

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



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

Tema:

*Migración del servidor de correo electrónico Microsoft Exchange
2003.*

Autor:

Delvis Mesa Chaviano.

Tutores:

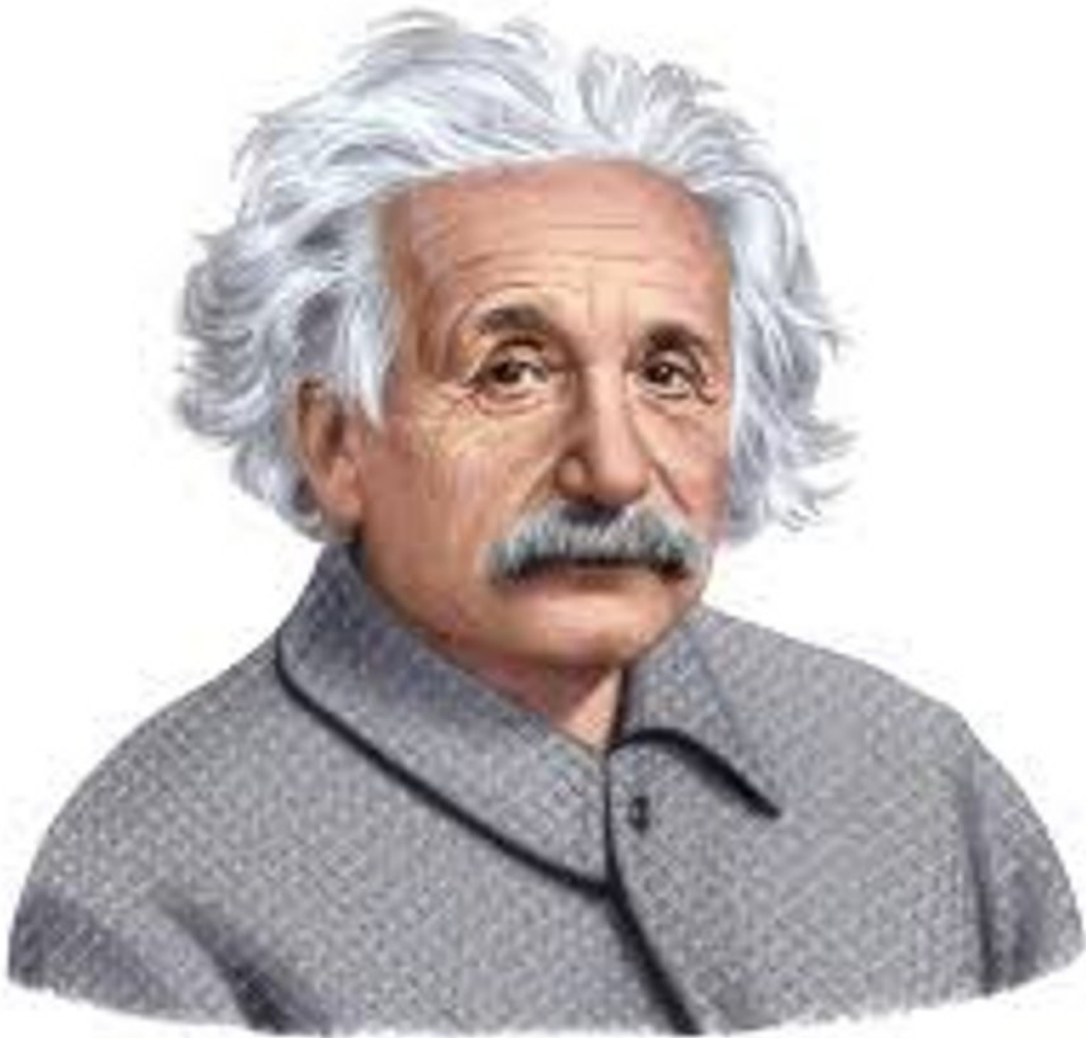
MSc. Amaury Viera Hernández.

Ing. Yadiel Pérez Villazón.

Ing. Miriela Velázquez Arias.

Ciudad de la Habana, Cuba.

Año 57 de la Revolución.



“Nunca consideres el estudio como una obligación sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.”

Albert Einstein

Declaración de autoría

Declaro ser el único autor de este trabajo y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 25 días del mes de Junio del año 2015.

Delvis Mesa Chaviano

MSc. Amaury Viera
Hernández

Ing. Yadiel Pérez
Villazón

Ing. Miriela Velázquez
Arias

Agradecimientos

A mi mamá por entregarme todo su amor y dedicación sin pedir nada a cambio y por ser la luz que me ilumina cada día de mi vida.

A mi papá por ser mi ejemplo a seguir y apoyarme en todas las decisiones que he tomado.

A mi hermano, que aunque sea menor que yo siempre me ha apoyado y me ha entregado su amor.

A mi familia en general que siempre me ha apoyado para que pueda cumplir mi sueño de ser ingeniero.

A mis tutores que me han brindado su ayuda incondicionalmente durante este curso. A Miriela que dedicó largas horas de su tiempo a ayudarme siempre con una sonrisa y haciendo parecer que todo era más fácil.

A mis amigos que han compartido buenos y malos momentos durante estos cinco años y entre todos nos hemos apoyado para salir adelante.

Dedicatoria

A mi mamita querida que siempre ha luchado para que yo y mi hermano seamos felices, que sin importar los problemas que nos hayan invadido siempre ha sabido combatirlos haciéndonos sentir seguros. Mamá deseo que la vida me alcance para retribuirte todo el amor que me has brindado y por ese amor hoy te dedico este trabajo.

Resumen

La Universidad de las Ciencias Informáticas juega un papel fundamental en el proceso de migración a software libre que se lleva a cabo en nuestro país. El departamento SIMAYS comenzó a desarrollar la herramienta HMAST con el objetivo de migrar y administrar servicios telemáticos de forma remota, dicha herramienta cuenta con un módulo que permite la administración del servicio de correo electrónico, sin embargo, no posee un mecanismo que permita transferir información desde un servidor Exchange en su versión 2003. En la presente investigación se realiza un estudio de las principales herramientas que son empleadas para llevar a cabo la migración de un servidor de este tipo hacia las distintas tecnologías de servidores de correo en GNU/Linux (Postfix, Dovecot y Davical) con el objetivo de diseñar un módulo para HMAST que permita transferir la información almacenada en dicho servidor de forma remota. Además se documentan las tecnologías, herramientas y lenguajes de programación propuestos, así como elementos asociados al análisis y diseño. Entre las principales funcionalidades identificadas durante el análisis se encuentran las asociadas a importar la información correspondiente a las cuentas de correo electrónico, crear las cuentas correspondientes en la base de datos que esté utilizando el módulo de correo (directorio activo, MySQL, etc) y obtener las configuraciones del servidor Exchange. La ejecución de pruebas de validación estáticas permitió validar el correcto diseño del módulo, así como la especificación de requisitos funcionales. Se desarrolló una aplicación de prueba que posibilitó validar la propuesta de solución. Dicha aplicación, utilizando la librería libmapi, importa los mensajes, contactos, calendarios y tareas de una cuenta de correo electrónico desde un servidor Exchange garantizando la integridad de toda la información.

Palabras clave: HMAST, migración, administración, cuentas de correo electrónico.

Índice de contenidos

| | |
|---|-----|
| Declaración de autoría..... | II |
| Agradecimientos..... | III |
| Dedicatoria..... | IV |
| Resumen..... | V |
| Introducción..... | 1 |
| Capítulo 1: Fundamentación teórica..... | 7 |
| 1. 1 Correo electrónico..... | 7 |
| 1. 1. 1 Conceptos asociados al correo electrónico..... | 7 |
| 1. 2 Microsoft Exchange..... | 8 |
| 1. 2. 1 Protocolos de Exchange..... | 9 |
| 1. 2. 3 Método de almacenamiento..... | 10 |
| Grupos de almacenamiento..... | 11 |
| 1. 3 Herramienta para la migración y administración de servicios telemáticos..... | 12 |
| 1. 3. 1 Arquitectura..... | 13 |
| 1. 3. 2 Tecnologías asociadas a HMAST..... | 15 |
| 1. 4 Estudio de herramientas homólogas..... | 16 |
| 1. 4. 1 Zimbra..... | 16 |
| 1. 4. 2 Zentyal..... | 18 |
| Arquitectura..... | 19 |
| 1. 4. 3 Openchange..... | 21 |
| Compatibilidad con Microsoft Exchange..... | 22 |
| 1. 4. 4 Zarafa..... | 23 |
| 1. 5 Otras herramientas para migrar servidores IMAP..... | 24 |
| 1. 5. 1 ImapCopy..... | 24 |
| 1. 5. 2 Imapsync..... | 24 |

| | |
|---|----|
| 1. 5. 3 Larch..... | 24 |
| Técnica utilizada para la comparación de las herramientas investigadas..... | 25 |
| Comparación entre las herramientas analizadas..... | 26 |
| 1. 6 Metodología de desarrollo de software..... | 28 |
| 1. 7 Herramientas y tecnologías de modelado..... | 29 |
| 1. 7. 1 Lenguaje para el modelado..... | 29 |
| 1. 7. 2 Herramientas para el diseño..... | 30 |
| Conclusiones parciales..... | 31 |
| Capítulo 2: Análisis y diseño..... | 32 |
| 2. 1 Modelo del dominio..... | 32 |
| 2. 2 Propuesta del módulo de migración..... | 34 |
| 2. 3 Captura de Requisitos..... | 36 |
| 2. 3. 1 Requisitos funcionales..... | 36 |
| 2. 3. 2 Requisitos no funcionales (RNF)..... | 38 |
| 2. 4 Historias de usuario (HU)..... | 39 |
| 2. 5 Herramientas y tecnologías a utilizar..... | 42 |
| 2. 6 Arquitectura del módulo..... | 42 |
| 2. 7 Diagrama de paquetes..... | 43 |
| 2. 8 Patrones de diseño..... | 46 |
| Conclusiones parciales..... | 48 |
| Capítulo 3: Pruebas..... | 49 |
| 3.1 Pautas de codificación..... | 49 |
| 3.2 Pruebas..... | 50 |
| 3. 2. 1 Pruebas a través de Técnicas Estáticas..... | 51 |
| 3. 2. 2 Validación de Requisitos..... | 51 |

| | |
|--|----|
| Técnicas de Validación de Requisitos..... | 52 |
| 3. 2. 3 Pruebas con la aplicación experimental..... | 53 |
| Escenario de pruebas..... | 53 |
| Windows..... | 53 |
| 3. 2. 4 Pruebas con la herramienta experimental..... | 54 |
| 3. 3 Resultados de las pruebas..... | 58 |
| Conclusiones parciales..... | 59 |
| Conclusiones generales..... | 60 |
| Recomendaciones..... | 61 |
| Referencias Bibliográficas..... | 62 |
| Glosario de términos..... | 66 |

Índice de Figuras

| | |
|--|----|
| Figura 1: Estructura de los grupos de almacenamiento..... | 12 |
| Figura 2: Arquitectura de HMAST..... | 14 |
| Figura 3: Componentes y protocolos que integra Zentyal..... | 19 |
| Figura 4: Consumo de recursos de Zentyal..... | 21 |
| Figura 5: Proceso de migración del servidor Exchange 2003..... | 33 |
| Figura 6: Diagrama de procesos de negocio..... | 36 |
| Figura 7: Arquitectura del módulo de migración..... | 43 |
| Figura 8: Diagrama de paquetes del módulo migración..... | 45 |

Índice de tablas

| | |
|--|----|
| Tabla 1: Comparación entre las herramientas..... | 26 |
| Tabla 2: Comparación entre las herramientas..... | 27 |
| Tabla 3: Encuesta para obtener configuraciones de servidor..... | 35 |
| Tabla 4: Lista de Reserva del Producto del módulo..... | 38 |
| Tabla 5: Comparación entre los resultados de la herramienta y los datos de Exchange..... | 57 |

Introducción

El avance de las Tecnologías de la Información y las Comunicaciones (TIC's), enmarcado en una era de constante desarrollo tecnológico, ha permitido la optimización de los servicios telemáticos en todo el mundo. El término telemático define la ciencia que reúne y combina las posibilidades técnicas y los servicios de la telecomunicación y la informática [1].

Un servicio telemático es un servicio destinado a la difusión, almacenamiento y tratamiento de la información en cualquiera de sus formas, ya sea voz, video, datos o una combinación de todas ellas (multimedia). Ejemplos de ellos lo constituyen el correo electrónico, telnet, la web y el protocolo de transferencia de archivos (FTP por sus siglas en inglés) [2].

El correo electrónico es un servicio de red que permite que dos o más usuarios se comuniquen entre sí por medio de mensajes que son enviados y recibidos a través de una computadora o dispositivo similar. Este es uno de los servicios más utilizados de Internet, ya que favorece las comunicaciones veloces y confiables [3].

Los servicios telemáticos están presentes en la mayoría de las organizaciones, y en muchos casos juegan un papel estratégico para las mismas. Existe un gran número de empresas que brindan estos servicios sobre sistemas operativos de la familia Microsoft Windows. Sin embargo, para muchas de estas la migración hacia plataformas GNU/Linux es de suma importancia debido a características como la reducción de los costos de licencias, tanto por el sistema operativo instalado como por las aplicaciones que se ejecutan en el mismo, la oportunidad de alcanzar una alta disponibilidad y tiempo efectivo de funcionamiento, el aseguramiento de una mayor estabilidad, fiabilidad y seguridad, entre otras [4].

El proceso de migración a software libre y código abierto se hace cada vez más cotidiano y va alcanzando mayor popularidad a nivel mundial, pues constituye una alternativa para el desarrollo de la informática, tanto en el marco público como en el privado. Dicho proceso es una vía factible para Cuba por ser un país

bloqueo económicamente, lo que representa una alternativa para superarse económicamente y lograr la independencia tecnológica [5].

En muchos de los Organismos de la Administración Central del Estado (OACE) en Cuba se utilizan servicios de correo electrónico como una de las vías principales para la comunicación, utilizando Microsoft Exchange Server como una de las tecnologías para la administración de dicho servicio. La migración de este servicio hacia las tecnologías de correo GNU/Linux es un proceso que puede tornarse complejo y con un consumo de tiempo excesivo.

Las razones que respaldan esta afirmación se exponen a continuación:

- En este proceso es necesario configurar las tecnologías del servicio de correo sobre los sistemas operativos GNU/Linux con las configuraciones homólogas en Microsoft Exchange Server, tarea que en ocasiones puede tornarse compleja, pues muchas veces es necesario introducir comandos en consola y realizar cambios en los ficheros de configuración de los servicios, todo esto de forma manual.
- Se requieren de otras tareas, como la creación de cuentas en el servidor de correo de GNU/Linux, creación de la estructura de los buzones, y la más importante, la transferencia de grandes volúmenes de datos (mensajes de correo, contactos, calendarios y tareas), todo lo anterior para cada usuario existente en Microsoft Exchange Server, proceso que se realiza también de forma manual.

Cuba no está exenta de los temas vinculados a la migración a software libre. Para ello la Universidad de las Ciencias Informáticas (UCI) cuenta, dentro de su infraestructura docente-productiva, con el Centro de Software Libre (CESOL). El Departamento de Servicios Integrales en Migración, Asesoría y Soporte (SIMAYS) de este centro tiene a su cargo la misión de brindar servicios y desarrollar aplicaciones de apoyo al proceso de migración en los OACE del país. En este proceso, una de las principales tareas que se lleva a cabo es la migración de los servicios telemáticos, y dentro de ellos el de correo electrónico.

Una de las soluciones de SIMAYS es la Herramienta para la Migración y Administración de Servicios Telemáticos (HMAST), dirigida a administrar y migrar los servicios telemáticos de forma remota. Durante los años 2013 y 2014 se desarrolló un módulo para HMAST que permite la administración del servicio de correo electrónico, sin embargo, este no cuenta con un mecanismo que permita realizar la transferencia de datos desde un servidor de correo Microsoft Exchange en su versión 2003 hacia un servidor de correo electrónico en GNU/Linux de forma automatizada, lo que trae consigo que se realice de forma manual dicho proceso.

Partiendo de la situación problemática identificada se determina como **problema de la investigación:** ¿Cómo transferir los datos almacenados en un servidor de correo Microsoft Exchange 2003 hacia la plataforma de correo electrónico de HMAST de forma tal que se reduzca el tiempo durante este proceso y se garantice la integridad de los datos?

Teniendo en cuenta el problema definido con anterioridad se determina como **objeto de estudio** las herramientas para la migración de servidores de correo electrónico hacia tecnologías libres de dicho servicio. Enmarcando como **campo de acción** las herramientas para la migración de servidores de correo electrónico Microsoft Exchange 2003 hacia tecnologías libres de dicho servicio en las organizaciones cubanas.

Se propone como **objetivo general** diseñar un módulo para HMAST que posibilite tras su implementación la transferencia de datos de un servidor de correo Microsoft Exchange 2003 hacia la plataforma de correo electrónico de dicha herramienta de forma tal que se reduzca el tiempo durante este proceso y se garantice la integridad de los datos.

Se definen los siguientes **objetivos específicos:**

- Analizar las herramientas que permiten la migración del servidor de correo electrónico Microsoft Exchange en su versión 2003 hacia un servidor de correo en GNU/Linux.

- Definir el procedimiento de migración del servidor de correo Microsoft Exchange 2003 hacia un servidor de correo en GNU/Linux a utilizar.
- Analizar y diseñar la solución propuesta.
- Validar el diseño del módulo.

Para cumplir los objetivos anteriores, se plantean como **tareas de la investigación** las siguientes:

- Definición de aspectos teóricos asociados a la migración de los servicios de correo hacia plataformas libres.
- Análisis del estado del arte referente a la migración del servidor de correo Microsoft Exchange en su versión 2003 hacia un servidor de correo en GNU/Linux.
- Descripción de la estrategia de migración a emplear.
- Análisis de la estructura y las características de HMAST.
- Especificación de requerimientos funcionales y no funcionales del módulo.
- Definición de los patrones de diseño.
- Realización de pruebas estáticas para validar el diseño.
- Realización de pruebas con la herramienta experimental desarrollada.

Se define como **idea a defender**:

El diseño de un módulo para HMAST que permita migrar el servidor de correo Microsoft Exchange 2003 hacia un servidor de correo electrónico en GNU/Linux dará paso a su posterior implementación y permitirá reducir el tiempo requerido para el proceso de migración y garantizar la integridad de los datos.

Los métodos científicos utilizados son:

Análisis Histórico-Lógico: Este método se utiliza para determinar los antecedentes históricos relacionados con las herramientas para la migración del servidor de correo Microsoft Exchange en su versión 2003.

Analítico-Sintético: Es utilizado en la investigación de las herramientas y mecanismos existentes para la migración del servidor de correo Microsoft Exchange 2003, estudiando sus principales características y funcionalidades, lo que permite seleccionar la herramienta más apropiada para guiar el diseño del módulo de migración.

Experimento: El método experimento se utiliza en el desarrollo de la herramienta de pruebas con el objetivo de verificar la validez de la propuesta de solución brindada, creando las condiciones para transferir información desde un servidor Exchange en su versión 2003 hacia un servidor de correo electrónico en GNU/Linux en un escenario controlado.

El presente trabajo de diploma consta de tres capítulos estructurados de la siguiente forma:

Capítulo I Fundamentación teórica: Se exponen definiciones asociadas al correo electrónico y a la migración de dicho servicio. Se analizan las herramientas que permiten la migración del servidor de correo electrónico Microsoft Exchange 2003, así como sus características generales, funcionalidades y configuraciones. Se fundamenta la selección de la metodología de desarrollo de software, las herramientas y lenguajes involucrados en el diseño.

Capítulo II Análisis y Diseño: Se realiza la descripción del negocio. Se especifican los requisitos funcionales y no funcionales de la aplicación propuesta y se describen las historias de usuario presentes

en el sistema. Se proponen las tecnologías y lenguajes para el modelado y diseño del módulo. Se define la arquitectura del módulo así como los patrones de diseño a utilizar.

Capítulo III Pruebas: En este capítulo se valida la propuesta de solución. Se emplean técnicas estáticas para validar el correcto diseño del módulo y la especificación de los requisitos del mismo, se emplea una herramienta experimental para verificar el correcto funcionamiento del mecanismo de migración propuesto.

Capítulo 1: Fundamentación teórica

En la actualidad la migración a software libre es un proceso que se hace cada vez más cotidiano y va alcanzando mayor auge a nivel mundial debido a las facilidades que brinda, tales como libertad de uso y distribución, independencia tecnológica y soporte. Un paso fundamental dentro del proceso de migración a software libre en una organización es migrar los servicios telemáticos y dentro de ellos el servicio de correo electrónico, el cual es usado en la mayoría de las organizaciones cubanas como vía principal para la comunicación, pues constituye una de las formas más rápidas y sencillas por la cual se puede enviar mensajes y archivos [6].

1. 1 Correo electrónico

El correo electrónico comenzó a utilizarse en 1965 en una súper computadora de tiempo compartido y para 1966 se había extendido rápidamente para utilizarse en las redes de computadoras. En 1971 Ray Tomlinson incorporó el uso de la arroba (@) como divisor entre el usuario y la computadora en la que se aloja el correo porque no existía este símbolo en ningún nombre ni apellido. En inglés la arroba se lee «at» (en). Así, *usuario@servidor.com* se lee “usuario en servidor punto com”.

El nombre correo electrónico proviene de la analogía con el correo postal: ambos sirven para enviar y recibir mensajes y se utilizan "buzones" intermedios (servidores), en donde los mensajes se guardan temporalmente antes de dirigirse a su destino, y antes de que el destinatario los revise [7].

1. 1. 1 Conceptos asociados al correo electrónico

A continuación se definirán algunos conceptos asociados al correo electrónico con el objetivo de alcanzar un mayor entendimiento de este servicio.

Lista de contactos

Información privada o personal de uno o varios individuos que otra persona, negocio o entidad pueden utilizar para llegar a dicho(s) individuo(s) [8].

Lista de correo

Las listas de correo son una función especial del correo electrónico que permite la distribución de mensajes entre múltiples usuarios de forma simultánea. Al enviar un mensaje que tiene como destinatario una lista de correo este llegará a la dirección de todas las personas inscritas en ella. Dependiendo de la configuración de la lista de correo el receptor tendrá o no la posibilidad de enviar mensajes [9].

Alias de correo

Un alias de correo es un enlace a una dirección de correo electrónico real. No ocupa espacio en ningún sitio y no permite el envío de correos, solo la recepción de los mismos. Su principal característica es que la cuenta alias no necesariamente debe existir [10].

Buzón de correo electrónico

Es el destino al que se entregan mensajes de correo electrónico. Es el equivalente a un buzón en el sistema postal.

1. 2 Microsoft Exchange

Microsoft Exchange es un producto de software de colaboración entre usuarios desarrollado por Microsoft. Es parte de la línea de productos para servidores de Microsoft ya que es una de las aplicaciones destinadas para el uso de servidores. Microsoft Exchange permite manipular mensajes de correo electrónico, calendarios, contactos y tareas [11].

El servidor de Exchange se utiliza usualmente en conjunto con Microsoft Outlook para aprovechar las características de colaboración de Outlook, tales como la capacidad para compartir calendarios y listas de contactos.

1. 2. 1 Protocolos de Exchange

Exchange emplea el protocolo **MAPI** (Message Application Programming Interface), a veces denominada API de mensajería como método de acceso a mensajes. Microsoft comenzó a desarrollar MAPI poco después que se desarrollaran los protocolos IMAP y POP, y tiene usos más abarcadores que el simple correo electrónico. MAPI es una vía para que las aplicaciones y clientes de correo electrónico puedan comunicarse con los servidores de Microsoft Exchange, y es capaz de sincronizar los correos electrónicos al estilo **IMAP**, además de contactos, calendarios y otros elementos, todo ligado a clientes de correo electrónico locales o aplicaciones. Microsoft Exchange puede ser configurado para emplear los protocolos POP3 e IMAP para el acceso a los mensajes [12].

Perfil MAPI

Un perfil MAPI es un conjunto de valores de configuración de MAPI almacenados en el registro de Exchange. Con un perfil MAPI es posible la conexión con diferentes servicios de mensajería, como por ejemplo Microsoft Exchange Server, Microsoft Outlook y Outlook Express [13].

Implementaciones de MAPI de código abierto

Hasta el momento las implementaciones de código abierto de MAPI han sido pocas, pero hay al menos tres proyectos de código abierto que trabajan en la implementación del protocolo MAPI en librerías para ser usadas en otras aplicaciones de código abierto. Esta lista incluye el proyecto OpenMapi, MAPI4Linux de Zarafa (también parte de OpenMapi) y libmapi, subproyecto del proyecto OpenChange.

1. 2. 2 Exchange 2003 y Active Directory

Exchange 2003 utiliza Active Directory¹ para almacenar y compartir información con Microsoft Windows. La estructura de unidades organizativas y los grupos, así como la información de directorios que crea y mantiene Windows, puede utilizarse también desde Exchange. Con Active Directory puede almacenar y organizar información de usuarios como nombres, direcciones de correo electrónico y números de teléfono. Esta información se almacena en forma de atributos del objeto de usuario. Dicha información puede utilizarla Exchange y otras aplicaciones [14].

1. 2. 3 Método de almacenamiento

El servidor Exchange 2003 almacena los datos en dos archivos: un archivo *.edb* y un archivo *.stm*. Ambos archivos forman un repositorio de almacén de dicho servidor. Por ejemplo, el almacén de buzones predeterminado de un servidor Exchange en su versión 2003 utiliza los archivos *Priv1.edb* y *Priv1.stm*, el almacén de carpetas públicas predeterminado utiliza los archivos *Pub1.edb* y *Pub1.stm*. El archivo *.edb* contiene tablas que albergan metadatos de todos los mensajes de correo electrónico y otros elementos del almacén de Exchange, así como el contenido de los mensajes MAPI. El archivo *.edb* es una base de datos ESE (Motor de almacenamiento extensible) que se utiliza principalmente para almacenar mensajes. El archivo *.stm* es también una base de datos ESE, denominada base de datos de secuencias. Los archivos *.edb* y *.stm* funcionan a la par, y la firma de base de datos (un número aleatorio de 32 bits combinado con la hora de creación de la base de datos) se almacena como encabezado en ambos archivos [14].

- El archivo *.edb* sirve de repositorio principal para los datos de buzones almacenados por Microsoft Exchange 2003 y está asociado a una base de datos de buzones o a una base de datos de carpetas públicas. Cada archivo *.edb* puede contener varios buzones o carpetas públicas. Contiene encabezados de mensajes, textos de mensajes y documentos adjuntos.

¹ Active Directory: En español significa Directorio Activo.

- Los mensajes de correo electrónico enviados recientemente, que no han sido accedidos por un usuario a través del cliente de correo correspondiente, se almacenan en archivos *.stm*. Cuando los clientes MAPI intentan acceder a los mensajes de correo electrónico almacenados en estos archivos en el servidor de correo, este contenido se transmite y almacena en los archivos *.edb* asociados.

Existen dos tipos de bases de datos disponibles en Exchange Server 2003:

- **Bases de datos de almacén privado:** Almacenan buzones y colas de mensajes.
- **Bases de datos del almacén público:** Almacenan jerarquías y contenido de carpetas públicas.

Grupos de almacenamiento

Un grupo de almacenamiento de Exchange es un contenedor lógico para bases de datos y sus archivos asociados del sistema y de registro de transacciones. Cada grupo de almacenamiento está formado por un conjunto de archivos de registro y archivos auxiliares (bases de datos temporales internas, el archivo de punto de control y los registros reservados) para todas las bases de datos (archivos *.edb* y *.stm*) del grupo. Exchange Server 2003 admite varios grupos de almacenamiento y varias bases de datos en cada uno de ellos.

En Exchange Server 2003 un servidor admite hasta cuatro grupos de almacenamiento y un grupo de almacenamiento admite hasta cinco bases de datos. Debido a la compatibilidad con varias bases de datos se puede distribuir un gran número de buzones y carpetas públicas entre muchas bases de datos más pequeñas y, de ese modo, facilitar la administración de las mismas [15].

Como se ilustra en la *Figura 1* cada grupo de almacenamiento se representa por una instancia de ESE.

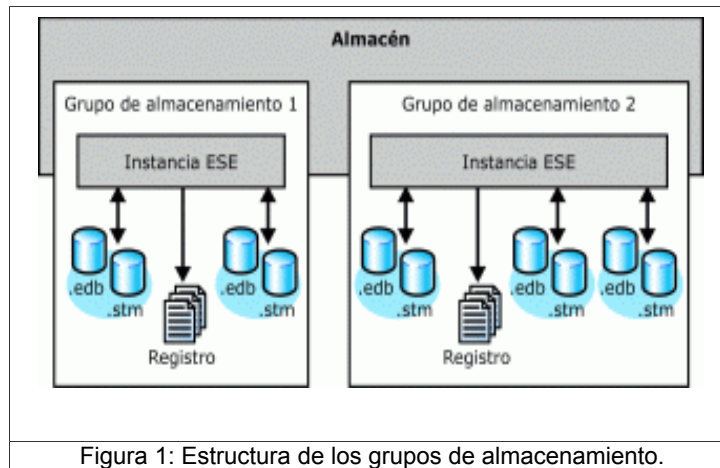


Figura 1: Estructura de los grupos de almacenamiento.

1. 3 Herramienta para la migración y administración de servicios telemáticos.

HMAST es un sistema web para la administración de servicios telemáticos desarrollado en la UCI. Su objetivo consiste en apoyar el proceso de migración a software libre que se realiza en los OACE.

El sistema se desarrolló bajo el concepto de poder administrar desde un único servidor a los restantes servidores de la institución mediante el protocolo SSH2, por lo que se optimizaría el consumo de recursos en los mismos, ya que para administrarlo sólo se requiere tener instalado OpenSSH en los servidores.

El diseño está dirigido a que el sistema sea usado tanto por usuarios avanzados como por aquellos que poseen pocos conocimientos asociados a los servicios telemáticos en GNU/Linux, pues toda la interfaz está en idioma Español, las variables de configuración tienen nombres intuitivos que facilitan la inferencia de su funcionalidad, cada parámetro de configuración tiene una ayuda que indica los valores que puede tener y una recomendación de cuál es el valor más adecuado en cada caso. Al introducir errores en el sistema no se guardan los cambios en el servidor y se le notifica al usuario para que lo modifique.

En su primera versión la herramienta cuenta con dos módulos dirigidos a administrar los servicios DHCP y MySQL. El módulo de DHCP se encarga de gestionar los ámbitos y las asignaciones estáticas, así como

de configurar de manera general el servicio. Por otra parte, el módulo de MySQL administra tanto las configuraciones generales como las avanzadas y las configuraciones de los registros del servicio.

1.3.1 Arquitectura

La herramienta HMAST presenta una arquitectura N-capas basada en el dominio compuesta por cinco capas. Dichas capas son descritas a continuación.

- **Capa de presentación:** presenta al usuario los conceptos de negocio mediante una interfaz de usuario.
- **Capa de aplicación:** realiza llamadas a servicios de la capa inferior y tiene la responsabilidad de adaptar la información que recibe de los servicios de dominio.
- **Capa de dominio:** es responsable de las validaciones. Define las interfaces de persistencia a datos (contratos de repositorio) pero no los implementa, y está compuesta por entidades del dominio que representan objetos del dominio y están definidas fundamentalmente por su identidad, servicios de dominio que contienen la lógica que trata a las entidades como un todo y los contratos de repositorios que son interfaces que especifican las operaciones que deben implementar los repositorios.
- **Capa de persistencia:** contiene el código necesario para persistir los datos y como componente los repositorios, estos últimos son clases que implementan los contratos de repositorios definidos en la **capa de dominio**.

- **Capa Infraestructura transversal:** promueve la reutilización de código. Alberga las operaciones de seguridad, *logging*, monitoreo del sistema, mecanismos de persistencia reutilizables, validadores genéricos y todas aquellas operaciones que se puedan utilizar desde otras capas (Ver *Figura 2*).

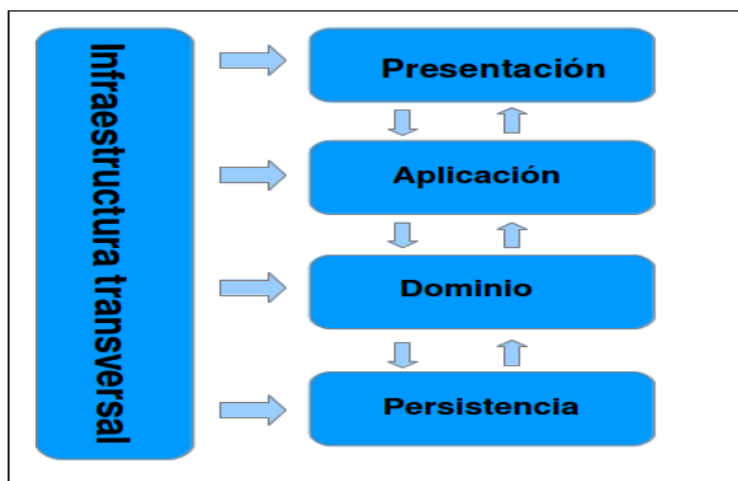


Figura 2: Arquitectura de HMAST.

Funcionalidades que brinda:

- **Gestión de servidores lógicos:** Permite la adición, edición y eliminación de los datos de un servidor lógico, además permite la conexión y desconexión remota a un servidor seleccionado.
- **Gestión de servicios telemáticos asociados a un servidor lógico:** Permite la adición, edición y eliminación de los datos de un módulo, así como activación y desactivación de los mismos.
- **Gestión de las variables de configuración asociadas a un servidor lógico:** Permite cargar y salvar las variables de configuración de los servicios telemáticos encontradas en un servidor lógico (ficheros de configuración, nombre de módulos, demonios, etc).

1. 3. 2 Tecnologías asociadas a HMAST

A continuación se mencionan un conjunto de tecnologías asociadas al desarrollo de HMAST. Dichos aspectos se encuentran especificados en el documento *Plantilla de Concepción del Sistema HMAST* recogido en el expediente de proyecto de dicho sistema.

Java

Java es un lenguaje de programación orientado a objetos de fácil aprendizaje para los usuarios provenientes de C/C++. Ofrece el código de casi todas sus librerías nativas para que los desarrolladores puedan conocerlas y estudiarlas en profundidad, o bien ampliar su funcionalidad, beneficiándose a ellos mismos y a los demás [16].

XML

El lenguaje de marcas extensible XML es una tecnología sencilla estandarizada por la W3C. Es empleado para la persistencia temporal de información en la computadora donde se encuentra el servidor de aplicaciones mientras se mantiene establecida una conexión [17].

Spring framework

Spring es un framework de código abierto que ofrece un modelo de programación y configuración para las modernas aplicaciones empresariales basadas en Java para cualquier tipo de plataforma de despliegue [18].

Netbeans

NetBeans es un entorno de desarrollo integrado libre dirigido principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso [19].

1. 4 Estudio de herramientas homólogas

En la actualidad existen varias tecnologías que posibilitan la migración de un servidor de correo electrónico Microsoft Exchange 2003 hacia un servidor de correo electrónico de los existentes en GNU/Linux. A continuación se realiza un análisis de las principales aplicaciones que pueden llevar a cabo este proceso, así como sus características y funcionalidades. Se resumen además sus principales ventajas y desventajas.

Entre las herramientas más usadas para la migración del servicio en cuestión se determinó que fueran analizadas Zimbra, Zentyal, OpenChange y Zarafa.

1. 4. 1 Zimbra

Zimbra es una suite de correo electrónico y mensajería de código abierto. Zimbra Server es el núcleo de Zimbra Collaboration Suite. Está diseñado sobre una arquitectura estable y modular usando tecnologías de código abierto, tales como:

- Postfix: Agente de transferencia de mensajes (MTA) que implementa el protocolo SMTP (Simple Mail Transfer Protocol).
- OpenLDAP: Servidor de directorios incorporado.

Zimbra emplea un asistente de migración para transferir cuentas de correo electrónico del servidor Microsoft Exchange en su versión 2003 al servidor Zimbra Collaboration Suite y para importar correos electrónicos, calendarios, libretas de direcciones, tareas, contactos y la información de los usuarios de Exchange seleccionados.

Durante el proceso de migración se ejecutan las siguientes tareas:

- Crear las cuentas de usuarios en el servidor Zimbra (si estas no existen).

- Importar los correos, calendarios, contactos, tareas y usuarios.

Esta tarea puede realizarse a través de dos vías:

1. Exportar la información de cada cuenta dentro del servidor Microsoft Exchange hacia un fichero *.pst*, y luego utilizar el asistente de importación para Outlook que copia los datos desde dicho archivo hacia el servidor Zimbra.

2. Utilizar el asistente de migración para Microsoft Exchange, el cual usa un perfil MAPI para guiar la migración de las cuentas. La utilización de este asistente requiere de los siguientes requerimientos:

- Debe ser ejecutado desde una computadora con Microsoft Windows que tenga Microsoft Outlook instalado.
- Debe tener permisos de administrador sobre los buzones que desea importar.
- El usuario con el que se inició sesión en Windows debe tener permiso para abrir el perfil MAPI.
- Dicho usuario debe ser miembro del grupo Administradores.

- Crear un archivo de registro para verificar los errores de importación o advertencias.

Principales ventajas:

- Rentable.
- Basada en el navegador (con la opción de cliente de escritorio): Proporciona a los usuarios finales una interfaz basada en navegador que les permite conectarse sin problemas a sus nubes personales en cualquier dispositivo o plataforma.

- Conectividad con Microsoft Outlook: Permite a los usuarios continuar utilizando Microsoft Outlook, manteniendo todos los beneficios de Zimbra. También se sincroniza con los clientes Thunderbird.
- Gestión de contactos: Listas de direcciones personales y globales que se pueden compartir a través de toda la organización.

Principales desventajas:

- Zimbra necesita un servidor con altas prestaciones para su correcto funcionamiento.

Consumo de recursos

Para un sistema en producción se recomienda la siguiente configuración:

- CPU Intel/AMD de 32 bits a 2.0GHZ o superior.
- Mínimo 2 GB de RAM.
- 10 GB de disco duro para actualizaciones, logs y software, con RAID para redundancia de datos.
- Espacio de disco adicional para el almacenamiento de los buzones de correo y las bases de datos.

Para instalaciones de más de 2000 cuentas de usuario se recomienda CPU de 64 bits y una memoria RAM de 4 GB como mínimo.

1. 4. 2 Zentyal

Zentyal (anteriormente conocido como eBox Platform) es un servidor GNU/Linux para pequeñas y medianas empresas. Todos los servicios de red gestionados por Zentyal están estrechamente integrados, lo que permite automatizar la mayoría de las tareas. Esto contribuye a evitar errores en la configuración de

la red y permite ahorrar tiempo. Es un sistema de código abierto distribuido bajo la licencia GNU General Public License (GPL) [20].

Zentyal ofrece compatibilidad con Microsoft Exchange Server 2003 en términos de correo, contactos y calendarios. Integra OpenChange un reemplazo nativo para las tecnologías y protocolos presentes en Microsoft Exchange Server. Gracias a Openchange puede transferir información almacenada en un servidor Exchange, incluyendo usuarios, mensajes de correo electrónico, contactos, calendarios y tareas [20].

Arquitectura

Se puede obtener una visión del lugar que ocupa OpenChange en relación a los demás componentes de Zentyal, además de sus interacciones y protocolos a partir de la siguiente figura (Ver Figura3).

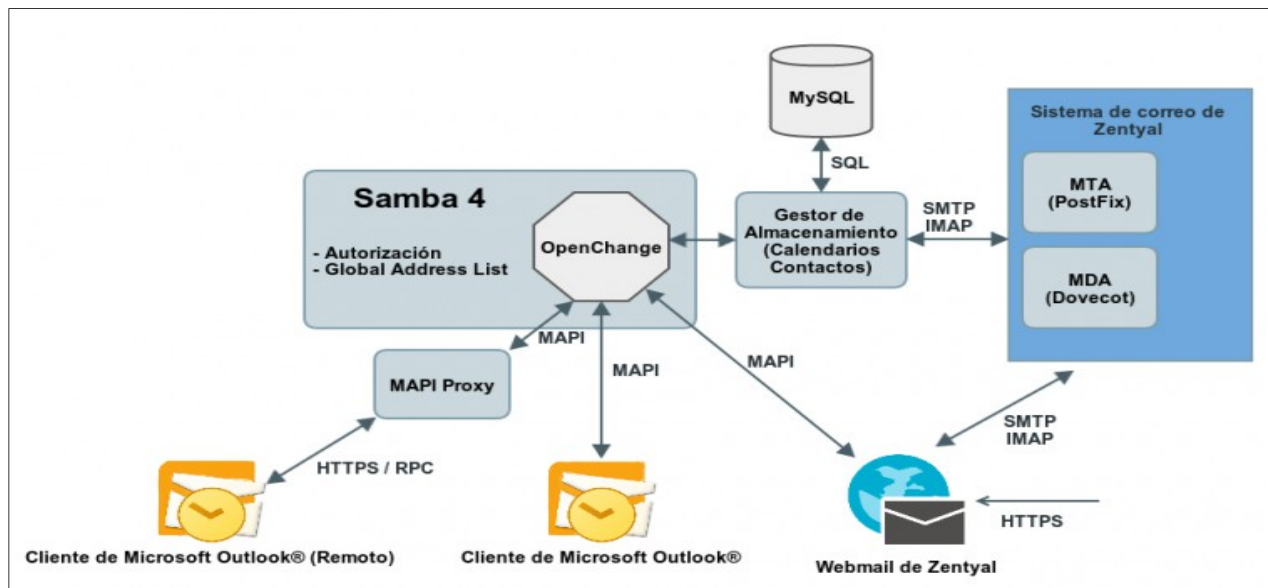


Figura 3: Componentes y protocolos que integra Zentyal.

Entre las tecnologías que emplea Zentyal para su funcionamiento se encuentran las siguientes:

- OpenLDAP como servidor de datos optimizado para la realización rápida de consultas de lectura y almacenamiento de datos de usuarios a modo de directorio.
- Zentyal usa Postfix como MTA para el envío/recepción de correos.

Principales ventajas:

- Permite gestionar todos los servicios de red, tales como acceso a Internet, seguridad de redes, intercambio de recursos, infraestructura de red o de comunicaciones de una manera fácil a través de una única plataforma.
- Ofrece una interfaz intuitiva que incluye las funciones más necesarias. Incorpora aplicaciones independientes que automatizan la mayoría de las tareas. Está diseñado para ahorrar tiempo de gestión de sistemas.

Principales desventajas:

- Administración local de los servicios.

Como todos los servicios pueden estar en una misma computadora si existe algún problema como por ejemplo desbordamiento de buffer y hay que reiniciar el servidor se reinician todos los servicios también.

Consumo de recursos

Los requerimientos de hardware para un servidor Zentyal dependen de los módulos que se instalen, de cuántos usuarios utilizan los servicios y de sus hábitos de uso. Algunos módulos tienen bajos

requerimientos, como Cortafuegos, DHCP o DNS, mientras que otros como el Filtrado de correo o antivirus necesitan más memoria RAM y CPU (Ver figura 4).

| PERFIL DE ZENTYAL | USUARIOS | CPU | MEMORIA | DISCO | TARJETAS DE RED |
|-----------------------|-----------|------------------------------|---------|-------|-----------------|
| Oficina | <50 | P4 o superior | 1G | 250G | 1 |
| | 50 ó más | Xeon Dual core o superior | 2G | 500G | 1 |
| Comunicaciones | <100 | Xeon Dual core o equivalente | 4G | 250G | 1 |
| | 100 ó más | Xeon Dual core o equivalente | 8G | 500G | 1 |

Figura 4: Consumo de recursos de Zentyal

El módulo **Oficina** incluye dominios y directorio de usuarios de la red local, ficheros, impresoras, calendarios, contactos, perfiles de usuarios y grupos, mientras que el módulo **Comunicaciones** incluye correo, mensajería instantánea y plataformas de trabajo en grupo.

1. 4. 3 Openchange

OpenChange es una implementación de código abierto del servidor Microsoft Exchange. Proporciona una solución para interoperar con clientes de Microsoft Outlook o servidores de Microsoft Exchange [21].

OpenChange consigue compatibilidad nativa con Microsoft Exchange debido a que implementa los mismos protocolos que los clientes de correo electrónico y trabajo en grupo: MAPI y, opcionalmente, ActiveSync. Estos protocolos gestionan no sólo el correo electrónico, sino también calendarios, listas de contactos y tareas [21].

El proyecto OpenChange tiene como objetivos fundamentales:

- Proporcionar una librería para la interoperabilidad con protocolos de Exchange.
- Proporcionar una alternativa a Microsoft Exchange Server que utiliza protocolos de Exchange.

Compatibilidad con Microsoft Exchange

OpenChange proporciona interoperabilidad con protocolos de Exchange a través del desarrollo de la librería MAPI (*libmapi*), la cual puede ser usada en los clientes de mensajería existentes y ofrecer la compatibilidad nativa con el servidor de Microsoft Exchange.

OpenChange implementa perfiles MAPI. Dichos perfiles son administrados a través de la interfaz *lprofAdmin*, suministrada en la librería *libmapi*, y se almacenan en una base de datos LDB², estos perfiles se utilizan para configurar una cuenta de Exchange en las estaciones de trabajo y obtener información necesaria del servidor Exchange, proceso que se utiliza en las comunicaciones Outlook-Exchange:

Información de conexión: dirección ip, nombre del Netbios, nombre del dominio Windows.

Credenciales: usuario y contraseña de la cuenta de Exchange.

Información del usuario: dirección de correo, base de datos de los buzones de correo.

Ejemplo de cómo trabaja la librería *libmapi* para conectarse con el servidor Exchange:

1. La primera función de la librería MAPI llamada es *MAPIInitialize()*. Como su nombre indica, esta función inicializa la librería MAPI: crea el contexto global MAPI, abre el almacén de la base de datos del perfil (ruta de base de datos pasada como parámetro de la función). Esta función debe ser ejecutada antes de cualquier otra operación de la librería.
2. Una vez que se inicializa MAPI se debe crear una conexión con Exchange y abrir la sesión MAPI con las credenciales de usuario. Este es el propósito de *MAPILogonEX()*. Esta función toma un puntero sobre una estructura *mapi_session*, el nombre de usuario del perfil y una contraseña.
3. Finalmente se llama *MAPIUnitialize()* antes de abandonar la función. Esta función va a limpiar la memoria asignada durante la sesión.

2 LDB: Los archivos *.ldb* son usados para determinar qué registros están bloqueados y por quien, en una base de datos compartida.

Limitaciones

- Sólo los usuarios dentro de la “OU³ Users” pueden tener cuentas de correo OpenChange.
- No se debe utilizar espacios vacíos para los nombres de usuario de Active Directory.
- No soporta carpetas públicas.

1. 4. 4 Zarafa

Zarafa es una aplicación de código abierto que se originó en la ciudad de Delft en Holanda. Proporciona almacenamiento de correo electrónico en el servidor y ofrece su propio cliente de correo basado en Ajax llamado WebAccess. Pretende ser una alternativa a Microsoft Exchange Server.

La herramienta de migración de Zarafa realiza la migración de cada usuario en el origen de datos (*exchange.pst*), migrando cada mensaje y carpeta. Esto significa que se transfiere toda la información del origen de datos, incluyendo correos, datos adjuntos, citas, tareas y contactos. La herramienta de migración Zarafa también comprueba si ya existe la *carpeta/mensaje* en el destino (un servidor Zarafa o un archivo *.pst*) para evitar duplicados [22].

Para exportar la información de los mensajes de correo almacenados en el servidor Exchange se emplea una herramienta desarrollada por Microsoft llamada *ExMerge*, la cual permite exportar dicha información hacia un fichero *.pst*, con el cual la herramienta de migración de Zarafa podrá importar los datos hacia su servidor.

Zarafa emplea las siguientes tecnologías dentro de su infraestructura:

- El servidor Zarafa puede obtener información de sus usuarios desde una base de datos OpenLDAP y Active Directory.

³ OU: Unidad organizativa

- Tiene la posibilidad de emplear otras tecnologías de código abierto como Postfix, Exim, Qmail y Sendmail, las cuales se usan como MTA para el envío/recepción de correos.

1. 5 Otras herramientas para migrar servidores IMAP.

Existen otras herramientas que son capaces de migrar buzones desde un servidor IMAP a otro, sin embargo, no migran usuarios, calendarios, tareas u otros elementos importantes que se almacenan en un servidor de correo, a continuación se mencionan algunas de estas herramientas:

1. 5. 1 ImapCopy

IMAPCopy es una herramienta de línea de comandos para copiar mensajes de varios usuarios de un servidor IMAP a otro. Para llevar a cabo este proceso se deben cumplir los siguientes pasos:

- Introducir la información relativa a la cuenta de correo que se quiere transferir.
- Introducir la información relativa a la nueva cuenta de correo.
- Una vez aceptada la sincronización el correo se copiará a la cuenta destino.

1. 5. 2 Imapsync

Imapsync es una herramienta de línea de comandos que permite transferencias incrementales y recursivas de correos IMAP de un buzón a otro, tanto en cualquier lugar de Internet o en su red local. Es incremental debido a que puede detener la transferencia en cualquier momento y reiniciarla más tarde de manera eficiente y recursiva ya que se puede copiar la jerarquía de carpetas completas [23].

Técnica utilizada para la comparación de las herramientas investigadas

Para establecer las comparaciones entre aplicaciones de código abierto son muy utilizados los modelos de madurez referentes a este tipo de aplicaciones. Para seleccionar los criterios de comparación entre las

herramientas analizadas se utiliza el método de calificación y selección de software Qualification and Selection of Open Source Software (QSOS). El proceso consiste en cuatro pasos: definición, evaluación, calificación y selección. A continuación se presenta una comparación entre las herramientas anteriormente descritas teniendo en cuenta los siguientes aspectos:

- Licencia de software: Contrato entre el licenciante y el licenciatario para utilizar el software según las condiciones establecidas por el primero.
- Rendimiento: Efectividad para utilizar de la forma más eficiente posible las prestaciones de la computadora. (RAM, disco duro, tarjetas de red, etc).
 - 0: Alto consumo de recursos.
 - 1: El consumo de recursos es mediano.
 - 2: Bajo consumo.
- Configuración: Facilidad de configuración del software.
 - 0: El software es difícil de configurar.
 - 1: Cuenta con niveles de configuración medios.
 - 2: Es una de las herramientas que con mayor facilidad se configura entre sus homólogas.
- Soporte: Respuesta de cambio del software ante nuevas necesidades o problemas que surjan.
 - 0: No cuenta con soporte.
 - 1: Se actualiza poco.
 - 2: Cuenta con un soporte activo.

Comparación entre las herramientas analizadas

Las aplicaciones estudiadas serán comparadas utilizando la técnica QSOS explicada anteriormente. Para seleccionar la herramienta adecuada para llevar a cabo el proceso de migración se tendrá en cuenta la que mayor puntaje obtenga, teniendo en cuenta los parámetros comparados.

| | Zimbra | Zentyal | OpenChange | Zarafa |
|-----------------------------|---|---|-------------------------------------|--|
| Licencia de Software | YPL⁴ Libre/ Código Abierto | GPL⁵ Libre/ Código Abierto | GPL Libre/ Código Abierto | AGPL⁶ Libre/ Código Abierto |
| Configuración | 1 | 1 | 1 | 1 |
| Consumo de recursos | 0 | 1 | 2 | 1 |
| Soporte | 2 | 2 | 2 | 2 |
| Total | 3 | 4 | 5 | 4 |

Tabla 1: Comparación entre las herramientas.

En el proceso de migración del servidor Microsoft Exchange 2003 hacia un servidor de correo electrónico en GNU/Linux es de vital importancia la transferencia de elementos como usuarios, alias, correos, calendarios, listas de correo, contactos, tareas y configuraciones propias del servidor. A continuación se realiza una comparación entre las tecnologías analizadas con el objetivo de determinar la herramienta que posibilite migrar el mayor número de elementos.

4 YPL: Yahoo Public Licence.

5 GPL: General Public Licence.

6 AGPL: Affero General Public Licence.

| | Zimbra | Zentyal | Openchange | Zarafa |
|-------------------------|--------|---------|------------|--------|
| Usuarios | ✓ | ✓ | ✓ | |
| Alias | | | | |
| Correos | ✓ | ✓ | ✓ | ✓ |
| Calendarios | ✓ | ✓ | ✓ | ✓ |
| Listas de correo | | | | |
| Contactos | ✓ | ✓ | ✓ | ✓ |
| Tareas | ✓ | ✓ | ✓ | ✓ |
| Configuraciones | | | | |

Tabla 2: Comparación entre las herramientas.

Luego de realizar una comparación entre las tecnologías estudiadas y analizar el mecanismo que emplean para migrar el servidor Exchange 2003, se concluye que la más adecuada para guiar el diseño del módulo de migración es Openchange debido a las siguientes razones:

- Zimbra es una herramienta que consume gran cantidad de recursos. Utiliza un asistente para la migración que requiere que los buzones del servidor Exchange sean exportados a un fichero *.pst* para después ser importados desde el servidor Zimbra.
- Zarafa emplea un asistente de migración que importa los datos almacenados en un fichero *.pst* hacia su servidor. Para exportar la información almacenada en Exchange hacia el fichero *.pst* es utilizada la herramienta ExMerge, la cual es privativa.

Una de las cuestiones más importantes en el desarrollo de un producto de software es sin duda la elección de la metodología de desarrollo de software. Su uso está orientado a la estructuración, planificación y control del proceso de desarrollo.

La metodología ágil SXP es la unión de XP y SCRUM. XP (Programación Extrema por sus siglas en inglés) es el más destacado de los procesos ágiles de desarrollo de software. Su particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para alcanzar el éxito del proyecto. Por su parte SCRUM es también una metodología ágil y está basada en un proceso iterativo e incremental. SXP fue desarrollada en el 2007 en la UCI. Está especialmente indicada para proyectos pequeños, con rápidos cambios de requisitos donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad [24].

- Openchange implementa el protocolo MAPI a través de la librería libmapi, la cual es usada para conectarse al servidor Exchange y acceder a toda la información almacenada en el mismo. Posibilita transferir información referente a mensajes, contactos, calendarios y tareas.

1.6 Metodología de desarrollo de software

SXP está dirigida a:

- Proyectos compuestos por pequeños equipos de trabajo.
- Proyectos con probabilidades de rápidos cambios de requisitos o requisitos imprecisos.
- Proyectos donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad.

SXP cuenta con cuatro fases:

- Planificación-Definición.

En esta fase se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.

➤ Desarrollo.

Es la fase donde se realiza la implementación del sistema hasta que esté listo para ser entregado.

➤ Entrega.

Fase donde se pone en marcha el producto desarrollado y se hace la entrega al cliente.

➤ Mantenimiento.

Esta fase está dirigida a garantizar el soporte para el cliente.

1. 7 Herramientas y tecnologías de modelado

El proceso de desarrollo de software se apoya en el uso de diferentes herramientas y tecnologías, las cuales conforman el ambiente de desarrollo de un sistema. A continuación se realiza un estudio sobre herramientas y tecnologías de diseño actuales, lo que permitirá seleccionar aquellas que apoyarán el diseño del componente.

1. 7. 1 Lenguaje para el modelado

El modelado de un software es una parte de vital importancia en el proceso de desarrollo del mismo, pues proporciona un conjunto de herramientas y artefactos que permiten a los desarrolladores tener una perspectiva visual del sistema.

UML

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, medios y dominios de aplicación.

Está dirigido a ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos [25].

BPMN

Modelo y Notación de Procesos de Negocio es un lenguaje de modelado de procesos de negocio que permite representar gráficamente las diferentes etapas del proceso del mismo. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes. Mientras que UML toma un perfil orientado a objetos enfocado al diseño de software, BPMN toma un perfil orientado a procesos de negocio, por lo que ambas notaciones son totalmente compatibles entre sí [26].

1. 7. 2 Herramientas para el diseño

Las herramientas CASE (Ingeniería de Software Asistida por Computadora) constituyen aplicaciones informáticas dirigidas a aumentar la productividad en el desarrollo de software, reduciendo el costo del mismo en términos de tiempo y de dinero. Estas herramientas pueden soportar todos los aspectos del ciclo de vida de desarrollo del software en tareas como diseño del proyecto, cálculo de costos, implementación de parte del código de forma automática a partir del diseño, compilación automática, documentación o detección de errores [27].

Visual Paradigm

Visual Paradigm es una herramienta CASE para el modelado de lenguaje UML que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML contribuye a una rápida construcción de aplicaciones de mejor

calidad y menor coste. Permite construir todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación [28].

Conclusiones parciales

A partir del estudio realizado sobre el proceso de migración del servidor de correo Microsoft Exchange en su versión 2003 hacia servidores de correo electrónico en GNU/Linux, se seleccionó OpenChange como herramienta guía para el diseño del módulo definiendo la librería libmapi como factor fundamental en este proceso, la cual permite transferir datos asociados a las cuentas de correo como mensajes, contactos, calendarios y tareas, favoreciendo la automatización del proceso de migración.

Capítulo 2: Análisis y diseño

En el proceso de desarrollo de software es de vital importancia el correcto análisis y diseño del producto que permita identificar y recopilar los requisitos del mismo. En el desarrollo de este capítulo se describen las Historias de Usuario del módulo. Además, se expone la arquitectura y los patrones de diseño que se utilizarán.

2. 1 Modelo del dominio.

Un modelo del dominio es una representación de las clases conceptuales del mundo real. Es utilizado con frecuencia como fuente de inspiración para el diseño de los objetos software. El modelo del dominio muestra (a los modeladores) clases conceptuales significativas en un dominio del problema [29].

Para transferir la información de un servidor de correo electrónico Microsoft Exchange 2003 hacia un servidor de correo electrónico en GNU/Linux pueden ser utilizadas varias herramientas de migración. Dichas herramientas piden al cliente la información necesaria para crear una conexión con el servidor Exchange, si la información es correcta accede a dicho servidor de forma satisfactoria, en caso contrario envía un mensaje de error al usuario. Luego el servidor Exchange accede a la información de las cuentas de correo almacenada en una base de datos *.edb*. Para finalizar el proceso la herramienta de migración extrae la información referente a las cuentas, según las características de la herramienta en cuestión, desde el servidor Exchange y la transfiere hacia el servidor de correo de GNU/ Linux (*Ver figura 5*).

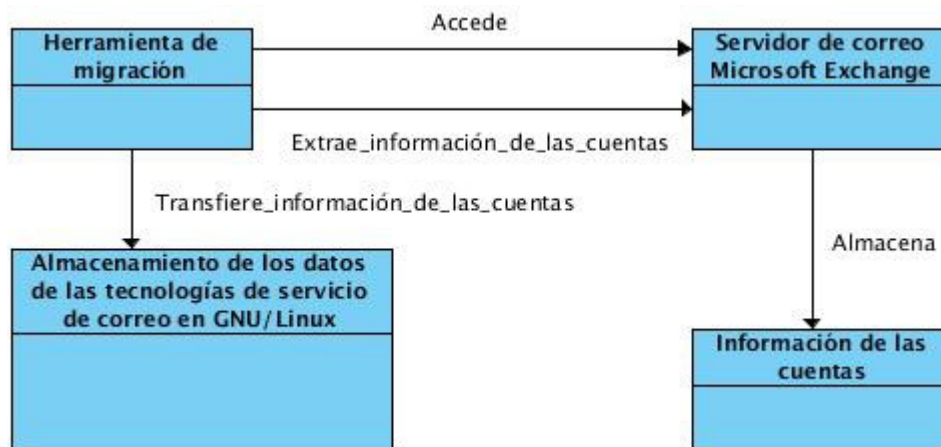


Figura 5: Proceso de migración del servidor Exchange 2003.

- **Herramienta de migración:** Es la herramienta utilizada para transferir la información desde un servidor hacia otro (por ejemplo Zimbra, Openchange, Zarafa).
- **Servidor de correo Microsoft Exchange 2003:** Es el servidor de correo electrónico privativo, del cual se desea transferir la información de las cuentas de correo.
- **Información de las cuentas:** Se refiere a los datos asociados a las cuentas de correo electrónico del servidor Exchange referente a mensajes, contactos, calendarios tareas y usuarios.
- **Almacenamiento de los datos de las tecnologías del servicio de correo en GNU/Linux:** Es la estructura de los buzones de las cuentas de correo en el servidor en GNU/Linux donde se almacena la información referente a dichas cuentas.

2. 2 Propuesta del módulo de migración

Las dificultades que trae consigo el proceso de migración del servicio de correo electrónico, sumado a la carencia en HMAST de un mecanismo que permita transferir la información asociada a un servidor de correo electrónico Exchange 2003 a un servidor de correo electrónico en GNU/Linux, da lugar a la necesidad de desarrollar un módulo para dicha herramienta que permita llevar a cabo este proceso.

El presente trabajo propone el diseño de un módulo para la herramienta HMAST que permita migrar la información almacenada en un servidor de correo electrónico Exchange 2003. El módulo contará con las funcionalidades necesarias para transferir los datos asociados a las cuentas de correo, dígame usuarios, contactos, mensajes, calendarios, tareas y configuraciones propias del servidor. Dicho módulo usará una estrategia similar a la herramienta OpenChange, donde la librería libmapi juega un papel fundamental. El proceso seguirá los siguientes pasos:

- 1- El módulo se conecta al servidor de correo Exchange en su versión 2003 a través del uso de perfiles MAPI creados previamente por el especialista.
- 2- Haciendo uso de dicha librería se accede a la información de las cuentas de correo electrónico, dicha información es almacenada en bases de datos *.edb*. De cada cuenta se extrae la información referente a correos, contactos, tareas y calendarios.
- 3- Se crean en el servidor GNU/Linux cuentas para cada uno de los usuarios almacenados en el servidor Exchange. Antes de crear las cuentas se verifica su previa existencia para evitar duplicados.
- 4- Se transfiere la información de las cuentas de Exchange hacia el servidor GNU/Linux.
- 5- Se obtienen las configuraciones del servidor Exchange.
- 6- Se editan los ficheros de configuración de los servicios (Postfix, Dovecot, etc) según las configuraciones obtenidas del servidor Exchange.

Para obtener las configuraciones del servidor Exchange se emplearán dos vías. En primer lugar la introducción del comando `get-mailboxstatistics | ft displayname, totalitemsize > c:\buzones.txt` en la consola de administración del servicio, en el servidor privativo, permite obtener la capacidad consumida por cada buzón y generar un fichero de texto con dicha información. En segundo lugar, partiendo de la imposibilidad de obtener otras configuraciones de forma automatizada, se hará uso de encuestas solicitando estos datos al administrador del servicio en la entidad en cuestión. En la tabla 3 se muestra el modelo de la encuesta.

| Encuesta de administración del servidor de correo Exchange 2003 | | |
|---|---|-----------------------------|
| Aplica: //nombre del administrador del servidor | | Fecha: |
| Solicitud de datos: | | |
| Tamaño de los buzones: | // tamaño máximo de los buzones | |
| Ruta de almacenamiento de los buzones en el servidor GNU/Linux: | // ruta donde se almacenarán los buzones de los usuarios en el servidor Linux | |
| Redes permitidas: | // redes con permiso de acceso al servidor | |
| Permitir autenticación SSL: | Sí <input type="checkbox"/> | No <input type="checkbox"/> |

Tabla 3: Encuesta para obtener configuraciones de servidor.

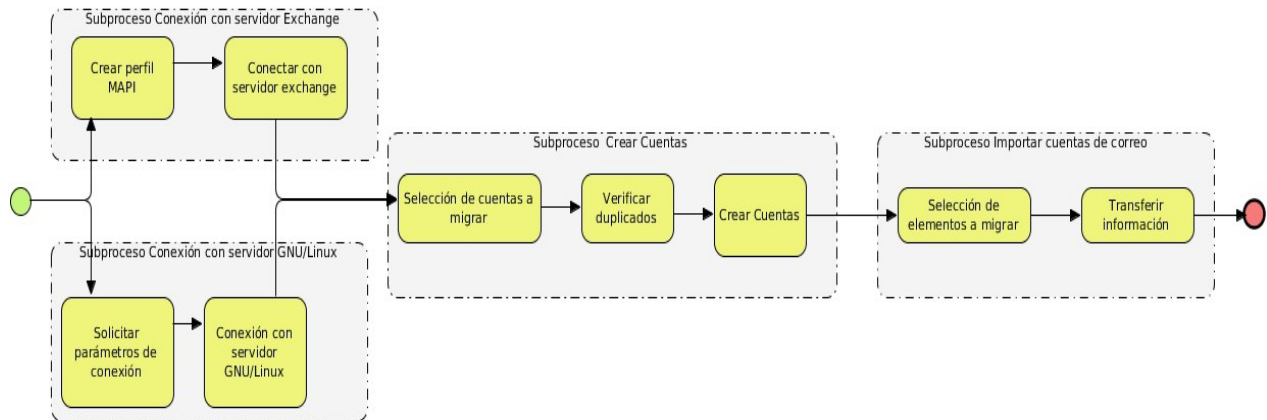


Figura 6: Diagrama de procesos de negocio.

2.3 Captura de Requisitos

La captura de requisitos es uno de los pasos fundamentales en el desarrollo de cualquier software. Esta tarea está encaminada a identificar lo que el cliente quiere, analizar las necesidades y especificar los requerimientos de la solución sin ambigüedades.

Los requisitos funcionales describen las funciones que el software va a ejecutar, se conocen también como capacidades. Los requisitos no funcionales son los que actúan para obligar la solución. Los requisitos no funcionales se conocen a veces como apremios o requisitos de calidad [30].

2.3.1 Requisitos funcionales

A continuación se muestran los requerimientos funcionales identificados en el módulo de migración a través de la Lista de Reserva del Producto (LRP).

| Prioridad | Ítem * | Descripción | Estimación | Estimado por |
|----------------------------|--------|---|------------|----------------------|
| Prioridad: Muy Alta | | | | |
| | 1 | Crear cuenta de usuario | 0.1 | Delvis Mesa Chaviano |
| | 2 | Importar correos | 0.1 | Delvis Mesa Chaviano |
| | 3 | Importar calendarios | 0.1 | Delvis Mesa Chaviano |
| | 4 | Importar tareas | 0.1 | Delvis Mesa Chaviano |
| | 5 | Importar contactos | 0.1 | Delvis Mesa Chaviano |
| Prioridad: Alta | | | | |
| | 6 | Habilitar SSL para la autenticación en el servidor | 0.1 | Delvis Mesa Chaviano |
| | 7 | Especificar datos del servidor destino | 0.1 | Delvis Mesa Chaviano |
| | 8 | Especificar datos del servidor Exchange | 0.1 | Delvis Mesa Chaviano |
| | 9 | Especificar datos del usuario para crear el perfil MAPI | 0.1 | Delvis Mesa Chaviano |
| | 10 | Crear perfil MAPI | 0.1 | Delvis Mesa Chaviano |
| | 11 | Especificar número de buzones a migrar simultáneamente | 0.1 | Delvis Mesa Chaviano |
| | 12 | Especificar dirección de almacenamiento de los buzones | 0.1 | Delvis Mesa Chaviano |
| | 13 | Determinar tamaño de los buzones de usuarios | 0.1 | Delvis Mesa Chaviano |

| | | | |
|----|---|-----|----------------------|
| 14 | Verificar si las cuentas existen en el servidor destino | 0.1 | Delvis Mesa Chaviano |
| 15 | Importar carpetas personales | 0.1 | Delvis Mesa Chaviano |
| 16 | Crear registro para la solución de errores | 0.1 | Delvis Mesa Chaviano |
| 17 | Mostrar notificaciones del estado de la importación | 0.1 | Delvis Mesa Chaviano |
| 18 | Listar perfiles MAPI existentes. | 0.1 | Delvis Mesa Chaviano |
| 19 | Establecer contraseñas por defecto | 0.1 | Delvis Mesa Chaviano |

Tabla 4: Lista de Reserva del Producto del módulo.

2. 3. 2 Requisitos no funcionales (RNF)

Software:

- La aplicación se va a ejecutar en sistemas operativos libres.
- Se debe tener instalado Windows Server 2003 SP2 como sistema operativo, en el cual estará alojado el servidor de correo electrónico Exchange Server 2003.

Hardware:

- Estación de trabajo con 512 MB mínimo de RAM.

Restricciones en el diseño y en la implementación:

- Usar como lenguaje de programación java con el framework de desarrollo Spring.
- Utilizar como entorno de desarrollo integrado Netbeans en su versión 8.0.

- El módulo se debe ajustar a la arquitectura definida en la herramienta HMAST.
- Debe ser desarrollado haciendo uso de tecnologías libres.

2.4 Historias de usuario (HU).

Las HU constituyen la técnica utilizada en SXP para especificar los requisitos del software.

A continuación se muestran una de las historias de usuarios del módulo de migración.

| Historia de Usuario | |
|--|--|
| Número: HMAST_Migración_HU_4 | Nombre Historia de Usuario: Importar cuentas de correo. |
| Modificación de Historia de Usuario Número: Ninguna | |
| Usuario: Delvis Mesa Chaviano | Iteración Asignada: 1 |
| Prioridad en Negocio: Alta | Puntos Estimados: 0.1 semana |
| Riesgo en Desarrollo: Muy Alto | Puntos Reales: 0.1 semana |
| Descripción: EL software permite importar la información asociada a las cuentas de correo desde el servidor Exchange hacia el servidor administrado por HMAST. | |
| Observaciones: La aplicación le permitirá al usuario seleccionar qué información desea migrar desde las cuentas alojadas en el servidor Exchange, luego se conectará al servidor a través de un perfil MAPI creado con anterioridad e importará toda la información seleccionada. | |
| Los datos que pueden ser seleccionados para importar a través de un <i>checkbox</i> son: | |
| <ul style="list-style-type: none"> • Correos • Contactos • Tareas • Calendarios | |

- Configuraciones del servidor

Para transferir la información de las cuentas se emplean los siguientes comandos:

--fetchmail: recupera los correos de los usuarios de Exchange.

--storemail=/dirección: almacena los datos adjuntos en la dirección especificada.

--fetch-items=STRING: recupera elementos específicos de las carpetas de Exchange.

STRING puede ser correo, cita, contacto, tarea o nota.

La información referente a los mensajes de correo electrónico transferidos es exportada a un fichero mbox a través del comando: **exchange2mbox --database=/dirección de la base de datos de perfiles --profile=/nombre del perfil --update**, este fichero no puede ser utilizado por la infraestructura de correo de HMAST debido a problemas de compatibilidad por lo que debe ser convertido al formato maildir el cual es empleado por la herramienta antes mencionada, para llevar a cabo el proceso de conversión se emplea una aplicación de líneas de comandos llamada mb2md, la cual emplea el siguiente comando para convertir de un formato a otro:

mb2md -s /dirección del archivo mbox -R -d /dirección destino

La información referente a los calendarios y tareas es exportada a un fichero con formato ical a través del comando: **exchange2ical --database=/dirección de la base de datos de perfiles --profile=/nombre del perfil --icalsync=/fichero.ical**. Dicho formato es compatible con HMAST.

Para transferir las configuraciones del servidor el usuario deberá completar un cuestionario con la información necesaria para transferir dichas configuraciones, las cuales serán introducidas en el servidor Linux. Los datos que se deben introducir son los siguientes:

Tamaño de los mensajes: es un valor entero que se escribe en bytes y hace referencia al tamaño máximo que podrán tener los mensajes enviados.

Ejemplo: mailbox_size_limit="valor"

Redes permitidas: se pueden introducir dominios, direcciones IP, etc.

Ejemplo: mynetworks = 127.0.0.1/8, 10.53.3.125, smtp.uci.cu

Ruta de almacenamiento de los buzones: se refiere a la dirección donde se guardaran los buzones que serán migrados.



Ejemplo: virtual_mailbox_base = /var/vmail

Luego de ser seleccionada la información que se desea migrar se selecciona el botón *Aceptar*, la aplicación muestra un mensaje notificando el estado de la importación: si la misma fue completada correctamente o hubo algún fallo durante el proceso. En caso de presentarse un fallo el mismo se registra en un fichero de registro (log).

Prototipo de interfaz:

| | | | | | |
|-------------------|----------------------|--------------------|-------------------|---------------------|-------------------------|
| DNS 196.2.1.23 | Correo 196.2.1.23 | DHCP 196.2.1.23 | Web 196.2.1.23 | Proxy 196.2.1.23 | Migración 196.2.1.23 |
|-------------------|----------------------|--------------------|-------------------|---------------------|-------------------------|

Módulo para la migración del servidor de correo electrónico Microsoft Exchange.

Importar cuentas de correo
 Seleccione cuáles cuentas desea migrar  

Seleccione las cuentas de usuario que desea migrar desde el servidor Exchange.

| Cuentas Exchange | Cuentas HMAST |
|-----------------------|---------------------|
| dmesa@exchange.local | No existe |
| delvis@exchange.local | delvis@hmast.uci.cu |
| gisel@exchange.local | gisel@hmast.uci.cu |
| osvel@exchange.local | osvel@hmast.uci.cu |

Buscar cuenta de correo electrónico

Nombre v

Dirección de correo v Buscar

< Anterior
Siguiente >
Cancelar ✘
Ayuda ?

2. 5 Herramientas y tecnologías a utilizar

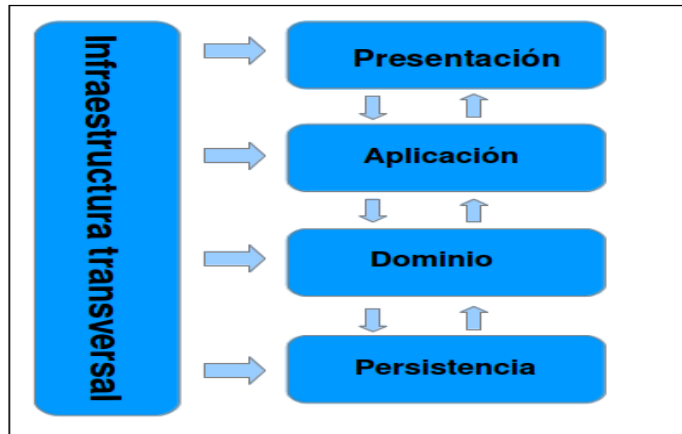
Para que el módulo de migración pueda integrarse correctamente a la herramienta HMAST se proponen como tecnologías para su desarrollo las siguientes:

- ✓ El módulo debe ser implementado usando el lenguaje de programación java.
- ✓ Como marco de trabajo se recomienda emplear Spring framework.
- ✓ Se propone utilizar Netbeans como IDE de desarrollo para la implementación.
- ✓ El modelado del software se realice usando los lenguajes UML y BPMN para especificar, construir y definir de forma gráfica y documental el diseño de la solución.
- ✓ La herramienta CASE a utilizar para el proceso de modelado sea Visual Paradigm Enterprise Edition ya que esta constituye un software de alta eficiencia que permite realizar ingeniería tanto inversa como directa, es de software libre y permite la generación de documentación de forma automática en diferentes formatos.

2. 6 Arquitectura del módulo

La descripción de la arquitectura es un elemento necesario para el desarrollo de un software pues permite una mejor comunicación entre los clientes y los desarrolladores.

Para lograr una completa integración del módulo con la herramienta HMAST se propone utilizar una arquitectura distribuida en 5 capas: capa de presentación, capa de aplicación, capa de dominio, capa de persistencia y la capa Infraestructura transversal (Ver Figura 7).



2.7 Diagrama de paquetes

El diagrama de paquetes muestra una distribución de los subpaquetes y clases del módulo, este contiene las relaciones entre ellas, así como los atributos y métodos que componen cada una.

El diagrama de la Figura 7 representa la distribución de las clases en el módulo mailMigration. A continuación se describe el diagrama con el objetivo de comprender como ocurre la interacción entre las capas.

En la capa de aplicación (application) se encuentra la clase interfaz IExchangeMailMigrationAppService que contiene los métodos que permiten el acceso desde la capa de presentación, los cuales son implementados por la clase ExchangeMailMigrationAppService que es responsable de adaptar la información que le llega desde la interfaz de usuario a los requerimientos de los servicios del dominio (domain). Estos servicios son accedidos a través del atributo iExchangeMailMigrationAppService que contiene esta clase, en esta capa se encuentra el paquete DTO (Data Object Transfer) que contiene clases que representan la información que llega desde la capa de presentación.

En la capa de dominio (domain) existen tres subpaquetes: entitys, service y repositorysContracts. En el subpaquete Entitys se encuentra las clases DataMigration, MapiProfiles, ConfigurationService y ExchangeMigration. En el subpaquete Service se encuentra la clase IExchangeMailMigrationService que contiene métodos que son accedidos desde la capa de aplicación, los cuales son implementados por la clase ExchangeMailMigrationService que es la responsable de realizar las validaciones de los datos antes de realizar la escritura o lectura del repositorio. En el subpaquete repositorysContracts se encuentra la clase IExchangeMailMigrationRepository que tiene definidos los métodos que son accedidos desde el subpaquete Service, los cuales son implementados por la clase ExchangeMailMigrationRepository encontrada en el paquete Persistens, esta clase es responsable de cargar y salvar la información en los repositorios.

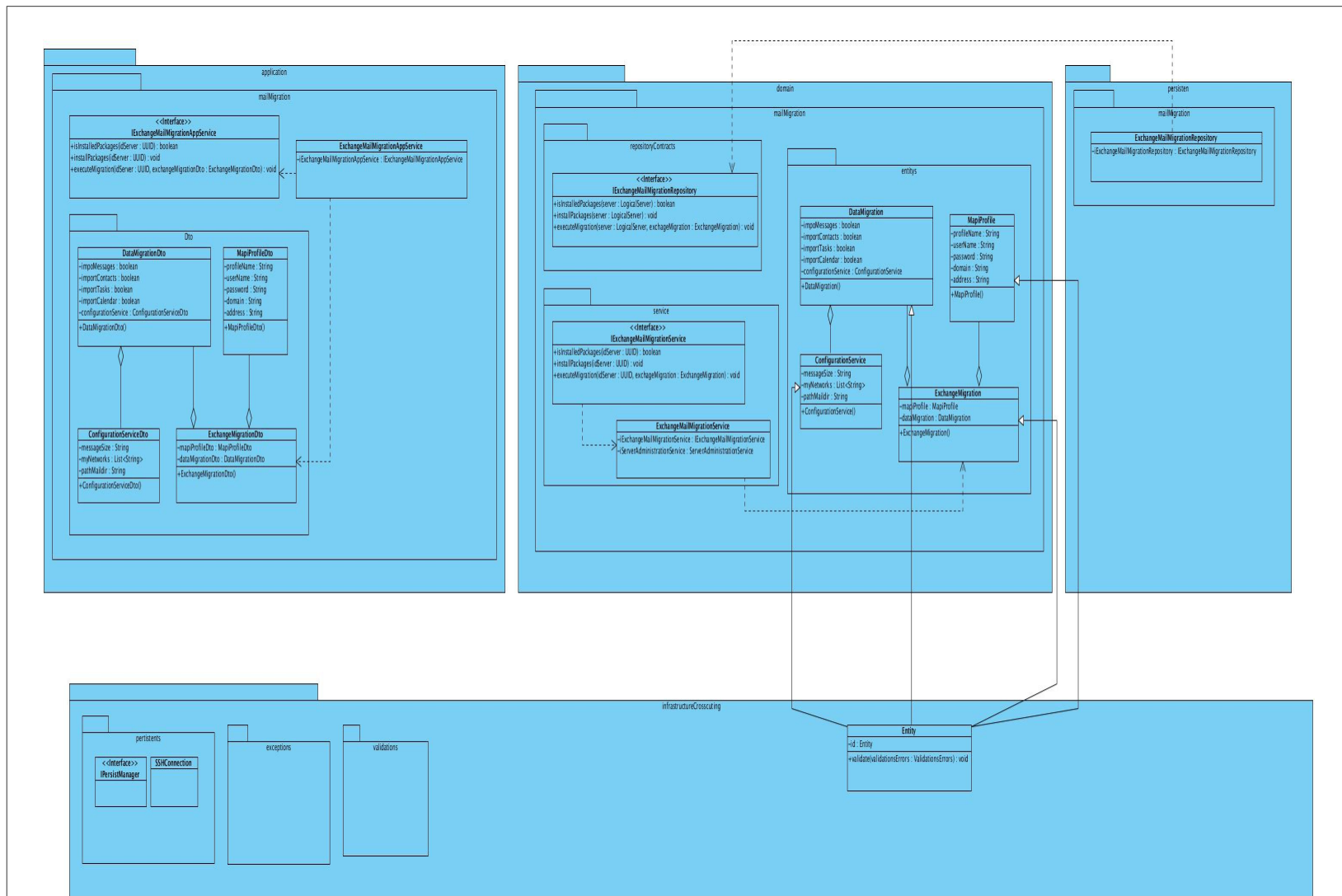


Figura 8: Diagrama de paquetes del módulo migración.

2. 8 Patrones de diseño

Una de las cuestiones más complicadas en la programación orientada a objetos consiste en elegir las clases adecuadas y cómo estas clases deben interactuar, así como elegir cuidadosamente las responsabilidades de cada clase.

Un patrón de diseño es una descripción de la comunicación personalizada entre objetos y clases para resolver un problema de diseño general en un contexto particular. Identifica clases, instancias, roles, colaboraciones y la distribución de responsabilidades [31]. Se presentan como pares de problema-solución con nombre, sugiriendo aspectos relacionados con la asignación de responsabilidades.

Los patrones GRASP (Patrones Generales de Software para Asignar Responsabilidades) representan los principios básicos de la asignación de responsabilidades a objetos, expresados en forma de patrones.

Patrón Experto: se usa más que cualquier otro al asignar responsabilidades, es un principio básico que suele utilizarse en el diseño orientado a objetos [31]. Consiste en la asignación de una responsabilidad a la clase que cuenta con la información necesaria para llevarla a cabo. El uso de este patrón da pie a un bajo acoplamiento y una alta cohesión, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento. Con la utilización de este patrón, se hizo posible definir dónde colocar en cada clase del módulo las funcionalidades que necesitan de esa información, dicha clase sería el experto en información.

Patrón Bajo Acoplamiento: el acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras [31]. El bajo acoplamiento soporta el diseño de clases más independientes y reutilizables, lo cual reduce el impacto de los cambios y acrecienta la oportunidad de una mayor productividad.

En el módulo propuesto el bajo acoplamiento se logra mediante el uso de interfaces utilizando inyecciones de dependencia. En este caso, cada vez que se necesite instanciar un objeto para ser llamado desde otras clases o capas.

Patrón Alta Cohesión: en la perspectiva del diseño orientado a objetos, la cohesión (o más exactamente, la cohesión funcional) es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme, clases con responsabilidades moderadas en un área funcional que colaboran con las otras para llevar a cabo las tareas [31]. Este patrón es utilizado para garantizar que las clases no contengan una amplia cantidad de funcionalidades, siguiendo el propósito de distribuir de forma equitativa el peso de la complejidad de las mismas.

Creador: El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. Este patrón es utilizado en la creación de las instancias de las clases.

La utilización de este patrón es evidenciado cuando las clases presentes en la capa Aplicación crean instancias de las entidades para la conversión de los objetos DTO a objetos del dominio. Un ejemplo es cuando la clase ExchangeMailMigrationAppService crea un nuevo objeto de tipo ExchangeMailMigration para enviarlo a la capa de dominio para su posterior validación.

Patrones GoF

Los patrones de diseño, conocidos como GoF se clasifican en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento. A continuación, es detallado el que ha sido utilizado en el diseño de la solución.

Patrón Solitario (Singleton): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Es usado debido a la necesidad de trabajar con el

mismo objeto en distintos momentos. Este patrón se evidencia en la conexión a un servidor ya que con una única instancia de este objeto se realizan todas las operaciones sobre el servidor.

Conclusiones parciales

El uso de SXP como metodología de desarrollo de software permitió generar artefactos del diseño que sientan las bases para una futura implementación del módulo propuesto, tales como la LRP, Historias de Usuario y el diagrama de paquetes. Fue propuesta la arquitectura a emplear en el desarrollo del módulo, así como los patrones de diseño a utilizar.

Capítulo 3: Pruebas

En todo proceso de diseño de software es importante realizar una correcta validación con el objetivo de mejorar la calidad de los productos de trabajo generados durante el desarrollo del software. Se debe corregir todo posible fallo y alcanzar cierto grado de perfección, así mismo se debe garantizar la confiabilidad y utilidad del software.

3.1 Pautas de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, se debe establecer un estándar de codificación para asegurar que todos los programadores del proyecto trabajen de forma coordinada [32]. Las técnicas de codificación incorporan muchos aspectos del desarrollo del software. Aunque generalmente no afectan a la funcionalidad de la aplicación, sí contribuyen a una mejor comprensión del código fuente [33].

Para el desarrollo del módulo se debe usar el Estándar de Codificación [34], el cual guía la codificación de HMAST. Dicho estándar contempla los siguientes elementos:

➤ **Asignación de nombres.**

➤ **Ficheros de código fuente.**

Cada fichero contiene una única clase o interfaz. Si hay una clase privada o una interfaz asociada a una clase pública se puede poner en el mismo fichero. La clase pública debe ser la primera.

➤ **Indentación.**

La unidad de indentación de bloques de sentencias son 4 espacios.

➤ **Longitud de la línea.**

Limitar todas las líneas a un máximo de 120 caracteres.

➤ **Comentarios.**

Los comentarios deben añadir claridad al código. Deben contar el por qué y no el cómo. Deben ser concisos.

➤ **Declaraciones**

Se debe declarar cada variable en una línea distinta, de esta forma cada variable se puede comentar por separado.

➤ **Espacio en blanco.**

➤ **Sentencias.**

➤ **Continuidad de las líneas largas.**

Cuando una sentencia no quepa en una única línea se debe fraccionar atendiendo a estos principios generales:

- Fraccionar después de una coma.
- Fraccionar después de un operador.
- Alinear la nueva línea con el comienzo de la expresión al mismo nivel de la línea anterior.

3.2 Pruebas

La prueba del software es un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. Con el objetivo de validar la

propuesta de solución se aplican pruebas estáticas para validar el diseño y se utiliza una aplicación de prueba para validar el funcionamiento de la estrategia de migración.

3. 2. 1 Pruebas a través de Técnicas Estáticas

Las técnicas de evaluación estática de artefactos del desarrollo o revisiones pretenden detectar manualmente (sea requisito, diseño, código) defectos en cualquier producto. Los revisores analizan ese producto mediante la lectura del mismo sin ejecutarlo [35]. Consiste en realizar revisiones y análisis estáticos a la documentación del proyecto. Dentro de las revisiones se pueden abordar especificaciones de requerimientos y de prueba, guías de usuario, entre otros.

Tipos de técnicas estáticas

- **Revisión informal:** Es un proceso donde se analiza un producto de software usando una técnica de lectura con el propósito de detectar defectos antes de la etapa de pruebas.
- **Revisiones formales o inspección:** Consiste en examinar en detalle los requisitos de software, diseño o codificación para detectar defectos o disconformidades con las normas de desarrollo.
- **Auditoría:** Contrastan los artefactos generados durante el desarrollo con estándares generales o de la organización. Pretenden comprobar formatos de documentos e inclusión de toda la información necesaria. Es decir, se tratan de gestión o administración del proyecto.

Para validar el diseño propuesto se utilizaron técnicas de tipo *Revisión Formal* para examinar los requisitos de software, y de tipo *Auditoría* para comprobar la correspondencia de la documentación generada con los estándares de la institución.

3. 2. 2 Validación de Requisitos

La validación de requisitos se realiza con el objetivo de demostrar que la definición de los requerimientos define realmente el sistema que el usuario necesita o el cliente desea. El proceso de validación de

requisitos comprende actividades que generalmente se realizan una vez obtenida una primera versión de la documentación de requisitos.

Los parámetros a comprobar por la especificación son:

1. **Validez:** No basta con preguntar a un usuario, todos los potenciales usuarios pueden tener puntos de vista distintos y necesitar otros requisitos.
2. **Consistencia:** No debe haber contradicciones entre unos requisitos y otros.
3. **Completitud:** Deben estar todos los requisitos.
4. **Verificabilidad:** Tiene que existir alguna forma de comprobar que cada requisito se cumple.

Técnicas de Validación de Requisitos

Los requisitos una vez definidos necesitan ser validados. Es necesario asegurar que el análisis realizado y los resultados obtenidos de la etapa de definición de requisitos son correctos. Para llevar a cabo este proceso se emplean las siguientes técnicas de validación:

- **Reviews o Walk-throughs:** Esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida.
- **Matrices de trazabilidad:** Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos.
- **Prototipos:** Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario.

3. 2. 3 Pruebas con la aplicación experimental

La aplicación de prueba fue implementada utilizando el lenguaje de programación Java a través de NetBeans IDE. Cuenta con una interfaz principal que guía el proceso de migración y hace uso de una base de datos de perfiles MAPI.

Para realizar las pruebas haciendo uso de la aplicación desarrollada se deben tener en cuenta una serie de pre-requisitos, los cuales son de vital importancia para el correcto funcionamiento de la misma.

Escenario de pruebas

Windows

Se debe tener instalado Windows Server 2003 SP2 como sistema operativo, en el cual estará alojado el servidor de correo electrónico Exchange Server 2003. Debe estar instalado y configurado un servidor DNS para establecer la conexión entre la aplicación y el servidor Exchange.

Configuraciones del servidor DNS:

- 1- Crear una zona de búsqueda directa, la cual permitirá hacer consultas al DNS, en las cuales proporcionando un host DNS previamente registrado devolverá en qué IP está alojado. En dicha zona se deberá crear un registro de tipo A y asociarlo a la dirección IP del servidor.
- 2- Crear una zona de búsqueda inversa permitirá generar registros PTR (Registro de Recursos de Punteros), los cuales permitirán asociar una dirección IP a un nombre.

GNU/ Linux

Para que la aplicación funcione correctamente en el sistema operativo GNU/ Linux se deben tener instalados los paquetes libmapi-dev y openchangeclient, los cuales son empleados para establecer la conexión al servidor Exchange y realizar las tareas deseadas.

Se debe configurar la red en la estación de trabajo para garantizar la conexión al servidor Windows, para esto en el campo **servidores DNS** se debe agregar la dirección IP de dicho servidor.

3. 2. 4 Pruebas con la herramienta experimental.

Las pruebas fueron realizadas en una estación de trabajo con las siguientes prestaciones:

- 4 GB de memoria RAM.
- 1 TB de disco duro.
- Procesador Core Intel 3 a 3.3 Ghz.

Para comprobar el correcto funcionamiento de la estrategia de migración propuesta, utilizando la aplicación de prueba desarrollada, se siguieron los siguientes pasos:

1. Crear conexión con el servidor Exchange.

Para crear la conexión con el servidor Exchange la aplicación hace uso de perfiles MAPI los cuales son implementados en la librería libmapi. A continuación se explican los pasos a seguir para llevar a cabo el proceso de conexión. El usuario debe seleccionar una base de datos de perfiles MAPI en la cual creará los perfiles que serán usados para conectarse al servidor privativo. Luego de seleccionada dicha base de datos el usuario tiene la posibilidad de listar perfiles existentes, adicionar un nuevo perfil o eliminar un perfil existente.

Una vez creado un perfil MAPI la aplicación es capaz de hacer peticiones al servidor Exchange haciendo uso de dicho perfil e importar la información de las cuentas de correo electrónico, así como listar los usuarios existentes en el Active Directory y la información de un buzón determinado.

2. Listar usuarios

Para listar los usuarios existentes en el Active Directory el usuario selecciona la opción “**Opciones de Buzón**” y luego “**Listar Usuarios**”, la aplicación muestra una ventana donde se debe introducir el nombre de un perfil MAPI existente, a través del cual se harán las peticiones al servidor privativo. Luego se selecciona el botón “**Mostrar Usuarios**” y la aplicación muestra una lista con todos los usuarios existentes.

3. Listar datos de Buzones

Para ver la información de un buzón determinado el usuario selecciona la opción “**Opciones de Buzón**” y luego “**Ver datos de buzón**”, la aplicación muestra una ventana donde se debe introducir el nombre de un perfil MAPI existente a través del cual se harán las peticiones al servidor privativo. Luego se selecciona el botón “**Mostrar datos**” y la aplicación muestra una lista con todos los datos del buzón de correo electrónico seleccionado, el cual corresponde al usuario especificado en el perfil previamente escogido.

4. Importar mensajes de correo electrónico

Para importar los mensajes de correo electrónico el usuario selecciona la opción “**Opciones de Buzón**” y luego “**Importar mensajes**”, la aplicación muestra una ventana donde se debe introducir el nombre de un perfil MAPI existente a través del cual se harán las peticiones al servidor privativo. Luego se selecciona el botón “**Ver Mensajes**” y la aplicación muestra una lista con todos los mensajes del buzón de correo electrónico del usuario correspondiente al perfil seleccionado. La aplicación transfiere todos los correos de forma satisfactoria garantizando la integridad de los mensajes. Para comprobar esta operación se muestran los mensajes del usuario presentes en el cliente de correo Outlook y son comparados con la respuesta que muestra la aplicación.

5. Importar contactos

Para importar los contactos el usuario selecciona la opción “**Opciones de Buzón**” y luego “**Importar contactos**”, la aplicación muestra una ventana donde se debe introducir el nombre de un perfil MAPI existente a través del cual se harán las peticiones al servidor privativo. Luego se selecciona el botón “**Mostrar Contactos**” y la aplicación muestra una lista con todos los contactos del buzón de correo electrónico del usuario correspondiente al perfil seleccionado. La aplicación transfiere todos los contactos de forma satisfactoria garantizando la integridad de los mismos. Para comprobar esta operación se muestran los contactos del usuario presentes en el cliente de correo Outlook y son comparados con la respuesta que muestra la aplicación.

6. Importar tareas

Para importar las tareas el usuario selecciona la opción “**Opciones de Buzón**” y luego “**Importar tareas**”, la aplicación muestra una ventana donde se debe introducir el nombre de un perfil MAPI existente a través del cual se harán las peticiones al servidor privativo. Luego se selecciona el botón “**Mostrar tareas**” y la aplicación muestra una lista con todas las tareas del buzón de correo electrónico del usuario correspondiente al perfil seleccionado. La aplicación transfiere todas las tareas de forma satisfactoria garantizando la integridad de las mismas. Para comprobar esta operación se muestran las tareas del usuario presentes en el cliente de correo Outlook y son comparadas con la respuesta que muestra la aplicación.

7. Importar Calendarios

Para importar los calendarios el usuario selecciona la opción “**Opciones de Buzón**” y luego “**Importar calendarios**”, la aplicación muestra una ventana donde se debe introducir el nombre de un perfil MAPI existente a través del cual se harán las peticiones al servidor privativo. Luego se selecciona el botón “**Mostrar calendarios**” y la aplicación muestra una lista con todas las notas en el calendario del buzón de correo electrónico del usuario correspondiente al perfil seleccionado. La aplicación transfiere todas las

notas del calendario de forma satisfactoria garantizando la integridad de las mismas. Para comprobar esta operación se muestran las notas en el calendario del usuario presentes en el cliente de correo Outlook y son comparadas con la respuesta que muestra la aplicación.

A continuación se muestra una comparación entre los datos importados por la herramienta experimental y los datos existentes en el servidor Exchange con el objetivo de evaluar la integridad de los mismos durante el proceso de importación. La comparación se realiza tomando como referencia el buzón de correo electrónico del usuario *Delvis*.

| | Mensajes | | | Contactos | | Tareas | | Calendarios | |
|---------------------------------|----------|------|----|-----------|------|--------|------|-------------|------|
| | # M | I M | DA | # C | IC | # T | IT | # N | IN |
| Herramienta experimental | 10 | 100% | Si | 5 | 100% | 6 | 100% | 4 | 100% |
| Servidor Exchange | 10 | - | - | 5 | - | 6 | - | 4 | - |

Tabla 5: Comparación entre los resultados de la herramienta y los datos de Exchange.

Donde:

M: cantidad de mensajes

C: cantidad de contactos

T: cantidad de tareas

N: cantidad de notas

IM: integridad de los mensajes

IC: integridad de los contactos

IT: integridad de las tareas

IN: integridad de las notas

DA: datos adjuntos recibidos

Luego de la comparación realizada se concluye que la herramienta desarrollada permite transferir mensajes de correo electrónico, tareas, calendarios y contactos desde el servidor Microsoft Exchange haciendo uso de los perfiles MAPI manteniendo la integridad de todos los datos transferidos.

3.3 Resultados de las pruebas

Las pruebas de validación del diseño aplicadas arrojaron los siguientes resultados:

- La confección de prototipos de interfaz evidenció que en los mismos se contemplaban la totalidad de los requerimientos, los cuales daban respuesta a las necesidades del cliente.
- En una primera revisión de las HU se detectaron un total de 16 No Conformidades (NC) asociadas a errores ortográficos (en redacción y prototipos), de redacción del documento y de trazabilidad de requerimientos, las cuales fueron resueltas. En una segunda revisión no se detectaron NC.
- La matriz de trazabilidad evidenció que existía correspondencia entre los requisitos especificados y los objetivos del sistema. La misma se muestra en los anexos.
- La ejecución de auditorías permitió comprobar la correspondencia de los artefactos generados durante la investigación con los formatos y estándares establecidos.
- Las pruebas ejecutadas con la aplicación experimental permitieron comprobar, en primer lugar, la validez de la estrategia de migración propuesta. En un ambiente simulado, donde la misión era migrar una única cuenta de correo electrónico con 10 mensajes, 4 notas en el calendario, 5 contactos y 6 tareas se comprobó que se mantiene la integridad de los datos durante el proceso de migración.

Conclusiones parciales

Se aplicaron pruebas al módulo permitiendo validar la propuesta de solución. Las técnicas de validación de requisitos permitió comprobar la correspondencia de los mismos con las necesidades del cliente y con la aplicación de auditorías se contrastaron los artefactos generados con los formatos establecidos en el proyecto. La aplicación experimental implementada evidenció el correcto funcionamiento de la estrategia de migración haciendo uso de la librería libmapi.

Conclusiones generales

A partir de la investigación realizada se concluye que:

- El estudio del proceso de migración del servidor de correo electrónico Microsoft Exchange hacia GNU/Linux, así como las herramientas más utilizadas para llevar a cabo esta tarea, permitió la comprensión del proceso y la selección de la herramienta más idónea para guiar la propuesta de solución.
- El uso de SXP como metodología de desarrollo de software permitió generar artefactos del diseño que sientan las bases para una futura implementación.
- La selección de OpenChange como herramienta guía para el diseño del módulo permitió definir la librería libmapi como factor fundamental en este proceso, la cual permite transferir datos asociados a las cuentas de correo como mensajes, contactos, calendarios y tareas, favoreciendo la automatización del proceso de migración.
- La aplicación de técnicas de validación de requisitos permitió comprobar la correspondencia de los mismos con las necesidades del cliente. La ejecución de auditorías posibilitó contrastar los artefactos generados con los formatos establecidos. Las pruebas con la aplicación experimental implementada evidenciaron el correcto funcionamiento de la estrategia de migración haciendo uso de la librería libmapi.

Recomendaciones

En busca de dar continuidad a la investigación realizada se identifican los aspectos siguientes:

- ✓ Implementar el módulo de migración que permita la transferencia de información desde Exchange.

Referencias Bibliográficas

- [1]. Definición de telemático. *WordReference.com Diccionario de la lengua española*. [En línea] [Accedido: octubre 22, 2014.] <http://www.wordreference.com/definicion/telem%C3%A1tico>.
- [2]. Baños, María Dolores Cano. Introducción a la Telemática. [En línea] [Accedido: octubre 22, 2014.] http://ocw.bib.upct.es/pluginfile.php/6599/mod_resource/content/1/Intro_Telematica.pdf.
- [3]. Definición de Correo electrónico. *Concepto en Definición ABC. Definicion abc*. [En línea] [Accedido: noviembre 2, 2014.] <http://www.definicionabc.com/tecnologia/correo-electronico.php>.
- [4]. Viera hernández, Amaury. Estrategia para la Migración de servicios de correo electrónico en Cuba. Universidad de las Ciencias informáticas, La Habana, Cuba, 2014.
- [5]. Pérez Benítez, Alfredo, Sánchez Ortiz, Susana and León Ramos, Dayana. *Portal Cubano para la Migración a Software Libre*. Universidad de las Ciencias Informáticas, La habana, Cuba : s.n., 2014.
- [6]. Que es el Software libre?. El sistema operativo GNU. [En línea]. [Accedido: diciembre 3, 2014.] <http://www.gnu.org/philosophy/free-sw.es.html>.
- [7]. Kirch, Olaf. Guía de Administración de Redes con Linux. [En línea] 1999. [Accedido: diciembre 3, 2014.] http://softwarelibre.mes.edu.cu/index_html/centro-de-asistencia-tecnica/manual/guia-de-administracion-de-redes-con-linux/garl-2.0.pdf.
- [8]. Contacto. *BusinessDictionary*. [En línea] [Accedido: enero 5, 2015.] <http://searchexchange.techtarget.com/definition/distribution-list>.
- [9]. Lista de distribucion. *Searchexchange*. [En línea] [Accedido: enero 5, 2015.] <http://searchexchange.techtarget.com/definition/distribution-list>.
- [10]. Alias de correo. *Softonic*. [En línea] [Accedido: enero 5, 2015.] <http://articulos.softonic.com/que-es-un-alias-de-correo-electronico>.

- [11]. Definición de Microsoft Exchange. *Diccionario de informática y tecnología*. [En línea] [Accedido: Enero 12, 2015.] <http://www.cavsi.com/preguntasrespuestas/que-es-microsoft-exchange-server/>.
- [12]. Microsoft Exchange MAPI. *Guatewireless*. [En línea] [Accedido: Enero 20, 2015.] <http://www.guatewireless.org/articulos/e-mail-cual-es-la-diferencia-entre-pop3-imap-y-exchange.html>.
- [13]. MAPI Prrofiles. *Msdn*. [En línea] [Accedido: Enero 20, 2015.] <https://msdn.microsoft.com/en-us/library/office/cc765895.aspx>.
- [14]. Descripción de la arquitectura de administración de Exchange. *Technet*. [En línea] 2005. [Accedido: Enero 20, 2015.] <http://technet.microsoft.com/es-es/library/5eae8ddd-14b9-44e1-9014-8f1e36e20ddf.aspx..>
- [15]. Arquitectura de almacenamiento de Exchange. *Technet*. [En línea] 2005. [Accedido: Enero 20, 2015.] <http://technet.microsoft.com/es-es/library/bb124808%28v=exchg.65%29.aspx>.
- [16]. Definición de java. *Definición de*. [En línea] [Accedido: Enero 22, 2015.] <http://definicion.de/java/>.
- [17]. Definición de XML. *Definición de*. [En línea] [Accedido: Enero 22, 2015.] [http://definicion.de/xml/..](http://definicion.de/xml/)
- [18]. Spring Framework. *SpringSource*. [En línea] [Accedido: Enero 28, 2015.] <http://www.springsource.org/spring-framework..>
- [19]. Netbeans. *Netbeans*. [En línea] [Accedido: Enero 28, 2015.] Disponible en: [http://netbeans.org/..](http://netbeans.org/)
- [20]. Introducción a la tecnología Openchange. *Zentyal*. [En línea] [Accedido: Febrero 5, 2015.] Disponible en: www.zentyal.org.
- [21]. Acerca de Openchange. *Openchange*. [En línea] [Accedido: Febrero 6, 2015.] Disponible en: www.openchange.org.
- [22]. Zarefa-Exchange Alternative. *Linux Magazine PRO*. [En línea] Febrero 2008. [Accedido: Febrero 12, 2015.] http://www.linux-magazine.com/issues/2008/87/exchange_alternative/.

- [23]. Que es ImapSync? *ImapSync*. [En línea] [Accedido: Febrero 12, 2015.] Disponible en: [http://imapsync.lamiral.info/.](http://imapsync.lamiral.info/)
- [24]. Zamudio, Esmeralda Villegas y Méndez, Alejandra Virrueta. Investigación documental. Metodologías de desarrollo de software. Instituto tecnológico superior de Apatzingán, Michoacan. Apatzingán Michoacan: s.n., 2010.
- [25]. Rumbaugh, James, Jacobson, Ivar y Booch, Grady. El lenguaje Unificado de Modelado, Manual de Referencia. s.l. : Addison Wesley, 2000. ISBN: 8478290370.
- [26]. ABIZTAR: Learning Technologies. [En Línea] [Accedido : Mayo 22, 2014.] <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>.
- [27]. Muller, Hausi A., Norman, Ronald J. y Slonim, Jacob . Computer Aided Software Engineering. New York : Springer, 2011.
- [28]. Visual Paradigm. *Visual Paradigm*. [En línea] [Accedido: Febrero 18, 2015.] <http://www.visual-paradigm.com/>.
- [29]. Modelo de Dominio. *Lamcom*. [En línea] [Accedido: Mayo 5, 2015.] <http://www.labcom.upcomillas.es/isw2/apuntes/01Tema3-Modelodedominio.pdf>.
- [30]. Abran, Alain. Software Engineering Body of Knowledge. 2004. ISBN 0-7695-2330-7.
- [31]. Larman, Craig. Uml y Patrones. México : Addison-Wesley, 1999. ISBN 0-13-748880-7.
- [32]. Revisiones de código y estándares de codificación. *Microsoft MSDN*. [En línea] [Accedido: Mayo 5, 2015.] [http://msdn.microsoft.com/es-es/library/aa291591\(v=vs.71\).aspx](http://msdn.microsoft.com/es-es/library/aa291591(v=vs.71).aspx).
- [33]. MSDN. Técnicas de codificación. [En línea] [Accedido: Mayo 5, 2015.] Disponible en: [http://msdn.microsoft.com/es-es/library/aa291593\(v=vs.71\).aspx](http://msdn.microsoft.com/es-es/library/aa291593(v=vs.71).aspx).

- [34]. Hommel Scott. Convecciones de código para el lenguaje de programación Java TM. <http://java.sun.com/docs/codeconv/>. 1999.
- [35]. Técnicas Estáticas. *Q-vision*. [En línea] [Accedido: Mayo 8, 2015.] <http://www.qvision.us/es/servicios/gestion-de-calidad-de-software/pruebas-estaticas-de-codigo>.

Glosario de términos

API: significa Interfaz de programación de aplicaciones (IPA). Es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Active Directory: significa directorio activo. Es un servicio establecido en uno o varios servidores en donde se crean objetos tales como usuarios, equipos o grupos, con el objetivo de administrar los inicios de sesión en los equipos conectados a la red, así como también la administración de políticas en toda la red.

SSH: (Secure SHell, en español: intérprete de órdenes segura) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante intérprete de comandos.

DHCP: significa protocolo de configuración de host dinámico. Es un protocolo de red que permite a los clientes de una red obtener sus parámetros de configuración automáticamente. Se trata de un protocolo de tipo cliente-servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas van estando libres.

W3C: es un consorcio internacional que produce recomendaciones para la World Wide Web.

MTA: (Agente de transferencia de mensajes, en español) es un programa ejecutado en el servidor que transfiere el correo desde un cliente de correo electrónico hasta un servidor de correo electrónico.

SMTP: (Protocolo de transferencia de mensajes simple, en español) es usado para transferir correos electrónicos a través de Internet.

RAM: Se refiere a la memoria de acceso aleatorio de un ordenador. Es la memoria desde donde el procesador recibe las instrucciones y guarda los resultados.

NetBIOS: (*Network Basic Input/Output System*) es una especificación de interfaz para acceso a servicios de red, es decir, una capa de software desarrollado para enlazar un sistema operativo de red con hardware específico.

IMAP: (Internet Message Access Protocol, por sus siglas en inglés). Protocolo de acceso a mensajes de Internet, permite a un cliente acceder y manipular los mensajes de correo electrónico en un servidor .

SSL: significa capa de conexión segura y su sucesor TLS (que significa seguridad de la capa de transporte) son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet.

Registro PTR: es el registro de recurso de un dominio que define las direcciones IP de todos los sistemas en una notación invertida. Esta inversión permite que se pueda buscar una IP en el DNS ya que a la notación de la IP invertida se le añade el dominio in-addr.arpa, convirtiendo la IP en un nombre de dominio.

ActiveSync: es un programa de sincronización de datos desarrollado por Microsoft para su uso con sus sistemas operativos Microsoft Windows, proporciona a los usuarios de Microsoft Windows una manera de transportar los documentos, calendarios, listas de contacto y correo electrónico entre la computadora de escritorio y un dispositivo móvil, como un PC de mano, teléfonos móviles o cualquier otro dispositivo portátil que soporte el protocolo de ActiveSync.

RAID: es un sistema de almacenamiento de datos en tiempo real que utiliza múltiples unidades de almacenamiento de datos (discos duros o SSD) entre los que se distribuyen o replican los datos. Dependiendo de su configuración (a la que suele llamarse «nivel»), los beneficios de un RAID respecto a un único disco son uno o varios de los siguientes: mayor integridad, mayor tolerancia a fallos y mayor capacidad.

Pruebas estáticas: Las técnicas de evaluación estática de artefactos del desarrollo o revisiones pretenden detectar manualmente (sea requisito, diseño, código) defectos en cualquier producto. Los revisores analizan ese producto mediante la lectura del mismo sin ejecutarlo. Consiste en realizar revisiones y análisis estáticos a la documentación del proyecto. Dentro de las revisiones se pueden abordar especificaciones de requerimientos y de prueba, guías de usuario, entre otros.

Archivo de punto de control: se encarga de determinar qué transacciones deben procesarse para moverlas desde los archivos de registro de transacciones a las bases de datos. Los archivos de punto de control se actualizan cuando ESE escribe una determinada transacción en un archivo de base de datos de un disco. Esta actualización hace que el archivo de punto de control apunte siempre a la última transacción que se ha transferido correctamente a la base de datos, lo que proporciona un mecanismo rápido de recuperación.

Archivo de registro reservado: son un repositorio de emergencia de las transacciones. Proporcionan espacio en disco suficiente para escribir una transacción desde la memoria al disco duro, aunque el disco de un servidor esté demasiado lleno para admitir nuevas transacciones en un archivo de registro. Los archivos de registro reservados se encuentran en el directorio de registro de transacciones. Se crean automáticamente cuando se inicializan las bases de datos.