Universidad de las Ciencias Informáticas

Facultad 2

Software para la interconexión de redes aisladas CID-555. Módulo Sincronización de archivos.

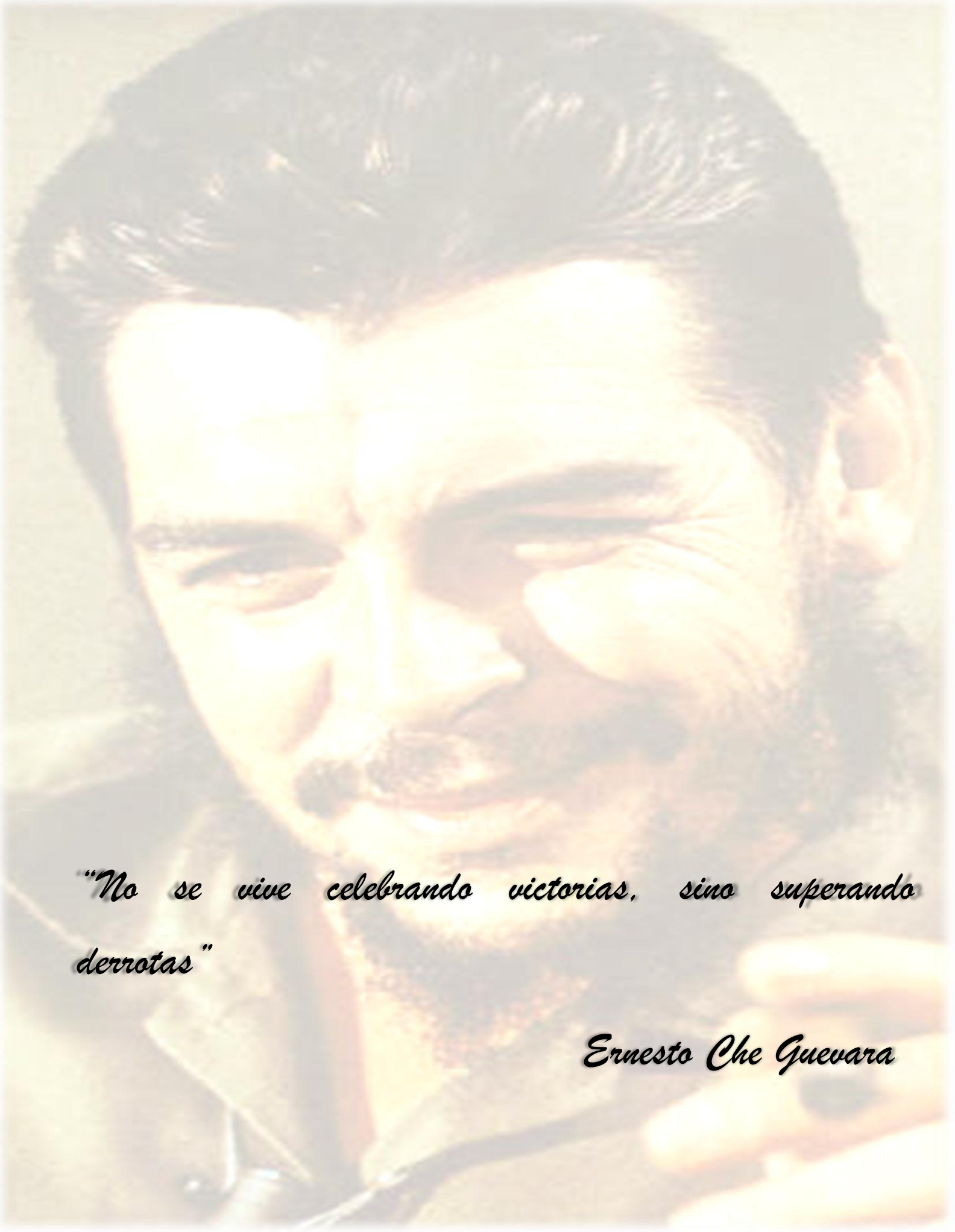
Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor:
Isel Ávila Santiesteban

Tutores:
Ing. Carlos Manuel Hernández Vega

Co-tutores:
Ing. Yosbel Morales Vazquez
Ing. Bárbara Daysi Bedoya López

La Habana
Enero, 2013



"No se vive celebrando victorias, sino superando derrotas"

Ernesto Che Guevara

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) y a la Facultad (2) para que hagan el uso que estimen pertinente con este trabajo.
Para que así conste firmamos la presente a los ____ días del mes de junio del 2013.

_____	_____
Autor:	Tutor:
Isel Ávila Santiesteban	Carlos Manuel Hernández Vega

Resumen

Actualmente en el Ministerio de las Comunicaciones se encuentra implementada la solución CID-555: Pasarela de correos para la intercomunicación de redes aisladas físicamente v2.0, que permite el envío de correos de forma automatizada desde la red interna a la red externa. Esta aplicación utiliza un dispositivo de hardware que posee dos memorias USB internamente para el intercambio de correos y es controlado por la aplicación desarrollada. A esta solución se necesita agregar un nuevo servicio que permita realizar la sincronización de archivos desde la red interna hacia la externa utilizando las funciones del dispositivo para el intercambio de los archivos en el entorno aislado de las redes.

El presente trabajo está encaminado a la solución de procesos para la sincronización de los sistemas de archivos ubicados en las redes aisladas, como un nuevo servicio que permita a diferentes empresas tener un control de la información que gestionan y además llegar a un nivel de seguridad que no pueda quebrantarse ante cualquier amenaza surgida desde Internet. El módulo se integra a la solución existente CID-555: Pasarela de correos para la intercomunicación de redes aisladas brindando oportunidades a instituciones, empresas y otros centros.

Como anteriormente se había mencionado desarrolla el servicio de sincronizar dos sistemas de archivos, pero desde redes aisladas físicamente entre sí. De esta forma se evitan peligrosas oportunidades de ataques a la información que se dispone. La presente solución de sincronización incluye la posibilidad de publicar una misma aplicación en ambas redes que realice la sincronización de sus datos. Además brinda la posibilidad de aplicarse a servidores FTP.

Palabras Clave: Redes aisladas, sincronización, seguridad, sistemas de archivos.

Índice

Introducción:

La necesidad actual de compartir información es cada vez mayor para un gran número de empresas e instituciones que hacen uso de aplicaciones que automatizan este proceso. Compartir la información que manejan las personas que pertenecen a estos centros les hace el trabajo más factible y menos complicado. No obstante, no solo representa una forma más viable sino que también trae aparejado una serie de dificultades y desventajas a la hora de intercambiar toda esta información.

Con el desarrollo creciente de la nueva revolución tecnológica y el surgimiento de Internet, la información se convierte en un elemento importante al ser manejada constantemente en la red. Su importancia está dada mayormente por el valor que le atribuyen las empresas que hacen uso de ella con fines de trabajo y de forma confidencial. Es por ello que para evitar cualquier tipo de afectación a la integridad de la información constituye una necesidad establecer su seguridad.

Actualmente existe un incremento de las amenazas que se originan desde Internet debido a la cantidad de información confidencial que viaja por la red. No existe un control total a las amenazas que vienen desarrollándose con un crecimiento acelerado. A pesar de que se han desarrollado conjuntamente tecnologías tales como los sistemas de detección de intrusos, los cortafuegos, entre otras, con el objetivo de lograr una mayor seguridad; la información y los recursos todavía son delicados y vulnerables a las intromisiones externas.

No deja de ser preocupante las acciones de personas llamadas hackers que intervienen en los sistemas de comunicación con fines malignos y no autorizadas con el objetivo de destruir datos y hacer que un sistema no funcione. Esto demuestra que aún los métodos existentes no son suficientes para satisfacer completamente la seguridad.

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

Para garantizar la integridad de todos los procesos y la seguridad de la información clasificada, que se maneja en el Ministerio de las Comunicaciones, se encuentra implementada una capa adicional de seguridad la cual establece precisamente que no existan conexiones físicas entre la red interna y la

externa conectada a Internet. Específicamente lo que se pretende evitar es todo tipo de conexión mediante los protocolos TCP/IP[1].

Al existir este aislamiento físico de las redes interna y externa, en el Ministerio surge la necesidad de comunicar el servicio de correo institucional con Internet. Para lograr establecer la comunicación, la Universidad de las Ciencias Informáticas (UCI) junto al Instituto Central de Investigaciones Digitales (ICID), la empresa más moderna y experimentada en el tema de la electrónica en Cuba desarrollaron la solución CID-555: Pasarela de correos para intercomunicación de redes aisladas.

La solución se compone de una aplicación de software y un dispositivo de hardware. El software es una aplicación que gestiona la entrada y salida de correos en un servidor de transporte de correos MTA[2] y controla el dispositivo de hardware.

Gracias a la separación física que implementa el CID-555 se impide automáticamente todas las formas conocidas de ataques en línea: spyware, puertas traseras, ataques a nivel de protocolo (como ataques al protocolo TCP), la construcción de túneles, encapsulación de mensajes, entre otros.

Conectar el correo institucional a Internet no solo constituyó una necesidad en su momento. Actualmente la sincronización de sistemas de archivos, también es una necesidad para el Ministerio cuando se tienen redes aisladas.

El Ministerio de las Comunicaciones necesita gestionar archivos desde la red interna a la externa, surgiendo la necesidad de publicar una misma aplicación en ambas redes que realice la sincronización de sus datos. Este servicio no se encuentra implementado en la Pasarela de correos, la cual solo se limita al intercambio de correos y no facilita el trabajo de los administradores en la gestión de los archivos de una red a otra.

Mantener un control de toda la información se convierte en una tarea ardua, pues si se guardan los archivos en ubicaciones diferentes no se puede estar seguro de estar trabajando con las versiones más recientes cuando se tienen diversas copias.

---

[1] **T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol, Protocolo de Control de Transmisión/Protocolo de Internet.

[2] **M**ail **T**ransfer **A**gent, Agente de Transferencia de Correo.

Se torna engorroso hacer actualizaciones de forma manual y directa por cada archivo, a partir de los cambios realizados en archivos específicos de la red interna. Esto incluye invertir un período de tiempo que puede ser reducido al automatizar el proceso de modificación de archivos (copia o eliminación). Además se ganaría en eficiencia y calidad del trabajo.

Por otra parte, para trasladar los archivos actualizados de una red a otra se tendrían que copiar de forma manual a un dispositivo de almacenamiento USB, trabajo que se torna complicado.

Por estas razones se necesita un nuevo servicio que sincronice los archivos ubicados en la red interna con una ubicación en la red externa, un servicio que sea aplicable a servidores FTP[3], archivos de aplicaciones como portales, aplicaciones de gestión y que permita además compartir recursos de una red a otra.

Dada la problemática descrita anteriormente y la necesidad de encontrar una solución para estas dificultades se plantea el siguiente **problema a resolver**:

¿Cómo realizar la sincronización de sistemas de archivos ubicados en dos redes aisladas físicamente entre sí?

Para dar solución a la problemática presentada se propone como **objeto de estudio** la sincronización de sistemas de archivos.

El **campo de acción** se enmarca: en los procesos de sincronización de dos sistemas de archivos ubicados en redes aisladas físicamente.

Se trazó como **objetivo general** de la investigación: El desarrollo de un módulo a la solución CID-555: Software para la interconexión de redes aisladas v2.0, para la sincronización de archivos ubicados en redes aisladas físicamente.

Desglosando el objetivo general en los siguientes **objetivos específicos**:

1. Elaborar la fundamentación teórica de la investigación.
2. Diseñar el módulo a la solución CID-555: Software para la interconexión de redes aisladas que sincronice dos sistemas de archivos, de principal a réplica, mediante la generación de un script con los cambios en el sistema de archivos principal.

[3] **F**ile **T**ransfer **P**rotocol, Protocolo de Transferencia de Archivos.

3. Implementar el módulo diseñado que gestione el servicio de sincronización de archivos, controlando el flujo formado por las etapas: generación del archivo de sincronización (script), empaquetado del archivo de sincronización y nuevos datos, intercambio, desempaquetado y ejecución de la sincronización en el sistema de archivos réplica.
4. Integrar el módulo a la solución CID-555: Software para la interconexión de redes aisladas v2.0.

Para darle cumplimiento a los objetivos definidos se propone la realización de las siguientes **tareas de investigación**:

- Identificación y estudio de las soluciones informáticas existentes para sistemas de sincronización de archivos en redes aisladas para una mayor profundización en el tema.
- Estudio de las herramientas y tecnologías a utilizar para el diseño e implementación del módulo de sincronización de sistemas de archivos. Selección de las que se van a usar durante el desarrollo.
- Estudio de las metodologías existentes para el desarrollo de aplicaciones. Selección de la que se va a usar durante el desarrollo.
- Análisis de los requisitos de software y del modelo de negocio, correspondientes al módulo de sincronización de sistemas de archivos.
- Diseño de los diagramas de secuencia correspondientes al módulo.
- Diseño de los diagramas de clases del diseño correspondientes al módulo.
- Implementación del módulo de sincronización de archivos.
- Diseño del diagrama de despliegue correspondiente al módulo.
- Diseño de los diagramas de componentes correspondientes al módulo.
- Diseño de los casos de prueba para el módulo implementado.
- Realización de pruebas al módulo de sincronización de archivos con el objetivo de obtener resultados satisfactorios.

- Estudio del entorno de despliegue para la instalación del módulo en el laboratorio de pruebas de la solución integral.

El presente documento se estructura en 4 capítulos:

Capítulo 1: Fundamentación Teórica.

En el Capítulo 1 se realiza un estudio de los conceptos y definiciones relacionados con el tema de la investigación, los cuales son abordados en su desarrollo. Incluye una profundización de las soluciones informáticas similares para la sincronización de archivos en el entorno de redes aisladas. Se hace referencia a las herramientas utilizadas para el desarrollo del sistema, así como las tecnologías, metodología y lenguajes existentes más utilizados e idóneos para el proceso de implementación.

Capítulo 2: Características del sistema.

En el Capítulo 2 se describe el negocio de la solución que se desarrolla. Se realiza una propuesta del sistema. Se desarrollan las especificaciones de requisitos de software, además son descritos los requerimientos funcionales y los requerimientos no funcionales que debe tener el sistema. Son definidos los casos de uso que tendrá la solución, representados por el diagrama de casos de uso correspondiente.

Capítulo 3: Diseño del sistema.

En el capítulo 3 se presentan también los diagramas de clases del diseño, secuencia y los diagramas que representan el diseño de la base de datos y la arquitectura del sistema así como la descripción de la misma.

Capítulo 4: Implementación y pruebas.

En el capítulo 4 se reflejarán los diferentes diagramas de despliegue y de componentes así como los diferentes casos de prueba y los resultados de los mismos.

Capítulo 1: Fundamentación Teórica

1.1. Introducción

Durante el desarrollo de este capítulo se exponen los conceptos y definiciones investigados como términos claves que serán utilizados durante el desarrollo de este documento para una mejor comprensión del contenido del trabajo. Del mismo modo el presente capítulo incluye un estudio de las metodologías y tecnologías utilizadas para el desarrollo del proyecto, así como las herramientas y los lenguajes existentes que serán significativos en el diseño y desarrollo posterior del sistema.

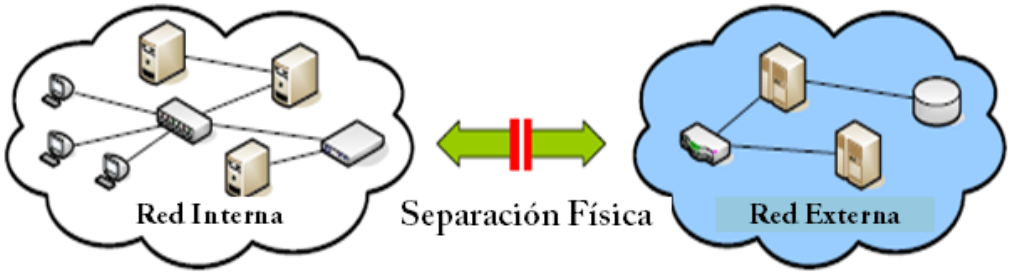
1.2. Términos claves

1.2.1 Redes aisladas

Cuando se menciona el término ¨**redes**¨ en la informática, se refiere a un sistema de comunicación entre un conjunto de ordenadores y dispositivos electrónicos conectados entre sí cuya finalidad es compartir recursos, información y servicios. (1)

Al existir infraestructura informática en centros como empresas y agencias gubernamentales que manejan información altamente sensible haciendo uso de las redes informáticas, es imprescindible la protección integral de sus redes y datos en pos de evitar o prever los ataques realizados al sistema de comunicación por personas malintencionadas. La protección de las redes y datos se aplica a una amplia gama de organismos en todo el mundo, incluida la policía, el ejército, los bancos, las compañías de seguros, empresas de telecomunicaciones así como oficinas de registro de residentes, entre otras. Algo importante para cualquier institución y organización es la confianza, el intercambio confidencial y seguro de datos entre las redes. Hoy en día se han desarrollado mecanismos de protección altamente radicales y seguros. Uno de esos mecanismos lo constituye la separación física de las redes. (2)

La separación física de las conexiones de redes o redes aisladas no es más que encontrar una manera de intercambiar datos entre dos redes. Generalmente clasificada como una alta seguridad de la red interna y una red externa menos segura sin establecer una conexión física directa. La figura 1 muestra una representación más precisa de redes aisladas. (3)



.

Fig 1. Representación de redes aisladas.

1.2.2 Sistemas de sincronización de archivos

La sincronización es la coordinación de procesos que se ejecutan simultáneamente para completar una tarea, con el fin de obtener un orden de ejecución correcto y evitar así estados inesperados. Es el proceso mediante el cual los archivos de dos o más ubicaciones se mantienen sincronizados. (4)

La sincronización puede ser en una o dos direcciones. En la **sincronización unidireccional**, cada vez que se agrega, cambia o elimina un archivo u otra información en una ubicación, se debe realizar la misma acción en la otra ubicación. Sin embargo, de realizar algún cambio en la segunda ubicación no se realiza ningún cambio en la primera ubicación porque la sincronización únicamente es en un solo sentido. (4)

En la **sincronización bidireccional**, los archivos se copian en ambas direcciones, por lo que los archivos se mantienen sincronizados en ambas ubicaciones. Cada vez que agrega, cambia o elimina un archivo en cualquier ubicación, ese mismo cambio también se realiza en la otra ubicación de sincronización. La sincronización bidireccional se suele usar en entornos de trabajo, en los que los archivos se actualizan con frecuencia en más de una ubicación y, a continuación, se sincronizan con otras ubicaciones. (4)

Los sistemas de sincronización son por tanto los que permiten gestionar los procesos de sincronización de forma automatizada, ya sea unidireccional o bidireccional. Estos sistemas velan por la estructura de un sistema de archivos y mantienen actualizada la réplica.

1.3. Sistemas similares

Lock Keeper

Sistema surgido en el año 2003 y comercializado por Siemens Switzerland. En los últimos años esta aplicación ha sido mejorada para ser más maduro y confiable. Es una solución de alto nivel basada en la simple idea de la "Separación Física de las Redes". Es un dispositivo de hardware que funciona como intermediario para proveer un intercambio de datos seguro entre las redes aisladas. Basado además en el principio de que el mejor método para garantizar la seguridad de una red es desconectarla. Debido a esta separación física Lock-Keeper evita automáticamente las formas conocidas de ataques en línea: spyware, puertas traseras, los ataques a nivel de protocolo (por ejemplo ataques TCP número de secuencia), construcción de túneles, mensaje encapsulado, entre otros, demostrando que ni los atacantes externos ni los que conocen acerca del tema pueden cambiar o pasar por alto la separación física de las redes que implementa esta solución. (5)

El Look Keeper ofrece diferentes módulos de aplicación, entre los que se encuentran:

- ✓ Tranferencia de correos.
- ✓ Intercambio de archivos.
- ✓ Replicación de bases de datos en línea. (6)

Sus principales características son:

- ✓ Alto nivel de protección de la red interna.
- ✓ Implementación del principio de Separación Física.
- ✓ Diseño flexible basado en el concepto de asegúralo primero, constrúyelo después.
- ✓ Simple de combinar con aplicaciones de terceros como firewalls, antivirus, etc. (6)

Look Keeper como solución brinda innumerables servicios de reconocida importancia y más si se habla de seguridad de redes. Pero desde el punto de vista económico esta solución tiene la desventaja de tener un precio muy elevado, y actualmente por las condiciones económicas que posee nuestro país resulta complicado hacer tal inversión para adquirir este software. Por estas razones se hizo necesario trazar un objetivo para obtener otras soluciones por medios propios, más factibles pero además con gastos reducidos y que contribuyan a la independencia tecnológica del país.

Basado en esta necesidad se decidió la construcción de un dispositivo para automatizar el envío de correos en redes aisladas, esta tarea le fue encomendada al ICID. Este dispositivo fue utilizado en la versión 1.0 del software para la interconexión de redes aisladas, es decir que ya existe un software elaborado con similares propósitos. (7)

Por otra parte Look Keeper no cuenta con una solución que gestione y satisfaga, de forma íntegra, todos los requerimientos que el presente módulo de sincronización de archivos será capaz de adicionar al software para la interconexión de redes aisladas v2.0, pues se necesitan otras funcionalidades más específicas de la aplicación como por ejemplo la sincronización de archivos ya que solamente provee intercambio de archivos.

1.4. Tecnologías utilizadas

1.4.1. Middleware Zeroc-ICE

ICE[4] es un middleware (software intermediario) orientado a objetos que soporta varios lenguajes: C++, Java, C#, Visual Basic, Python, PHP. ICE es un software libre, que está disponible en código fuente completo y ofrecido bajo los términos de Licencia Pública General de GNU (GPL) aunque también existen licencias comerciales para los clientes que deseen usar ICE para código propietario. (8)

ICE permite centrar los esfuerzos en la lógica de aplicación, y se encarga de todas las interacciones con bajo nivel de programación de interfaces de red. (7)

Los principales objetivos del diseño de ICE son:

- ✓ Proporcionar una plataforma middleware orientado a objetos adecuados para su uso en ambientes heterogéneos.
- ✓ Proporcionar un conjunto completo de características que apoyan el desarrollo de aplicaciones realmente distribuidas para una amplia variedad de dominios.
- ✓ Evitar la complejidad innecesaria, por lo que la plataforma es fácil de aprender y de usar.
- ✓ Proporcionar una implementación que sea eficiente en ancho de banda, consumo de memoria, y sobrecarga del CPU.

[4] **I**nternet **C**ommunications **E**ngine, Motor de Comunicaciones de Internet.

- ✓ Proporcionar una implementación que posea seguridad interna, por lo que es adecuado para su uso a través de redes públicas inseguras. (9)

1.4.2. Lenguajes de programación

Un lenguaje de programación describe un conjunto de acciones consecutivas que un equipo debe ejecutar. Estas acciones constituyen un conjunto de símbolos y reglas que permiten la comunicación con un computador. Las reglas permiten combinar dichos símbolos y se usan para escribir programas. Los lenguajes de programación al igual que un lenguaje natural se componen de un léxico, una estructura (sintaxis) y un significado (semántica). El léxico es un conjunto de símbolos permitidos o vocabulario que forman parte de un lenguaje específico, la sintaxis son las reglas que indican cómo construir frases correctas en un lenguaje, la semántica son las reglas que permiten determinar el significado de cualquier construcción del lenguaje, es decir, constituye el significado de las frases generadas por la sintaxis y el léxico. (10)

Cada lenguaje tiene sus instrucciones y enunciados verbales propios, que se combinan para formar los programas de cómputo. Los lenguajes de programación no son aplicaciones, sino tecnologías que permiten construir y adecuar esas aplicaciones. (11)

De acuerdo al número de instrucciones necesarias para desarrollar una tarea específica se tiene la clasificación de los lenguajes informáticos:

Lenguaje de bajo nivel.

Es el tipo de lenguaje que cualquier computadora es capaz de entender. Se dice que los programas escritos en forma de ceros y unos están en lenguaje de máquina, porque esa es la versión del programa que la computadora realmente lee y sigue. (10)

Lenguajes de alto nivel.

Son lenguajes de programación que se asemejan a las lenguas humanas usando palabras y frases fáciles de entender. (10)

La diferencia existente entre los lenguajes de bajo nivel y los lenguajes de alto nivel está en que, el primero cada instrucción corresponde a una acción ejecutable por el ordenador y el último una instrucción puede corresponder a varias acciones. (10)

Entre sus principales características se encuentran: (10)

- ✓ Son independientes de la arquitectura física de la computadora.
- ✓ Permiten usar los mismos programas en computadoras de diferentes arquitecturas (portabilidad), y no es necesario conocer el hardware específico de la máquina.
- ✓ La ejecución de un programa en lenguaje de alto nivel, requiere de una traducción del mismo al lenguaje de la computadora donde va a ser ejecutado.
- ✓ Una sentencia en un lenguaje de alto nivel da lugar, al ser traducida, a varias instrucciones en lenguaje entendible por el computador.
- ✓ Se suelen incluir instrucciones potentes de uso frecuente que son ofrecidas por el lenguaje de programación.

Teniendo en cuenta las características que poseen los lenguajes de alto nivel y la utilidad que tienen, se utiliza esta clasificación de alto nivel para aprovechar todas las ventajas que brinda para el desarrollo de programas o aplicaciones, pues es un tipo de lenguaje de programación avanzado que no está limitado por el tipo de computadora o para un trabajo en específico. Es entendido más fácilmente, es más fácil de manipular, escribir, mantener y de corregir errores, ya que utilizan instrucciones en palabras que describen con claridad la tarea a realizar. Ayudan a incluir rutinas de uso frecuente como son las de entrada/salida, funciones matemáticas, manejo de tablas etc. En resumen, facilitan el proceso de programación y a la misma vez le posibilita al programador ciertas libertades para ser creativo. Además las posibilidades son infinitas y solo están limitadas a la creatividad del programador.

1.4.2.1 Python 2.7

Python es un lenguaje de programación de alto nivel creado por Guido van Rossum a principios de los años 90. Es un lenguaje con una sintaxis simple, sencilla y favorece un código legible. Es interpretado o de script, con tipado dinámico y orientado a objetos. La legibilidad permite acceder al código y tener una mayor comprensión de la implementación. (12)

Es multiplataforma ya que está disponible en plataformas como UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS, etc. lo que posibilita la utilización de librerías específicas de cada plataforma y correr en todos estos sistemas sin grandes cambios. (12)

Ofrece gran soporte e integración con otros lenguajes y herramientas. Viene con una biblioteca estándar muy amplia que incluye herramientas matemáticas y dicímiles funcionalidades de gran ayuda

desde el más bajo nivel hasta el más alto, facilitando al programador la implementación de aplicaciones sin la necesidad de recurrir continuamente a bibliotecas externas. Además dispone de una extensa colección de bibliotecas libres disponibles en la mayoría de los repositorios de los sistemas GNU\Linux.

Debido a la portabilidad de su código; al programar las aplicaciones sin utilizar bibliotecas propias a los sistemas operativos, es posible ejecutarlas en cualquier plataforma sobre la que exista el intérprete de Python.

Estas características unidas a la simplicidad, interactividad, código abierto y excelente documentación, lo convierten en un lenguaje muy apropiado y codiciado para numerosas aplicaciones.

Ofrece un entorno interactivo que facilita la realización de pruebas. El entorno de ejecución de Python detecta muchos de los errores de programación que escapan al control de los compiladores y proporciona información para detectarlos y corregirlos. Posee un amplio juego de estructuras de datos que se pueden manipular de modo sencillo. Además una de las ventajas fundamentales de Python es la gratuidad de su intérprete. (13)

Se utiliza Python para el desarrollo de la aplicación por las características y ventajas que brinda. Además ofrece funcionalidades de gran utilidad que determinan las facilidades de uso del lenguaje y teniendo en cuenta principalmente que la solución Pasarela de Correos a la que se debe integrar el módulo de sincronización de archivos se encuentra desarrollada en el lenguaje de Python.

1.4.2.2 PHP[5] 5.3

PHP es un lenguaje script que corre del lado del servidor en la Arquitectura Cliente – Servidor. Fue desarrollado originalmente por Rasmus Cerdorf en 1994. Es utilizado para generar páginas Web dinámicas. Su programación es segura y confiable ya que de ninguna manera en el navegador se accede al código fuente en PHP, sino solo a su resultado en HTML[6]. (14)

[5] HyperText Preprocessor

[6] **H**yper**T**ext **M**arkup **L**anguage, Lenguaje de Marcado de Hipertexto.

Es extremadamente simple para alguien con poca experiencia, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales, esto demuestra su facilidad de aprendizaje. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. (14)

Con PHP se puede procesar la información de formularios, generar páginas con contenidos dinámicos, entre muchas más cosas. Permite aplicar técnicas de programación orientada a objetos. (14)

Como se ha diseñado para su uso en la Web, incorpora una gran cantidad de funciones integradas para realizar útiles tareas relacionadas con la Web. Puede generar imágenes GIF a un instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF. (14)

Una de sus características es su portabilidad ya que está disponible para diferentes sistemas operativos. (15)

Dispone de una conexión propia a todos los sistemas de base de datos. Además de MySQL, puede conectarse directamente a bases de datos como PostgreSQL, mSQL, Oracle, entre otras. (15)

Independientemente de todas las facilidades que aporta este lenguaje de programación, las funcionalidades que ofrece, la amplia documentación, las innumerables ventajas en desarrollo de aplicaciones Web, al igual que las características que hacen que sea un lenguaje de gran utilidad; es utilizado para el desarrollo de la aplicación porque la interfaz Web de la solución Pasarela de Correos está implementada en PHP y el módulo de sincronización de archivos se integrará a esta solución.

1.4.3 BPMN

Business Process Modeling Notation o BPMN (En español Notación para el Modelado de Procesos de Negocio), constituye un estándar que proporciona una notación gráfica para expresar procesos de negocio en un formato de flujo de trabajo (workflow). Estos procesos constituyen una colección de actividades estructuradas y relacionadas que se modelan mediante un diagrama de procesos de negocio (DPN), que modela también los sub-procesos y tareas que componen a estos procesos.

La notación BPMN fue desarrollada por la organización Business Process Management Initiative (BPMI) y es actualmente mantenida por el OMG (Object Management Group), luego de la fusión de las dos organizaciones en el año 2005. Su versión actual, en abril de 2011, es la 2.0. (16)

El principal objetivo de BPMN es proveer una notación estándar que sea fácilmente legible y entendible para el usuario. (17)

Dicha notación está planeada para dar soporte únicamente a aquellos procesos que sean aplicables a procesos de negocios. Esto significa que cualquier otro tipo de modelado realizado por una organización con fines distintos a los del negocio no estará en el ámbito de BPMN.

BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación; y para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente. BPMN permite también modelar los procesos de una manera unificada y estandarizada permitiendo un entendimiento a todas las personas de una organización específica. (18)

La notación BPMN es utilizada para expresar y representar los procesos de negocios definidos para el presente sistema. Gracias a las ventajas que posee se usa esta notación gráfica estándar ya que permite modelar todos los procesos de negocio que se definieron para el módulo de sincronización de archivos. Además ofrece varios elementos para realizar este proceso paso a paso y es más entendible cuando se trabaja en torno a informáticos de un equipo de desarrollo.

1.4.4 Framework de JavaScript

1.4.4.1 JQuery

JQuery es una librería de JavaScript. Fue publicado por primera vez en enero del 2006 en “BarCamp NYC” por John Resign. Es un conjunto de funciones que ya fueron desarrolladas y probadas. Estas funciones se utilizan de una manera muy simplificada y permiten lograr los mismos resultados en menos tiempo y sin necesidad de programar una funcionalidad completamente.

JQuery permite agregar efectos y funcionalidades complejas a una aplicación web, como por ejemplo: galerías de fotos dinámicas, validación de formularios, calendarios y mucho más. Otra ventaja es la posibilidad que nos brinda de trabajar con AJAX, sin tener en cuenta los detalles complejos de la programación. Además cuenta con la posibilidad de agregar plugins facilitando aún más el trabajo.

En resumen, es flexible y rápido para el desarrollo web. Tiene una excelente comunidad de soporte y plugins, además de su excelente integración con AJAX. Permite escribir menos código y funciona en todos los navegadores actuales incluyendo Internet Explorer 6+, Firefox 2+, Safari 3+, Chrome, and Opera 9+. Usa selectores CSS, aprovechando el conocimiento que la mayoría de los diseñadores web tienen. Es de

código libre licenciado a la vez con Licencia MIT y la Licencia Pública General GNU. Es relativamente fácil de aprender. (19)

Se utiliza el framework de Javascript JQuery porque independientemente de las ventajas que ofrece ya es usado en la solución Pasarela de correos v2.0, a la que se integrará el módulo en desarrollo.

1.5 Metodología de Desarrollo de Software

Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo. (20)

1.5.1 RUP[7]

La metodología de desarrollo seleccionada es RUP y se caracteriza por ser:

- **Dirigido por casos de uso:** Los casos de uso describen los requisitos funcionales del sistema desde la perspectiva del usuario y se usan para determinar el alcance de cada iteración y el contenido de trabajo de cada persona del equipo de desarrollo.
- **Centrado en la arquitectura:** La arquitectura permite ganar control sobre el proyecto para manejar su complejidad y controlar su integridad. Hace posible la reutilización a gran escala y provee una base para la gestión del proyecto.
- **Iterativo e incremental:** Se divide en 4 fases: Inicio, Elaboración, Construcción y Transición, y cada una de ellas se divide en iteraciones. Cada iteración realizada añade funcionalidades al producto de software o mejora las existentes. Además en cada iteración se trabaja en un número de disciplinas haciendo énfasis en algunas de ellas. Las disciplinas propuestas por RUP son: Modelado del negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas, entre otras. (21)

[7] **R**ational **U**nified **P**rocess, Proceso Unificado de Rational.

Se definió RUP como metodología de desarrollo por ser política del departamento de Seguridad Informática del centro de Telemática la utilización de la misma para el proceso de implementación de todos los productos de software. Además se considera que la utilización de RUP puede ser de suma importancia para el desarrollo de la aplicación de forma eficiente y con alta productividad debido a que define qué se tiene que hacer, cómo, quién y cuándo lo hace en cada momento del proceso de desarrollo de software.

Constituye un estándar en el desarrollo del software actualmente, de forma que se adapta a un amplio rango de proyectos y organizaciones. Representa una guía en el desarrollo de proyectos que requieren seguimiento y control. Al estar dividido en fases brinda una mayor organización del trabajo. Es la metodología más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. Permite además obtener un producto con calidad.

1.6 Herramientas de software para el desarrollo de la solución

1.6.1 Herramienta CASE

Las herramientas CASE[8] nacen para auxiliar a los desarrolladores de software, lo que permite el apoyo computarizado en todo o en parte del ciclo de vida del desarrollo de un sistema de software.

CASE comprende un amplio abanico de diferentes tipos de programas que se utilizan para ayudar a las actividades del proceso de software, como el análisis de requerimientos, el modelado de sistemas, la depuración y las pruebas. (22)

Las herramientas CASE han surgido para dar solución a varios problemas inherentes al diseño del software, principalmente nacen para solucionar el problema de la mejora de la calidad del desarrollo de sistemas de mediano y gran tamaño, y en segundo término, por el aumento de la productividad.

En el presente trabajo se utilizará Visual Paradigm como herramienta CASE. (23)

1.6.1.1 Visual Paradigm 8.0

[8] **C**omputer **A**ided **S**oftware **E**ngineering, Ingeniería de Software Asistida por Computadora.

Visual Paradigm 8.0 es una herramienta de modelado. Su característica más importante es su flexibilidad ya que es multiplataforma, es decir, tiene la capacidad de ejecutarse sobre diferentes sistemas operativos además de constituir una herramienta de software libre.

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

Esta herramienta soporta todo el ciclo de vida del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Como lenguaje de modelado utiliza UML[9], ofreciendo soluciones de software que permiten a las organizaciones desarrollar las aplicaciones con calidad más rápido, de forma satisfactoria y más baratas. Posee gran facilidad de uso y el entorno gráfico que brinda es agradable para los usuarios. Presenta una buena interoperabilidad con otras herramientas CASE y con los principales Entornos de Desarrollo Integrado. (24)

Visual Paradigm 8.0 fue la herramienta de modelado de software elegida para representar los artefactos de la aplicación debido a las facilidades que brinda para la modelación de sistemas ya que agiliza la creación de los diagramas definidos en la metodología de desarrollo RUP. Se puede utilizar en sistemas operativos GNU/Linux incluyendo además que el equipo de desarrollo está familiarizado con el uso de esta herramienta.

1.6.2 Entorno de Desarrollo Integrado

1.6.2.1 Eclipse Helios 3.6

La plataforma Eclipse consiste en un Entorno de Desarrollo Integrado (IDE, Integrated Development Enviroment) abierto y extensible. Un IDE es un programa compuesto por un conjunto de herramientas útiles para un desarrollador de software. Como elementos básicos, un IDE posee un editor de código, un compilador/intérprete y un depurador. Eclipse cuenta con numerosas herramientas de desarrollo de

[9] **U**nified **M**odeling **L**anguage, Lenguaje Unificado de Modelado.

software. También da soporte a lenguajes de programación, como C/C++, Cobol, Fortran, Java, PHP o Python. A la plataforma base de Eclipse se le pueden añadir extensiones (plugins) para extender las funcionalidades de la herramienta. (25)

PyDev es una de las extensiones que existe para Eclipse, permite integrar a la plataforma un IDE para Python, el cual provee entre otras funcionalidades:

- Completamiento de Código.
- Resaltado de Sintaxis.
- Análisis del Código.
- Refactorizar.
- Debuguear.
- Consola Interactiva. (26)

A través de la investigación realizada para Eclipse, se pudieron apreciar características significativas que demuestran que Eclipse constituye un IDE idóneo para el desarrollo de la aplicación que se desarrolla. Además es importante resaltar su alto nivel de integración con el leguaje a utilizar Python. Por estas razones y las ventajas que proporciona se decidió la utilización de este entorno de desarrollo integrado.

1.7 Conclusiones parciales

Durante el desarrollo de este capítulo se realizó un estudio de sistemas similares como el Lock Keeper para el intercambio de archivos en redes aisladas, solución que no gestiona, ni satisface de forma automatizada las necesidades relacionadas con el proceso de sincronización de archivos en el entorno de redes aisladas en el Ministerio, a partir de esta dificultad incluyendo además su costo elevado se hace necesario diseñar e implementar un sistema que sea capaz de gestionar completamente la sincronización de los archivos para el Ministerio de las Comunicaciones.

Por otra parte para facilitar el desarrollo de la aplicación se hizo un análisis y descripción de las herramientas y tecnologías ya propuestas con el objetivo de explotar las ventajas y las facilidades que ofrecen para la posterior implementación del sistema. Como parte de las tecnologías será utilizado Python como lenguaje de programación, soportado por la herramienta Eclipse. Además Visual Paradigm en su versión 8.0 como herramienta CASE, que utiliza UML como lenguaje de modelado y se utilizó la notación BPMN para el modelado de negocio.

También se definió como metodología de desarrollo de software RUP pues se necesita generar la documentación necesaria durante todo el ciclo de vida del software para de esta forma obtener un resultado con calidad y que satisfaga las necesidades del cliente que interactúa con el producto.

Capítulo 2: Características del sistema

2.1 Introducción

El presente capítulo abordará principalmente la propuesta del sistema así como el modelado de los procesos de negocio. Se definirán los requisitos del software especificándose los requerimientos funcionales, entrada fundamental para la automatización del proceso de sincronización de archivos. Del mismo modo se definirán los requerimientos no funcionales que tendrá la solución describiéndose las características del sistema, seguridad, soporte, confiabilidad, usabilidad, portabilidad, ayuda y documentación en línea de la aplicación de forma general. También serán definidos los casos de uso del sistema, representándose en el diagrama de casos de uso correspondiente.

2.2 Propuesta del sistema

La solución se desarrolla sobre la base de lo implementado en el Ministerio de las Comunicaciones: la solución CID-555: Pasarela de correos para la intercomunicación de redes aisladas v2.0, que permite el envío de correos de forma automatizada. Además utiliza un dispositivo de hardware de conexión USB[10] con almacenamiento de tecnología Flash conectado a dos servidores de bajas prestaciones.

El dispositivo, también nombrado dispositivo de conmutación está conectado a ambos servidores por dos cables serie con conector DB9 y dos cables USB. Internamente tiene dos memorias Flash para el almacenamiento de los mensajes. Las memorias Flash son conmutadas entre ambos servidores por relés electromagnéticos, de tal manera que en un instante de tiempo cada memoria está conectada a un único servidor. (7)

La solución propuesta se integrará a la solución antes descrita como un nuevo servicio que permitirá gestionar la sincronización de archivos en las redes aisladas existentes en el Ministerio. La misma tendrá una aplicación de software que hará automático el proceso de sincronización de archivos en dos servidores aislados, además hará posible sincronizar servidores FTP. La aplicación usará el dispositivo electrónico de conexión USB para comunicar las redes. Los archivos que deben ser sincronizados serán

[10] **U**niversal **S**erial **B**us, Bus Universal en Serie.

almacenados en las memorias y el dispositivo conmutará las memorias para mover los archivos desde el servidor interno hasta el servidor externo. Lo antes descrito se representa en la figura 2.

Fig 2. Representación de la propuesta del sistema. Módulo sincronización de archivos.

2.3 Modelo de negocio

El modelamiento en BPMN se realiza mediante diagramas muy simples con un conjunto muy pequeño de elementos gráficos. A estos diagramas se les conoce como DPN (Diagramas de Procesos de negocio). Con esto se busca que para los usuarios del negocio y los desarrolladores técnicos sea fácil entender el flujo y el proceso.

Para mostrar el flujo de los procesos de negocio del sistema: Módulo de sincronización de archivos se aprovechan las características y ventajas de la notación BPMN, y se modela el siguiente Diagrama de Procesos de Negocio con el objetivo de visualizar con claridad cada proceso de negocio que interviene.

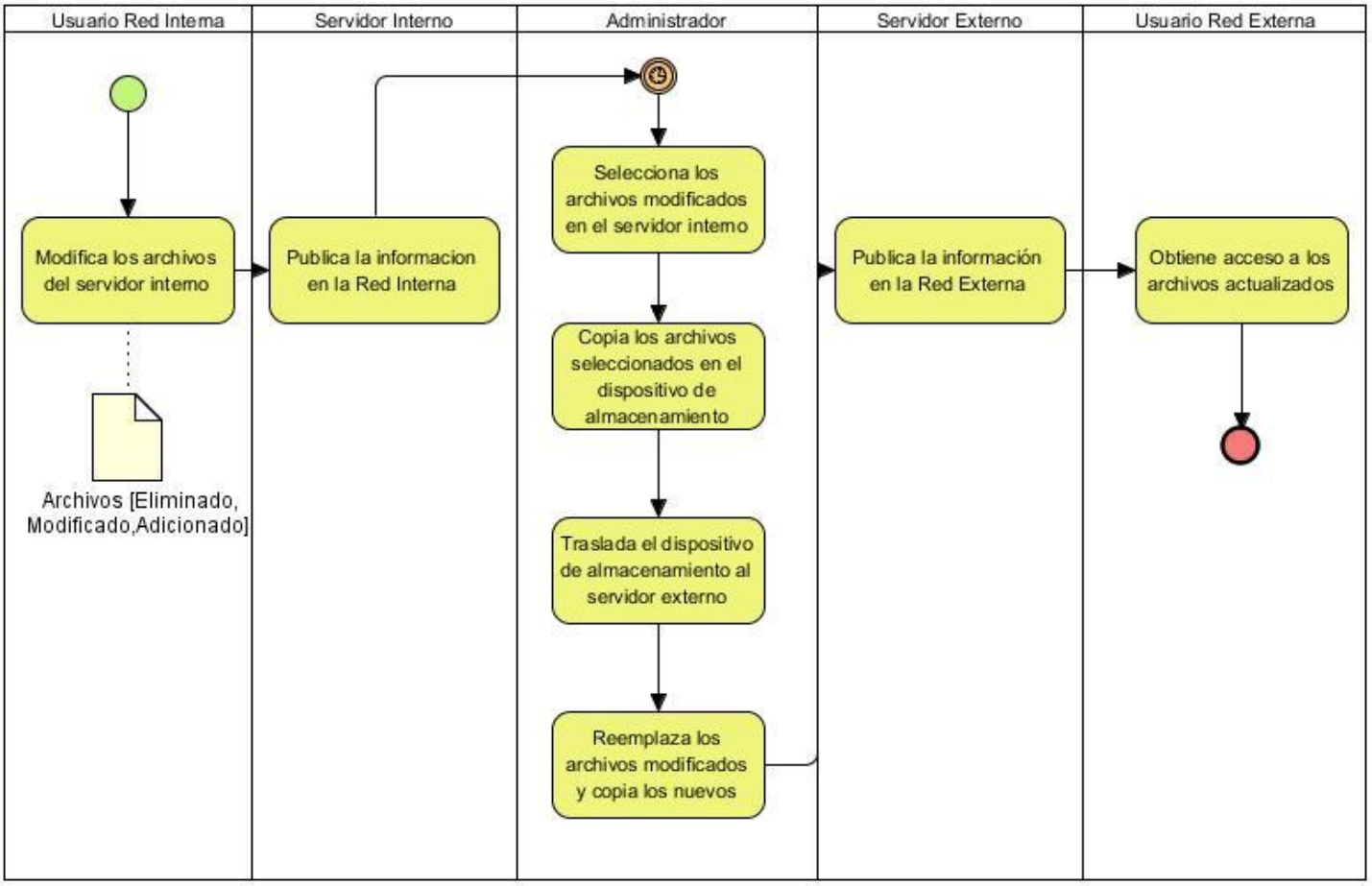


Fig. 3. Diagrama de Procesos de Negocio. Módulo sincronización de archivos.

2.4 Descripción del diagrama de procesos del negocio. Módulo sincronización de archivos.

El proceso se inicia cuando el usuario modifica el sistema de archivos principal ubicado en el servidor interno, el cuál publica la información en la red interna después de realizado los cambios. El administrador cada cierto tiempo selecciona los archivos modificados en el servidor interno y secuencialmente copia los archivos seleccionados en el dispositivo de almacenamiento. Luego traslada el dispositivo al servidor externo, reemplaza los archivos que han sido modificados y copia los

nuevos archivos. Posteriormente el servidor externo publica la información y finalmente el usuario de la red externa accede a los archivos actualizados.

2.5 Especificación de los requisitos de software

Luego de realizarse el modelo de negocio se procede a ejecutar el proceso de captura de requisitos del sistema. Los requisitos especifican qué es lo que el sistema debe hacer, para lo cual son identificadas las funcionalidades requeridas. (22)

2.5.1 Requerimientos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, es decir, especifican acciones que el sistema debe ser capaz de realizar. En los requisitos funcionales se definen los servicios que el sistema debe proporcionar. (22)

La presente solución se compone de varios requisitos funcionales. Teniendo en cuenta también los requisitos funcionales pertenecientes a la solución CID-555: Pasarela de correos para la intercomunicación de redes aisladas V2.0 que sufren modificaciones, con la integración de los requisitos funcionales que debe cumplir el módulo de sincronización de archivos. Las tablas siguientes reflejan los requisitos funcionales que cumplirá el sistema en su solución integral.

Requisitos funcionales a modificar:		
N°	**Funcionalidad**	**Descripción**
RF1.	Crear carpetas en las memorias flash y en la ubicación local de la pc si no están previamente creadas.	El sistema debe crear la estructura de carpetas necesarias en el sistema de archivos locales y en el dispositivo USB cada vez que se realice la sincronización.
RF2.	Detener el sistema.	El sistema permite detener por completo su funcionamiento, a esto se le agrega además, detener el módulo de sincronización.
RF3.	Mostrar Logs.	El sistema permite mostrar el registro de sucesos brindando de esta forma información referente a las operaciones que se van

		realizando sobre los archivos (adicionar, eliminar).
RF4.	Cifrar la información.	El sistema cifra los archivos empaquetados para ser copiados al USB.
RF5.	Descifrar la información.	El sistema descifra los archivos para ser desempaquetados en el servidor externo antes de realizar la sincronización.
RF6.	Aplicar suma de verificación.	El sistema realiza una suma de verificación al fichero cifrado para verificar su validez al pasar de una red a otra.
RF7.	Chequear capacidad de la memoria flash.	El sistema chequea la capacidad de la memoria flash antes de copiar el archivo empaquetado y cifrado.
RF8.	Crear logs.	El sistema registra información referente a cada acción que realiza para una mejor detección de posibles errores.
RF9.	Alertar eventos inesperados.	El sistema alerta sobre posibles eventos que deben ser chequeados por el administrador.

Tabla 1 : Descripción de los requisitos funcionales a modificar. CID-555: Pasarela de correos para la intercomunicación de redes aisladas.

Requisitos funcionales del Módulo sincronización de archivos:		
N°	**Funcionalidad**	**Descripción**
RF10.	Generar archivo de sincronización en el sistema de archivos principal.	El sistema genera un script con todas las operaciones (adicionar, eliminar) que se realizan en el sistema de archivos principal y que se deben ejecutar sobre el sistema de archivos réplica.
RF11.	Empaquetar archivo de sincronización y	El sistema empaqueta los archivos que van a

	nuevos datos.	ser sincronizados junto al script con las operaciones de la sincronización para ser cifrado y copiado al USB.
RF12.	Intercambiar archivos.	El sistema intercambiará los datos de una red a otra haciendo uso del conmutador de dispositivos USB.
RF13.	Desempaquetar archivo de sincronización y nuevos datos.	El sistema desempaquetará los archivos que recibe dado que estos son previamente empaquetados.
R14.	Ejecutar la sincronización en el sistema de archivos réplica.	El sistema haciendo uso del script y los archivos a sincronizar realizará la sincronización en el sistema de archivos réplica.
RF15.	Configurar opciones de sincronización.	El sistema permitirá configurar la dirección del servidor, el puerto, usuario y contraseña para acceder al servidor ftp que será sincronizado.
RF16.	Mostrar última actualización de la sincronización.	El sistema mostrará información acerca de la sincronización como cantidad de archivos sincronizados y el tamaño total de los mismos.

Tabla 2 : Descripción de los requisitos funcionales. Módulo sincronización de archivos.

2.5.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, es decir, son las características que hacen al producto atractivo, usable y confiable. Normalmente, están vinculados a los requerimientos funcionales, o sea, una vez que se conozca lo que el sistema debe hacer, se puede determinar cómo ha de comportarse, las cualidades que debe tener o cuán rápido o grande debe ser. La solución posee la descripción de los siguientes requisitos no funcionales: (27)

2.5.2.1 Apariencia o interfaz externa

- El sistema informático contará con un diseño de interfaz sencillo, agradable, sugerente e intuitivo, de fácil entendimiento. Las páginas de la aplicación estarán poco cargadas, sólo contendrá la información requerida para el usuario. (7)

2.5.2.2 Usabilidad

- La aplicación podrá ser manejada por cualquier usuario con conocimientos básicos del tema. La interfaz y los mensajes para interactuar con el usuario, así como los mensajes de error, serán en el lenguaje español. Los mensajes de error deben ser lo suficientemente informativos para dar a conocer la severidad del error. Los elementos de navegación deben ser orientados al usuario.
- La aplicación podrá ser utilizada en entornos similares donde se requiera este nivel de seguridad. (7)

2.5.2.3 Soporte

- El sistema debe contar con un manual de usuario que permita al usuario hacer más entendible el funcionamiento de la aplicación con el objetivo de ganar una mayor experiencia en el trabajo y uso de sus funcionalidades.
- El sistema debe contar con un manual de instalación el cual podrá ser consultado para los elementos necesarios en el despliegue del sistema (configuraciones, dependencias y elementos a instalar). (7)

2.5.2.4 Seguridad

- El sistema solo debe permitir interactuar con la aplicación a los usuarios que tengan acceso según el rol definido.
- La contraseña de los usuarios del sistema será cifrada.
- El sistema solo interactuará con la memoria flash destinada para esta función.
- La información de las memorias flash estará cifrada.
- A los paquetes recibidos en cada iteración se le realiza una suma de verificación para comprobar la integridad de la información. (7)

2.5.2.5 Legales

- El sistema es desarrollado con herramientas libres lo que permite independencia tecnológica. (7)

2.5.2.6 Confiabilidad

- El sistema debe ser capaz de recuperarse ante la ocurrencia de un fallo y alertar al personal encargado de la administración del mismo, así como proteger la información y el contenido de los dispositivos de almacenamiento. (7)

2.5.2.7 Software

- El servidor Web es Apache 2.22 o superior con soporte para SSL y PHP 5.3.x.
- El servidor de la aplicación requiere el intérprete de Python en su versión 2.7.
- Las computadoras clientes deben tener instalado un navegador web Mozilla Firefox versión 10.02 o superior.
- Se requiere la instalación de un Sistema Operativo GNU/Linux, se recomienda Ubuntu 11.04 o superior. (7)

2.5.2.8 Hardware

Se requieren 2 servidores que tengan como mínimo 1 GB de memoria RAM, procesador Pentium IV o superior y 100 GB de disco duro disponibles. (7)

2.6 Definición de los casos de uso

2.6.1 Diagrama de casos de uso del sistema

Los diagramas de casos de uso del sistema describen y documentan el comportamiento de un sistema desde el punto de vista del usuario, bajo la forma de acciones y reacciones. (28)
Los casos de uso representan las funciones que un sistema puede ejecutar. Son descripciones de la funcionalidad del sistema independientes de la implementación, basados en un lenguaje natural, que le

proporciona como ventaja principal la facilidad para interpretarlos además de ser muy útiles en la comunicación con el cliente. (29)

Luego de la realización de la captura de requisitos se desarrolló utilizando UML el diagrama de casos de uso del sistema: Módulo sincronización de archivos, en el cuál se reflejan los casos de uso:

1. Configurar módulo de sincronización de archivos.
2. Mostrar última actualización de la sincronización.
3. Administrar sistema.
4. Sincronizar servidores FTP.

El diagrama de casos de uso también incluye los casos de uso propios de la solución CID-555: Pasarela de correos para la intercomunicación de redes aisladas v2.0 que no sufren modificaciones. Esto se debe a la integración del módulo a la solución mencionada anteriormente, pues las funcionalidades que implementa son necesarias para la realización del proceso de sincronización. No obstante las funcionalidades del módulo son independientes de la aplicación a la que estará integrado.

Los casos de uso correspondientes a la solución integral CID-555 son los siguientes:

1. Autenticar usuario.
2. Gestionar usuarios.
3. Configurar sistema.
4. Mostrar sucesos del sistema.
5. Mostrar las colas del MTA.

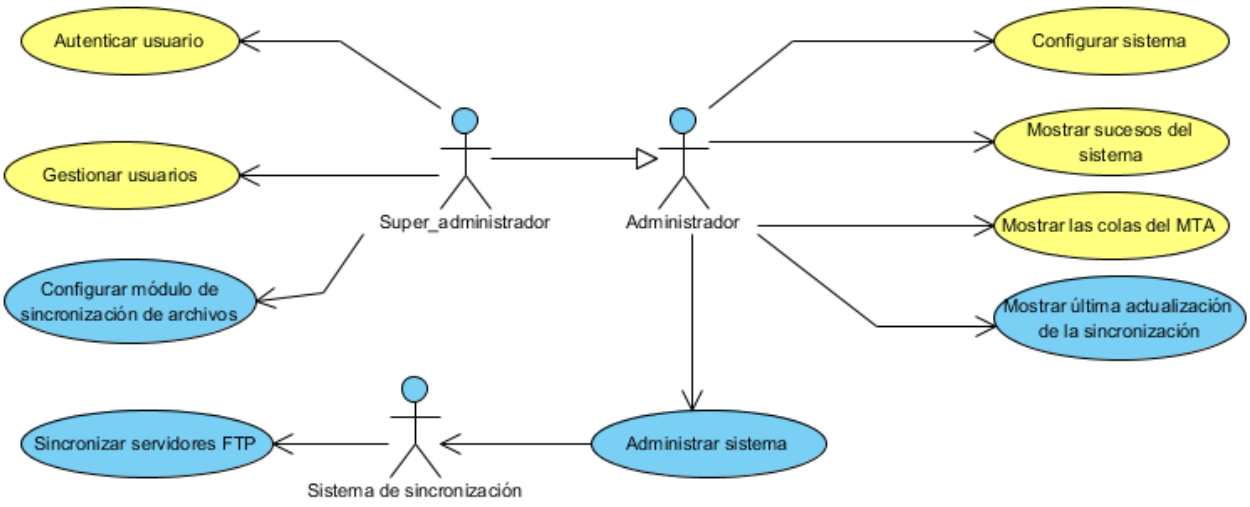


Fig. 4. Diagrama de Casos de Uso del sistema.

Nota: *Para visualizar y diferenciar los casos de uso que pertenecen al Módulo de Sincronización de archivos, de los pertenecientes a la solución CID-555: Pasarela de correos para la intercomunicación de redes aisladas v2.0 que no sufren modificaciones, son representados por el color azul para los casos de uso del módulo y amarillo para la Pasarela de correos.*

Actor	Descripción
Administrador	Representa un rol del sistema que está destinado en su mayoría a usuarios con el cargo de administrador de red.
Super_administrador	Representa un rol del sistema que además de poder realizar las mismas acciones del administrador es el único que puede gestionar los usuarios y establecer las configuraciones del sistema.
Sistema de Sincronización	Representa un actor ficticio, es el evento que realiza las acciones de sincronización. Es iniciado por el caso de uso Administrar sistema en su sección “Iniciar”.

Tabla 3 : Descripción de las responsabilidades de cada actor.

2.6.2 Descripción de los casos de uso del sistema

La descripción de los casos de uso del sistema tiene la finalidad de brindar una explicación detallada de las funcionalidades asociadas a cada caso de uso. Su elaboración debe mostrar claridad para lograr una mejor comprensión sobre lo que debe realizar el sistema. Expresa de forma precisa las acciones que se realizan durante la interacción entre el actor y el sistema, describe el flujo de actividades que desarrolla el actor al hacer uso del sistema y las respuestas que emite el mismo. La descripción de los casos de uso propios del módulo de sincronización de archivos se muestran a continuación y la descripción de los casos de uso pertenecientes a la solución general se pueden encontrar en los anexos.

CU 1. Configurar Módulo de Sincronización de Archivos

Objetivo	Configurar la sincronización de archivos.	
Actores	Super Administrador	
Resumen	El super administrador establece la ruta de los archivos a sincronizar.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	El usuario debe estar autenticado.	
Postcondiciones		
Flujo de eventos		
Flujo básico < Configurar el Sistema >		
	Actor	Sistema
	El actor selecciona la opción configurar del módulo de sincronización de archivos.	
		El sistema muestra una interfaz con los valores actuales de la configuración con los campos directorio a sincronizar, ip, puerto, usuario y contraseña del servidor ftp. Además el botón “Editar”.
	El actor selecciona en la opción editar las opciones de configuración.	
		El sistema muestra un cuadro de diálogo con los datos requeridos: - Ip del servidor ftp. - Puerto del servidor ftp. - Usuario del servidor ftp.

		- Contraseña del servidor ftp. Además los botones “Aceptar” y “Cancelar”
	El actor llena los datos y selecciona la opción Aceptar.	
Flujos alternos		
4º Evento <Los datos son incorrectos>		
		El sistema muestra una alerta de información sobre el error “Datos incorrectos”. Se muestra la interfaz inicial del caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		
5º Evento <Los datos son correctos>		
		El sistema registra los cambios en la base de datos. Se muestra la interfaz inicial del caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

CU 2. Mostrar última actualización de la sincronización.

Objetivo	Brindar información sobre la última actualización de la sincronización.	
Actores	Administrador	
Resumen	El administrador consulta la información referente a la última fecha de la sincronización.	
Complejidad	Baja	
Prioridad	Media	
Precondiciones	El usuario debe estar autenticado.	
Postcondiciones		
Flujo de eventos		
Flujo básico < Configurar el Sistema >		
	Actor	Sistema
	El actor selecciona la opción para mostrar la información referente a la última actualización del módulo de sincronización de archivos.	
		El sistema muestra una interfaz con la información completa de las últimas sincronizaciones. (Fecha, Hora, Descripción, Cantidad de Archivos, Tamaño total)
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		

Asuntos pendientes	

CU 3. Administrar sistema

Objetivo	Controlar y monitorear todo el funcionamiento del flujo central del sistema.	
Actores	Administrador	
Resumen	El caso de uso permite al administrador Iniciar, Reiniciar, Detener el sistema, y Cambiar Id USB.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	El usuario debe estar autenticado en el sistema.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Administrar sistema>		
	Actor	Sistema
	El administrador selecciona la opción“Control del sistema”.	
		El sistema muestra interfaz de visualización de “Control del sistema” con las siguientes opciones: Botón: Iniciar Botón: Detener Botón: Reiniciar

		Botón: Cambiar Id USB
	El administrador selecciona uno de los servicios disponibles para el “Control del sistema” • Si elige “Iniciar”, ir a la Sección 1: “Iniciar”. • Si elige “Detener”, ir a la Sección 2: “Detener”. • Si elige “Reiniciar”, ir a la Sección 3: “Reiniciar”. • Si elige “Cambiar Id USB”, ir a la Sección 4: “Cambiar Id USB”.	
		Termina el caso de uso.
Sección 1: “Iniciar”		
Flujo básico < Iniciar >		
	Actor	Sistema
	El administrador presiona el botón“Iniciar”.	
		El sistema comprueba que haya conexión con el servidor.
		El sistema muestra el mensaje “Sistema Iniciado”, pone inactivo el botón “Iniciar”.
		El sistema crea el archivo cid.log si no existe.
		El sistema dispara el evento “Sistema de sincronización”.

Flujos alternos		
7º Evento <Id no está registrado en la Base de datos>		
		El sistema escribe en los log suplantación de memoria, envía una notificación a los administradores que tengan el campo notificación habilitado en la base de datos y se detiene el sistema.
Sección 2: “Detener”		
Flujo básico < Detener >		
	Actor	Sistema
	El administrador presiona el botón “Detener”.	
		Elsistemacompruebaque haya conexión con el servidor.
		El sistema muestra el mensaje “Sistema detenido”, pone inactivo el botón “Detener”. y pone activo el botón “Iniciar”.
Flujos alternos		
2º Evento <No hay conexión con el servidor>		
	Actor	Sistema
		El sistema presenta una ventana con el siguiente mensaje:

		"Error del sistema. No se ha podido establecer la conexión con el Servidor." Conel botón Cancelar.
		El sistema envía al actor de vuelta al paso 2 de la Sección
Sección 3: "Reiniciar"		
Flujo básico < Reiniciar >		
	Actor	Sistema
	El administrador presiona el botón "Reiniciar".	
		El sistema comprueba que haya conexión con el servidor.
		El sistema ejecuta la Sección 2 y seguidamente la Sección 1.
Flujos alternos		
2º Evento <No hay conexión con el servidor>		
	Actor	Sistema
		El sistema presenta una ventana con el siguiente mensaje: "Error del sistema. No se ha podido establecer la conexión con el Servidor." Conel botón Cancelar.
		El sistema envía al actor de vuelta

		al paso 2 de la Sección
Sección 4: “Cambiar Id USB”		
Flujo básico < Cambiar Id USB>		
	El administrador presiona el botón “Cambiar Id USB”.	
		El sistema presenta un cuadro de diálogo de confirmación: “La acción cambiará un id guardado en el sistema por el de la memoria flash actual. ¿Está seguro de realizar este cambio? ” conlosbotones“Sí, Cambiar Id” y “Cancelar”.
	El administrador presiona el botón “Sí, Cambiar Id”.	
		Elsistemacompruebaque haya conexión con el servidor.
		El sistema cambia el Id del USB almacenado en la base de datopor el del USB conectado.
	El administrador presiona el botón “Cancelar”.	
		El sistema regresa a vista principal.
		Termina caso de uso.

Flujos alternos		
2º Evento <No hay conexión con el servidor>		
	Actor	Sistema
		El sistema presenta un cuadro de diálogo con el siguiente mensaje: "Error del sistema. No se ha podido establecer la conexión con el Servidor." Conel botón Cancelar.
		El sistema envía al actor de vuelta al paso 2 de la Sección.
Flujos alternos		
Relaciones	CU Incluidos	No procede
	CU Extendidos	No procede
Requisitos no funcionales		
Asuntos pendientes		

CU 4. Sincronizar Servidores FTP.

Objetivo	Sincronizar el servidor FTP de la red interna con el de la red externa.
Actores	Sistema de sincronización
Resumen	El sistema realizará la sincronización del servidor FTP interno con el externo realizando una serie de operaciones con los archivos que

	serán sincronizados y haciendo uso del dispositivo CID-555.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones		
Postcondiciones		
Flujo de eventos		
Flujo básico <Ejecutar sincronización>		
	Actor	Sistema
	Se dispara el evento que realiza la sincronización.	
		Genera un archivo con comandos referentes a las acciones de sincronización correspondientes a los cambios realizados por el usuario en el sistema de archivos del servidor FTP interno; añadir, modificar y eliminar archivos. Además copia los archivos a añadir o modificar para una ubicación local.
		El archivo de sincronización y los archivos adjuntos a él son empaquetados en un solo archivo.
		Cifra el archivo empaquetado.
		Genera la suma de verificación del archivo cifrado.
		Se registra en la base de datos del servidor interno información referente

		a la sincronización en curso.
		Realiza el cambio del dispositivo USB de una red a la otra.
		Descifra el archivo recibido desde la red interna dejando el archivo empaquetado en una ubicación local.
		Desempaqueta el archivo recibido desde la red interna dejando el archivo de la sincronización y los archivos correspondientes para ser sincronizados.
		Haciendo uso del archivo generado en el servidor interno y los archivos adjuntos a este, realiza la sincronización de archivos en el servidor externo.
		Registra en la base de datos del servidor externo información referente a la sincronización completada.
		Espera el tiempo definido en la base de datos para realizar el próximo ciclo.
Flujos alternos		
2º Evento <No está conectado el dispositivo USB>		
		El sistema registra en los logs una entrada “No hay USB conectado.”
Flujos alternos		
3º Evento <No hay conexión con el servidor FTP>		

		El sistema registra en los logs una entrada “No hay conexión con el servidor FTP.”
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

2.7 Conclusiones parciales

Durante el desarrollo de este capítulo se realizó una descripción del negocio de la solución, se efectuó un análisis de las características del sistema que se presenta, lo cual contribuye a la documentación y correcta implementación de la aplicación. Se definieron los requisitos, tanto funcionales, como no funcionales y la representación del diagrama de casos de uso que hará posible el comienzo de la construcción del sistema. Para un mayor entendimiento de lo que debe realizar el sistema se realizaron las descripciones de los casos de uso con el objetivo de que cualquier usuario que interactúe con el sistema no tenga ninguna dificultad con su funcionamiento.

Capítulo 3: Diseño del sistema

3.1 Introducción

Este capítulo está dirigido al diseño de la aplicación, haciendo uso de la metodología RUP. Se generan los artefactos: diagramas de secuencia y diagramas de clases del diseño. Se fundamenta la arquitectura del sistema, así como todo el marco de trabajo de desarrollo de la aplicación. Se detallan los patrones de diseño utilizados en la presente solución y se diseña además la base de datos con las descripciones de las tablas.

3.2 Arquitectura del sistema

El diseño del sistema CID-555: Software para la interconexión de redes aisladas v2.0. Módulo sincronización de archivos está dividido en front-end y back-end.

3.2.1 Front-end

Una aplicación de tipo front-end es aquella que se muestra al usuario final, o sea, es la parte con la que el usuario interactúa directamente. La misma se encarga de proveer una intefaz agradable al usuario simplificando el uso de algún componente que funciona en segundo plano. (32)

En la presente propuesta de solución se utilizará el front-end como administración vía web, usando para la misma el patrón Modelo – Vista – Controlador (MVC), el cuál separa los datos y la lógica del negocio de una aplicación de interfaz de usuario. Para ello MVC propone tres componentes, el modelo, la vista y el controlador, es decir, se define por una parte la representación de la información y por otra la interacción con el usuario. Esto permite un mayor nivel de desacoplamiento y brinda facilidades al desarrollo de la aplicación como por ejemplo simpleza, detección de errores y reutilización del código. (33)

La siguiente imagen ilustra la representación del patrón arquitectónico: Modelo - Vista– Controlador aplicado.

Fig. 6. Arquitectura front-end.

- **Descripción de los componentes del patrón MVC (figura 6):**

Controladores: Son las clases que reciben las entradas, usualmente como eventos, e interpretan las operaciones del usuario. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista.

Vistas: Muestran la información al usuario. Son archivos con contenido visual que permiten la interacción entre el usuario y el sistema, para finalmente retornar una respuesta.

Modelos: Son las clases que procesan la lógica del negocio, son las que acceden a los datos persistentes, es decir, las que interactúan con la Base de Datos y además con el Back-end.

3.2.2 Back-end

El back-end es el encargado de las operaciones con las que pocas veces el usuario entra en contacto. Presta servicios que componen una función de negocio tales como administración, procesamiento de datos, comunicaciones, entre otras. (34)

En el diseño de la presente solución se utilizará el patrón arquitectónico N-Capas para el desarrollo del back-end, el cual tiene como objetivo principal la separación de la lógica de negocio y la lógica del diseño en capas con un nivel de dependencia bajo entre ellas, dado que en cada capa se agrupan elementos con responsabilidades similares, permitiendo así el desarrollo de las capas de forma individual. Esto incluye además, una mayor reutilización de las clases pertenecientes a cada capa, simplificando la detección de errores y posibilitando futuros mantenimientos o cambios. (34)

En el diseño de la propuesta de solución se utilizarán 3 capas y 2 aspectos, como se muestra en la siguiente imagen:

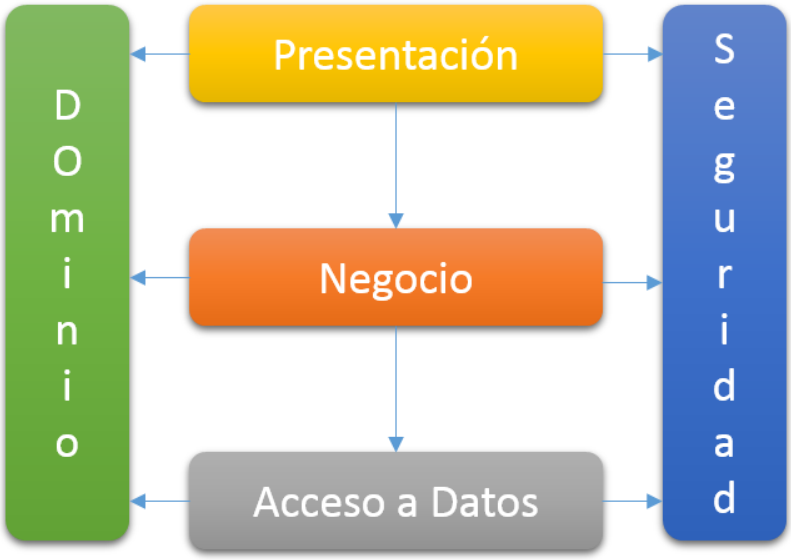


Fig 7. Arquitectura back-end.

- **Descripción de los componentes del patrón n-capas (figura 7):**

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

Interfaz: Es la capa que permite la comunicación entre la interfaz web y la aplicación de control del proceso de intercambio de archivos.

Negocio: En esta capa está la lógica, se reciben las peticiones del usuario, y tras ejecutar una acción se le envía las respuestas del proceso. Es la capa encargada de iniciar y manejar todo el proceso del sistema, es la que realiza el procesamiento de la información.

Acceso a Datos: Es donde se accede a los datos, es la capa que se encarga de la interacción con la Base de Datos del sistema.

Seguridad: Es el aspecto donde se encuentran todos los elementos necesarios para la implementación de la seguridad del sistema. (7)

3.3 Patrones de diseño utilizados

Los patrones de diseño son soluciones estándares para problemas comunes, estos representan un conjunto de reglas, probadas y documentadas anteriormente, que describen cómo afrontar tareas y solucionar problemas que surgen durante el desarrollo de software.

La solución debe ser lo más reutilizable posible para que pueda ser incluida en otros entornos similares. Para el diseño de la aplicación se utilizaron los Patrones Generales de Software para Asignar Responsabilidades (GRASP). Los patrones de GRASP usados fueron: (35)

- **Alta cohesión:** La cohesión 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. Se aplica en la mayoría a las clases del diseño, ya que en cada una solo se implementan las funcionalidades que le corresponden.
- **Bajo acoplamiento:** Ya que se asigna la responsabilidad de que cada clase se comunica con un número relativamente pequeño de clases.
- **Creador:** Las clases que tienen la responsabilidad de crear una nueva instancia de alguna clase que contiene toda la información necesaria para llevar a cabo sus tareas.
- **Experto:** Pues propone que la responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados, es decir, cuando los objetos utilizan su propia información para llevar a cabo sus tareas.

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

Creacionales

- **Singleton:** Este patrón consiste en crear una única instancia de un objeto para una aplicación y la creación de un mecanismo de acceso global a dicha instancia. En el caso de la implementación del backend se utiliza por ejemplo en la clase Configuracion, de la cual se necesita una instancia única.

3.4 Diagrama de clases del diseño

Un **diagrama de clases** muestra las diferentes clases que componen un sistema. Los diagramas de clases se conocen como estáticos porque muestran las clases junto con sus métodos y atributos, así como las relaciones estáticas entre ellas, pero no muestran los métodos mediante los que se invocan las clases. (36)

Los **diagramas de clases del diseño** constituyen diagramas de clases a los que se le aplican extensiones UML llamados estereotipos (pequeñas etiquetas que aplicadas a los elementos o relaciones de un diagrama indican significado adicional). Entre los estereotipos aplicados se encuentran Server Page (Página servidor), Client Page (Página cliente) y Form (Formulario). (37)

Los diagramas de clases del diseño muestran también las vistas con los formularios que las componen, así como las clases JavaScript correspondientes a cada vista. En estos diagramas también se representan las clases controladoras, clases servicios y clases del dominio necesarias para la posterior implementación del sistema.

A continuación se presentan los diagramas de clases del diseño (DCD) del Módulo de sincronización de archivos:

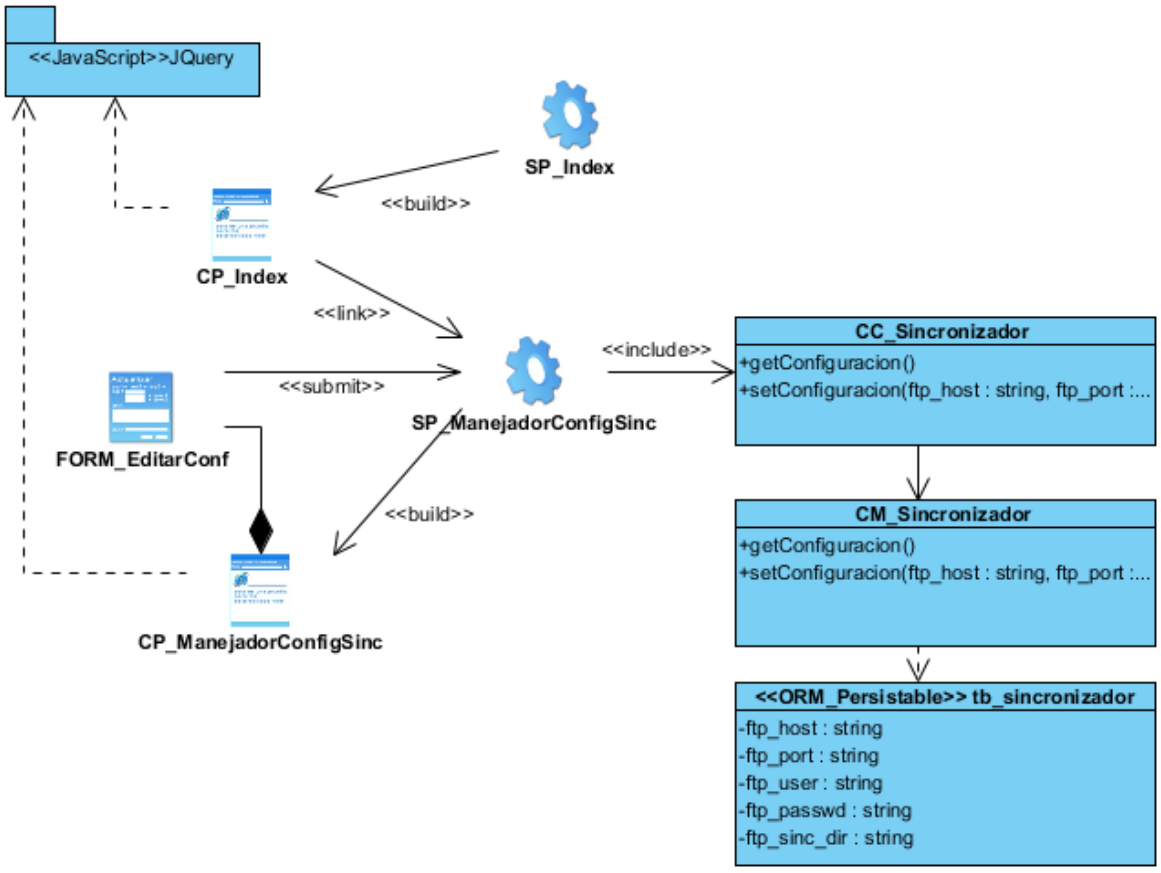


Fig. 8. DCD Configurar módulo sincronización de archivos.

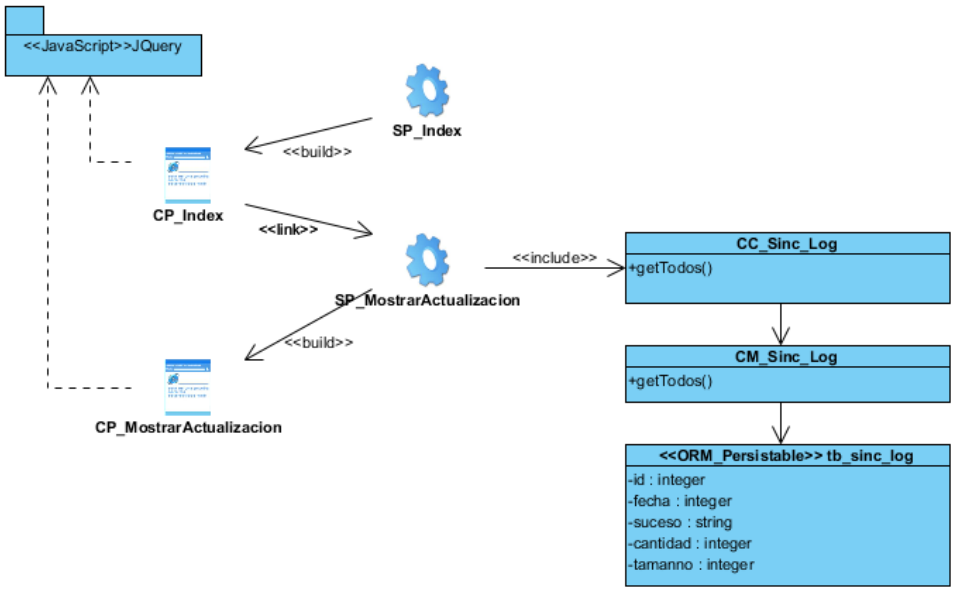


Fig. 9. DCD Mostrar última actualización de la sincronización.

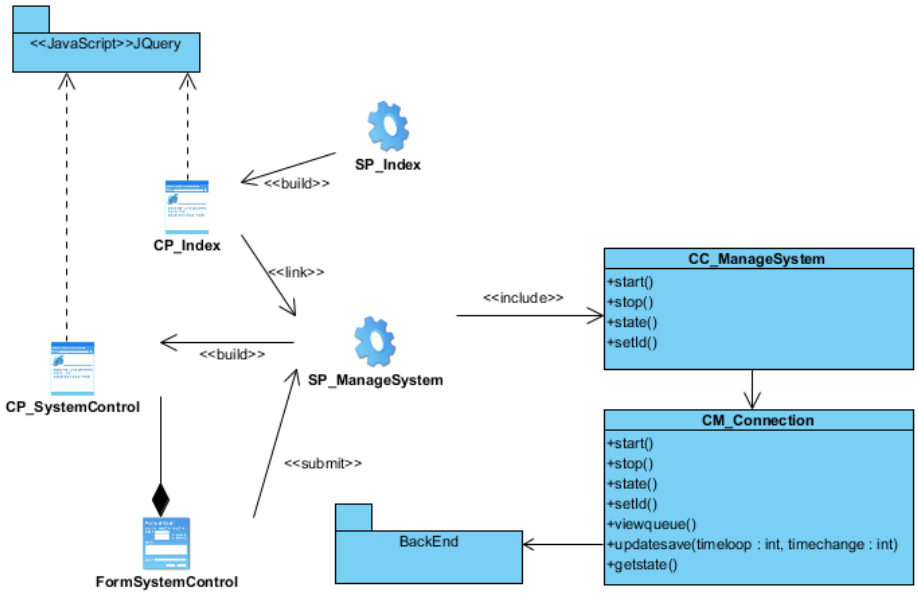


Fig. 10. DCD Administrar Sistema.

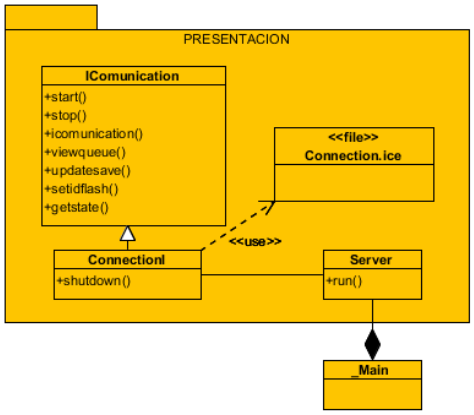


Fig. 11 DCD Capa Presentación

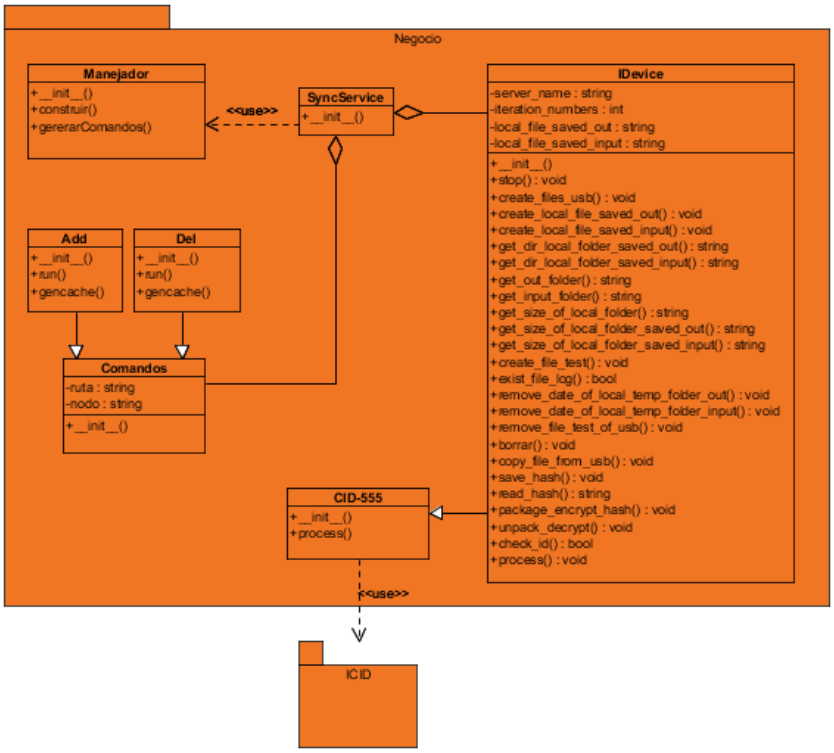


Fig. 12 DCD Capa Negocio

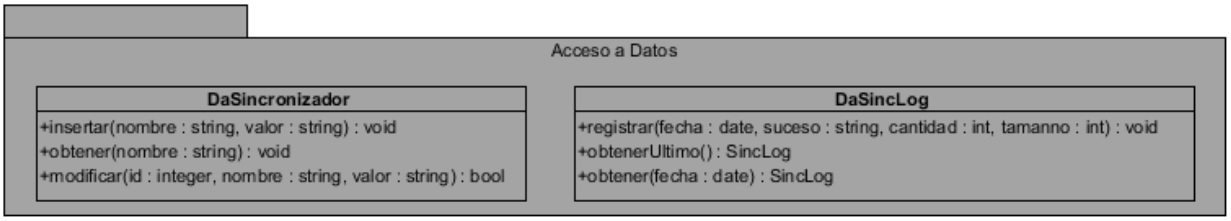


Fig. 13 DCD Capa Acceso a Datos

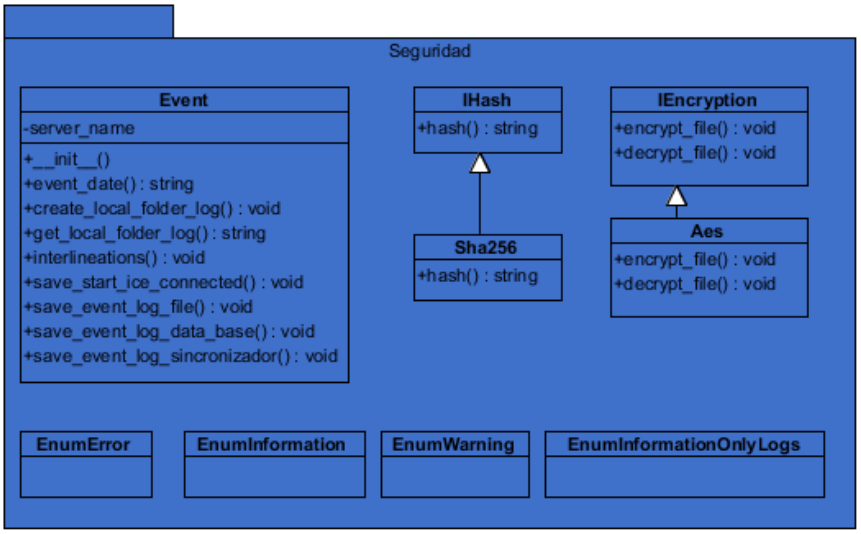


Fig. 14 DCD Capa Seguridad

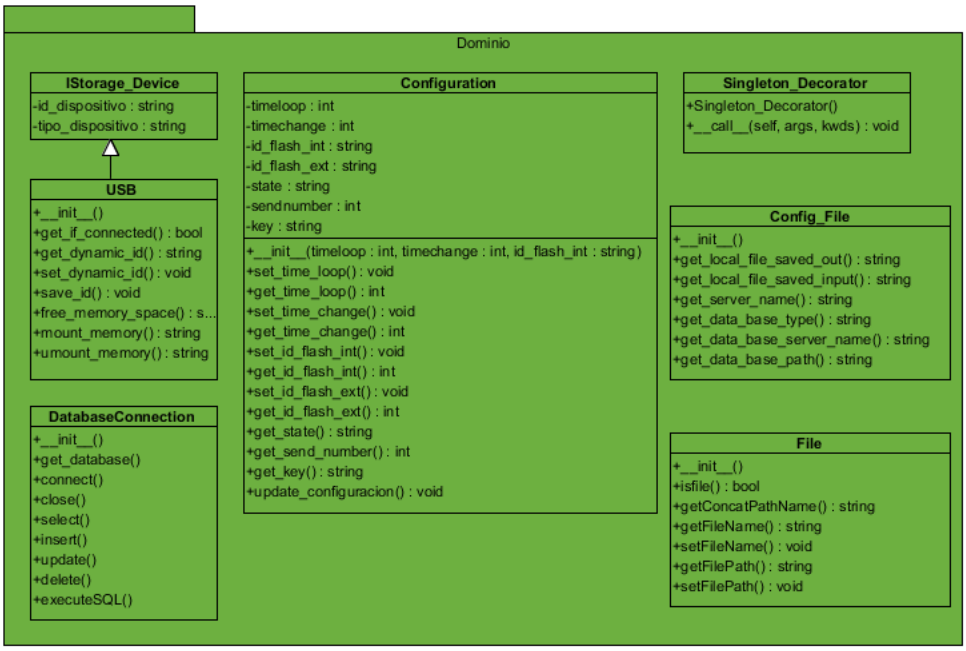


Fig. 15 DCD Capa Dominio

3.4.1 Descripción de los diagramas de clases del diseño.

3.4.1.1 DCD Configurar módulo sincronización de archivos.

El flujo comienza en el momento que se genera la página Index (Inicio) del sistema, el usuario accede desde ahí a la página ManageConfigSinc que contiene un formulario con las opciones de configuración del módulo de sincronización de archivos. En caso de realizar algún cambio se enviará la información a la página correspondiente en el servidor mediante un submit. La misma procesará la información y la almacenará en la base de datos haciendo uso de la clase controladora ConfigSinc que contiene a la clase modelo ConfigSinc y utiliza la clase persistente tb_sincronizador para finalizar el flujo.

3.4.1.2 DCD Mostrar última actualización de la sincronización.

El flujo comienza al generar la página Inicio del sistema, desde donde el usuario accede a la página MostrarActualizacion. Haciendo uso de la clase controladora Sincronizador, que a su vez contiene la clase

modelo Sincronizador y utiliza la clase persistente tb_sinc_log, recuperará de la base de datos la información sobre la última actualización que posteriormente se mostrará al usuario en la página cliente MostrarActualizacion.

3.4.1.3 DCD Administrar Sistema.

El flujo comienza al generar la página de inicio del sistema, desde donde el usuario selecciona la opción Administrar Sistema y accede a la página correspondiente que muestra las opciones que permiten controlar el sistema. Se usa la controladora ManageSystem que a su vez contiene la clase modelo ManageSystem y utiliza la clase CM_Connection. Esta última usa el middleware para comunicarse con el backend y ejecutar las acciones Iniciar, Detener, Reiniciar y Cambiar Id de la Flash.

3.5 Diagrama de secuencia

El diagrama de secuencias es un esquema conceptual que permite representar el comportamiento de un sistema, para lo cual emplea la especificación de los objetos que se encuentran en un escenario y la secuencia de mensajes intercambiados entre ellos, con el fin de llevar a cabo una transacción del sistema. Muestra la forma en que los objetos se comunican entre sí al transcurrir el tiempo.

Se muestran los diagramas de secuencia (DS) del diseño para los caso de uso (CU) Administrar sistema en las secciones iniciar y detener; y Configurar módulo de sincronización de archivos.

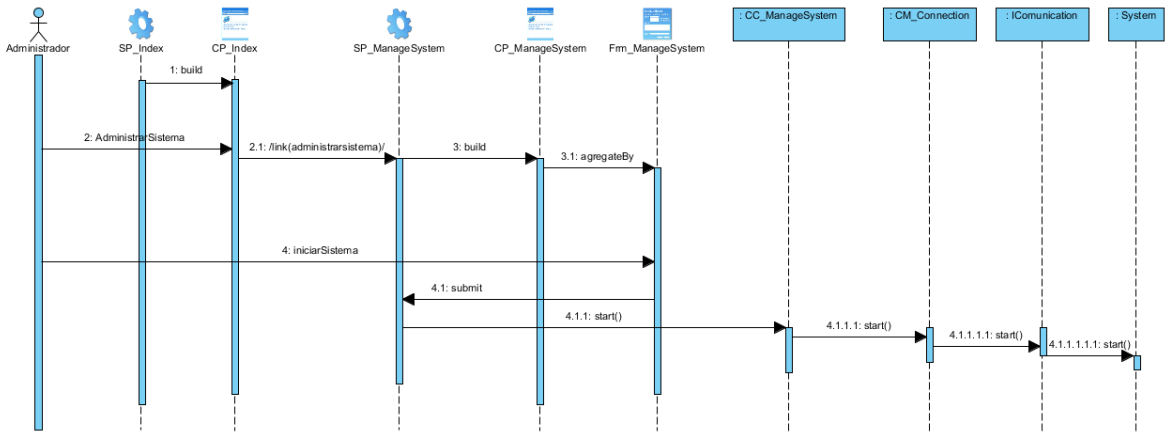


Figura 16. DS CU Administrar sistema. Sección Iniciar.

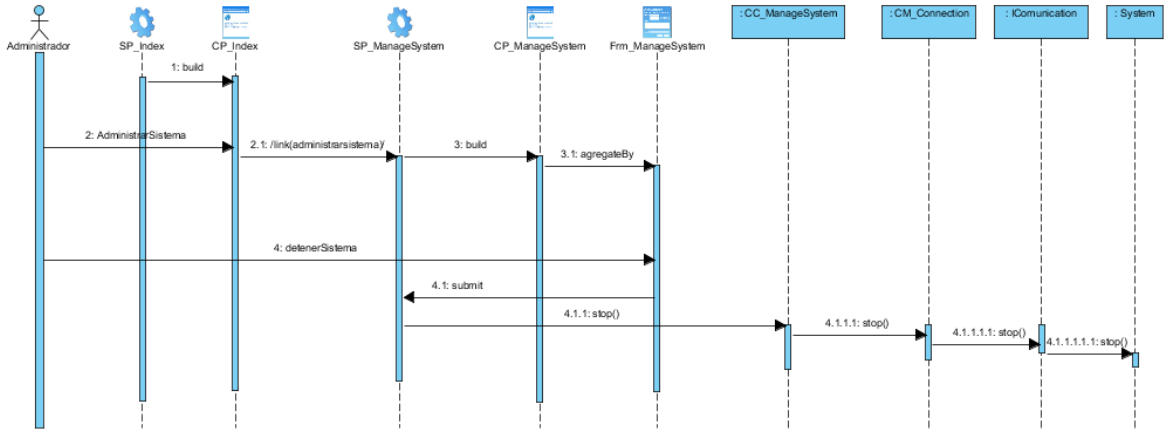


Fig. 17. DS CU Administrar sistema. Sección Detener.

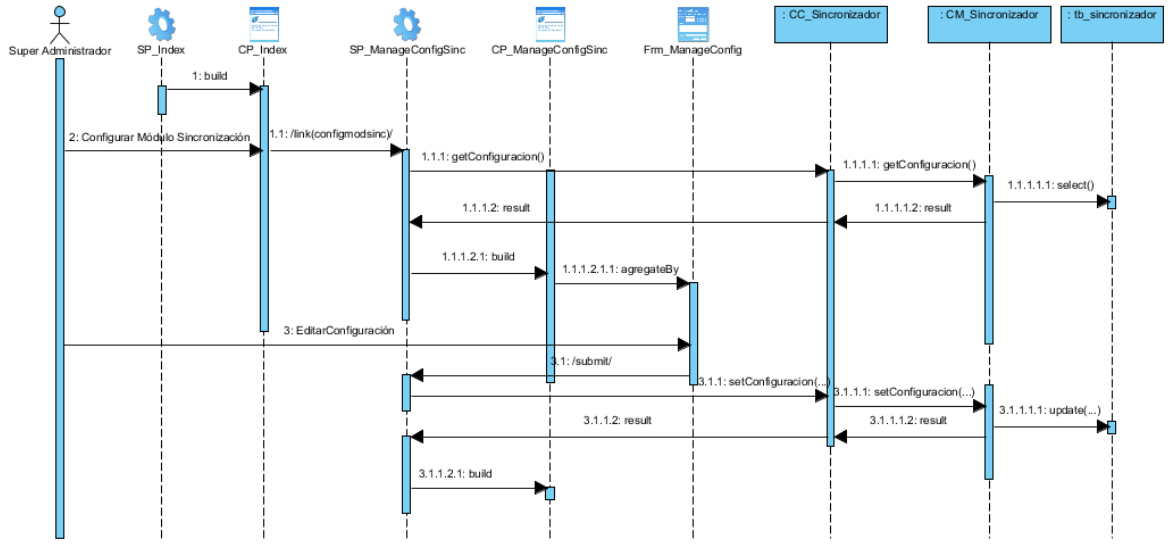


Fig. 18. DS CU Configurar módulo de sincronización de archivos.

3.6 Diseño de la Base de Datos

El diseño de bases de datos es el proceso de analizar un problema para el que se necesita una solución, desarrollando el modelo lógico de los datos disponibles para la misma. Agrupando los datos en tablas

relacionadas para así almacenar, manejar y recuperar dichos datos referentes a la solución. El modelo relacional se centra en esta idea: la organización de los datos en colecciones de tablas bidimensionales llamadas relaciones que reflejan las entidades existentes y las relaciones entre ellas. (39)

Para lograr un buen diseño de la base de datos se hace necesario seguir una serie de pasos, primeramente se definen los diagramas de clases persistentes y el modelo entidad relación. Las figuras 18 y 19 muestran los diagramas de la base de datos de todo el sistema donde se incluyen las tablas del módulo sincronización de archivos:

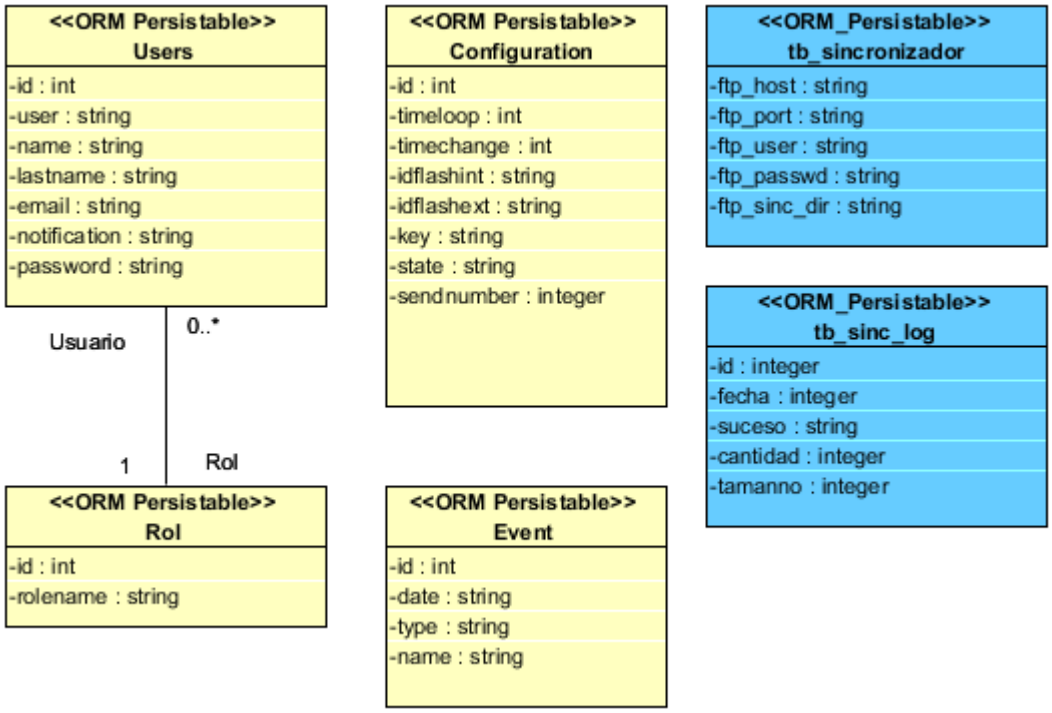


Figura 19. Diagrama de clases persistentes.

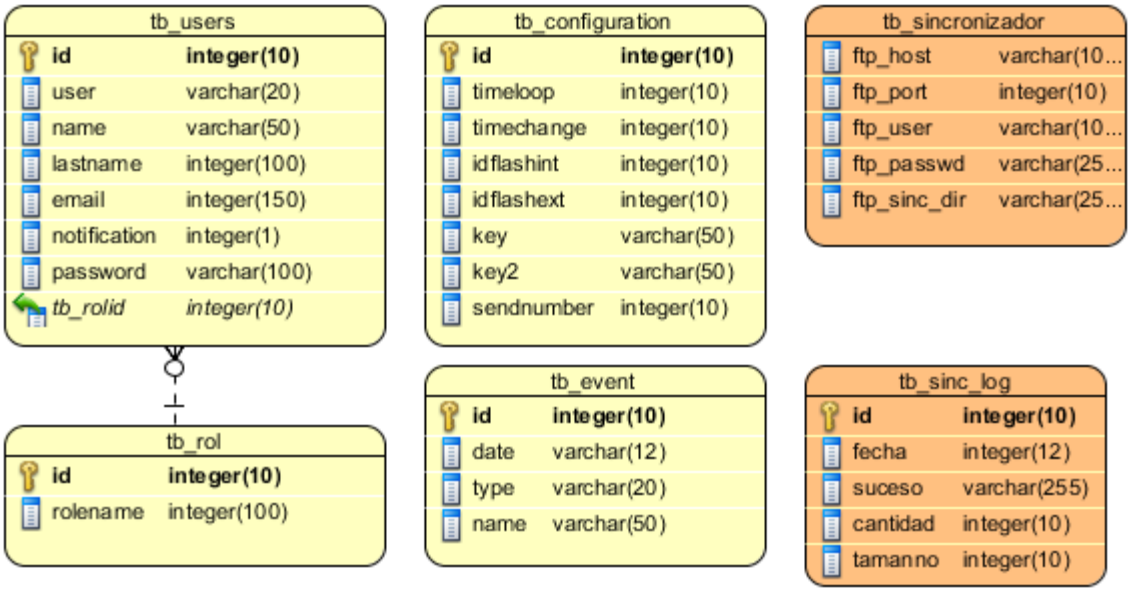


Fig. 20 Modelo Entidad-Relación.

3.6.1 Descripción de las tablas del diagrama de Base de datos.

Para la presente solución se requiere hacer persistentes las configuraciones del módulo y tener un registro de las sincronizaciones realizadas. Por tanto, se utilizarán las tablas tb_sicronizador en la cual se guardarán elementos de configuración como por ejemplo: dirección del servidor ftp y todos los datos necesarios para realizar la sincronización; y la tabla tb_sinc_log donde se almacenará información referente a la sincronización, última fecha, tamaño, cantidad de ficheros y una breve descripción.

3.7 Conclusiones parciales

En el presente capítulo se han modelado los diagramas de clases del diseño y los diagramas de secuencia, artefactos obtenidos en el diseño del sistema, elemento fundamental para el comienzo de la implementación. Se obtuvieron además los artefactos del diseño de la base de datos perteneciente a la solución integral que incluye el módulo sincronización de archivos. El modelado de todos los artefactos generados reflejan una idea inicial de lo que será el software realmente.

Gracias al diseño de la arquitectura se brinda una visión de cómo se estructura el sistema y cómo se agruparán los componentes, al igual que la manera en que interactuarán entre ellos. De esta forma se organiza y se comprende mejor el sistema que se desarrolla.

Capítulo 4: Implementación y pruebas

4.1 Introducción

La implementación del sistema comienza después de haber terminado los flujos de trabajo: análisis y diseño. Con los artefactos obtenidos se inicia la implementación, fase que es de vital importancia para el desarrollo de cualquier sistema. En el capítulo se presenta el diagrama de despliegue y el diagrama de componentes correspondientes al módulo. Además se realizarán las pruebas correspondientes para comprobar que el producto posee la calidad requerida.

4.2 Generalidades de la implementación

La implementación es un flujo de trabajo de RUP que se comienza con el resultado del diseño y se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. Se describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. Entre los principales objetivos de este fjujo de trabajo se encuentran:

- Definir la organización del sistema en términos de Subsistemas de Implementación organizados en capas.
- Probar los componentes desarrollados independientemente como unidades: cada implementador es responsable de probar las unidades que produzca.
- Integrar los resultados producidos por desarrolladores independientes o equipos en un sistema ejecutable.

El flujo de implementación está fuertemente determinado por el lenguaje de programación utilizado y en la fase de elaboración va encaminado a implementar la arquitectura que se ha definido. El resultado final de la implementación es un sistema ejecutable. Además implementa las clases y subsistemas encontrados durante el diseño.

4.3 Diagrama de despliegue

El Diagrama de Despliegue describe la topología del sistema, la estructura de los elementos de hardware y el software que ejecuta cada uno de los nodos que conforma la estructura. Los nodos son objetos físicos que existen en tiempo de ejecución, y que representan algún tipo de recurso computacional (capacidad de memoria y procesamiento). En otras palabras un diagrama de despliegue describe la arquitectura física del sistema durante la ejecución, en términos de procesadores, dispositivos y componentes de software.
A continuación se muestra el diagrama de despliegue correspondiente al módulo:

Fig. 21 Diagrama de despliegue

4.3.1 Descripción del diagrama de despliegue

El sistema funcionará en dos servidores de aplicaciones que se comunicarán por medio del conmutador de dispositivos USB mediante el cual se realizará el intercambio de la información de una red a otra. En cada uno de estos servidores estará presente el sistema de archivos a sincronizar. Además se tendrán dos servidores FTP, uno en cada lado de la red interna y externa. Estos servidores se sincronizarán mediante el uso del presente sistema para lo cual se necesita una conexión de tipo TCP/IP entre dichos servidores FTP y el servidor de aplicación correspondiente.

4.4 Diagrama de componentes

El diagrama de componentes es utilizado para modelar los componentes de un sistema, incluyendo los artefactos que implementan dichos componentes. Un componente es la implementación física de un conjunto de elementos lógicos, como por ejemplo las clases.

Un diagrama de componentes muestra la estructura de alto nivel del modelo de implementación, especificando los subsistemas de implementación y sus dependencias a la hora de importar código.
A continuación se presentan los diagramas de componentes del sistema, que incluyen un diagrama de componentes general organizado en capas con el diagrama de componentes correspondiente a cada capa:

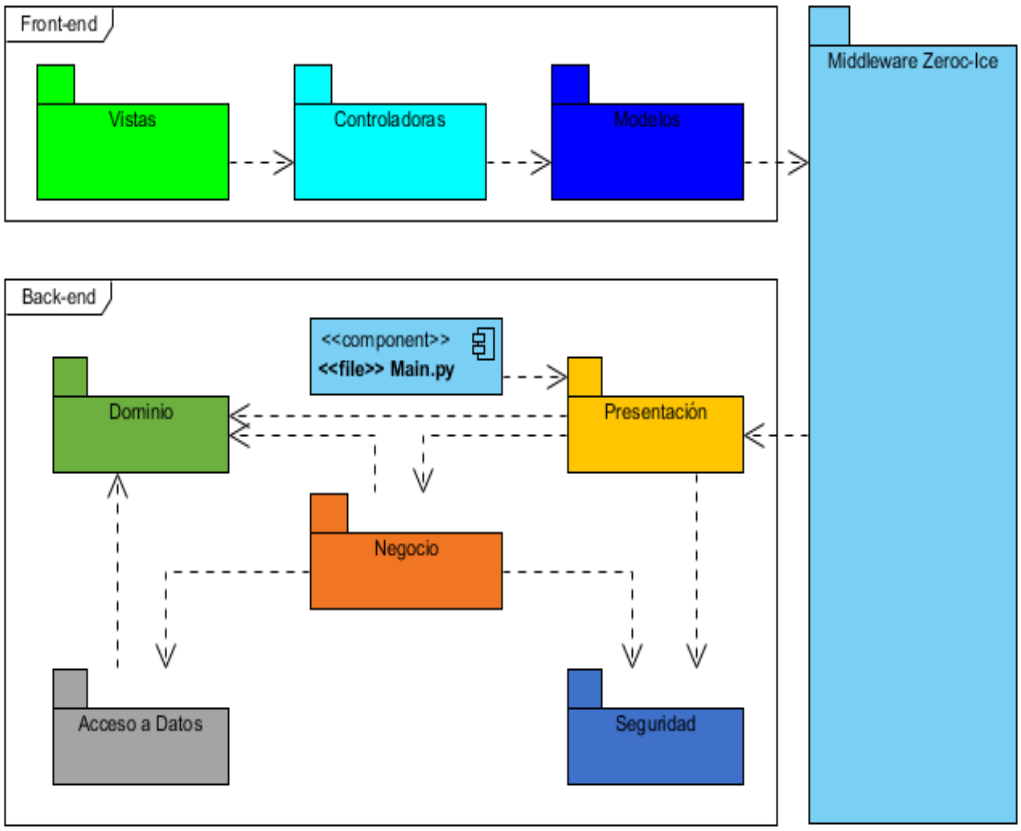


Fig. 22 Diagrama de componentes del sistema en general.

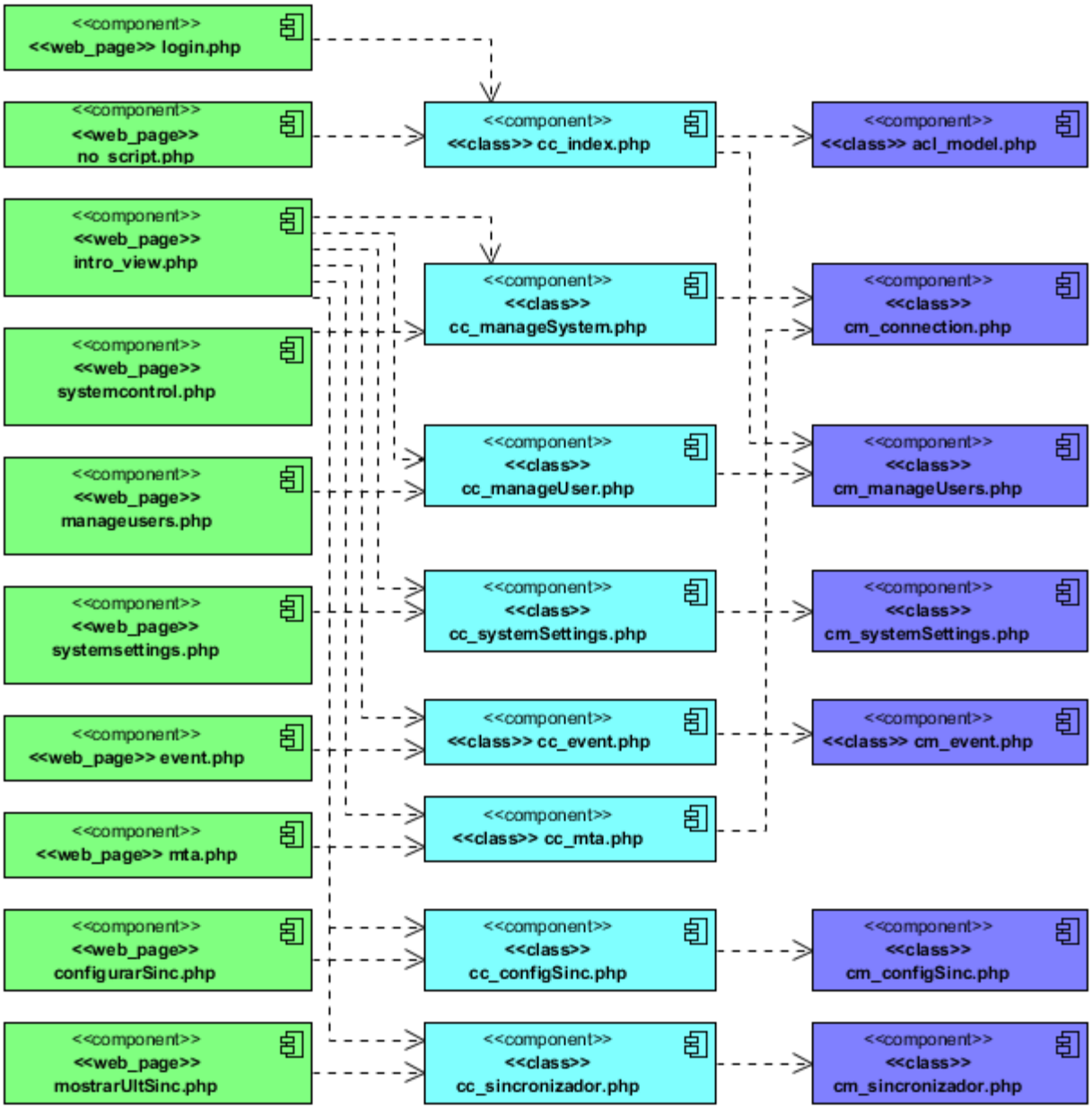


Fig. 23. Diagrama de componentes del front-end, componentes Web.

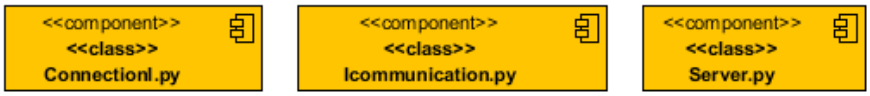


Fig. 24. Diagrama de componentes, Capa de presentación.

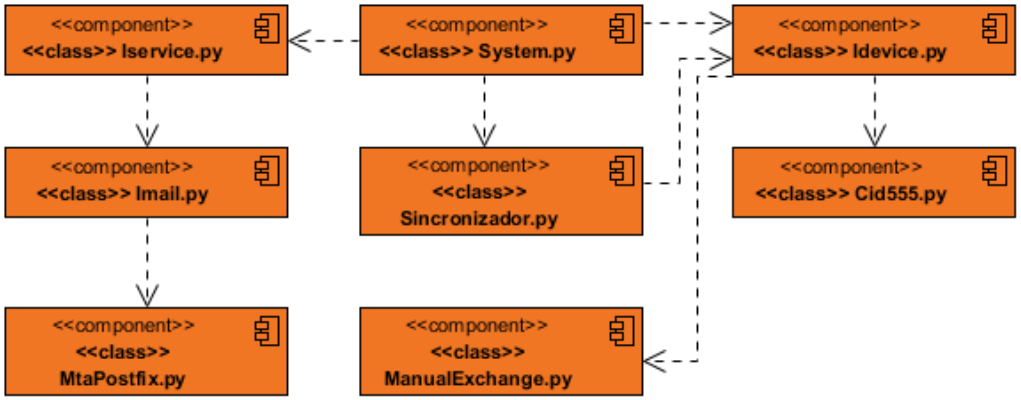


Fig. 25. Diagrama de componentes, Capa de Negocio.

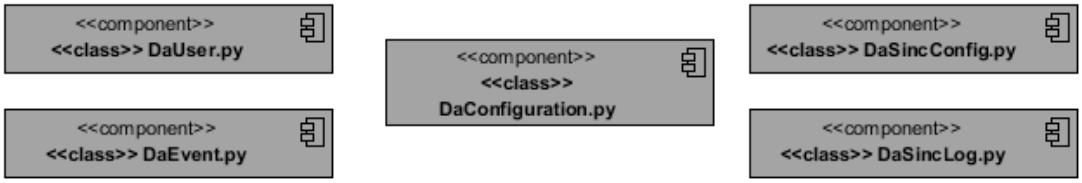


Fig. 26. Diagrama de componentes, Capa de Acceso a Datos.

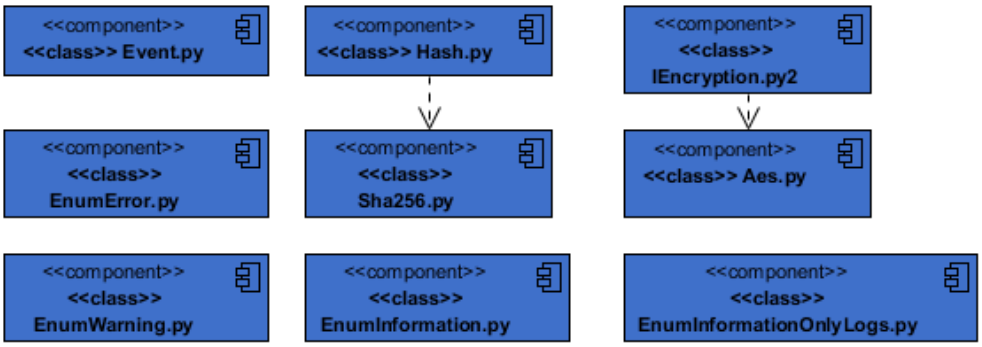


Fig. 27. Diagrama de componentes, Capa de Seguridad.

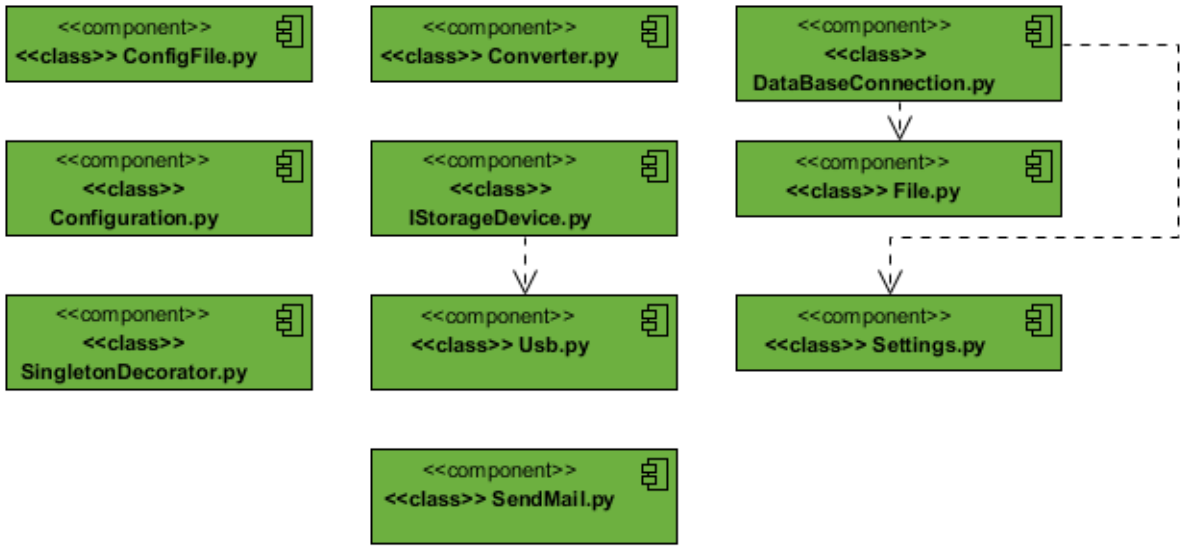


Fig. 28. Diagrama de componentes, Capa de Dominio.

4.5 Pruebas de software

Las pruebas de software son los procesos que permiten verificar y revelar la calidad de un producto de software. Son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un sistema informático. Básicamente es una fase en el desarrollo de software, consistente en probar las aplicaciones construidas. Para determinar el nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto a las especificaciones iniciales del sistema. (40)

4.5.1 Métodos de pruebas: Pruebas de caja negra

Se denominan pruebas de caja negra a un tipo de prueba de software que se realizan sobre la interfaz del software, tiene como objetivo verificar que la entrada se acepta de forma adecuada y que se produce un resultado correcto, sin tener en cuenta su funcionamiento interno. En estos casos el probador proporciona las entradas y estudia las salidas para ver si son o no las esperadas. Estas pruebas se centran en los requisitos funcionales del software. (40)

4.5.1.1 Técnica de pruebas de caja negra.

Para desarrollar las pruebas de caja negra existen varias técnicas, entre ellas se encuentra la **"Técnica de la Partición de Equivalencia",** esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software. Es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así en número de clases de prueba que hay que desarrollar.

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución, resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada. La entidad más simple que siempre es ejecutada como una unidad, desde el comienzo hasta el final.

Los casos de pruebas deben verificar:

Si el producto satisface los requerimientos del usuario, tal y como se describe en las especificación de los requerimientos.

Si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño.

Un caso de prueba se diseña según las funcionalidades descritas en los casos de uso. Este diseño se elabora previo a la realización de las pruebas funcionales de la aplicación. Cada planilla de caso de prueba recoge la especificación de un caso de uso, dividido en secciones y escenarios, para hacer más fructífera la ejecución de las pruebas. Los casos de prueba se encuentran a partir del anexo número 7.

4.6 Resultados de las pruebas.

Luego de haber aplicado las pruebas de caja negra al módulo realizando pruebas funcionales se detectaron 9 no conformidades en un total de 3 iteraciones, las cuales fueron resueltas a medida que fueron realizándose las mismas.

No	No conformidad	Ubicación	Estado
1	El campo IP del servidor FTP permite registrar datos en blanco.	Configurar módulo de sincronización de archivos. (Font-end)	Resuelta
2	El campo usuario del servidor FTP	Configurar módulo de	Resuelta

	validado como obligatorio.	sincronización de archivos. (Font-end)	
3	No se muestra la lista de las sincronizaciones realizadas.	Mostrar información de la sincronización. (Font-end)	Resuelta
4	No se eliminan las carpetas del servidor externo al realizar la sincronización.	Sincronizar (Back-end)	Resuelta
5	La lista de las sincronizaciones realizadas no se muestran ordenadas de la última a la primera.	Sincronizar (Back-end)	Resuelta
6	No se muestran las entradas del módulo de sincronización de archivos en el log.	Sincronizar (Back-end)	Resuelta
7	Cuando se presiona el botón reiniciar, el módulo de sincronización de ficheros no se reinicia correctamente.	Administrar Sistema (Front-end) Manejador de Sistema (Back-end)	Resuelta
8	No se comprueba el tamaño de la memoria antes de copiar los archivos.	Sincronizar (Back-end)	Resuelta
9	Los mensajes registrados en el log se muestran en inglés.	Sincronizar (Back-end)	Resuelta

Tabla 4: Resultados de las Pruebas.

4.7 Conclusiones parciales.

En este capítulo se mostraron y explicaron los artefactos generados durante la fase de implementación, el diagrama de despliegue, el diagrama de componentes y la implementación del sistema. Además se realizaron las pruebas necesarias para demostrar que el sistema funciona correctamente y cumple con los

requisitos especificados. Se detectaron un total de 9 no conformidades las cuales fueron resueltas a medida que se realizaron las iteraciones. Finalmente se culmina la implementación de la aplicación, dándose cumplimiento al objetivo general trazado inicialmente.

Conclusiones Generales:

Como resultado de la realización de este trabajo, se diseñó e implementó el Módulo sincronización de archivos dotando así al sistema de un mecanismo que facilita la sincronización de dos sistemas de archivos ubicados en redes aisladas físicamente entre sí. Además se le ha dado cumplimiento a los objetivos propuestos inicialmente obteniendo los siguientes resultados:

- Se logró automatizar la gestión de archivos desde un servidor interno a un servidor réplica externo, sincronizando la información en redes aisladas.
- Se realizó un análisis de soluciones informáticas similares existentes para el manejo de la información en redes aisladas.
- Se describieron las herramientas y tecnologías a utilizar para la implementación del módulo.
- Se generaron los artefactos relacionados con el diseño y la implementación del módulo sincronización de archivos.
- A la aplicación le fueron realizadas las pruebas de caja negra obteniéndose un total de 9 no conformidades, las cuales fueron resueltas en su totalidad.

Recomendaciones:

Luego de haber concluido el desarrollo del presente trabajo se recomienda:

- Incluir en soluciones posteriores la implementación de sincronización de servidores de control de versiones.

Referencias bibliográficas:

1. Definición. [En línea] http://es.scribd.com/doc/74942047/Redes-.

2. **Vega, Carlos Manuel Hernández.** CID-555: Pasarela de correos para la intercomunicación de redes aisladas. La Habana : s.n.

3. **Christoph Meinel.** HPI. [En línea] http://www.hpi.uni-potsdam.de/fileadmin/hpi/FG_ITS/lock-keeper/HPI-LK-Intro.pdf.

4. Windows. [En línea] http://windows.microsoft.com/es-ar/windows-vista/how-to-keep-your-information-in-sync.

5. **Dr.Christoph Meinel.** HPI. [En línea] 01 de Enero de 2013. http://www.hpi.uni-potsdam.de/meinel/projekte/lock_keeper.html .

6. **Cheng, Feng.** "Lock-Keeper - A High Security Solution based on Physical Separation" . 2008.

7. **Vazquez, Yosbel Morales y Bedoya, Bárbara Daisy.** Software para la interconexión de redes aisladas. Módulo correo electrónico v2.0. La Habana : s.n., 2012.

8. **Laura Barros Bastante.** Docstoc. [En línea] Julio de 2008. http://www.docstoc.com/docs/120926737/Implementaci%EF%BF%BDn-de-la-tecnolog%EF%BF%BDa-de-componentes-CCM-sobre-el.

9. **Henning, Michi y Spruiell, Mark.** Distributed Programming with Ice. 2006.

10. **Maria Elena de Lobos .** Emagister. [En línea] http://www.emagister.com/curso-aprende-programar/concepto-lenguaje-programaci.

11. Fcasua. [En línea] http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/4/informatica_4.pdf.

12. **Duque, Raúl Gonzáles.** Python para todos. España : s.n.

13. **Marza, Andrés y García, Isabel.** Introducción a la programación en Python.

14. **Enrique González .** Aprenderaprogramar. [En línea] http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70:tutorial-basico-programador-web-php-desde-cero&Itemid=193.

15. EVA. [En línea] http://eva.uci.cu/course/view.php?id=106.

16. **Rui-Perez, Ulises Cáceres.** Business Process Modeling Notation.

17. Business Process Modeling Notation.

18. **Owen, Martin.** BPMN and Business Process Management.

19. Activ. [En línea] http://activ.com.mx/introduccion-a-jquery-parte-1/.

20. **Aragón, Adys.** Diseño e Implementación del módulo Evaluación. La Habana : s.n., 2012.

21. **Pérez, Isaías Carrillo.** Metodología de Desarrollo de Software.

22. **Sommerville, Ian.** Ingeniería de software.

23. Herramientas Case. [En línea] 2 de Abril de 2011. http://fds-herramientascase.blogspot.com/.

24. V. Paradigm, Visual paradigm for uml. 2010.

25. Curso-sobre. [En línea] http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.html/eclipse.html.

26. **Appcelerator.** PyDev. [En línea] http://pydev.org/.

27. Ingeniería de Requisitos: conceptos,. Madrid : s.n., 2005.

28. **Dra. Anaisa Hernández González.** Scribd. [En línea] http://es.scribd.com/doc/13500172/actividad2-diagrama-de-casos-de-uso-del-negocio-y-del-sistema.

29. **Tello, Jesús Cáceres.** Diagramas de Casos de Uso.

30. **P´erez, Juan Diego.** Notaciones y lenguajes de procesos.

31. Dpto de lenguajes y sistemas informáticos. [En línea] http://www.lsi.us.es/~javierj/cursos_ficheros/metricaUML/EAActividades.pdf.

32. Bussines Dictionary. [En línea] http://www.businessdictionary.com/definition/front-end-application.html.

33. Comusoft. [En línea] http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas.

34. Alegsa. [En línea] http://www.alegsa.com.ar/Dic/back-end.php.

35. **Larman, Craig.** UML y Patrones.

36. Kde. [En línea] http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html.

37. Tecnología y synergix. [En línea] http://synergix.wordpress.com/2008/07/20/estereotipos-en-uml/.

38. Yoquese. [En línea] http://www.yoquese.com.ar/resources/external/material_analisis_y_diseno_de_sistemas/ExtensionUMLparaAplicacionesWeb.pdf.

39. Infolab. [En línea] http://infolab.stanford.edu/~ullman/focs/ch08.pdf.

40. EVA. [En línea] http://eva.uci.cu/mod/resource/view.php?id=9066..

41. Python. [En línea] [En línea] http://www.python.org/doc/essays/blurb.html ..

42. Sitio Oficial de PHP . . [En línea] [En línea] http://php.net/manual/en/intro-whatis.php..

43. Manual de PHP. [En línea] http://www.manualdephp.com/.

44. Rational Unified Process. [En línea] http://www.rational.com.ar/herramientas/rup.html..

45. **Laguna, Miguel A.** Requisitos.

46. Windows. [En línea] http://windows.microsoft.com/es-XL/windows-vista/How-to-keep-your-information-in-sync.

Bibliografía:

1. Definición. [En línea] http://es.scribd.com/doc/74942047/Redes-.

2. **Vega, Carlos Manuel Hernández.** CID-555: Pasarela de correos para la intercomunicación de redes aisladas. La Habana : s.n.

3. **Christoph Meinel.** HPI. [En línea] http://www.hpi.uni-potsdam.de/fileadmin/hpi/FG_ITS/lock-keeper/HPI-LK-Intro.pdf.

4. Windows. [En línea] http://windows.microsoft.com/es-ar/windows-vista/how-to-keep-your-information-in-sync.

5. **Dr.Christoph Meinel.** HPI. [En línea] 01 de Enero de 2013. http://www.hpi.uni-potsdam.de/meinel/projekte/lock_keeper.html .

6. **Cheng, Feng.** "Lock-Keeper - A High Security Solution based on Physical Separation" . 2008.

7. **Vazquez, Yosbel Morales y Bedoya, Bárbara Daisy.** Software para la intercomunicación de redes aisladas. Módulo correo electrónico v2.0. La Habana : s.n., 2012.

8. **Laura Barros Bastante.** Docstoc. [En línea] Julio de 2008. http://www.docstoc.com/docs/120926737/Implementaci%EF%BF%BDn-de-la-tecnolog%EF%BF%BDa-de-componentes-CCM-sobre-el.

9. **Henning, Michi y Spruiell, Mark.** Distributed Programming with Ice. 2006.

10. **Maria Elena de Lobos .** Emagister. [En línea] http://www.emagister.com/curso-aprende-programar/concepto-lenguaje-programaci.

11. Fcasua. [En línea] http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/4/informatica_4.pdf.

12. **Duque, Raúl Gonzáles.** Python para todos. España : s.n.

13. **Marza, Andrés y García, Isabel.** Introducción a la programación en Python.

14. **Enrique González .** Aprenderaprogramar. [En línea] http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70:tutorial-basico-programador-web-php-desde-cero&Itemid=193.

15. EVA. [En línea] http://eva.uci.cu/course/view.php?id=106.

16. **Rui-Perez, Ulises Cáceres.** Business Process Modeling Notation.

17. Business Process Modeling Notation.

18. **Owen, Martin.** BPMN and Business Process Management.

19. Activ. [En línea] http://activ.com.mx/introduccion-a-jquery-parte-1/.

20. **Aragón, Adys.** Diseño e Implementación del módulo Evaluación. La Habana : s.n., 2012.

21. **Pérez, Isaías Carrillo.** Metodología de Desarrollo de Software.

22. **Sommerville, Ian.** Ingeniería de software.

23. Herramientas Case. [En línea] 2 de Abril de 2011. http://fds-herramientascase.blogspot.com/.

24. V. Paradigm, Visual paradigm for uml. 2010.

25. Curso-sobre. [En línea] http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.html/eclipse.html.

26. **Appcelerator.** PyDev. [En línea] http://pydev.org/.

27. Ingeniería de Requisitos: conceptos,. Madrid : s.n., 2005.

28. **Dra. Anaisa Hernández González.** Scribd. [En línea] http://es.scribd.com/doc/13500172/actividad2-diagrama-de-casos-de-uso-del-negocio-y-del-sistema.

29. **Tello, Jesús Cáceres.** Diagramas de Casos de Uso.

30. **P´erez, Juan Diego.** Notaciones y lenguajes de procesos.

31. Dpto de lenguajes y sistemas informáticos. [En línea] http://www.lsi.us.es/~javierj/cursos_ficheros/metricaUML/EAActividades.pdf.

32. Bussines Dictionary. [En línea] http://www.businessdictionary.com/definition/front-end-application.html.

33. Comusoft. [En línea] http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas.

34. Alegsa. [En línea] http://www.alegsa.com.ar/Dic/back-end.php.

35. **Larman, Craig.** UML y Patrones.

36. Kde. [En línea] http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html.

37. Tecnología y synergix. [En línea] http://synergix.wordpress.com/2008/07/20/estereotipos-en-uml/.

38. Yoquese. [En línea] http://www.yoquese.com.ar/resources/external/material_analisis_y_diseno_de_sistemas/ExtensionUMLparaAplicacionesWeb.pdf.

39. Infolab. [En línea] http://infolab.stanford.edu/~ullman/focs/ch08.pdf.

40. EVA. [En línea] http://eva.uci.cu/mod/resource/view.php?id=9066..

41. Python. [En línea] [En línea] http://www.python.org/doc/essays/blurb.html ..

42. Sitio Oficial de PHP . . [En línea] [En línea] http://php.net/manual/en/intro-whatis.php..

43. Manual de PHP. [En línea] http://www.manualdephp.com/.

44. Rational Unified Process. [En línea] http://www.rational.com.ar/herramientas/rup.html..

45. **Laguna, Miguel A.** Requisitos.

46. Windows. [En línea] http://windows.microsoft.com/es-XL/windows-vista/How-to-keep-your-information-in-sync.

47. EVA. [En línea] eva.uci.cu/mod/resource/view.php?id=9066..

48. **Pressman, Roger S.** Ingeniería del Software.Un enfoque práctico.Cuarta edición.

49. Msdn. [En línea] http://msdn.microsoft.com/es-es/library/dd409360.aspx.

50. Slideshare. [En línea] http://www.slideshare.net/camiloan40/diagrama-de-actividades-uml.

51. **Rolando Jaldin Rosales.** [En línea] http://rolandojaldin.blogspot.com/2010/10/implementacion-fase-de-construccion-rup.html.

52. EVA. [En línea] http://eva.uci.cu/mod/resource/view.php?id=4342.

53. eva. [En línea] http://eva.uci.cu/file.php/104/Tema_1/Conferencia_1/Materiales/Apuntes_de_metodologia_de_la_investigacion_Velez_2005.pdf.

Glosario de términos:

Middleware: Es un software de conectividad que consiste en un conjunto de servicios que permiten interactuar a múltiples procesos que se ejecutan en distintas máquinas a través de una red. Ocultan la heterogeneidad y proveen de un modelo de programación conveniente para los desarrolladores de aplicaciones.

Lenguaje interpretado o de script: Un lenguaje interpretado o de script es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje máquina que pueda comprender y ejecutar directamente una computadora (lenguajes compilados).

Spyware (software espía): Es un software que recopila información de un ordenador y después transmite esta información a una entidad externa sin el conocimiento o el consentimiento del propietario del ordenador.

Cookies: Una cookie es un fragmento de información que un navegador web almacena en el disco duro del visitante a una página web. La información se almacena a petición del servidor web, ya sea directamente desde la propia página web con JavaScript o desde el servidor web mediante las cabeceras HTTP, que pueden ser generadas desde un lenguaje de web scripting como PHP. La información almacenada en una cookie puede ser recuperada por el servidor web en posteriores visitas a la misma página web.

GoF: Gang of Four hace referencia al nombre con el que comúnmente se conoce a los autores del libro Design Patterns. Estos patrones de diseño se clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento.

GRASP: Son Patrones Generales de Software para Asignación de Responsabilidades (General Responsibility Assignment Software Patterns), aunque son considerados que más que patrones propiamente dichos pues constituyen una serie de buenas prácticas recomendables en el diseño de software.