satisfacer completamente la seguridad en entornos altamente críticos como el Ministerio de la Informática y las Comunicaciones.

Una de las alternativas más radicales para evitar estos problemas lo constituye la separación física de los activos involucrados en el sistema a proteger el cual se ha propuesto desde hace años como una solución de alto de nivel de seguridad, que espera evitar por completo los ataques de intrusos mediante la separación física de las redes.

El Ministerio de la Informática y las Comunicaciones es el organismo encargado de regular, dirigir, supervisar y controlar la política del estado y el gobierno en cuanto a las actividades detecnologías informáticas, telecomunicaciones, redes de infocomunicaciones, servicios de valor agregado en infocomunicaciones, radiodifusión, espectro radioeléctrico, automatización, servicios postales e industria electrónica.

Para garantizar la integridad de todos los procesos y la seguridad de la información clasificada y altamente importante, que se maneja en el ministerio, se encuentraimplementada una capa adicional de seguridad la cual establece precisamente que no existan conexiones físicas entre la red interna y la externa conectada a Internet. Específicamente lo que se pretende evitar es todo tipo de conexión mediantelos protocolos TCP/IP.

En el Ministerio existe el servicio de correo institucional, que permite la comunicación electrónica entre los trabajadores y los diferentes directivos del centro.

Para prestar el servicio de correo disponible desde la red interna, al no existir conexión entre las redes, se emplea una pasarela de correos manual en la cual existen buzones de correos intermedios a los cuales los usuarios les envían los correos que van hacia el exterior.

Existe un servidor de correos externo con acceso a Internet, en el que están definidos usuarios genéricos con acceso a enviar y recibir correos con el exterior.Igualmente existen servidores de correos internos con los buzones de los usuarios del Ministerio y otros usuarios genéricos homólogos a los del servidor externo. Los servidores internos están aislados, por lo que los correos se envían a los usuarios genéricos internos. Frecuentemente los buzones de estos usuarios son revisados por los operadores de la plataforma, los cuales copian manualmente los correos en una memoria externa de tecnología flash y conexión USB conectada a una estación de trabajo en la red interna, luego se conecta la memoria en otra estación de trabajo en la red externa y se reenvían los correos desde los usuarios genéricos externos.

El procedimiento pararecibir correos desde el exterior se realiza con el proceso contrario, donde los operadores copian los correos que reciben desde el exterior en la estación de trabajo externa y los trasladan en la memoria flash hasta una estación de trabajo interna y de allí se envían para los usuarios genéricos internos.

La problemática descrita demuestra lo engorroso que se torna el envío y recepción entre ambas redes, la interna y la externa, de los correos electrónicos. Se presentan algunas de

las desventajas más significativas en el desarrollo actual de dicho proceso con el objetivo de tener una mayor claridad de las mismas:

- Procesamiento manual en la manipulación de la información. Esto introduce posibilidad de errores humanos y abre una brecha de seguridad producto a que las operadoras tienen acceso a la información que se maneja en los correos.
- Empleo de múltiplesrecursos. Recursos de hardware: 8 computadoras, 2 servidores (correo externo y correo interno auxiliar), 3 memorias flash. Recursos humanos: 3 personas y 3 puestos de trabajo.
- Engorrosa implementación del sistema, debido a que se emplean muchas reglas en el servicio de correo.
- Despersonalización del remitente del mensaje debido a que el remitente es suplantado por un mensaje que envía un usuario genérico. Esto imposibilita el seguimiento a los mensajes y los servicios de interacción con el remitente.
- Incomodidad tanto para usuarios internos como externos. Esto se debe a que hay que extraer los mensajes adjuntos y responder al usuario final a través de un remitente genérico.
- Retraso excesivo en la llegada de los mensajes a su destino.

En resumen en el MIC no se implementaba un servicio de correos institucional adecuado a las necesidades de la institución.

Para dar solución a estos problemas el Proyecto Centro de Datos del Departamento de Seguridad Informática del Centro de Telemática de la Universidad de las Ciencias Informáticas tuvo como tarea el desarrollo de una versión 1.0 del producto. La misma fue implementada en bash, no contiene los requerimientos necesarios de seguridad para una solución de su tipo, dígase comprobar identificador de la memoria asegurándose que no haya una suplantación de la misma, además de algún mecanismo de encriptación para la protección de la información. La interfaz web no brinda las funcionalidades necesarias para la administración de la aplicación de gobierno del dispositivo, no tiene escalabilidad ni flexibilidad, además de que no implementa recuperación ante cualquier tipo de fallo o excepción.

Tomando en cuenta lo anteriormente expuesto y la necesidad del MIC de encontrar una solución para estas dificultades se plantea como problema a resolver: ¿Cómo hacer robusto, confiable y escalableel software para la interconexión de redes aisladas?

Para dar respuesta al problema científico se propone como objeto de estudio los Servicios de Correo Electrónico.

Se enmarca el campo de acciónde la investigación en los Agentes de Transporte de Correo.

Como objetivo generalde la investigación se proponedesarrollar una nueva versión del software para la interconexión de redes aisladas, que posea escalabilidad, seguridad y robustez.

Desglosando el objetivo general en los siguientesobjetivos específicos:

1. Implementar un servicio de transporte de correos MTA que almacena los correos enviados hacia determinada red en una memoria Flash en aras de resolver los problemas de la versión 1.0.
2. Desarrollarel software de gobiernodel dispositivo de hardware que intercambia las memorias Flash entre puertos USB de servidores en redes diferentes.
3. Desarrollar una aplicación de control y administración para brindar unasolución integral con el software que gobierna el servicio de transporte de correos y el dispositivo.

Para cumplir con los objetivos se propone la realización de las siguientes tareas de investigación:

1. Investigación acerca de las tecnologías, técnicas y productos para pasarelas de correos.
2. Definición del entorno de desarrollo del software, las tecnologías a usar y su arquitectura.
3. Investigación sobre los servidores de correos y configuraciones para establecer un flujo de mensajes desde y hacia almacenamiento local.
4. Investigación acerca de la interacción y control de dispositivos de hardware.
5. Análisisdel software de gobierno del servidor de correos y el dispositivo de hardware.

6. Implementación la interfaz web de administración del software.
7. Despliegue de la solución en un entorno controlado.
8. Despliegue de la solución en el entorno de producción final.
9. Investigación de diferentes algoritmos de encriptación para la implementación de la seguridad del sistema.

Queda definida como idea a defender que la implementación de la nueva versión del software para la interconexión de redes aisladas mejorará el servicio de correo institucional en el MIC.

Métodos Teóricos:

- Análisis-síntesis: Este método fue utilizado en todo el proceso investigativo, ya que a través de la descomposición del problema en varias partes, se precisan características y particularidades de cada una de ellas lo que posibilita llegar a un mejor entendimiento y comprensión del problema en cuestión, para luego con los resultados obtenidos del análisis, buscar sus relaciones y similitudes.

Métodos empíricos:

- Observación: Este método permitirá obtener conocimiento acerca del comportamiento de un sistema de supervisión. Permitirá observarcómo funcionan, cómo se desarrollan así como las ventajas que brindan y sus especificidades de acuerdo al lugar donde se implemente.

Técnicas de investigación:

Las técnicas de investigación es más que nada la recopilación de datos para verificar los métodos empleados en lo investigado, para llegar a la verdad del suceso estudiado, teniendo las pruebas y una serie de pasos que se llevan a cabo para comprobar la hipótesis planteada.

- Entrevista: La entrevista es una técnica antigua, pues ha sido utilizada desde hace mucho en psicología y, desde su notable desarrollo, en sociología y en educación. De hecho, en estas ciencias, la entrevista constituye una técnica indispensable porque permite obtener datos que de otro modo serían muy difícil conseguir.

  En el anexo 13 se refleja todos los detalles de la entrevista.

El presente documento se estructura en 4 capítulos:

En el Capítulo 1 se abordan los conceptos y definiciones investigados durante el estudio delas tendencias actuales en el contexto en el cual se desarrolla este trabajo. Incluye además un estudio de las herramientas utilizadaspara el desarrollo del proyecto, así como las tecnologías, herramientas y lenguajes existentesútiles para el proceso de implementación.

En el Capítulo 2 se describirán las características del sistema a desarrollar, así como el objeto de automatización, la información que se maneja. Se realizará una propuesta de sistema y el modelo de negocio, las especificaciones de requisitos de software, las dependencias reales y relaciones con otrossoftwares, los requerimientos funcionales y no funcionales y la definición de los casos de uso.

En el capítulo 3 se abordarán todos los elementos referidos al diseño del sistema.

En el capítulo 4, dedicado a la implementación y a las pruebas del software, se reflejarán los diferentes diagramas de despliegue, de componentes, etc., asi como los diferentes casos de prueba y los resultados de los mismos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En este capítulo se abordan los conceptos y definiciones investigados durante el estudio delas tendencias actuales en el contexto en el cual se desarrolla este trabajo con los cuales se pretende lograr un basamento teórico. Se argumenta sobre el concepto de Pasarela de Correos, se describe el ciclo completo de funcionamiento de la pasarela de correos CID-555. Del mismo modo incluye un estudio de las herramientas utilizadaspara el desarrollo del proyecto, así como las tecnologías, herramientas y lenguajes existentes para tales propósitos.

1.2. Redes aisladas

Las empresasy agencias gubernamentalescon una infraestructura de tecnología informática altamente sensibleno puede prescindir dela protección integral desus redes y datos. Esto se aplica auna amplia gama deorganismos en todo el mundo, incluida la policía, el ejército, los bancos, las compañías deseguros, empresasde aviacióny el transporte, de telecomunicaciones yproveedor de energía, así comooficinas de registro deresidentesy las autoridades fiscales. Algo importante para cualquierinstitución y organización es la confianza, el intercambio confidencial y seguro de datos entrelas redes. Un mecanismo de protección altamente radica y seguro lo constituye la separaciónfísica de las redes.

La separaciónfísica delas conexiones de redesgarantizaun intercambio de datoscontrolado.Es complejo configurar una red que no contenga errores,y esigualmente complejo el control y protección de las mismas ante cualquier ataque o penetración.[1]

1.3. Pasarela de correo

Una pasarela de correono es más que una o variascomputadorasque se utilizanpara conectar a dos o más sistemas de correo electrónico, transfiriendo mensajes entre ellos. Una pasarela de correo recoge, guarda y envía mensajes a un servidor de dominio principal; habitualmente, no dispone de cuentas individuales. Las pasarelas de correo pueden ayudar a empresas grandes y pequeñas a resolver algunos problemas de los servicios de correo por internet. En una empresa pequeña, una pasarela puede hacer que un dominio de correoprivado sea asequible. Las empresas más grandes pueden utilizar pasarelas de seguridad más estricta para proporcionar protección y filtrar los contenidos de sus sistemas de correo corporativo, más vulnerable. Las pasarelas también pueden configurarse para proveer el backup automático del correo en tiempo real cuando un dominio primario se desactiva por algún motivo.

¿Cómo funciona?

Consiste en un equipo de filtrado antivirus y antispam que se instala en elprincipal Centro de Datos y que protege todo el tráfico de correo. Realiza exclusivamente la función de filtro, por lo que acepta el correo de entrada, lo procesa con diferentes herramientas de filtrado y lo entrega a un servidor de correo externo (tanto de alojamiento compartido, como servidor dedicado o un equipo propio en su centro de trabajo).[2]

1.4. Funcionamiento del Correo Electrónico

El correo electrónico gira alrededor del uso de las casillas de correo electrónico. Cuando se envía un correo electrónico, el mensaje se enruta de servidor a servidor hasta llegar al servidor de correo electrónico del receptor. Más precisamente, el mensaje se envía al servidor del correo electrónico (llamado MTA, del inglés Mail Transport Agent [Agente de Transporte de Correo]) que tiene la tarea de transportarlos hacia el MTA del destinatario. En Internet, los MTA se comunican entre sí usando el protocolo SMTP, y por lo tanto se los llama servidores SMTP (o a veces servidores de correo saliente).

Luego el MTA del destinatario entrega el correo electrónico al servidor del correo entrante (llamado MDA, del inglés Mail Delivery Agent [Agente de Entrega de Correo]), el cual almacena el correo electrónico mientras espera que el usuario lo acepte. Existen dos protocolos principales utilizados para recuperar un correo electrónico de un MDA:

POP3 (Post Office Protocol [Protocolo de Oficina de Correo]), el más antiguo de los dos, que se usa para recuperar el correo electrónico y, en algunos casos, dejar una copia en el servidor.

IMAP (Internet Message Access Protocol [Protocolo de Acceso a Mensajes de Internet]), el cual se usa para coordinar el estado de los correos electrónicos (leído, eliminado, movido) a través de múltiples clientes de correo electrónico. Con IMAP, se guarda una copia de cada mensaje en el servidor, de manera que esta tarea de sincronización se pueda completar.

Por esta razón, los servidores de correo entrante se llaman servidores POP o servidores IMAP, según el protocolo usado.[3]

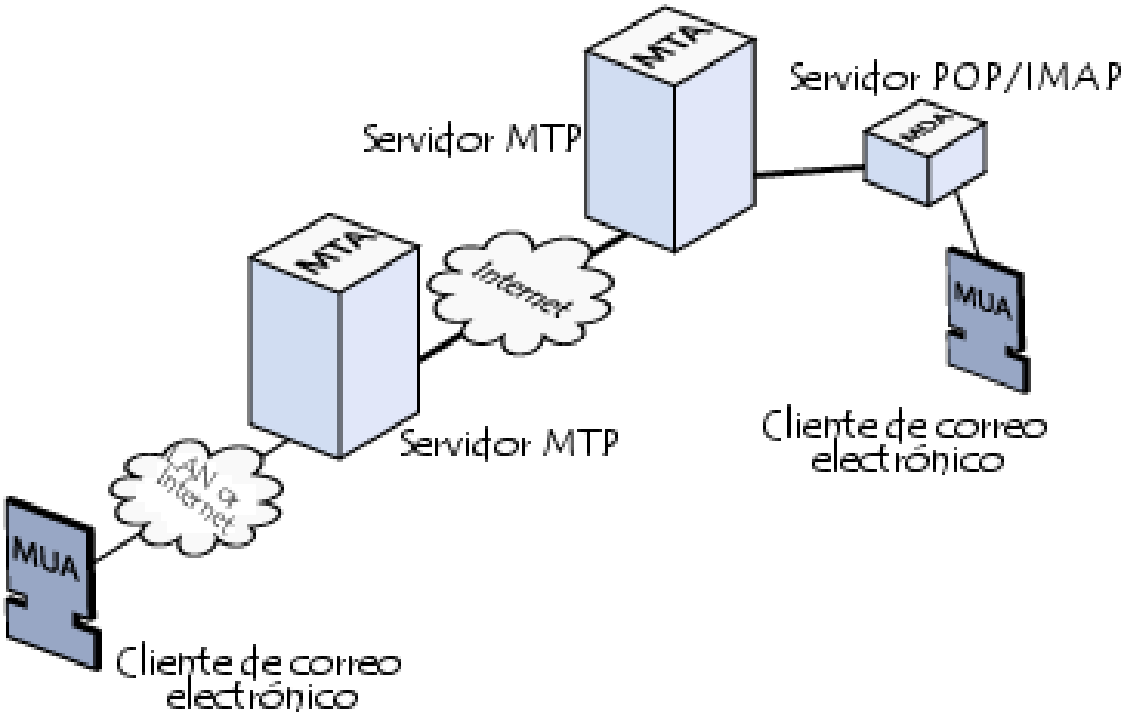


Figura 1.Representación del funcionamiento del correo electrónico.

Usando una analogía del mundo real, los MTA actúan como la oficina de correo (el área de clasificación y de transmisión, que se encarga del transporte del mensaje), mientras que los MDA actúan como casillas de correo, que almacenan mensajes (tanto como les permita su volumen), hasta que los destinatarios controlan su casilla. Esto significa que no es necesario que los destinatarios estén conectados para poder enviarles un correo electrónico.[4]

La recuperación del correo se logra a través de un programa de software llamado Mail User Agent (MUA [Agente Usuario de Correo]).

Cuando el MUA es un programa instalado en el sistema del usuario, se llama cliente de correo electrónico, tales como, Mozilla Thunderbird, Microsoft Outlook, Eudora Mail, Incredimail o Lotus Notes.

Cuando se usa una interfaz de web para interactuar con el servidor de correo entrante, se llama correo electrónico. [5]

1.5. Agentes de Transporte de Correo

Para la implementación de esta pasarela es necesario un Agente de Transporte de Correos (MTA).Dentro del mundo de código abierto existen varios MTAs, entre ellos, los más conocidos y populares son:

- Sendmail[6]
- Qmail[7]
- Postfix[8]
- Exim[9]

Cada uno de dichos MTAs tienen características funcionales similares, todos pueden manejar cantidades grandes de tráfico de e-mail, pueden interactuar con sistemas de Bases de Datos en varios formatos, manejan las variantes de SMTP que existen actualmente, no son trivialmente hackeables, entre otros.

Los criterios de selección más comúnmente utilizados para decidir la implantación de un MTA son:[4]

- Facilidad de administración
- Seguridad
- Desempeño
- Confiabilidad

Postfix(licenciado bajo la IBM PublicLicense)fue escrito por un especialista en seguridad, WietseVenema, y comenzó como una alternativa a Sendmail. Los objetivos de Postfix

fueron el ser rápido, de fácil administración y seguro, pero a la vez ser lo suficientemente compatible con Sendmail pensando en los administradores y usuarios de este último. Postfix es distribuido como servidor de correo predeterminado en algunas distribuciones comoMandrake, SuSE, RedHat y FedoraCore.[10]

Postfix consiste de varios programas aunque esta modularidad no se ve en la configuración, ya que consta de 2 archivos grandes (master.cf y main.cf). Postfix consta de muchas opciones de configuración, sin embargo los valores por defecto son en general una buena opción y solamente algunas pocas opciones deben ser modificadas para un sistema dado. También las opciones son más claras que aquellas encontradas en Sendmail.

A diferencia de Sendmail, los procesos de Postfix corren en el servidor con privilegios restringidos, solo los necesarios para realizar su tarea, y dichos procesos son aislados unos de otros haciendo que el trabajo de Postfix se divida en rutinas pequeñas y desconectadas.

Soporta:[10]

- Entrega a casillas de mail con formato Maildir y mbox.
- Enmascaramiento de usuario y host.
- Dominios virtuales.
- SMTP AUTH (utiliza biblioteca Cyrus SASL).
- Soporte para LDAP, MySQL, Berkley DB, PostrgeSQL
- IPv6.
- Cuota de usuario.
- Soporte para sesiones encriptadas vía TLS.
- Consultas a Base de Datos de listas negras y blancas de clientes SMTP.
- Control por host de SMTP Relay.

Por estas razones se decidió la utilización de Postfix como MTA para la implementación delsoftware para la interconexión de redes aisladas. Posteriormente y tomando las bases

de las soluciones existentes, se da la necesidad de investigar las herramientas a utilizar en el desarrollo de solución.

1.6. Criptografía

La criptografía se encarga del estudio de los algoritmos, protocolos y sistemas necesarios para la protección de la información. Su objetivo principal es diseñar, implementar, implantar y hacer uso de sistemas criptográficos para garantizar de alguna forma la seguridad. Por eso se ocupa de la confidencialidad de la información y que esté accesible únicamente a personal autorizado, para conseguirlo utiliza códigos y técnicas de cifrado, también garantiza la integridad ya que se realiza la corrección y completitud de la información, también proporciona protección frente a que alguna de las entidades implicadas en la comunicación pueda negar haber participado en toda o parte de la comunicación, esto es conocido como no repudio, además proporciona mecanismos de autenticación que permiten verificar la identidad del comunicante.[11]

1.6.1. Funciones HASH

Las funciones hash se utilizan para garantizar la integridad de los datos que viajan por la red, éstas ayudan a cumplir con uno de los principales objetivos de la criptografía.[12]

El resultado de aplicar una función resumen a un fichero es un número grande, el número resumen, que tiene las siguientes características:

- Todos los números resumen generados con un mismo método tienen el mismo tamaño sea cual sea el texto utilizado como base.
- Dado un fichero base, es fácil y rápido (para un ordenador) calcular su número resumen.
- Es imposible reconstruir el fichero base a partir del número resumen.
- Es muy difícil con la infraestructura computacional actual que dos ficheros base diferentes tengan el mismo número resumen.

Ejemplos de funciones hash:

- MD4
- MD5
- SHA1
- SHA256

El SHA (Secure Hash Algorithm) es uno de una serie de funciones criptográficas de hash.Un hash criptográfico es como una firma para un texto o un archivo de datos. es un algoritmo que implementa una función resumen que genera una cadena de 256 bits. SHA-256 Esto lo hace adecuado para la validación de contraseña, autenticación de desafío de hash, anti-sabotaje, las firmas digitales.

SHA-256 es una de las funciones de hash sucesores a SHA-1 , y es uno de los más fuertes funciones disponibles hash. [13]

1.7. Tipos de algoritmos criptográficos

1.7.1.1. Algoritmos Simétricos

Utilizan una clave con la cual se encripta y desencripta el documento. Todo documento encriptado con una clave, deberá desencriptarse, en el proceso inverso, con la misma clave. Es importante destacar que la clave debería viajar con los datos, lo que hace arriesgada la operación, imposible de utilizar en ambientes donde interactúan varios interlocutores.

Se presentan algunos ejemplos de este tipo de algoritmos:[14]

- Algoritmo DES
- DES Múltiple
- Idea (International Data Encryption Algorithm)
- Algoritmo de Rijndael (AES)

- BlowFish
- RC5
- CAST

El algoritmo AES, también conocido como Rijndael, es un algoritmo de cifrado de bloque que ha sido adoptado como estándar por el gobierno de los EE. UU. Se espera que sea usado ampliamente y analizado de forma extensa como fue el caso de su predecesor el Data Encryption Standard (DES). El AES en la actualidad es el algoritmo de cifrado estándar para todo tipo de aplicación que utilice cifrado de datos y se espera que permanezca varias décadas más. Por estas razones se utiliza este algoritmo en la encriptación de datos en la pasarela de correos.[15]

1.8. Sistemas Similares

Las amenazas procedentes de internet aumentan cada día y son cada vez mayores y están lejos de estar “bajo control”. La seguridad moderna está diseñada para proteger la amplia gama de servicios de comunicación empresarial de los intrusos externos e internos llamados “hackers”. Debido a esto se han desarrollado diversos mecanismos de defensa para proteger los datos internos y los sistemas de acceso no autorizados. Basándose en el simple principio de que “el método perfecto para asegurar una red es que esté desconectada”, “la separación física” se ha propuesto desde hace años como una solución de alto de nivel de seguridad, que espera evitar por completo los ataquesde intrusos mediante la separación física de las redes.

Lock-Keeperes una aplicación novedosa de separación física surgida en el año 2003 y comercializada por Siemens Switzerland, en los últimos años esta aplicación ha sido mejorada para ser más maduro y confiable. Debido a esta separación física Lock-Keeperevita automáticamente que todos los conocidos y hastaformas desconocidas de los ataques en línea: spyware,puertas traseras, los ataques a nivel de protocolo (por

ejemplo,como los ataques TCP número de secuencia), construcción de túneles, mensaje encapsulado, entre otros.[16]

El productoe-Gapesuna combinación única dehardware y softwareque permitela desconexiónde Internet.En lugar deconectarfísicamentetantola zona de distensióny las redes internas, los sistemas dee-Gap permite lacomunicacióna través deun banco de memoriade estado sólidoSCSIconectado a un hosten la DMZ yun host enel interior. El aparatotieneunconmutador analógicode alta velocidadque permite que sólounadelasdos máquinaspueda comunicarsecon éla la vez.Cuandouna petición llegadesde la red externa, el anfitrión en la DMZcon un softwareexternoacepta la solicitud,elimina todos losencabezados TCPy vuelcala carga útilde datos sóloen launidad de disco SCSI. Elconmutador analógicoladesconecta delhost externoy la conecta alhost internoen el aparato.El anfitrión,ejecutando el softwareinterno,procesa la solicitud, añadesuspropiascabeceras de TCPy la envíaal servidor dedispositivoreal.Este procesose realizaa la inversa cuando la respuesta se envíade vuelta al usuario inicial. El sistema también puedeserconfigurado en el modode un solo sentidoconuna llave físicaque bloqueael conmutador analógico, permitiendo la comunicación entrante osaliente.[17]

Estas soluciones tienen la desventaja de tener un precio muy elevado y por las condiciones económicas delpaís no se pueden asumir estos gastos, por lo que eran necesarias otras soluciones más factibles y que contribuyan a la independencia tecnológica.

Se decidió la construcción de un dispositivo para automatizar el envío de correo, esta tarea le fueencomendada al Instituto Central de Investigaciones Digitales (ICID).

Este dispositivo fue utilizado en la versión 1.0 del Software para la interconexión de redes aisladas, es decir que ya existe un software elaborado para el mismo.

El ICID (Instituto Central de InvestigacionesDigitales) es la empresa más moderna yexperimentada en el tema de la electrónica en Cuba. Incluye facilidades para la producción electrónicay electromecánica, soportado por décadas deexperiencia con las

más modernas tecnologías. Además fabrica partes mecánicas de alta calidad para equipos médicos y otros. Con más de 30 años de experiencia en la fabricación, la planta de circuitos impresos procesa una gran cantidad de códigos diferentes y fabrica pequeños y medianos lotes con alta eficiencia. El sistema de aseguramiento de la calidad de las plantas de producción del ICID está basado en las normas ISO9001-2000 y certificado por Bureau Veritas Quality Internacional, institución acreditada por UKAS. [18]

Debido a esto se le solicitó la creación del dispositivo de interconexión CID-555.

El dispositivo de conmutación está conectado a ambos servidores por dos cables serie con conector DB9 y dos cables USB. Internamente tiene dos memorias Flash para el almacenamiento de los mensajes.

Las memorias Flash son conmutadas entre ambos servidores por relés electromagnéticos, de tal manera que en un instante de tiempo cada memoria está conectada a un único servidor.

La comunicación con el dispositivo para su control se realiza por las terminales serie. Las interfaces permiten notificar el estado de la copia hacia la memoria por la solución de software y ordenarle conmutar los relés. Para conocer el estado de la copia, el dispositivo implementa un registro para el Servidor Externo. Para realizar la conmutación el dispositivo implementa un segundo registro que al ser escrito acciona los relés.

Hacia el Servidor Externo solo se presenta el registro bandera de estado copiando en modo escritura. Hacia el Servidor Interno se presenta el registro bandera de estado copiando en el Servidor Externo en modo lectura y el registro orden de conmutación en modo escritura.

Los componentes del dispositivo son:

- 2 Microcontroladores que establece la lógica de funcionamiento
- 2Módulos de comunicación
- 2 Módulos de conmutación
- 5 Relés electromagnéticos (1 para 1 conexión, 4 para 5 conexiones)

- 2 Puertos DB9
- 2 Puertos USB
- 2 Memorias Flash

Los terminales DB9 del dispositivo deben ser conectados correctamente al servidor correspondiente para el correcto funcionamiento del sistema. Los puertos USB pueden conectarse indistintamente uno a cada servidor.

La arquitectura del dispositivo garantiza que no exista comunicación de señales eléctricas entre el módulo Externo y el Interno. El estado se da a conocer mecánicamente por la acción del Relé de estado.

1.9. Tecnologías utilizadas

1.9.1. Middleware Zeroc-ICE

El motor de comunicaciones de Internet(ICE) es una herramienta orientada a objetos moderna que le permite construir aplicaciones distribuidas con el mínimo esfuerzo. ICE permite centrar los esfuerzos en la lógica de aplicación, y se encarga de todas las interacciones con bajo nivel de programación de interfaces de red. Con Ice, no hay necesidad de preocuparse por los detalles como la apertura de las conexiones de red, serialización y deserialización de datos para la transmisión de la red, o volver a intentar los intentos fallidos de conexión.

Es compatible con C + +, Java, .NET lenguajes como C # o Visual Basic, Objective-C, Python, PHP y Ruby en la mayoría de los principales sistemas operativos como Linux, Solaris, Windows y Mac OS X.

Es por todas estas ventajas y facilidades que se decide utilizar ICE para la comunicación con la interfaz Web de la aplicación.

1.9.2. Lenguaje de Programación

Es un conjunto de símbolos junto a un grupo de reglas para combinar dichos símbolos que se usan para expresar programas. Constan de un léxico, una sintaxis y una semántica.

El léxico no es más que un conjunto de símbolos permitidos o vocabulario, la sintaxis son las reglas que indican cómo realizar las construcciones del lenguaje, la semánticas son las reglas que permiten determinar el significado de cualquier construcción del lenguaje y los tipos de lenguajes son lo queatendiendo al número de instrucciones necesarias para realizar una tarea específica se puede clasificar los lenguajes informáticos en dos grandes bloques:

- Bajonivel
- Alto nivel

Lenguaje de bajo nivel

Es el tipo de lenguaje que cualquier computadora es capaz de entender. Se dice que los programas escritos en forma de ceros y unos están en lenguaje de máquina, porque esa es la versión del programa que la computadora realmente lee y sigue.

Lenguajes de alto nivel

Son lenguajes de programación que se asemejan a las lenguas humanas usando palabras y frases fáciles de entender.

En un lenguaje de bajo nivel cada instrucción corresponde a una acción ejecutable por el ordenador, mientras que en los lenguajes de alto nivel una instrucción suele corresponder a varias acciones.

Los lenguajes de alto nivel son:

Independientes de la arquitectura física de la computadora. Permiten usar los mismos programas en computadoras de diferentes arquitecturas (portabilidad), y no es necesario conocer el hardware específico de la máquina. La ejecución de un programa en lenguaje de alto nivel, requiere de una traducción del mismo al lenguaje de la computadora donde va a

ser ejecutado. Una sentencia en un lenguaje de alto nivel da lugar, al ser traducida, a varias instrucciones en lenguaje entendible por el computador. Utilizan notaciones cercanas a las usadas por las personas en un determinado ámbito. Se suelen incluir instrucciones potentes de uso frecuente que son ofrecidas por el lenguaje de programación.

1.9.2.1. Python

Es un lenguaje interpretado, orientado a objetos, de alto nivel que permite escribir código con una alta claridad y legibilidad, permitiendo así un rápido aprendizaje del mismo. La legibilidad permitirá en futuros accesos al código una clara comprensión de lo antes implementado, haciendo más fácil el mantenimiento de las aplicaciones.

Su biblioteca estándar es muy amplia, conteniendo funcionalidades de gran ayuda desde el más bajo nivel hasta el más alto, facilitándole al programador la implementación de aplicaciones sin la necesidad de recurrir continuamente a bibliotecas externas. Además dispone de una extensa colección de bibliotecas libres disponibles en la mayoría de los repositorios de los sistemas GNU\Linux.

Este lenguaje permite ser extendido, haciéndolo mucho más favorable para su uso, pues en caso de existir alguna región crítica del proyecto que al ser escrita en Python no sea lo más recomendable u óptimo, esta puede ser implementada en C/C++ y compilada de modo que sea accesible desde el intérprete de Python, aumentando considerablemente el rendimiento de esa sección a programar.

El intérprete de Python funciona en diversos sistemas operativos, tales como: GNU\Linux, Mac OS X y Windows. Debido a la portabilidad de su código; al programar las aplicaciones sin utilizar bibliotecas propias a los sistemas operativos, es posible ejecutarlas en cualquier plataforma sobre la que exista el intérprete de Python.

Teniendo en cuenta que para el desarrollo de los driver del hardware al cual implementaremos el software el lenguaje utilizado fue Python, y además por todas las

funcionalidades que brindan para la comodidad del desarrollo, y para seguir la misma línea, se propone como lenguaje de programación.

1.9.2.2. PHP

PHP es un lenguaje de scripting que permite la generación dinámica de contenidos en un servidor web. El significado de sus siglas es HyperText Preprocessor. Entre sus principales características cabe destacar su potencia, su alto rendimiento, su facilidad de aprendizaje y su escasez de consumo de recursos.

PHP es el lenguaje de lado servidor más extendido en la web. Nacido en 1994, se trata de un lenguaje de creación relativamente reciente, aunque con la rapidez con la que evoluciona Internet parezca que ha existido toda la vida. Es un lenguaje que ha tenido una gran aceptación en la comunidad de desarrolladores, debido a la potencia y simplicidad que lo caracterizan, así como al soporte generalizado en la mayoría de los servidores de hosting.

PHP permite embeber su pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz, combinando lo que ya se sabe del desarrollo HTML. Es decir, con PHP se escribenscripts dentro del código HTML. Por otra parte, PHP ofrece un sinfín de funciones para la explotación de bases de datos de una manera llana, sin complicaciones.

El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable. Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Permite aplicar técnicas de programación orientada a objetos. Además es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

1.9.3. Herramientas de software para el desarrollo de la solución

1.9.3.1. Herramienta CASE

Dentro de las herramientas claves en el desarrollo de aplicaciones informáticas se encuentran las herramientas de Ingeniería de Software Asistida por Ordenador (CASE), las cuales son las encargadas de ayudar en el ciclo de desarrollo, con el fin de aumentar la productividad y reduciendo el coste en términos de tiempo y dinero. En el ciclo de desarrollo pueden ayudar en el proceso de diseño del proyecto, en el cálculo de costes, pueden implementar una parte del código, compilación automática y documentación. En el presente trabajo se utilizaráVisual Paradigm para UML.

1.7.3.1.1. Visual Paradigm para UML

Una de las herramientas CASEmás usadas es la suite creada por Visual Paradigm International (VPI). VPIes un proveedor de soluciones informáticas que incluye organizaciones para desarrollar aplicaciones de calidad, rápidas y baratas. Sus soluciones se enfocan en eliminar la complejidad, aumentando así la productividad y disminuyendo el tiempo de desarrollo de las aplicaciones informáticas.

La suiteVisualParadigm está compuesta por productos que facilitan a las organizaciones la visualización y diseño de diagramas, así como su integración con lenguajes de programación y gestores de bases de datos. Es independiente de las plataformas, soportando varios entornos integrados de desarrollo, entre las que se pueden mencionar: Microsoft Visual Studio, Eclipse y Java.

La herramienta CASE Visual Paradigm fue escogida para desarrollo de la aplicación debido a que utiliza UML como lenguaje de modelado, agiliza la creación de los diferentes diagramas definidos en la metodología RUP, se integra con EasyEclipse, se puede utilizar en sistemas operativos GNU/Linux y genera una excelente documentación en varios formatos (jpg, html, pdf, etc.).

1.9.4. Metodología de Desarrollo de Software

RUP (Rational Unified Process) es Proceso Unificado de Rational es más que un simple proceso; es un marco de trabajo genérico que puede especificarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

RUP utiliza el Lenguaje Unificado de Modelado para preparar todos los esquemas de un sistema software. De hecho, UML es una parte esencial de RUP, sus desarrollos fueron paralelos.

Para llevar a cabo esta investigación, se implementará un gran número de funcionalidades con alta complejidad, por lo que es necesario agrupar las funcionalidades similares de tal manera que el desarrollo de la aplicación sea un proceso comprensible y quede bien documentado. Al ser RUP dirigido por casos de uso, su utilización garantiza que se satisfaga esta necesidad, ya que los casos de uso(CU) guían el proceso de desarrollo, pues los modelos que se obtienen como resultado de los diferentes flujos de trabajo, representan su realización.

Igualmente se necesita tener una visión del sistema completo, realizando primeramente los CU arquitectónicamente significativos, de forma que constituyan los cimientos del sistema, que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP es centrado en la arquitectura, característica que se encuentra en total concordancia con esta necesidad.

El sistema se debe desarrollar en varias iteraciones desde las mínimas funcionalidades hasta que se complete el ciclo de vida, reduciendo la posibilidad de que aparezcan riesgos que provoquen una compleja solución cuando esté terminado el producto. Además, se deben especificar las posibles funcionalidades futuras para otros ciclos de desarrollo. Para ello RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los

incrementos, al crecimiento del producto, lo que lo convierte en un proceso iterativo e incremental.

Se espera que el producto final contenga manuales de usuario y buena documentación que asegure una eficiente transferencia tecnológica. RUP es ideal para esto debido a que genera la documentación necesaria para validar los artefactos.

Se considera que la utilización de RUP puede ser de suma importancia para el desarrollo de la aplicación de forma eficiente y con alta productividad debido a que define qué se tiene que hacer, cómo y quién lo hace en cada momento del proceso de desarrollo.

Tambiés es política del departamento de Seguridad Informática del centro de Telemática la utilización de esta metodología de desarrollo para el proceso de implementación de todos los productos de software.

1.9.5. Entorno de Desarrollo Integrado

Una de las herramientas que juegan un papel importante en el desarrollo de soluciones informáticas son los Entornos de Desarrollo Integrado (IDE). Éstos ofrecen facilidades al equipo de desarrollo cuando se implementan las aplicaciones debido a que permite corrección de errores comunes que se comenten a diario.

1.9.5.1. EasyEclipse

El proyecto EasyEclipse tiene como principal propósito proveer una fácil instalación y uso de las distribuciones de Eclipse. Su arquitectura basada en plugins extiende las funcionalidades de la herramienta, por ejemplo, con la adición del soporte para varios lenguajes, como pueden ser: Java, Ruby, PHP, Python o Prolog. Fue creado en el 2003 por un grupo de desarrollo de software llamado nexB dedicados al análisis de código abierto.

Para desarrollar la investigación se decidió utilizar EasyEclipse debido a que actualmente se sitúa a la vanguardia de los entornos de desarrollo para Python existentes por su alto nivel de integración con este lenguaje, completamiento de código, resaltado de sintaxis,

depuración de la ejecución, funcionamiento en varias plataformas, así como diferentes funcionalidades que le facilitan el trabajo al programador como la refactorización del código. Además, posee una excelente integración con SubVersion.[19]

1.9.6. Framework de JavaScript

1.9.6.1. JQuery[20]

jQuery se está convirtiendo rápidamente en una herramienta que todo desarrollador de interfaces web debería de conocer. JQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos; al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Esta herramienta consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX.

La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX, aunque pose otras características como:

- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.

- Animaciones personalizadas.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones como trim() (elimina los espacios en blanco del principio y final de una cadena de caracteres), etc.

1.10. Conclusiones parciales

En el capítulo abordado se ofrecen consideraciones en torno a un sistema, el cual, una vez aplicado, mejoraría considerablemente el envío y recepción de correos del Ministerio de la Informática y las Comunicaciones. Se realizó también un análisis de conceptos y definiciones relacionadas con el estudio del estado del arte de herramientas de hardware y software que permiten lapasarela de correos como Lock-Keeper, e-Gap, etc., pero tienen un precio extremadamente alto y no realizan todas las funcionalidades que se requieren en el MIC.

Se seleccionó la metodología de desarrollo de software RUP, para el modelado de negocio se utilizará BPMN,UML como lenguaje de modelado, fue definido Python comolenguaje de programación a utilizar soportado por un IDE como Eclipse,Visual Paradigm como herramienta CASE.De igual forma, la propuesta de desarrollo del sistema demuestra un marcado uso en la utilización de herramientas de reconocida popularidad y bajo patentes libres, requisitos indispensables en el sistema que se propone.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción

En este capítulo se trataráproblema y situación problemática así como la propuesta del sistema, características, seguridad, soporte, rendimiento, requerimientos funcionales y no funcionales, todo esto para la automatización del envío y recepción de los correos en el MIC. Se definirán los casos de uso y se especificarán los requisitos del software, del mismo modo incluye una síntesis de las dependencias y relaciones con otros software, usabilidad, portabilidad confidencialidad, ayuda y documentación en línea.

2.2. Problema y situación problemática

En el Ministerio de Informática y Comunicaciones (MIC) se implementa una pasarela de correos que aísla físicamente las redes interna y externa. La pasarela requiere de personas que manualmente mueven los mensajes de una subred a otra almacenando los archivos de los mensajes en memorias externas tipo Flash con puerto USB.

Este procesamiento se torna engorroso y muchas veces problemático ya que se realiza una manipulación manual de la información donde se introduce la posibilidad de errores humanos y abriendo una brecha de seguridad. También se emplean múltiples recursos en el proceso así como una engorrosa implementación del sistema pues se utilizan muchas reglas en el servicio de correo. También existe el problema de la despersonalización del remitente del mensaje, incomodidad tanto para usuarios internos como externos, además del retraso de la llegada de los mensajes a su destino.

Teniendo en cuenta todos estos problemas se desarrolló una versión 1.0 que resuelve en gran medida todas estas problemáticas, pero no implementa ni recuperación ante errores o fallos, ni ningún tipo de seguridad para el sistema.

2.3. Objeto de automatización

Las ideas expresadas anteriormente permiten constatar la necesidad de desarrollar un sistema para aportar a la automatización del envío y recepción de los correos en el Ministerio de la Informática y las Comunicaciones, para lograr rapidez en el proceso, además de hacerlo transparente al usuario y con los requerimientos mínimos de seguridad para este sistema.

2.4. Información que se maneja

Se trabaja con la información de los servidores de correo encriptando y decriptando la misma así como el intercambio de los datos entre los mismos a través de las memorias flash.

2.5. Propuesta de sistema

La solución parte de mantener lo logrado en el Organismo Central del MIC, disminuyendo los recursos humanos, materiales y el tiempo de entrega de los mensajes.

La solución se compone de un dispositivo de hardware y una aplicación de software. El hardware es un dispositivo electrónico de conexión USB con almacenamiento de tecnología Flash conectado a dos servidores de gama baja. El software es una aplicación que gestiona la entrada y salida de correos en un servidor de transporte de correos MTA y controla el dispositivo de hardware. El sistema puede ser conectado y desconectado de la infraestructura tecnológica de servicios sin afectar el flujo de información.

2.6. Modelo de negocio

Business Process Modeling Notation o BPMN (En español Notación para el Modelado de Procesos de Negocio), es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (workflow). BPMN fue inicialmente

desarrollada por la organización Business Process Management Initiative (BPMI), y es actualmente mantenida por el OMG (Object Management Group), luego de la fusión de las dos organizaciones en el año 2005. El principal objetivo de BPMN es proveer una notación estándar que sea fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio (stakeholders). Entre estos interesados están los analistas de negocio (quienes definen y redefinen los procesos), los desarrolladores técnicos (responsables de implementar los procesos) y los gerentes y administradores del negocio (quienes monitorean y gestionan los procesos).

En síntesis BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación.[21]

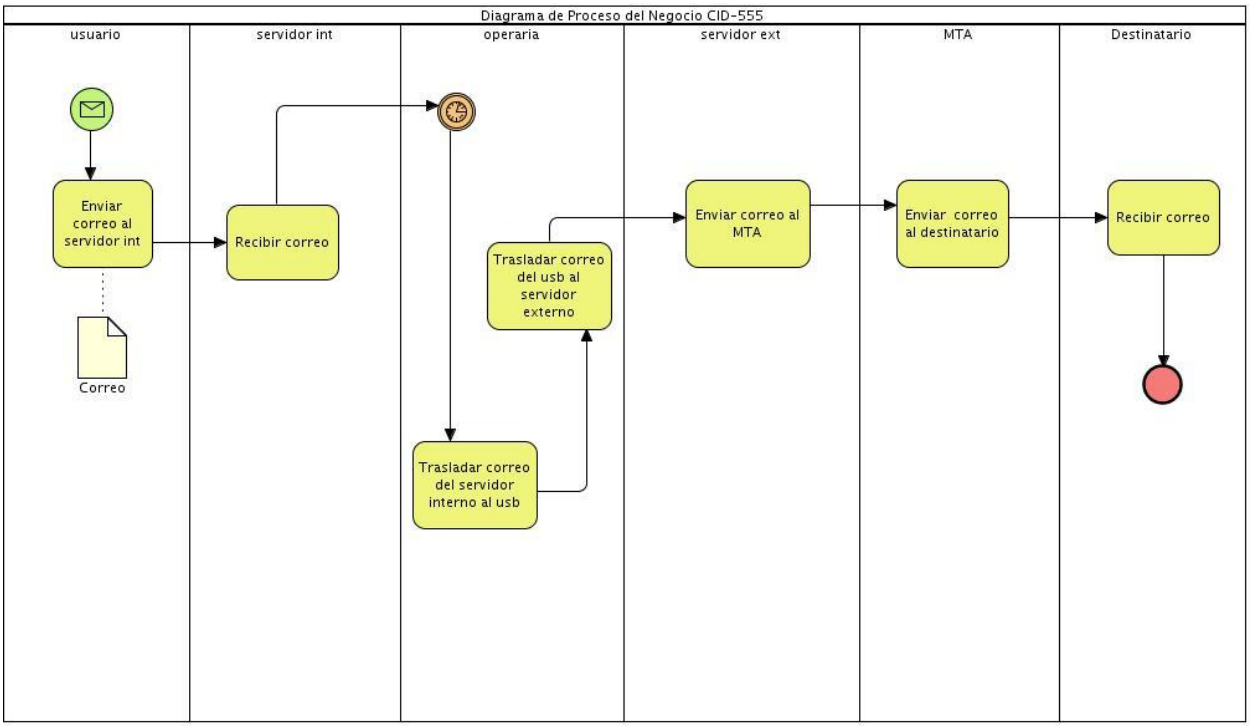


Figura 2. Diagrama de Procesos de Negocio.

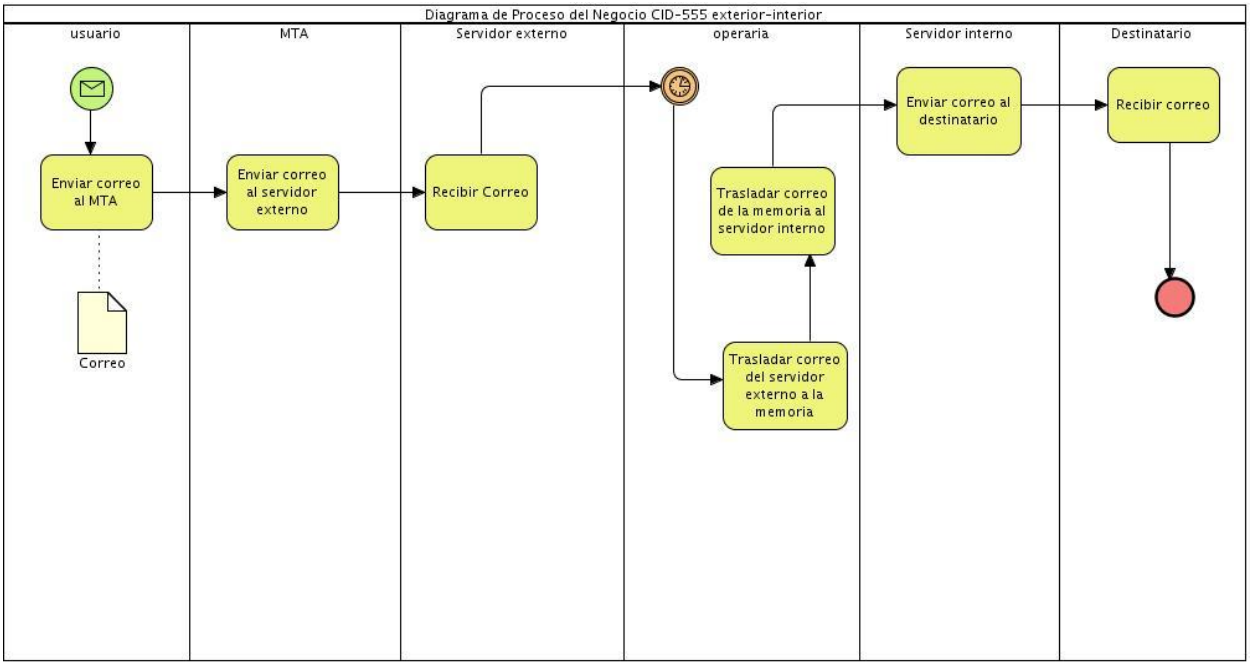


Figura 3. Diagrama de Procesos de Negocio proceso inverso.

2.7. Especificación de los requisitos de software

Después de la realización del modelo de negocio se procede a ejecutar el proceso de captura de requisitos del sistema, para el cual se visitó en varias ocasiones el MIC. Los requisitos son en sí, lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

2.7.1. Requerimientos Funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, es decir especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. No alteran la funcionalidad del producto, lo cual quiere decir que éstos se mantienen invariables sin importarle con quépropiedades o cualidades se relacionen. Los requisitos funcionales que debe cumplir el sistema son:

R1. Crear carpetas en las memorias flash y en la ubicación local de la pc si no están previamente creadas

R2.Copiarlos correos en las memorias flash.

R3. Detener el sistema

R4.Mostrar Logs

R5.Encriptar la información.

R6. Decriptar la información

R7.Aplicar suma de verificación

R8.Empaquetar mensajes

R9.Desempaquetar mensajes

R10.Chequear capacidad dela memoria flash.

R11.Crear logs.

R12.Mostrarestado de las colas.

R13.Alertar eventos inesperados.

R14.Autenticar usuarios.

R15.Establecer tiempo de espera del ciclo de iteraciones.

R16.Establecer tiempo de espera de intercambio de memorias flash.

2.7.2. Requerimientos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, es decir, son las características que hacen al producto atractivo, usable y confiable. Normalmente, están vinculados a los requerimientos funcionales, o sea, una vez que se conozca lo que el sistema debe hacer, se puede determinar cómo ha de comportarse, las cualidades que debe tener o cuán rápido o grande debe ser.

2.7.2.1. Apariencia o interfaz externa

- El sistema informático contará con un diseño de interfaz sencillo, agradable, sugerente e intuitivo, de fácil entendimiento. Las páginas de la aplicación estarán poco cargadas, sólo contendrá la información requerida para el usuario.

2.7.2.2. Usabilidad

- La aplicación podrá ser manejada por cualquier usuario con conocimientos básicos del tema. La interfaz y los mensajes para interactuar con el usuario, así como los mensajes de error, serán en el lenguaje español. Los mensajes de error deben ser lo suficientemente informativos para dar a conocer la severidad del error. Los elementos de navegación deben ser orientados al usuario.

2.7.2.3. Soporte

- El sistema debe contar con un manual de usuario que permita al usuario un mayor uso de sus funcionalidades y una mayor experiencia en el trabajo con la aplicación.
- El sistema debe contar con un manual de instalación el cual podrá ser consultado para los elementos necesarios en el despliegue del sistema (configuraciones, dependencias y elementos a instalar).

2.7.2.4. Seguridad

- El sistema solo debe permitir interactuar con la aplicación a los usuarios que tengan acceso según el rol definido.
- La contraseña de los usuarios del sistema será encriptada para que no pueda ser leída en el fichero donde será guardada.
- El sistema solo interactuará con la memoria flash destinada para esta función, previendo de esta forma la suplantación de la memoria.

- La información de las memorias flash estará encriptada, previendo de esta forma el robo de información.
- A los paquetes recibidos en cada iteración se le realiza una suma de verificación para comprobar la integridad de la información.

2.7.2.5. Legales

- El sistema y la documentación del mismo, pertenecen a la Universidad de las Ciencias Informáticas.
- El sistema es desarrollado con herramientas libres lo que permite independencia tecnológica.

2.7.2.6. Confiabilidad

El sistema debe ser capaz de recuperarse ante la ocurrencia de un fallo y alertar al personal encargado de la administración del mismo, así como proteger la información y el contenido de los dispositivos de almacenamiento.

2.7.2.7. Software

- El servidor Web es Apache 2.22o superior con soporte para SSL.
- El lenguaje de programación del sistema es Python en su versión 2.7.
- Los lenguajes de programación de la interfaz son PHP en su versión 5.3.x, y Jquery 1.7.2.
- Las computadoras clientes deben tener instalado un navegador web Mozilla Firefox versión 10.02 o superior
- Se requiere la instalación de un Sistema Operativo GNU/Linux, se recomienda Ubuntu 11.04 osuperior.

2.7.2.8. Hardware

Se requieren 2 servidores que tengan como mínimo 1 GB de memoria RAM, procesador Pentium IV o superior y 100 GB de disco duro disponibles.

2.8. Definición de los casos de uso

2.8.1. Diagrama de casos de uso del sistema

El diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores. Los casos de uso son artefactos que describen, mediante acciones y reacciones, el comportamiento que tendrá el sistema desde el punto de vista del usuario.

Establece un acuerdo entre los clientes y los desarrolladores en cuanto a las condiciones y posibilidades que debe cumplir el sistema.

En el diagrama se representan 7 casos de uso determinadosdespués de haber identificado los requisitos del sistema,estos son:

- Autenticar usuario.
- Administrar sistema.
- Mostrar sucesos del sistema.
- Mostrar las colas del MTA.
- Configurar sistema.
- Gestionar usuario.

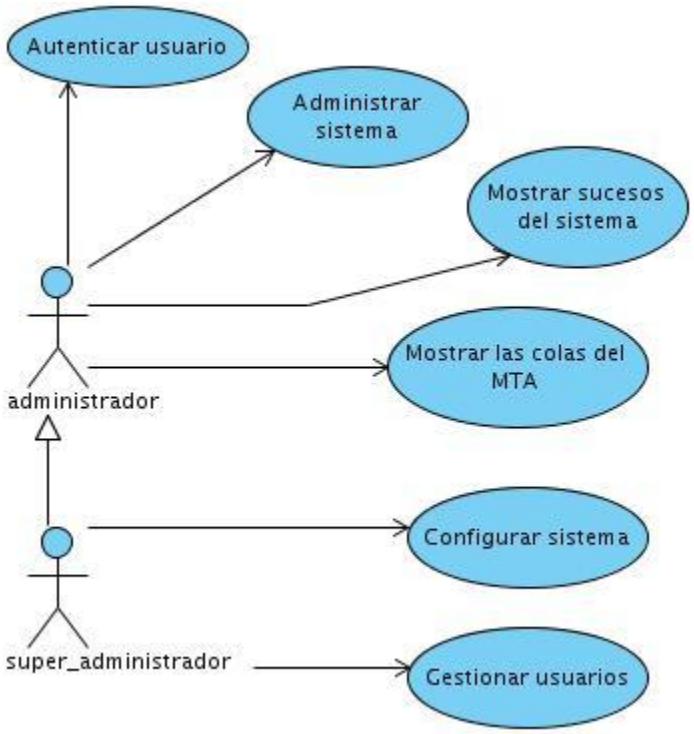


Figura 4. Diagrama de Casos de Uso del sistema.

Actor	Descripción
Administrador	Representa un rol del sistema, debido a que el mismo está destinado en su mayoría a usuarios con el cargo de administrador de red.
Super_administrador	Representa un rol del sistema que además de poder realizar las mismas acciones del actor administrador es el único que puede gestionar los usuarios y establecer las configuraciones el sistema.

Tabla 1 : Descripción de las responsabilidades de cada actor.

2.8.2. Descripción de los casos de uso del sistema

La descripción de los casos de uso del sistema se realiza con el objetivo de entender la funcionalidad asociada a cada uno de ellos. Ésta debe ser elaborada de forma breve o extendida para lograr una mejor comprensión sobre lo que debe realizar el sistema y ayudar a un mejor entendimiento. Expresa de forma clara y precisa las acciones que se realizan durante la interacción entre el actor y el sistema, describe el flujo de actividades que realiza el actor al hacer uso del sistema y las respuestas que emite el mismo.La descripción de los casos de uso se pueden encontrar en los anexos del número 1 al número 6.

2.9. Conclusiones parciales

En el capítulo abordado se ha efectuado un análisis ingenieril de las características del sistema que se presenta, lo cual contribuye a la documentación y correcta implementación de la aplicación. Se realizó la definición de los requisitos, tanto funcionales, como no funcionales y la representación visual de las funcionalidades a través del Diagrama de Casos de Uso del Sistema. En este orden, además del cumplimiento de los requisitos expresados, es posible el comienzo de la construcción del sistema que se presenta.

CAPÍTULO 3: DISEÑO DEL SISTEMA

3.1. Introducción

El presente capítulo está enfocado en el diseño e implementación de la aplicación mediante la metodología RUP.Determinados los requerimientos funcionales y casos de uso del sistema, entrada fundamental para eldiseño,seprocedeaanalizarsiesposibledarunasoluciónquesatisfagalosrequisitossignificativos de la arquitectura. El diseño contribuye a una arquitectura sólida y estable quesoporte las funcionalidades del sistema correlacionadas a requerimientos funcionales.

Teniendo en cuenta la propuesta de solución descrita en el capítulo anterior, es necesario definir cómose desarrollará el sistema. De esta operación se encarga el flujo de trabajo de diseño.

3.2. Arquitectura del sistema

El diseño del software “Software para la Interconexión de Redes Aisladas. Módulo correo electrónico V2.0” está dividido en front-end y back-end.

3.2.1. Front-end

Es la parte del software que interactúa con el usuario final, es decirla interfaz de administración web.Para el diseño de la propuesta de solución, se utilizael patrón arquitectónico Modelo - Vista- Controlador, con lo cual se separan los datos de la aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

El siguiente diagrama representael patrón arquitectónico: Modelo - Vista– Controlador aplicado.

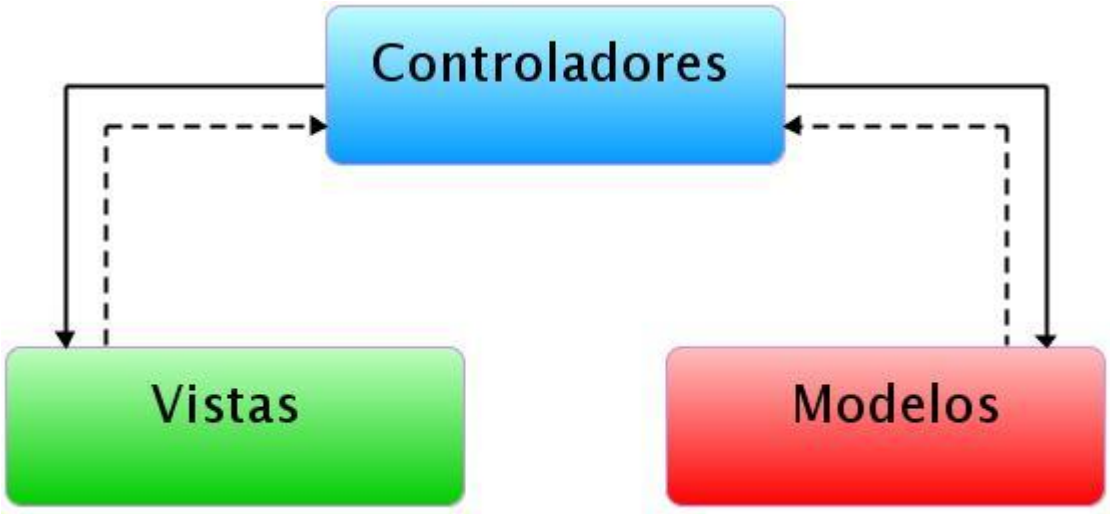


Figura 5. Diagrama de la arquitectura front-end.

Detalles de los componentes del diagrama:

Controladores: Es donde se encuentran las clases que atienden, responden y enrutan las acciones solicitadas por el usuario, constituyen además todo lo referente a verificacion de entrada de datos y cómo se presentará la información en las vistas.

Vistas: Es donde se encuentran los archivos que permite la interacción entre el usuario y el sistema, para finalmente retornar una respuesta.

Modelos: Es donde se encuentran las clases que procesan la logica del negocio, éstos interactúan con la Base de Datos y con el Back-end.

3.2.2. Back-end

Es la parte del software que procesa la entrada desde el front-end. Para el diseño de la propuesta de solución del mismo, se utiliza el patrón arquitectónico: N-Capas, en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario. Las capas puedan ser manejadas e implementadas de forma independiente, y

poseerán responsabilidades específicas que no dependan del funcionamiento de las otras o al menos que su dependencia sea mínima. Este aspecto constituye una ventaja considerable pues proporciona una amplia reutilización de las clases implementadas al hacer abstracciones de las distintas funcionalidades o responsabilidades del sistema agrupándolas en capas.

El diagrama representa la distribución de los componentes del sistema en N-Capas.

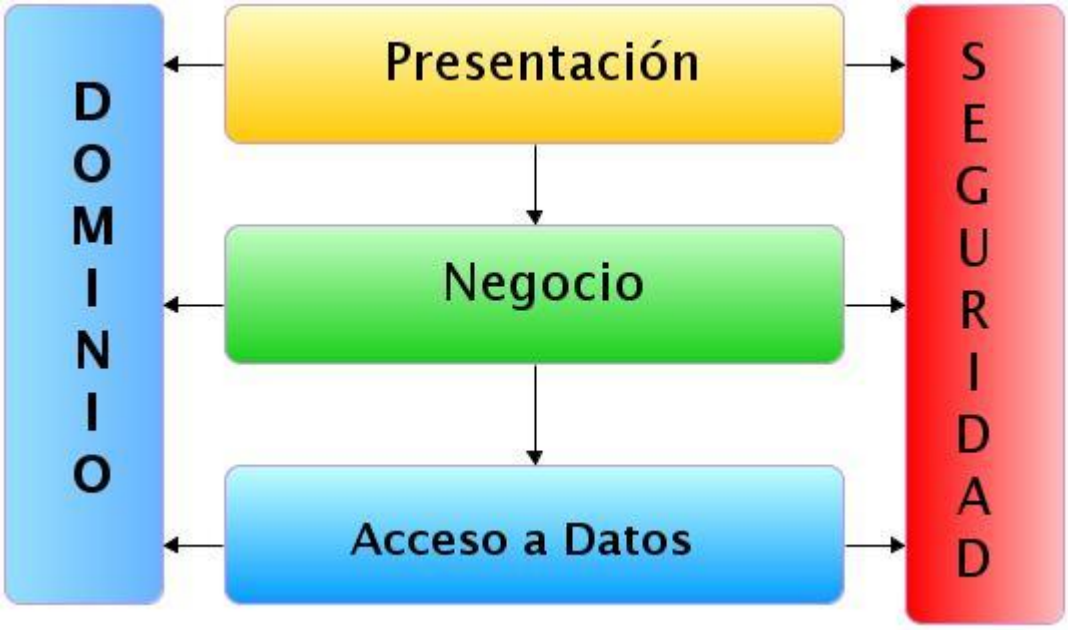


Figura 6. Diagrama de la arquitectura back-end.

Detalles de los componentes del diagrama:

Dominio: Es la capa que brinda la posibilidad del manejo de los objetos del dominio del sistema, de manera tal que todas las capas del sistema puedan acceder a las funcionalidades que esta capa brinda.

Interfaz: Capa que permite la comunicación entre la interfaz web y la aplicación de control del proceso de intercambio de mensajes.

Negocio: Es la capa encargada de iniciar y manejar todo el proceso del sistema, es la que realiza los cálculos de procesamiento.

Acceso a Datos: Capa encargada de interactuar con la Base de Datos.

Seguridad: Es la capa que contiene todos los elementos necesarios para la implementación de la seguridad del sistema.

3.3. Patrones de diseño utilizados

Es una solución reutilizable de problemasrecurrentes que ocurren durante eldesarrollo del software". Ayudan a entender las soluciones delproblema con un vocabulario igual lo quepermite un mejor entendimiento.

Para el diseño de la aplicación se hizo uso de los Patrones Generales de Software para Asignar Responsabilidades (GRASP). Los patrones de GRASP utilizados fueron:

- Alta cohesión: Se aplica en la mayoría las clases del diseño, ya que en cada una solo se implementan las funcionalidades que le corresponden.
- Bajo acoplamiento: Ya que cada clase se comunica con un número relativamente pequeño de clases.
- Creador:Las clases que tienen la responsabilidad de crear objetos contienen toda la infromación necesaria para construir los mismos.
- Experto: Se mantiene el encapsulamiento, los objetos utilizan su propia información para llevar a cabo sus tareas. Se distribuye el comportamiento entre las clases que contienen la información requerida. Son más fáciles de entender y mantener.

Se utilizaron también patrones del Gang of Four (GoF):

Creacionales

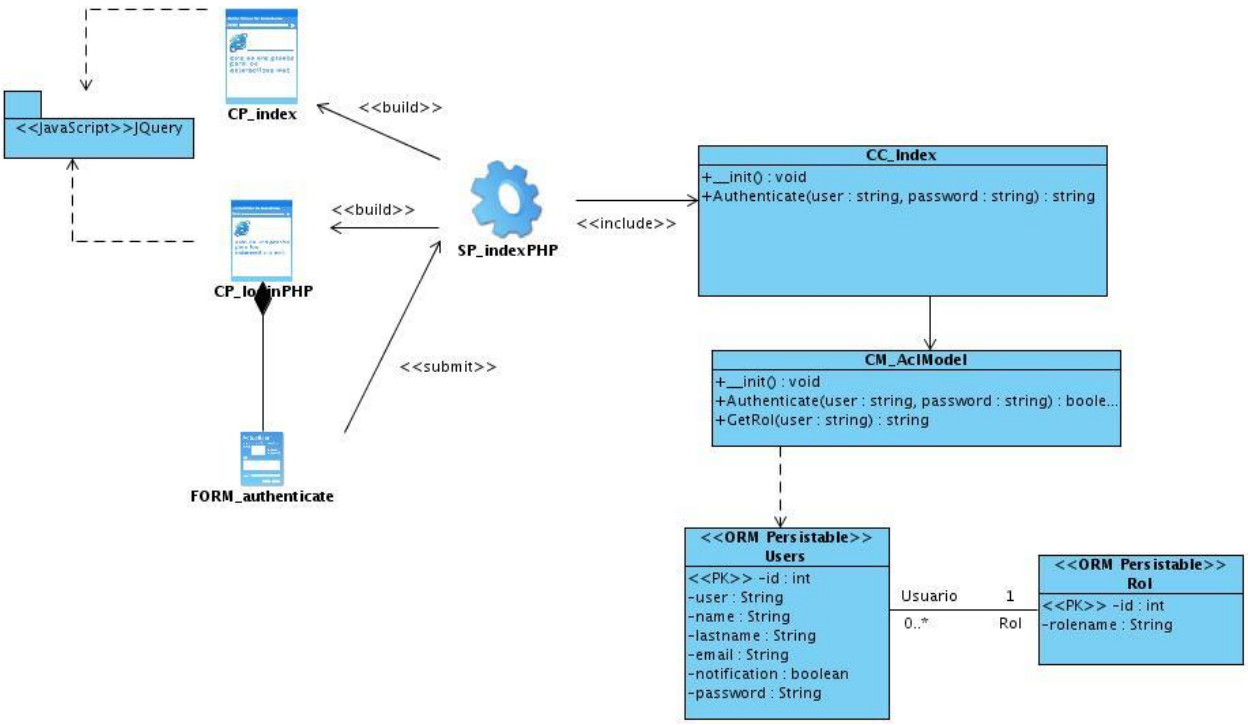
- Singleton: Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. En el caso de la

implementación del backend se utiliza por ejemplo en la clase *System* y en la clase *Configuration*, de las cuales se necesita una instancia única de las mismas.

3.4. Diagrama de clases del diseño

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y las relaciones entre ellas; muestran lo que el sistema puede hacer y cómo puede ser construido. Cuando se crea un diagrama de clases, se modela una parte de los elementos y relaciones que configuran la vista de diseño del sistema

Se presentan los diagramas de clases del Diseño (*DCD*) del Sistema:



.

Figura 7. DCD Autenticar usuario.

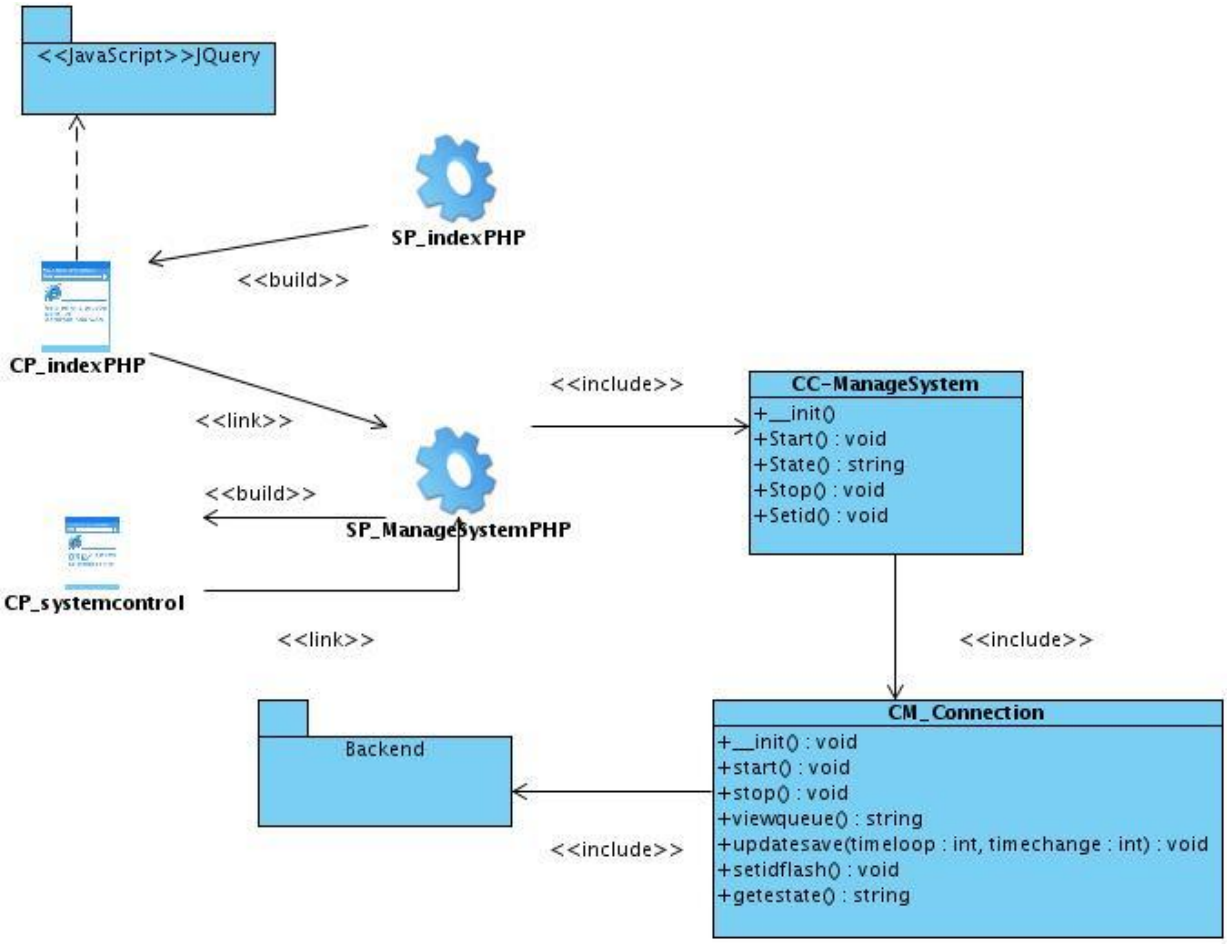


Figura 8. DCD Administrar sistema.

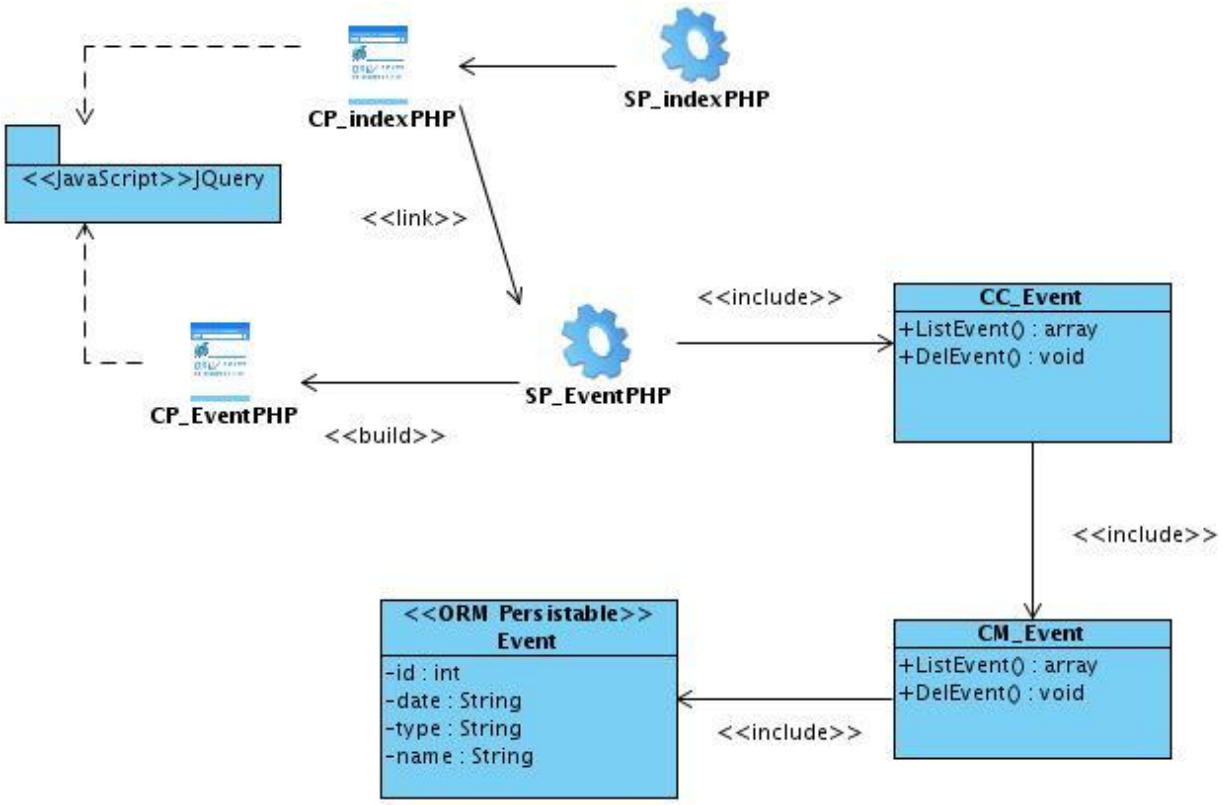


Figura 9. DCD Mostrar sucesos del sistema.

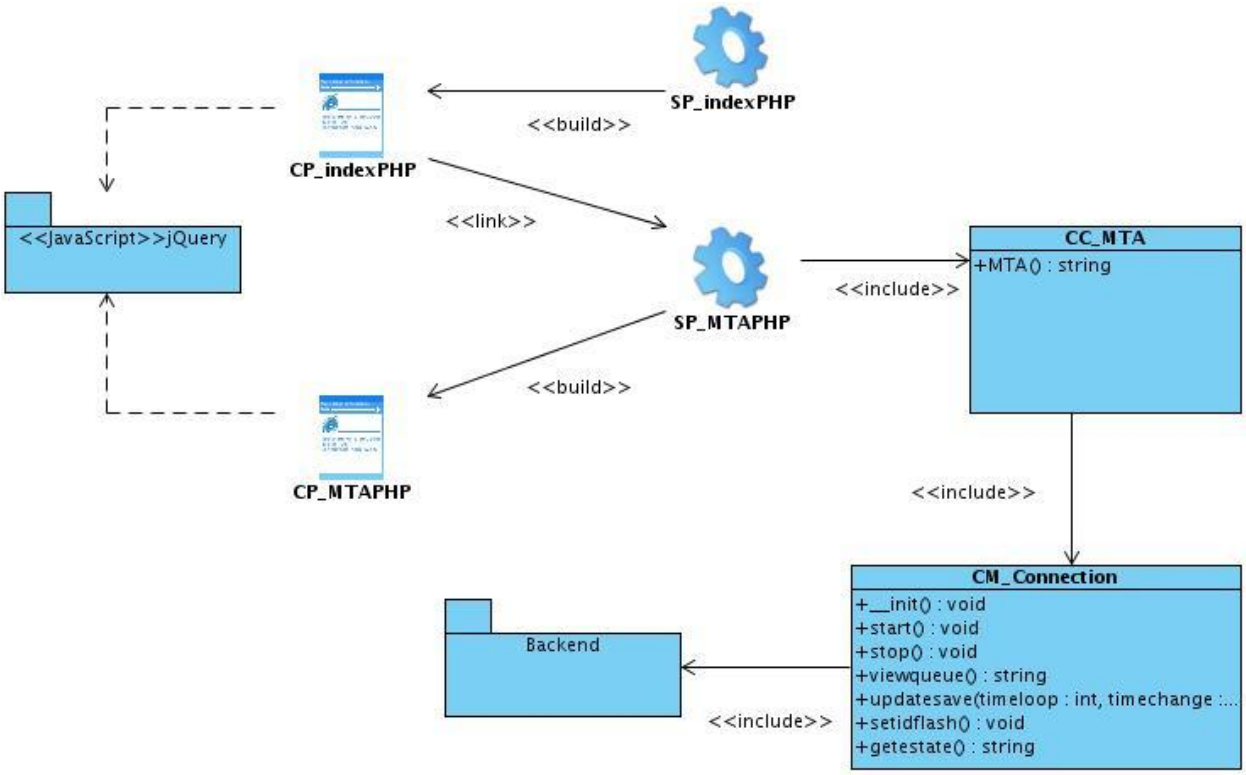


Figura 10 . DCD Mostar lascolas del MTA.

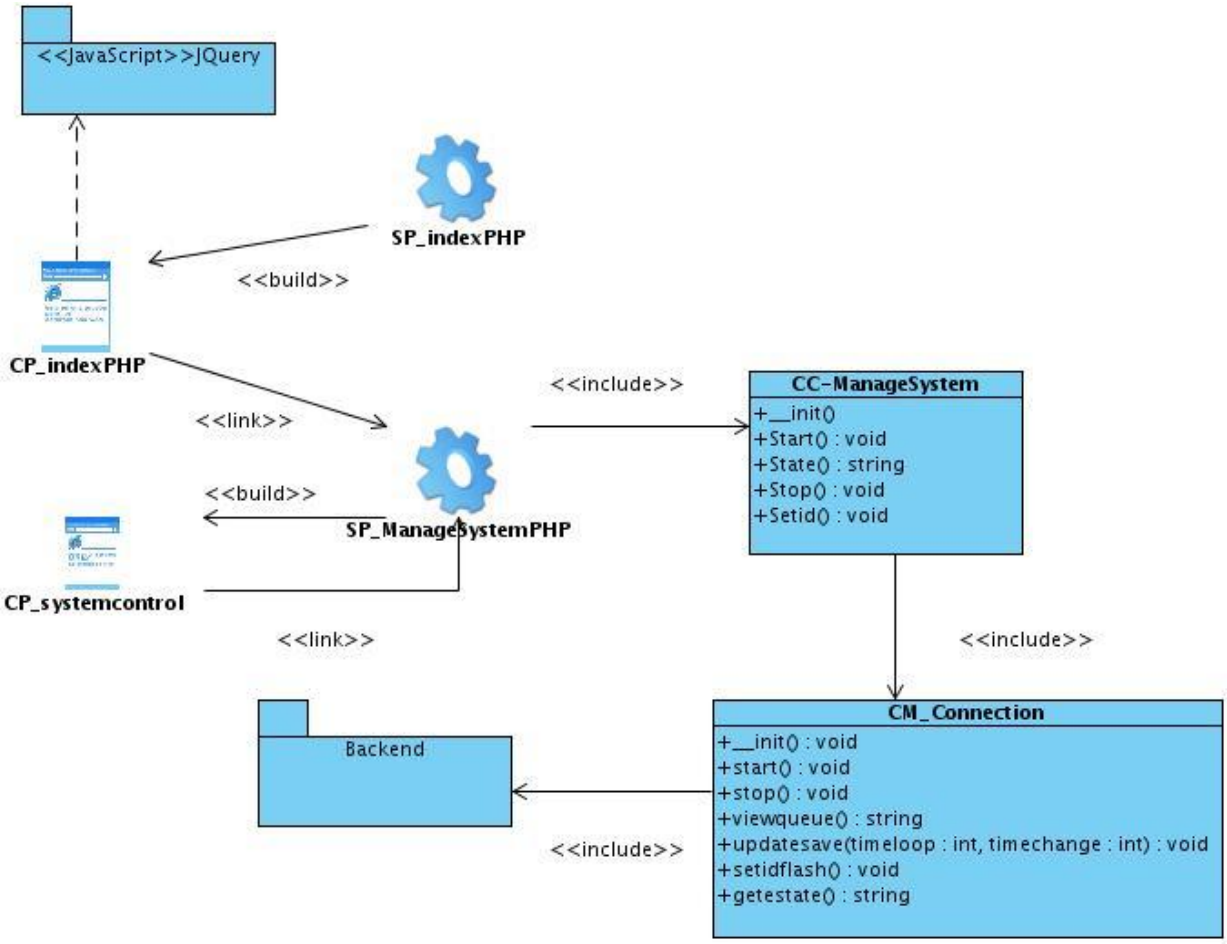


Figura 11. DCD Administrar sistema.

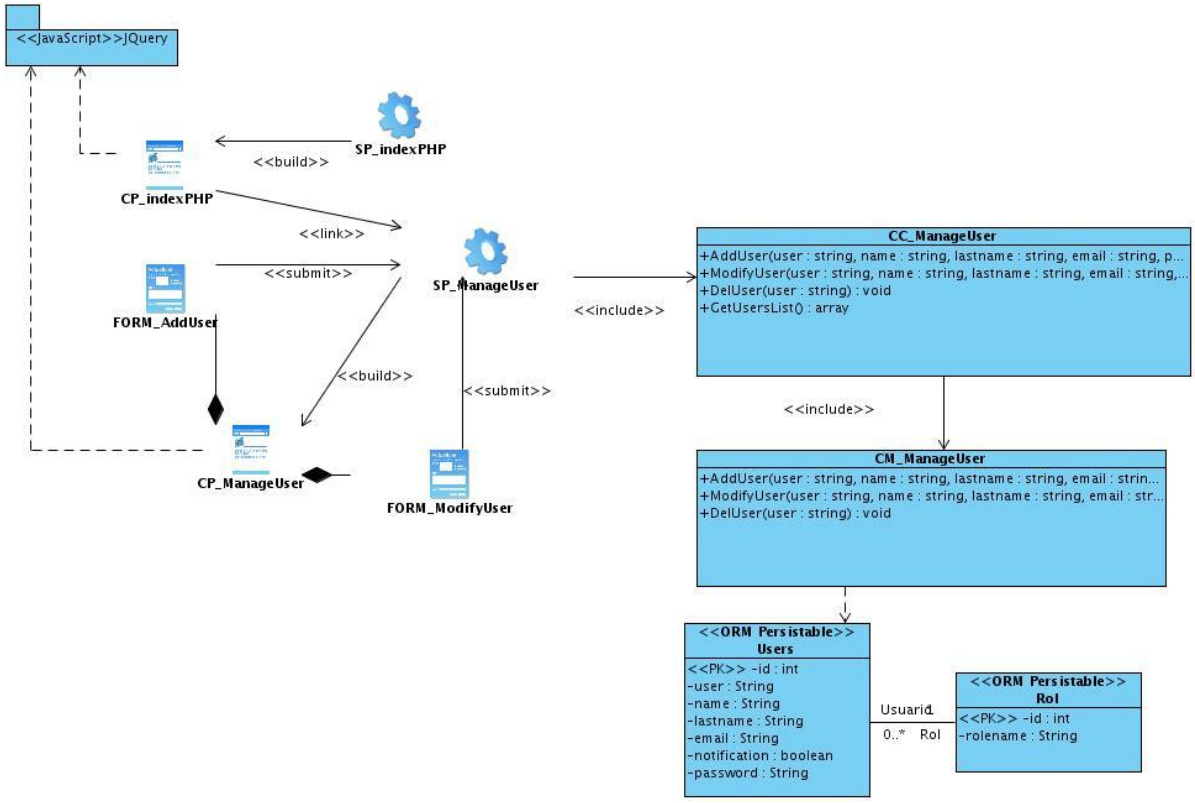


Figura 12. DCD Gestionar usuario.

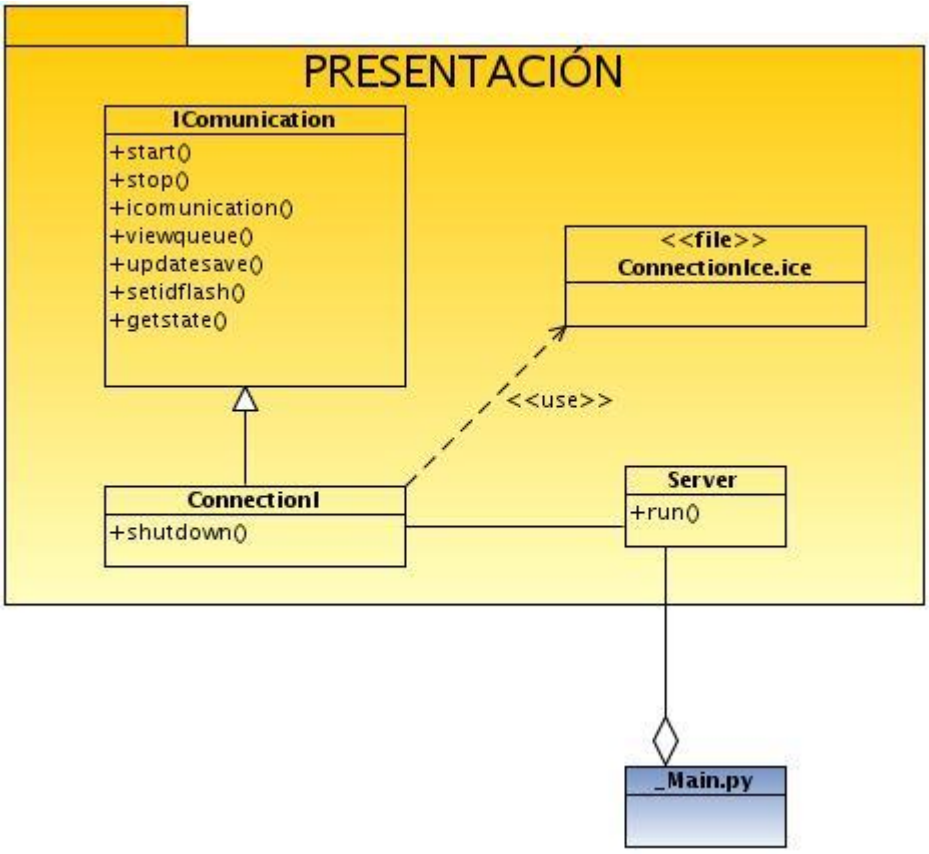


Figura 13.DCD, Capa de Interfaz.

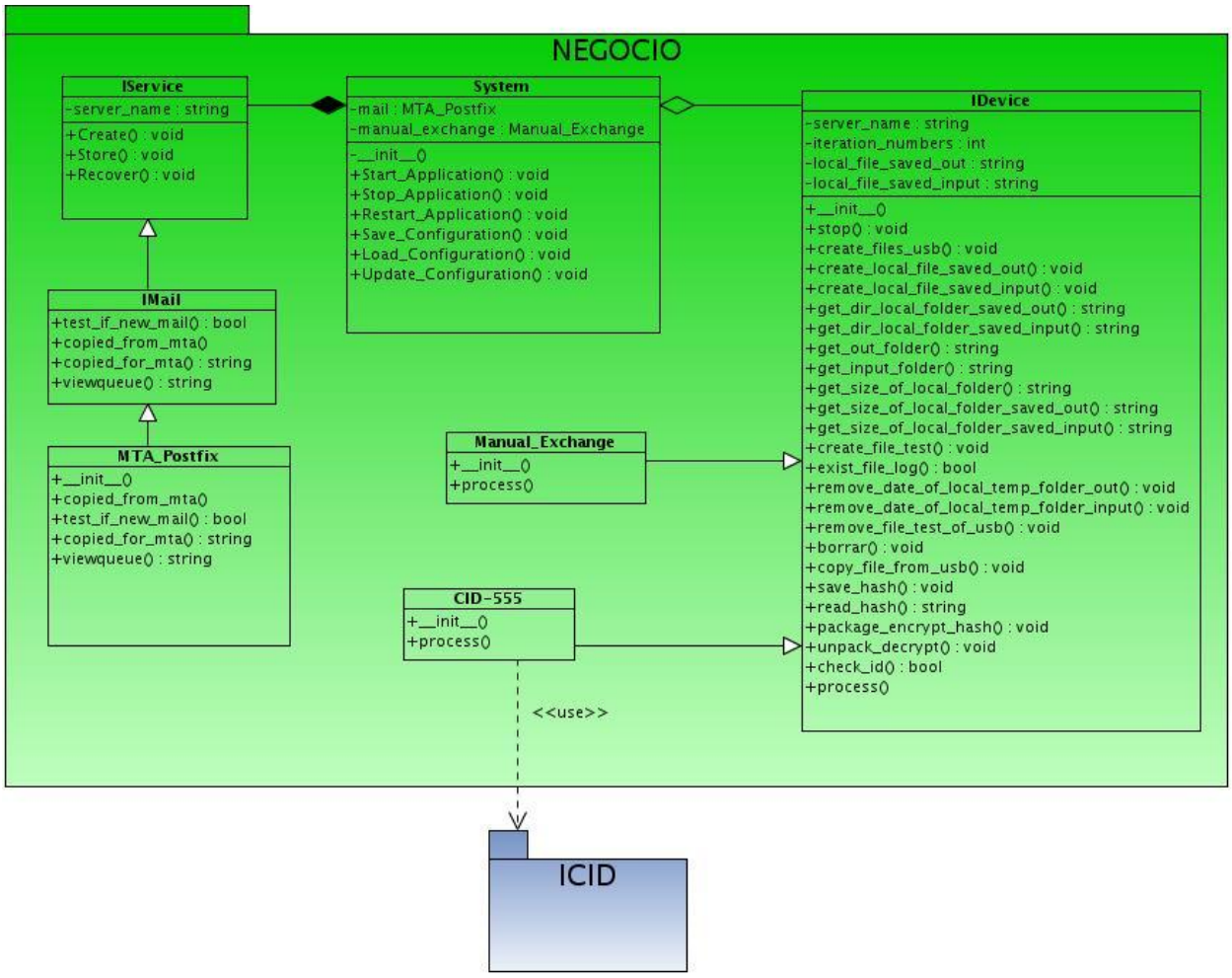


Figura 14.DCD, Capa de Negocio.

Figura 15.DCD, Capa de Acceso a Datos.

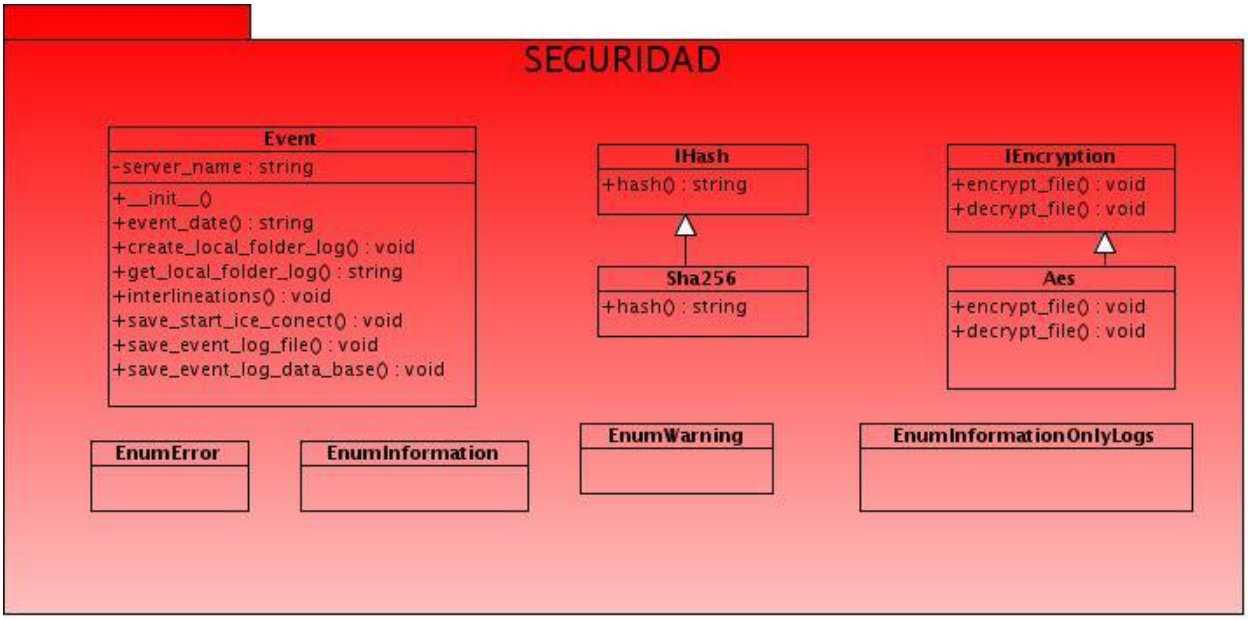


Figura 16.DCD, Capa de Seguridad.

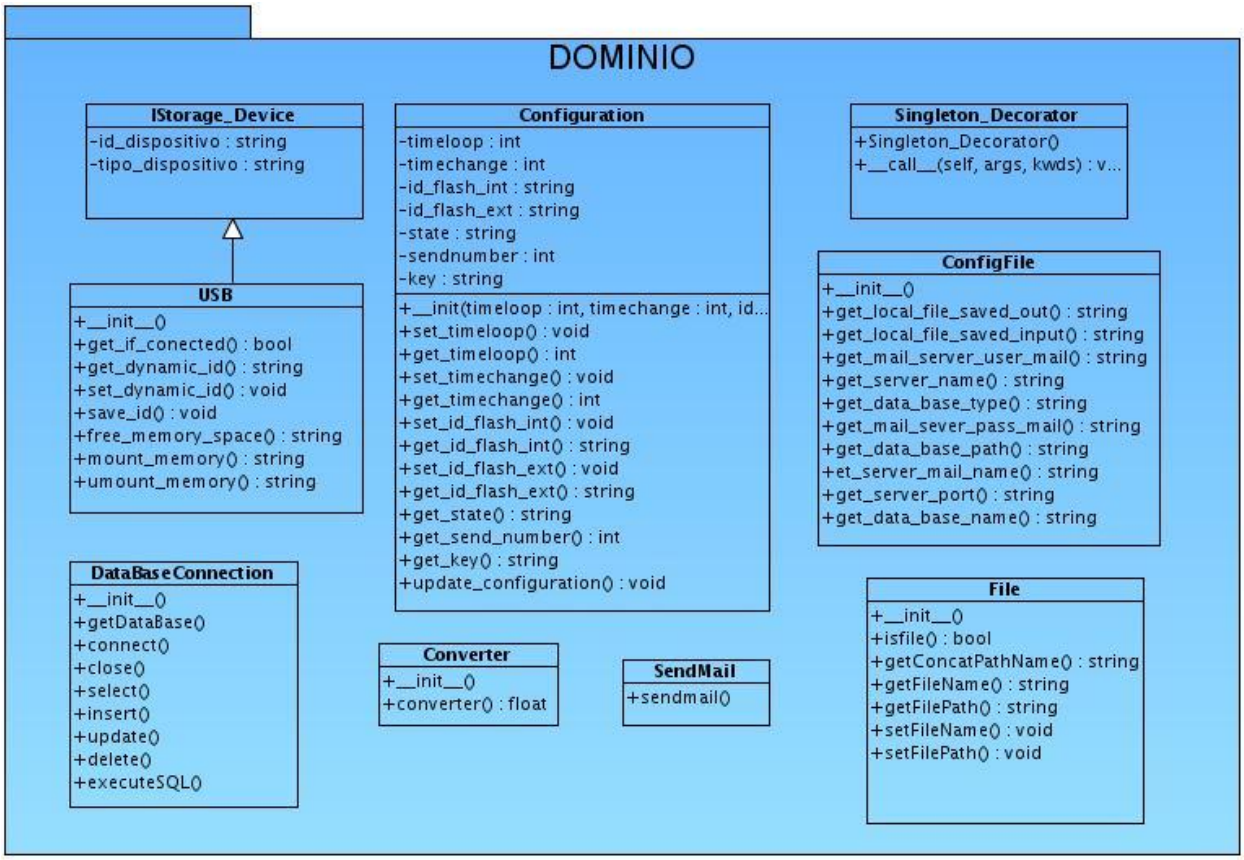


Figura 17.DCD, Capa de Dominio.

3.5. Diagramas de secuencia

En el diseño es preferible representar la secuencia de las acciones de un caso de uso, con diagramas de secuencia, ya que el centro de atención principal es encontrar secuencias de interacciones detalladas y ordenadas en el tiempo.

Se muestran los diagramas de secuencia (DS) del diseño para el caso de uso (CU) Administrar sistema.

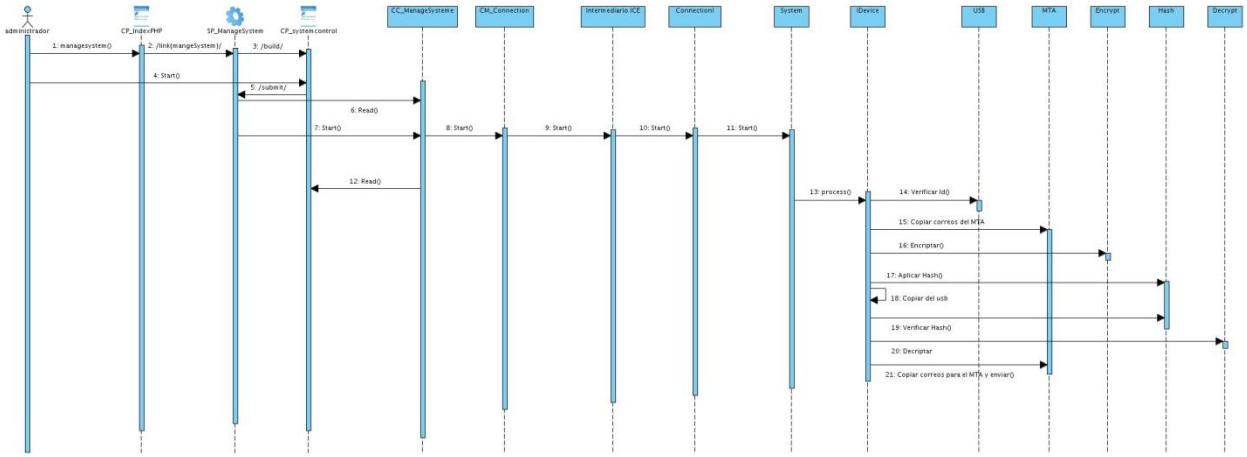


Figura 18. DS CU Administrar sistema. Sección Iniciar.

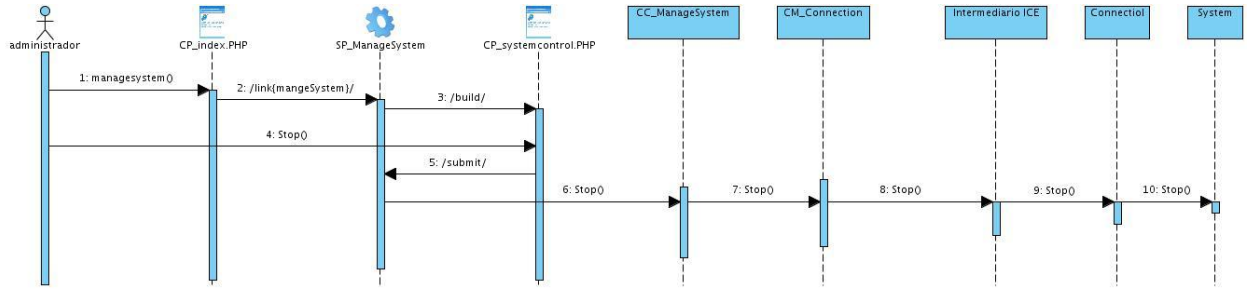


Figura 19. DS CU Administrar sistema. Sección Detener.

3.6. Diseño de la base de datos

El primer paso para la construcción de una base de datos es definir su estructura, de forma tal que permita un adecuado mecanismo para almacenar los datos y posteriormente recuperarlos. Para lograr un buen diseño de la base de datos es necesario seguir un conjunto de pasos que comienzan con definir los diagramas correspondientes de clases persistentes yel modelo entidad-relación.

A continuación se muestra el diagrama de clases persistentes y el modelo entidad-relación de la misma.

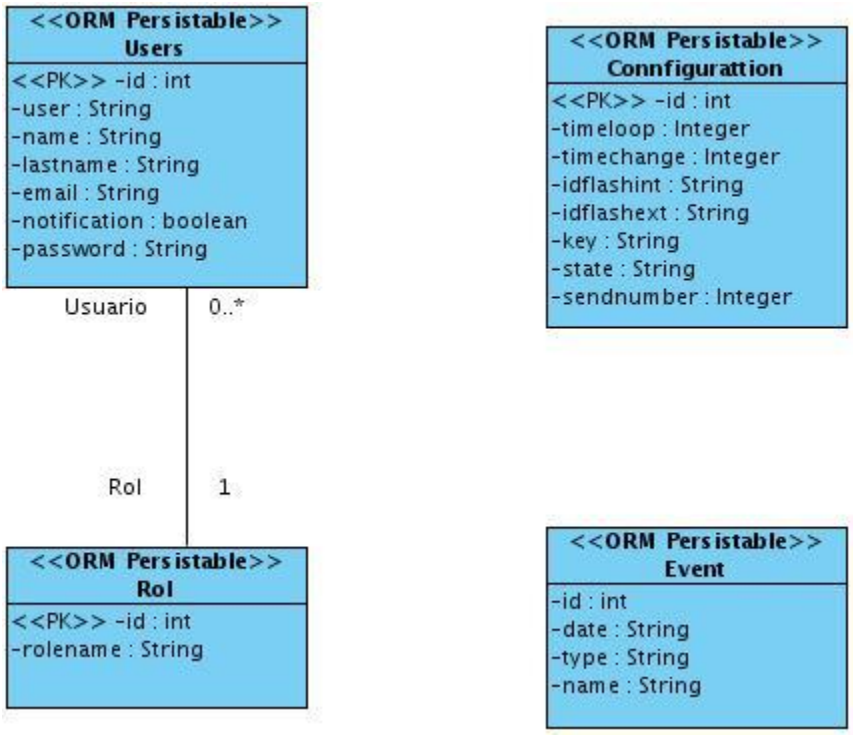


Figura 20. Diagrama de clases persistentes

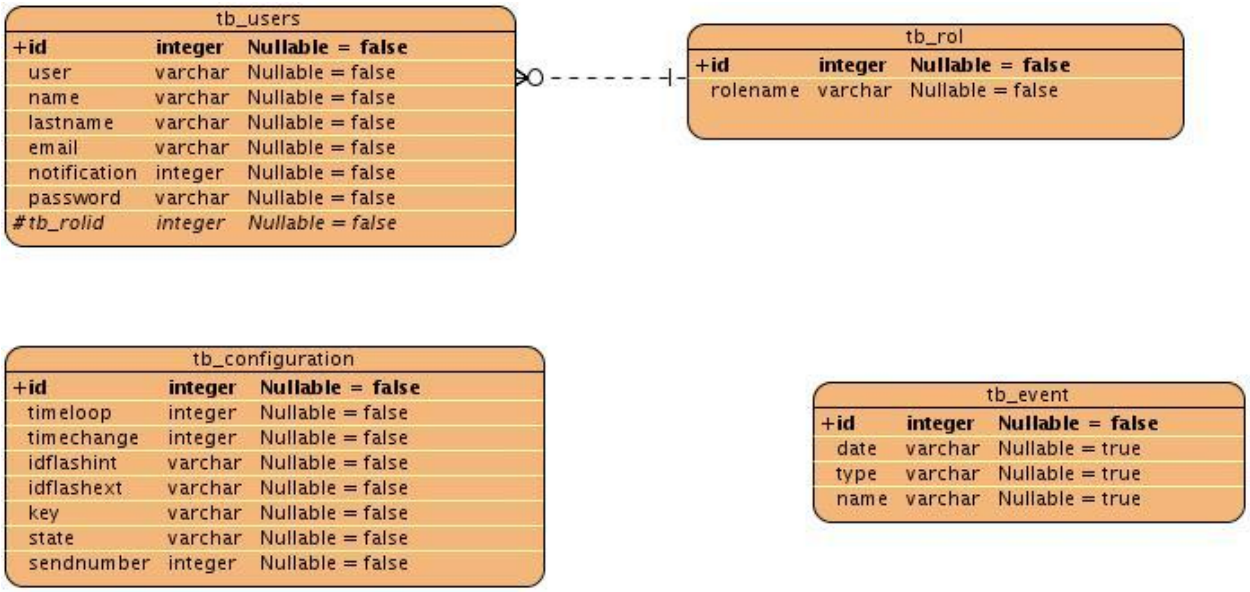


Figura 21. Modelo entidad-relación

3.7. Conclusiones parciales

En este capítulo se ha presentado un estudio relacionado con el diseño de la aplicación que sirve de base fundamental en la implementación del sistema. El diseño de software materializó los requerimientos del cliente. Se ha logrado modelar todos los procesos que han sido objeto de estudio durante el transcurso de la investigación, lo que proporciona una idea completa de lo que realmente es el software que se presenta,permitiendo describir todos los aspectos del futuro sistema. Los diagramas y especificaciones de diseño que se proponen constituyen una guía que puede ser fácilmente comprendida por los desarrolladores con el objetivo de implementar la aplicación que se ha diseñado.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1. Introducción

En la fase de implementación del sistema es donde ya se pueden ver concretamente los componentes y subsistemas en un modelo de implementación. En este capítulo ya se vislumbra un sistema mucho más íntegro en términos de componentes, se lleva a cabo la implementación de las clases y las relaciones entre ellas usando para ello varios lenguajes de programación.

4.2. Generalidades de la implementación

En la fase de construcción se implementa el sistema, se crea el código adecuado al resultado de la fase de diseño por lo que sigue los patrones y la arquitectura escogida. El modelo de implementación permite planificar las integraciones de sistemas necesarias en cada iteración, distribuye el sistema al asignar componentes ejecutables a nodos en el diagrama de despliegue, implementa las clases y subsistemas encontrados durante el diseño y posibilita probar los componentes individualmente para después integrarlos.

4.3. Diagrama de despliegue

Los diagramas de despliegues muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional que generalmente tiene algo de memoria y a menudo, capacidad de procesamiento. Los nodos se utilizan para modelar la topología del hardware sobre el que se ejecuta el sistema. Representa típicamente un procesador o un dispositivo sobre el que se pueden desplegar

los componentes. En la figura se muestra el modelo de despliegue para el sistema planteado:

Figura 29 .Diagrama de despliegue

4.4. Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas pueden ser simples archivos, paquetes, bibliotecas cargadasdinámicamente.

Acontinuaciónsepresentanlosdiagramasdecomponentes:

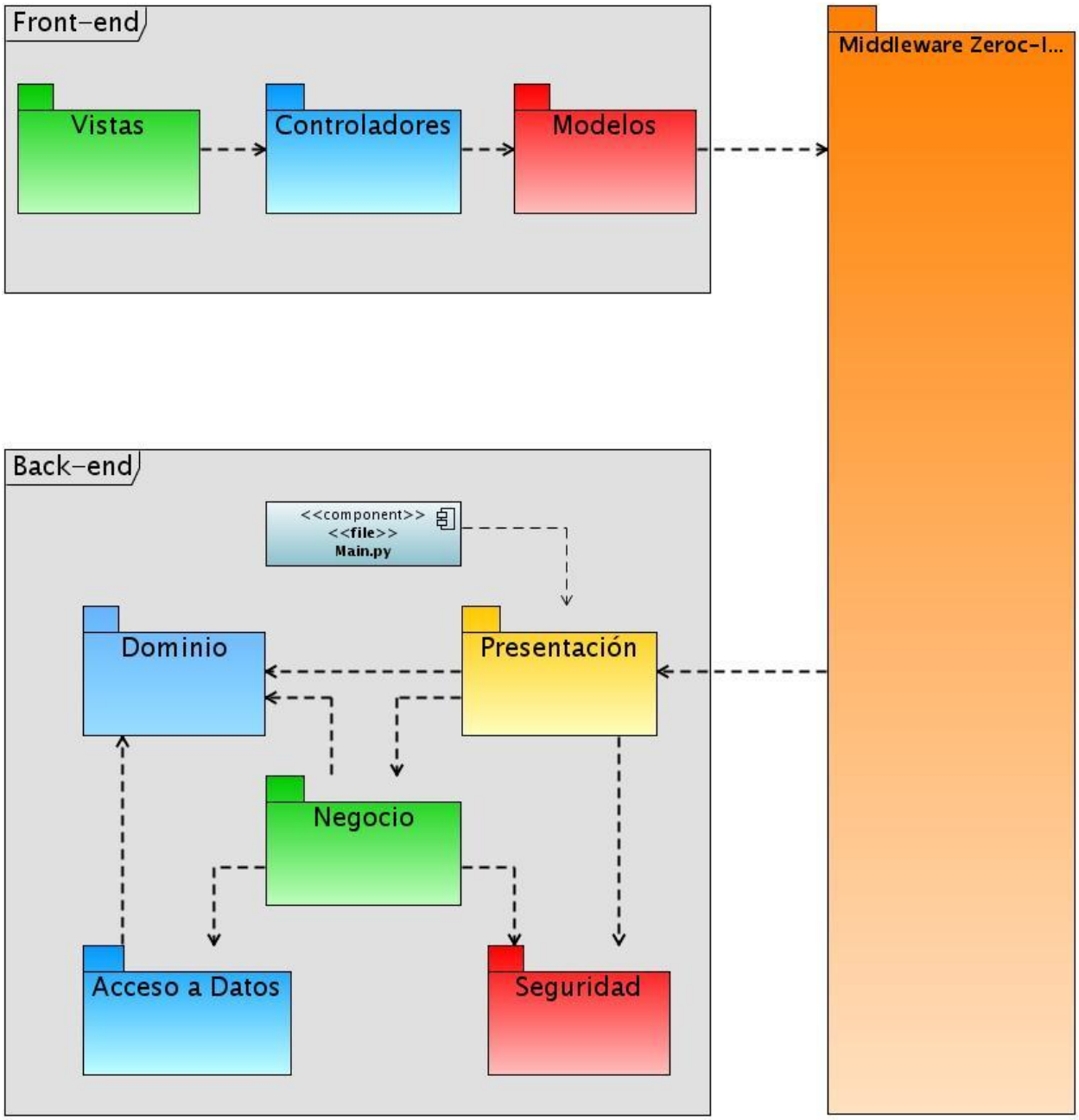


Figura 22.Diagrama de componentes de todo el sistema.

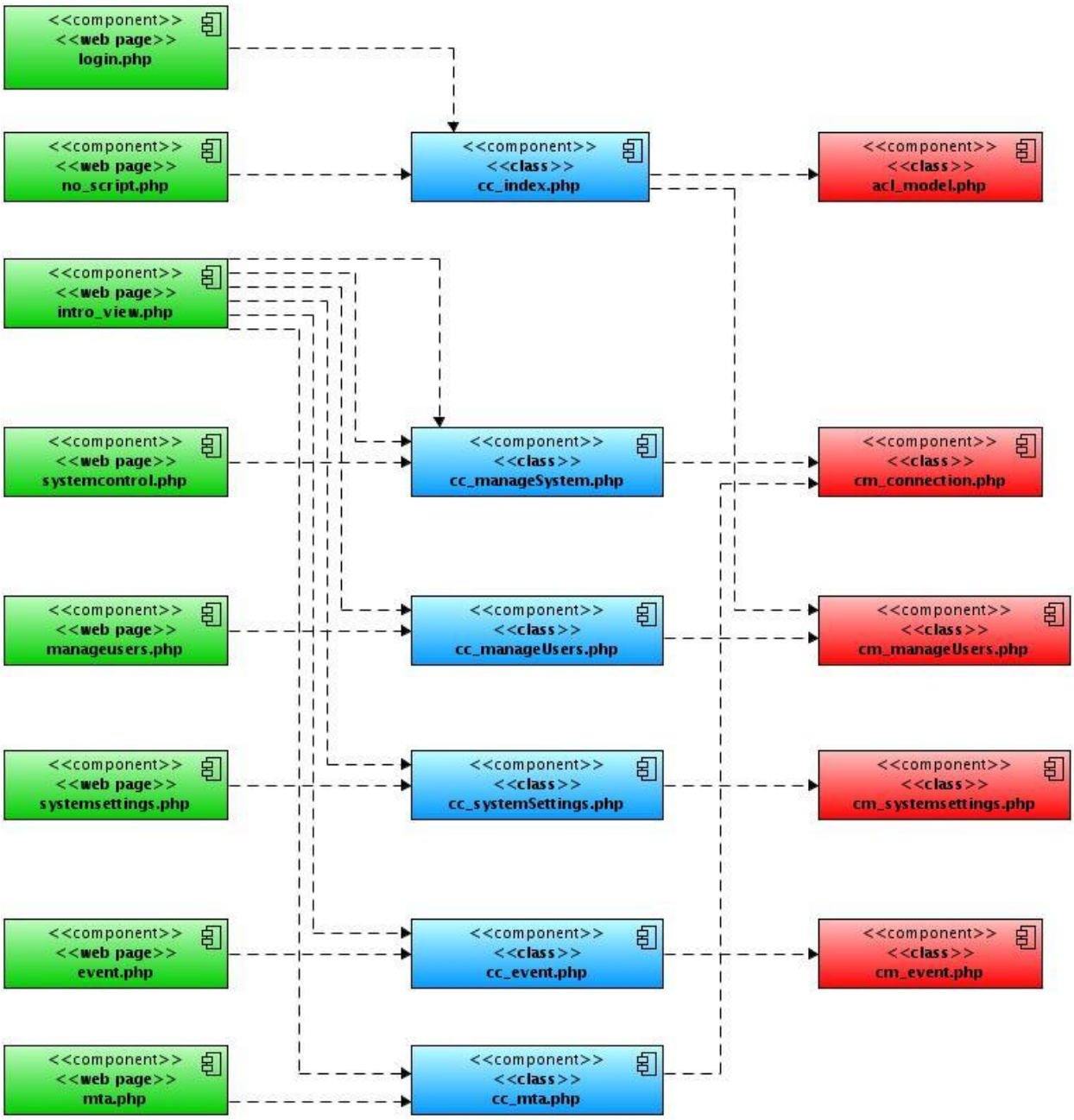


Figura 23.Diagrama de componentes del front-end, componentes Web.

Figura 24.Diagrama de componentes, Capa de Interfaz.

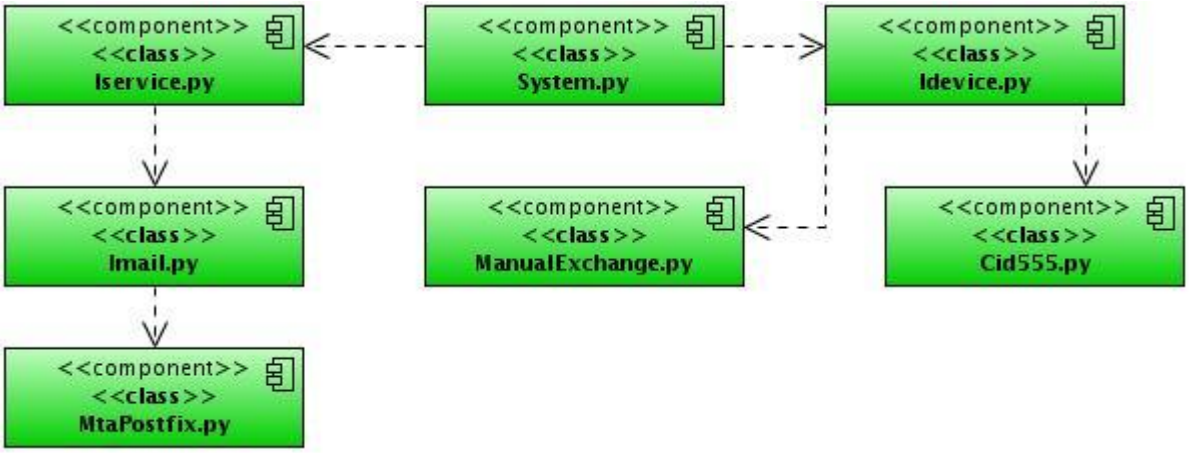


Figura 25.Diagrama de componentes, Capa de Negocio.

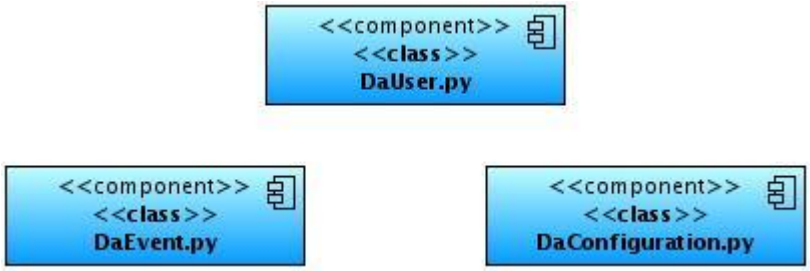


Figura 26.Diagrama de componentes, Capa de Acceso a Datos.

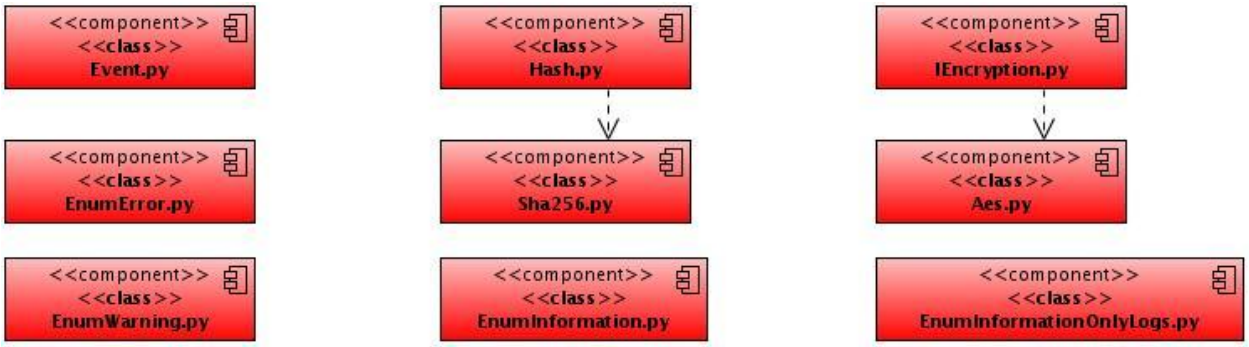


Figura 27.Diagrama de componentes, Capa de Seguridad.

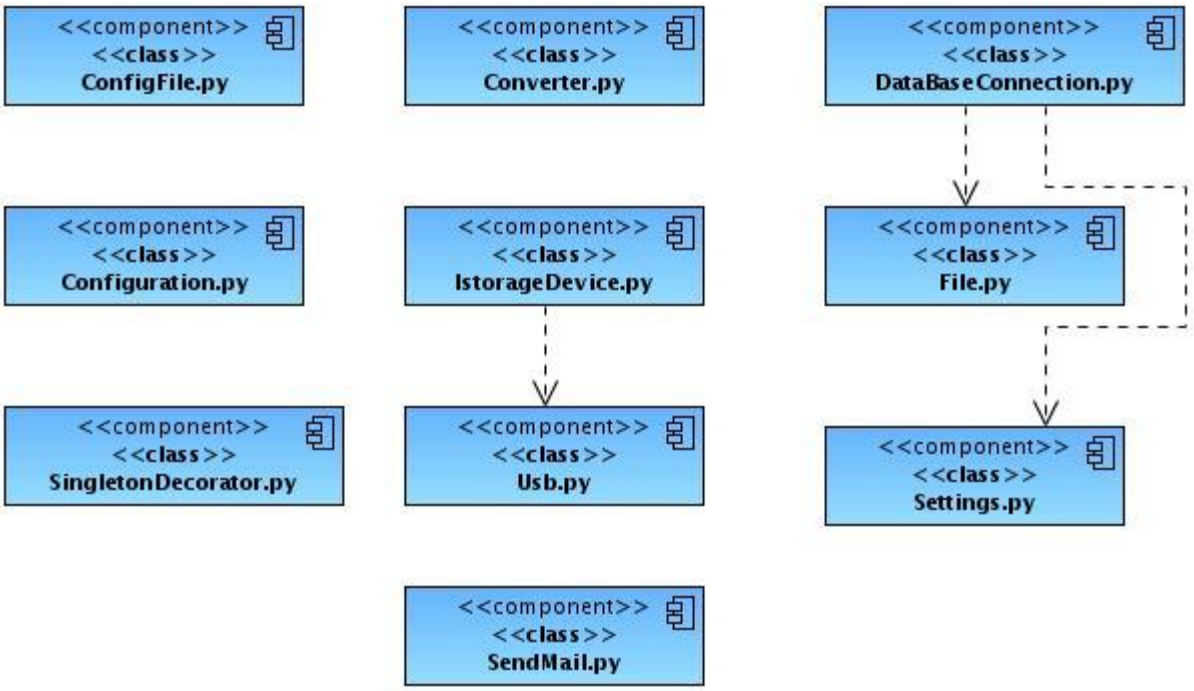


Figura 28.Diagrama de componentes, Capa de Dominio.

4.5. Caso de pruebas

Un caso de prueba no es más que un conjunto de condiciones, bajo las cuales se introducen datos, con el objetivo de obtener varios resultados, permitiendo determinar si se ha cumplido satisfactoriamente el desarrollo de las funcionalidades que se han estado probando. Se puede saber si un caso de prueba es aceptable, si el mismo presenta una alta probabilidad de detectar un error, que no haya sido encontrado hasta el momento. Cualquier tipo de prueba tiene el objetivo principal de hallar errores o defectos en el software. Para que las pruebas aplicadas a un software tengan éxito es necesario efectuar casos de pruebas con probabilidad de descubrir los errores en el sistema.

Se anexan los casos de prueba diseñados para ser aplicado a las funcionalidades con las que cuenta el sistema.

4.6. Métodos de prueba

4.6.1. Prueba de caja blanca

Mediante el método de prueba de caja blanca, se puede comprobar los caminos lógicos del software, se puede examinar el estado del mismo en varios puntos para determinar si el estado real coincide con el esperado. Para ello se requiere del conocimiento de la estructura interna del software y son derivadas del las especificaciones internas de diseño o el código.

```
def copied_from_postfix(self,  size_local_folder_out, usb):
    '''
    Metodo para copiar del MTA
    '''
    converter = Converter()                                                    1


    '''Comprobar si hay mensajes nuevos en el hold del postfix'''
    mensajes = self.test_if_new_mail()
    if not '0' in mensajes:                                                    2
        '''Comprobar si tiene espacio para los correos'''
        if (converter.convert(size_local_folder_out)) >= (converter.convert(usb.free_memory_space())):   3
            return "Memoria llena"                                             4
        '''Copiar correos para carpetas'''
        good = subprocess.call("sh -c \"for I in \`find /var/spool/postfix/hold/ -type f| cut -d '/' -f 6\`; do cp
/var/spool/postfix/hold/\$I "+out_file+" && postsuper -d \$I ; done\"",shell=True, stdout = subprocess.PIPE)
        return good                                                            5
    else:                                                                      6
        return "No hay correos nuevos"                                         7
```

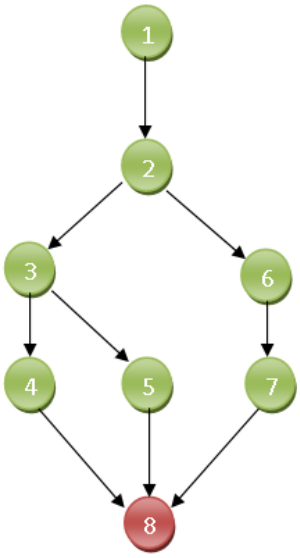


Figura 29: Prueba de Caja Blanca

Cálculo de la complejidad ciclomática

V(g) = A-N+2

=9-8+2

V(g) = 3

Descripción del método

- Se le asigna a la variable mensajes el valor de retorno del método que comprueba si hay nuevos correos en la cola de retenidos del mta.
- En caso de que hayan correos procede a verificar que todos quepan en el dispositivo de almacenamiento.

- Después de comprobado si el dispositivo de almacenamiento no está lleno procede a copiar los correos en el mismo dándole a la variable **good** el valor de retorno de la ejecución del comando.
- Si el valor de la variable **mensajes** es 0 procede a retornar el mensaje:¨No hay correos nuevos¨.

Con el cálculo de la complejidad ciclomática se puede asegurar que demostrando que asignándole valores a las variables se recorren al menos tres caminos. Esta demostración se ilustra en la tabla:

Caminos	Variables	Valores	Respuesta
1-2-3-4-8	converter.convert(size_local_folder_out)	5	Correcta
	converter.convert(usb.free_memory_space())	5	
	mensajes	3	
1-2-3-5-8	converter.convert(size_local_folder_out)	3	Correcta
	converter.convert(usb.free_memory_space())	5	
	mensajes	3	

1-2-6-7-8	mensajes	0	Correcta

Tabla 2 : Validación de los caminos

La variable **converter.convert(size_local_folder_out)**Devuelve un número entero que significa el tamaño que tiene el fichero que se encuentra en la ruta que se le pasa por parámetro.

La variable **converter.convert(usb.free_memory_space())** Devuelve la capacidad del dispositivo de almacenamiento usb.

4.7. Conclusiones parciales

En este capítulo se detalló el modelo de implementación de la solución,se mostró la relación entre los principales componentes del sistema, la organización de las clases y los objetos en términos de componentes. Se expone detalladamente la relación existente entre todos los ficheros del sistema,de igual forma, se constató la distribución del sistema mediante el Diagrama de Despliegue.

También se muestra la estrategia de pruebas definida para validar la solución, observándose una forma de programar con una complejidad ciclomática aceptable.

Los resultados se pueden categorizar como buenos en las pruebas de caja negra teniendo en cuenta el elevado número de operaciones que realiza la aplicación y la complejidad de las mismas.

Se culmina de esta forma la implementación de la aplicación, con lo que se da cumplimiento al objetivo general trazado inicialmente.

CONCLUSIONES

Luego de dar cumplimiento a los objetivos trazados al inicio de la investigación del software para la interconexión de redes aisladas. Módulo correo electrónico V2.0, se arriba a las conclusiones generales que se presentan:

- Con tecnología de software libre y un diseño ligero es posible automatizar el intercambio de mensajes entre dos redes físicamente aisladas entre sí.
- Se eliminaron las desventajas del sistema anterior sin modificar la infraestructura tecnológica actual del MIC.
- Se previene la suplantación de los dispositivos usb; copiando solo en los destinados al intercambio de datos.
- Se previene el robo de información contenida en los dispositivos usb mediante la encriptación de la misma.
- Se garantiza la integridad de los datos mediante la suma de verificación aplicadas al paquete que se transporta en la pasarela.
- Se diseñó e implementó un software para la interconexión de redes aisladas. Dicho software es extensible a las demás entidades que implementan una pasarela de correos manual en su infraestructura tecnológica.
- A la aplicación le fueron realizadas varias pruebas, entre las que se encuentran las pruebas de caja blanca y las de caja negra, obteniendo resultados satisfactorios.

RECOMENDACIONES

Luego de la realización del trabajo, se recomienda:

- Desplegar el sistema en el Ministerio de Informática y Comunicaciones (MIC).
- Implementar un sistema de reportes estadísticos e incidencias del sistema.
- Implementar nuevos módulos de comunicación en redes, por ejemplo: replicación de base de datos en línea y transferencia de ficheros.
- Establecer mecanismos de seguridad en el entorno donde se despliegue el sistema, teniendo en cuenta: sistema operativo, firewall de aplicación (SELinux), firewall de red en el host (Iptables), reglas de acceso a redes (ACL).
- Realizar un instalador de la aplicación.

REFERENCIAS BIBLIOGRÁFICAS

[1] **F. Cheng, P. Ferring, C. Meinel, G. Müllenheim, y J. Bern, «The dualgate Lock-Keeper: A highly efficient, flexible and applicable network security solution», *networks*, vol. 1, p. 5, 2007.**

[2] **D. Checkoway y N. Checkoway, *E-mail gateway system*. Google Patents, 2001.**

[3] **M. E. . Newman, S. Forrest, y J. Balthrop, «Email networks and the spread of computer viruses», *Physical Review E*, vol. 66, nº. 3, p. 035101, 2002.**

[4] **M. Hafiz, «Security patterns and evolution of MTA architecture», in *Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, 2005, pp. 142–143.**

[5] **LDC - Daniela A. Torres, «MUA, Agente Cliente de Correo», 24-mar-2010. [Online]. Available: http://ldc.usb.ve/~daniela/postfix/node4.html. [Accessed: 06-jun-2012].**

[6] **B. Costales, G. Jansen, C. Assmann, y G. N. Shapiro, *sendmail*. O'Reilly Media, Inc., 2008.**

[7] **M. Hafiz, R. Johnson, y R. Afandi, «The security architecture of qmail», in *Proceedings of the 11th Conference on Patterns Language of Programming (PLoP'04)*, 2004.**

[8] **E. Dekel y S. Sahni, «Parallel generation of postfix and tree forms», *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 5, nº. 3, pp. 300–317, 1983.**

[9] **P. Hazel, *Exim: the mail transfer agent*. O'Reilly Media, 2001.**

[10] M. Muck, M. De Courville, y P. Duhamel, «A pseudorandom postfix OFDM modulator-semi-blind channel estimation and equalization», *Signal Processing, IEEE Transactions on*, vol. 54, nº. 3, pp. 1005–1017, 2006.

[11] J. C. . Díaz, *Criptografía: Historia de la escritura cifrada*. Complutense SA Editorial, 1995.

[12] R. Palacios Hielscher y V. Delgado, «Introducción a la Criptografía: tipos de algoritmos», in *Anales de mecánica y electricidad*, 2006, vol. 83, pp. 42–46.

[13] C. Básicos, «Conceptos básicos», 2004.

[14] «Seguridad Informatica / Criptología - Algoritmos Simétricos Modernos (Llave Privada)». [Online]. Available: http://www.segu-info.com.ar/criptologia/simetricos.htm. [Accessed: 07-jun-2012].

[15] M. F. . Muñoz, S. F. . de Popayán, J. P. . Melenge, L. F. . Sanabria, y D. Investigador, «ANTOLOGÍA DEL ALGORITMO DE CIFRADO AES».

[16] F. Cheng y C. Meinel, «Research on the Lock-Keeper technology: Architectures, applications and advancements», *International Journal of Computer & Information Science*, vol. 5, nº. 3, pp. 236–245, 2004.

[17] C. Espionage, «SANS Institute InfoSec Reading Room», 2007.

[18] «Detalles de Empresa». [Online]. Available: http://www.mic.gov.cu/sitiomic/servlet/hempdetails?36,3. [Accessed: 21-jun-2012].

[19] N. Devillard, «ESO C Library for an Image Processing Software Environment (eclipse)», in *Astronomical Data Analysis Software and Systems X*, 2001, vol. 238, p. 525.

[20] B. Bibeault y Y. Katz, *jQuery in Action*. Manning Publications Co., 2008.

[21] S. A. White, «Introduction to BPMN», *IBM Cooperation*, p. 2008–029, 2004.

BIBLIOGRAFÍA

[1] «24.6. Agentes de usuario de correo». [Online]. Available: http://docs.redhat.com/docs/es ES/Red_Hat_Enterprise_Linux/5/html/Deployment_Guide/s1-email-mua.html. [Accessed: 10-jan-2012].

[2] G. Cevenini, M. Chesi, y G. Fantauzzi, «A multiprovider, universal, E-mail service for the secure exchange of legally-binding multimedia documents», in *EUROCOMM 2000. Information Systems for Enhanced Public Safety and Security. IEEE/AFCEA*, 2000, pp. 47–50. [Accessed: 10-jan-2012].

[3] K. E. Iverson, «A programming language», in *Proceedings of the May 1-3, 1962, spring joint computer conference*, 1962, pp. 345–351. [Accessed: 10-jan-2012].

[4] I. Soto Rodriguez y K. Vargas Campos, «Análisis de la eficiencia de los algoritmos de encriptación RSA, DES, IDEA y AES», *ANALES CIENTIFICOS-Universidad Nacional Agraria La Molina (Peru).(Ene-Abr 2004)*, vol. 57. [Accessed: 10-jan-2012].

[5] I. Soto Rodriguez y K. Vargas Campos, «Análisis de la eficiencia de los algoritmos de encriptación RSA, DES, IDEA y AES», *ANALES CIENTIFICOS-Universidad Nacional Agraria La Molina (Peru).(Ene-Abr 2004)*, vol. 57.[Accessed: 10-jan-2012].

[6] M. F. . Muñoz, S. F. . de Popayán, J. P. . Melenge, L. F. . Sanabria, y D. Investigador, «ANTOLOGÍA DEL ALGORITMO DE CIFRADO AES».[Accessed: 10-jan-2012].

[7] M. Pazmiño, J. Aviles, y C. L. Abad Robalino, «Captura y Análisis de los ataques informáticos que sufren las redes de datos de la ESPOL, implantando una honeynet con miras a mejorar la seguridad informática en redes de datos del Ecuador», 2009.

[8] «Componentes de software y hardware de servicios de correo - Guía de administración del sistema: servicios de red». [Online]. Available:

http://docs.oracle.com/cd/E24842_01/html/E22524/mailrefer-53.html. [Accessed: 29-feb-2012].

[9] V. Paradigm, *Consultado el: 9 de marzo de 2007*. . [Accessed: 10-jan-2012].

[10] J. Forné, J. L. Melús, y M. Soriano, «Criptografía y Seguridad en Comunicaciones», *Novática: Revista de la Asociación de Técnicos de Informática*, nº. 116, p. 20, 1995. [Accessed: 10-jan-2012].

[11] J. Forné, J. L. Melús, y M. Soriano, «Criptografía y Seguridad en Comunicaciones», *Novática: Revista de la Asociación de Técnicos de Informática*, nº. 116, p. 20, 1995.[Accessed: 10-jan-2012].

[12] M. Horkheimer, *Eclipse of reason*, vol. 1. Continuum Intl Pub Group, 1974. [Accessed: 10-jan-2012].

[13] I. E. Mendvil, «El ABC de los documentos electrónicos seguros», *Disponibleen: http://www. criptored. upm. es/guiateoria/gt_m163a. htm*, 2003.[Accessed: 10-jan-2012].

[14] «El lenguaje HTML. Manual de HTML. Tutorial de HTML. WebEstilo.» [Online]. Available: http://www.webestilo.com/html/cap2a.phtml. [Accessed: 10-jan-2012].

[15] D. Checkoway y N. Checkoway, *E-mail gateway system*. Google Patents, 2001. [Accessed: 10-jan-2012].

[16] M. E. . Newman, S. Forrest, y J. Balthrop, «Email networks and the spread of computer viruses», *Physical Review E*, vol. 66, nº. 3, p. 035101, 2002. [Accessed: 10-jan-2012].

[17] N. Devillard, «ESO C Library for an Image Processing Software Environment (eclipse)», in *Astronomical Data Analysis Software and Systems X*, 2001, vol. 238, p. 525. [Accessed: 10-jan-2012].

[18] «fetch.php (objeto application/pdf)». . [Accessed: 10-jan-2012].

[19] Rebecca Murphey, *Fundamentos de jQuery*. 2012. [Accessed: 10-jan-2012].

[20] P. G. Rodriguez, «Grupos de discusión sobre metodología de investigación cualitativa asistida por computadora en Ciencias Sociales.»[Accessed: 10-jan-2012].

[21] «HTML». [Online]. Available: http://www.hipertexto.info/documentos/html.htm. [Accessed: 10-jan-2012].

[22] «ICE Middleware». [Online]. Available: http://www.icemiddleware.com/page4.html. [Accessed: 10-jan-2012].

[23] F. Cheng, S. Roschke, y C. Meinel, «Implementing IDS Management on Lock-Keeper», *Information Security Practice and Experience*, pp. 360–371, 2009. [Accessed: 10-jan-2012].

[24] Rubén Alvarez, «Introducción a la programación en PHP», 01-ene-2001. [Online]. Available: http://www.desarrolloweb.com/articulos/303.php. [Accessed: 10-jan-2012].

[25] R. C. Gloria Cortés, «Introducción a los patrones de Software», *http://sistemas.uniandes.edu.co*. [Online]. Available: http://sistemas.uniandes.edu.co/~isis2701/dokuwiki/lib/exe/fetch.php?media=isis2701-patronesgrasp.pdf. [Accessed: 10-jan-2012].

[26] B. Bibeault y Y. Katz, *jQuery in Action*. Manning Publications Co., 2008. [Accessed: 10-jan-2012].

[27] «jQuery Project». [Online]. Available: http://jquery.org/. [Accessed: 10-jan-2012].

[28] «jQuery Tutorial». [Online]. Available: http://www.w3schools.com/jquery/default.asp. [Accessed: 10-jan-2012].

[29] «jQuery UI - Home». [Online]. Available: http://jqueryui.com/. [Accessed: 10-jan-2012].

[30] E. McCormick y K. De Volder, «JQuery: finding your way through tangled code», in *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, 2004, pp. 9–10. [Accessed: 10-jan-2012].

[31] «jQuery: The Write Less, Do More, JavaScript Library», 2012. [Online]. Available: http://jquery.com/. [Accessed: 10-jan-2012].

[32] J. Chaffer y K. Swedberg, *Learning jQuery*. Packt Publ., 2009. [Accessed: 10-jan-2012].

[33] D. Mullet y K. Mullet, *Managing Imap*. O'Reilly & Associates, Inc., 2000. [Accessed: 10-jan-2012].

[34] «Manual de PHP». [Online]. Available: http://www.manualdephp.com/. [Accessed: 10-jan-2012].

[35] «MéTodos De EncriptacióN». [Online]. Available: http://www.slideshare.net/erikitaalex/mtodos-de-encriptacin-2457832?src=related_normal&rel=2466251. [Accessed: 10-jan-2012].

[36] LDC - Daniela A. Torres, «MUA, Agente Cliente de Correo», 24-mar-2010. [Online]. Available: http://ldc.usb.ve/~daniela/postfix/node4.html. [Accessed: 10-jan-2012].

[37] «PHP: Hypertext Preprocessor», 2012. [Online]. Available: http://www.php.net/. [Accessed: 10-jan-2012].

[38] R. Mazo González, «Plataforma de correo y entorno colaborativo open source», 2011.[Accessed: 10-jan-2012].

[39] W. Venema, «Postfix», *Web page at http://postfix. org*, 2008. [Accessed: 10-jan-2012].

[40] C. Ghezzi y M. Jazayeri, *Programming language concepts*. John Wiley & Sons, Inc., 1997. [Accessed: 10-jan-2012].

[41] M. F. Sanner, «Python: a programming language for software integration and development», *J Mol Graph Model*, vol. 17, nº. 1, pp. 57–61, 1999. [Accessed: 10-jan-2012].

[42] I. Murwantara, «Rational Unified Process», 2004. [Accessed: 10-jan-2012].

[43] I. RUP, «Rational Unified Process», *Engenharia de Software*, p. 52, 2003. [Accessed: 10-jan-2012].

[44] «Red Hat Enterprise Linux 4: Manual de referencia.Capítulo 11. Correo electrónico». . [Accessed: 10-jan-2012].

[45] F. Cheng y C. Meinel, «Research on the Lock-Keeper technology: Architectures, applications and advancements», *International Journal of Computer & Information Science*, vol. 5, n^{o}. 3, pp. 236–245, 2004. [Accessed: 10-jan-2012].

[46] F. Cheng y C. Meinel, «Research on the Lock-Keeper technology: Architectures, applications and advancements», *International Journal of Computer & Information Science*, vol. 5, n^{o}. 3, pp. 236–245, 2004. [Accessed: 10-jan-2012].

[47] N. Shivakumar y H. Garcia-Molina, «SCAM: A copy detection mechanism for digital documents», 1995. [Accessed: 10-jan-2012].

[48] M. Hafiz, «Security patterns and evolution of MTA architecture», in *Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, 2005, pp. 142–143. [Accessed: 10-jan-2012].

[49] J. C. Mayes y B. W. Coile, *Security system for network address translation systems*. Google Patents, 1998. [Accessed: 10-jan-2012].

[50] M. G. Feal y J. L. . FERREIRO, «Seguridad en Internet», 2002. [Accessed: 10-jan-2012].

[51] «Seguridad Informatica / Criptología - Algoritmos Simétricos Modernos (Llave Privada)». [Online]. Available: http://www.segu-info.com.ar/criptologia/simetricos.htm. [Accessed: 10-jan-2012].

[52] S. Garfinkel, G. Spafford, y M. C. Riverol, *Seguridad y comercio en el web*. McGraw-Hill, 1999. [Accessed: 10-jan-2012].

[53] J. Klensin, «Simple mail transfer protocol», 2008. [Accessed: 10-jan-2012].

[54] F. Cheng, P. Ferring, C. Meinel, G. Müllenheim, y J. Bern, «The dualgate Lock-Keeper: A highly efficient, flexible and applicable network security solution», *networks*, vol. 1, p. 5, 2007. [Accessed: 10-jan-2012].

[55] P. Kroll y P. Kruchten, *The rational unified process made easy: a practitioner's guide to the RUP*. Addison-Wesley Professional, 2003. [Accessed: 10-jan-2012].

[56] P. Kruchten, *The rational unified process: an introduction*. Addison-Wesley Professional, 2004. [Accessed: 10-jan-2012].

[57] C. Larman, *UML y patrones*. Pearson, 1999. [Accessed: 10-jan-2012].

[58] C. Newman, «Using TLS with IMAP, POP3 and ACAP», 1999. [Accessed: 10-jan-2012].

[59] V. Paradigm, *Visual paradigm for uml*. 2010. [Accessed: 10-jan-2012].

[60] «ZeroC - The Internet Communications Engine (Ice)». [Online]. Available: http://www.zeroc.com/ice.html. [Accessed: 10-jan-2012].