

Universidad de las Ciencias Informáticas

Facultad 1

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

Título:

Herramienta para la Migración y Administración de Servidores (HMAS).

Autores:

Reidiel Castillo Arbelo.

Pablo Soria Acosta.

Tutores:

Msc. Siovel Rodríguez Morales.

Ing. Amaury Viera Hernández.

Consultante:

Ing. Yoandy Pérez Villazón.

Ciudad de la Habana, Cuba.
Junio de 2012.

Declaración de Autoría

Declaramos ser los únicos autores de este trabajo y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 12 días del mes de Junio del año 2012.

Reidiel Castillo Arbelo

Pablo Soria Acosta

MSc. Siovel Rodríguez Morales

Ing. Amaury Hernández

Agradecimientos

Reidiel Castillo Arbelo:

*A mi mamá **Adelaida** que me ha dedicado toda su vida. A mis hermanos, en especial a **Yunierkis, Elianet y Reiniel** que siempre me han apoyado en todas mis desiciones y nunca me han negado su ayuda. A toda mi familia, la cual incluye a mi cuñada **Yarima**.*

*A mi compañero de tesis **Pablo**, el mejor programador que conozco. A **Roberto** que en estos 5 años se ha comportado como un hermano. A **Indy** que ha sido mi amiga durante estos 5 años y siempre me ha brindado su amistad. A mis tutores **Amaury y Siovel** que me han brindado su ayuda durante el desarrollo de la tesis. A mis mejores amistades en estos últimos años **Mairim y Adis**. A mis compañeros de aula que han compartido momentos buenos y malos: **Goar, Ricardo, Hector, Guille, Jorge, Tomás, Machín, Yasmany, Carbajo, Sandra, Dairelis, Aylena, Zulema, Anneris** y otros. A mis compañeros de proyecto con los cuales he trabajado en estos 5 años.*

Pablo Soria Acosta:

*A toda mi familia, porque siempre me han puesto en lo más alto, dándome un gran voto de confianza, en especial a mis primos **Jaimito y Josué**. A **Adisleydis**, porque fue mi soporte, mi estímulo para seguir adelante, porque nunca me fallo cuando lo necesite, y sobre todo, por su paciencia para entenderme cuando me hizo falta.*

*A todas las personas maravillosas que he conocido durante estos cinco cursos. Gracias a ustedes: **Carlos Santana** por ser un amigo incondicional. A **Mairim** por su apoyo. A **Reidiel** por ayudarme y ser mi compañero de tesis. A **Yoandy y sus hermanos** porque me han sido de gran ayuda y nunca me han dado la espalda. A los amigos **Jenlys, Elier, Eddy, Maydelis**, en general a todos los del grupo, porque hemos sido una gran familia, nunca los voy a olvidar, son muy especiales. Mis tutores **Amaury y Siovel** porque realmente fueron de gran ayuda para la realización de este trabajo y porque siempre nos dieron su apoyo. A Todas las personas que desde Santiago siempre me ayudaron y me alentaron para que siguiera adelante: **Abelardo, Aleida, Lisbet, Jeidis, Miguelito**, en fin, a todos los que me impulsaron siempre hacia arriba. A mis compañeros de proyecto como **Lara, Jailen, Alexander, Javier, Carlos, Eduardo** y en general a todos con los cuales he trabajado en estos 5 años. A **Ana Evis, Analaisy (nana), Iván, Roberto Menendez y Ángel** por ser buenos amigos. A mi antiguo grupo 10106 y al actual 01508. A la **Revolución**, por darme la oportunidad de estudiar y de hacerme un ingeniero.*

Dedicatoria

Reidiel Castillo Arbelo:

*Le dedico este trabajo las personas que más amo en este mundo, a mi mamita **Adelaida**, a mi hija **Yeily Danay**, a mi hermana **Elianet** y a mis hermanos **Yunierkis** y **Reiniel**. Este título se lo debo a ustedes, muchas gracias.*

Pablo Soria Acosta:

*Muchas personas han intervenido de diversas formas para que yo haya podido llegar a este punto y por eso le dedico este trabajo a: Mi madre **Marina Acosta Rodriguez**, mi hermano **Jardiel Soria Acosta** y a mi tía **Aleida Acosta**, por haberme dedicado su vida, porque siempre han confiado en mí y me han dado su apoyo en todo momento.*

Resumen

Los procesos de migración realizados por el departamento de Servicios Integrales en Migración, Asesoría y Soporte (SIMAYS) en las diferentes instituciones cubanas, requieren el cambio de las plataformas privativas instaladas en los servidores para sistemas GNU/Linux. En la actualidad esta labor se está realizando manualmente, provocando que sea engorroso el mismo y ocasionando muchas veces la pérdida de datos. Además, debido a las bajas prestaciones que tienen los servidores en estas empresas, constituye un problema su administración sobre tecnologías libres ya que las herramientas existentes no cumplen con algunos aspectos importantes entre los cuales se destacan la seguridad, el consumo de recursos, la usabilidad, y otros.

En la presente investigación se desarrolla una aplicación de escritorio, sobre tecnologías libres, que automatiza el proceso de migración de los servidores y su administración. Para esto, se realiza un estudio de las principales herramientas existentes para la administración y migración de servidores con el objetivo de obtener sus mejores funcionalidades y adicionarlas al producto final. También se documentan las tecnologías, herramientas y lenguajes de programación utilizados, todo guiado por la metodología de desarrollo SXP.

La aplicación desarrollada cuenta con diferentes secciones para la administración de los servidores entre los cuales se encuentra la gestión de usuarios y grupos, la gestión de tareas programadas, configuración de la red, entre otros. Además, se le desarrollaron dos módulos relacionados con los servicios telemáticos, uno para la administración del SSH y otro para el DHCP, siendo este último, capaz de automatizar el proceso de migración del mismo.

Palabras claves: Administración, DHCP, Migración de servidores, Software Libre, SSH.

Índice

Introducción.....	1
Capítulo 1: Fundamentación teórica.....	6
1.1 Herramientas para la administración de servidores.	6
1.1.1 WebMin.....	6
1.1.2 YAST.....	8
1.1.3 Zentyal.....	10
1.2 Herramientas para la migración de servidores.....	14
1.3 Lenguaje de programación.....	15
1.4 Herramientas para el desarrollo.....	17
1.4.1 Bibliotecas.....	17
1.4.2 Herramientas de programación.	19
1.4.3 Herramientas CASE.....	19
1.5 Herramientas auxiliares.....	21
1.6 Metodología ágil utilizada.....	22
Capítulo 2: Análisis y Diseño.....	25
2.1 Propuesta del sistema a desarrollar.....	25
2.2 Propuesta de módulos para el sistema.....	26
2.3 Roles.....	26
2.4 Requerimientos funcionales.....	26
2.5 Historias de Usuario (HU).....	30
2.6 Estructuración de los componentes.....	36
2.6.1 Fundamentación de los Patrones de Arquitectura.....	36
2.6.2 Arquitectura de los módulos.....	40
2.7 Reutilización de componentes.....	40
Capítulo 3 Implementación y Pruebas.....	41
3.1 Implementación.....	41
3.1.1 Diagrama de Componentes.....	41
3.1.2 Diagrama de Despliegue.....	42
3.1.3 Tareas de Ingeniería.....	42
3.2 Prueba.....	48
Conclusiones.....	66
Recomendaciones.....	67
Referencia Bibliográfica.....	68
Glosario de Términos.....	71
Anexos.	74

Introducción.

En la actualidad, la informatización de la sociedad se ha convertido en uno de los principales eslabones en el desarrollo de la humanidad, ayudando a elevar la productividad y efectividad en las ramas a las cuales se aplica. Además, muestra varios aspectos que la singularizan con relación a otras actividades productivas pues, se trata de una industria generadora de intangibles¹, esencialmente globalizada e intensiva en capital humano.

Es, así mismo y de manera muy significativa, una industria de carácter transversal, que produce resultados y transforma mucha de las actividades que tienen lugar en la sociedad moderna. Se trata, por otro lado, de una industria que se encuentra en manos de grandes empresas transnacionales, fuertemente concentrada tanto en lo que respecta a la producción como al consumo, en Estados Unidos, Alemania y Japón. No obstante, hay países en desarrollo que cuentan con una industria de software significativa, desde el punto de vista económico. Se trata de países de ingreso tardío, presentes en la industria después de que las potencias mencionadas llevaban un largo camino andado [1].

Como dijo el compañero Fidel Castro en su discurso pronunciado en el Palacio Central de Computación por el 15 aniversario de ese centro: *La informática se convertirá en una poderosa fuerza científica, económica y política para Cuba* [2]. Por esta razón se han trazado diferentes estrategias en esta rama, con el fin de lograr la inserción del software cubano en el mercado mundial. Entre ellas, se destaca la creación de la Universidad de Ciencias Informáticas (UCI), la cual cuenta con un nuevo modelo (Formación-Investigación-Producción) que le permite la producción de software y servicios informáticos tanto para las instituciones cubanas como para la exportación.

La producción de la universidad se concentra en el desarrollo de proyectos pertenecientes a los 14 centros productivos existentes en la actualidad. Entre ellos, se encuentra el Centro de Software Libre (CESOL), el cual está conformado por el Departamento de Sistemas Operativos y el Departamento de Servicios

¹ **Intangible:** es un adjetivo que califica todo aquello que no tiene una presencia física, lo que no puede (o debe) ser tocado jamás.

Integrales en Migración, Asesoría y Soporte (SIMAYS). Este último especializado en brindar servicios relacionados con la migración a código abierto como *Asesoría, Consultoría, Capacitación y Soporte Técnico*. Además, desarrolla aplicaciones para apoyar el proceso de migración a Software Libre y aplicaciones de código abierto, define también los lineamientos que rigen el proceso de migración en las empresas cubanas.

En la actualidad, el país se encuentra inmerso en un proceso de migración. Este comenzó en Abril de 2004, con la aprobación del “Acuerdo 084/2004” por el Consejo de Ministros, donde se indica al Ministerio de la Informática y las Comunicaciones (MIC) ordenar el proceso paulatino de migración de Cuba a Software Libre. Posteriormente se crea en el año 2005 el Grupo Ejecutivo Nacional para la Migración presidido por el ministro del MIC, en el cual el departamento SIMAYS tiene un papel fundamental pues pertenece al Grupo Técnico para la Migración definido en el mismo.

El proceso de migración se ha realizado en entidades cubanas como: Empresa de Proyectos (EPRO), Empresa Nacional de Investigaciones Aplicadas (ENIA), Diseño Ciudad Habana (DCH), entre otras. Estas empresas se asemejan desde el punto de vista informático, por tener menos de 100 usuarios o computadoras y una infraestructura de red compuesta por servidores de bajas prestaciones². También presentan baja conectividad a Internet debido a que se efectúa mediante módem de bajos recursos y un ancho de banda reducido, siendo lenta además, las conexiones entre máquinas y servidores dentro de la institución. Las organizaciones que presenten estas características serán tratadas en lo adelante como PYMES³ cubanas, centros a los cuales va destinado el software a implementar.

Las PYMES cubanas tienen instalado en la mayoría de sus servidores sistemas operativos privativos, como Windows Server en alguna de sus variantes, lo que provoca la existencia de una gran dependencia tecnológica hacia estas plataformas. Por lo que uno de los retos para estos organismos lo constituye la

² **Servidor de bajas prestaciones:** servidor que cuenta con 1 o 2 GB de RAM, menos de 2 GHz de procesamiento, tarjeta de red de 100 Mbyte/s, etc.

³ **PYMES:** (Acrónimo de pequeña y mediana empresa). Empresa mercantil, industrial, etc., compuesta por un número reducido de trabajadores, y con un moderado volumen de facturación. tomado del diccionario de la Real Academia Española.

migración a GNU/Linux de los servicios telemáticos que brindan y la administración de los mismos.

Una vez planteado lo anterior, se identifica el siguiente **problema científico**:

¿Cómo brindar una solución a la medida para realizar la migración y administración de los servicios telemáticos necesarios en las pequeñas y medianas empresas cubanas?

El **objeto de estudio** del presente trabajo de investigación se centra en las herramientas para la migración y administración de los servicios telemáticos. Enmarcándose en el **campo de acción** las herramientas para la migración y administración de los servicios telemáticos en las pequeñas y medianas empresas cubanas.

Por lo que se propone como **objetivo general** desarrollar una herramienta que permita la migración y administración de servicios telemáticos en las pequeñas y medianas empresas cubanas.

En el marco de esta investigación se ha desglosado el objetivo general en los siguientes **objetivos específicos**:

- Sistematizar en el estudio de herramientas para la migración y administración de servicios telemáticos.
- Realizar el análisis y diseño de la aplicación a desarrollar.
- Implementar el software requerido.
- Realizar pruebas al sistema desarrollado.

Para dar cumplimiento a los objetivos específicos se planifican las siguientes **tareas de investigación**:

- Sistematización sobre las herramientas para la administración de servicios telemáticos.
- Sistematización sobre las herramientas para la migración de servicios telemáticos.
- Definición de la arquitectura del sistema y patrones de diseño a utilizar.

- Desarrollo de la herramienta definida.
- Realización de pruebas a la herramienta desarrollada.

Idea a defender:

La creación de una herramienta que permita migrar los servicios telemáticos a plataformas libres y administrarlos, eliminará la dependencia tecnológica de las herramientas privativas existentes en las PYMES cubanas.

En el desarrollo de la investigación es utilizado el siguiente **método científico**:

Analítico – Sintético: Se utiliza en la investigación de las herramientas para la administración y migración de servidores existentes. Ayuda a separar los mismos con el objetivo de realizar un estudio del funcionamiento y las configuraciones necesarias de estos, facilitando el desarrollo de la aplicación propuesta.

El documento está confeccionado por 3 capítulos, las conclusiones generales, la bibliografía general utilizada, el glosario de términos donde se explican los términos de difícil comprensión que se han utilizado en la elaboración de esta investigación y los anexos. La estructura de los capítulos se define a continuación:

Capítulo 1: Fundamentación teórica: Se realiza un estudio acerca de las herramientas que permiten migrar y administrar servidores, así como sus principales características y funcionalidades. Además, se abordan conceptos claves que serán usados durante el desarrollo de la investigación y se fundamentan las tecnologías, lenguajes de programación, metodología y herramientas utilizadas en el desarrollo del sistema.

Capítulo 2: Diseño del sistema. Se fundamentan los software que serán usados en los servicios a administrar. Además, se presenta la fase de Planificación-Definición definida por la metodología SXP para dar solución al problema científico y se exponen los requisitos funcionales y no funcionales del sistema

propuesto. También se realiza la descripción de las Historias de Usuario, se elabora la Lista de Riesgos y se explica la arquitectura del sistema.

Capítulo 3: Implementación y pruebas. Se implementa la solución propuesta. Además, se explican las clases principales del sistema, se confecciona el plan de pruebas y se realizan las pruebas necesarias para validar que las funcionalidades desarrolladas dan cumplimiento a los requisitos planteados.

Capítulo 1: Fundamentación teórica.

En este capítulo, se realiza un estudio de las herramientas que permiten administrar servicios telemáticos que se encuentran sobre sistemas operativos GNU/Linux. Se documentan las principales funcionalidades de los mismos, así como las deficiencias que impiden su utilización en las empresas cubanas. También se analiza de los lenguajes de programación cual es el más idóneo para el desarrollo del sistema, lo mismo se hace para las herramientas utilizadas. Además, se fundamenta el uso de la metodología de desarrollo SXP en la investigación.

1.1 Herramientas para la administración de servidores.

Actualmente se encuentran una enorme variedad de aplicaciones que permiten la administración de los servicios telemáticos, estas pueden ser web, de escritorio y otras que tiene su interfaz desarrollada para el trabajo en la terminal (Ncurses). A continuación se describen las principales herramientas que realizan esta función, detallándose las principales funcionalidades y desventajas de cada una.

1.1.1 WebMin.

Webmin es una interfaz web para administrar y configurar sistemas con distribuciones GNU/Linux de forma local y remota. Se puede abrir mediante cualquier navegador que soporte tablas, formularios y Java para el módulo de administración de archivos, usando una conexión mediante SSL por el puerto 10000. El mismo requiere de un servidor web y un gran número de script escritos en Perl, lenguaje en el cual está desarrollado.

Esta herramienta permite la configuración de la mayoría de los servicios telemáticos mediante módulos entre los cuales se encuentra: Apache WebServer, BIND DNS Server, CVS Server, DHCP Server, FTP, Proxy, Jabber, entre otros, para un total de 27 módulos. También cuenta con 15 módulos para la gestión de red de la computadora en la cual se encuentra instalado, 10 para la gestión del hardware, 9 para el

trabajo con Cluster, 20 para el trabajo con el sistema y otros 32 que no entran en ninguna de las categorías anteriores, siendo un total de 113 módulos que hasta el momento tiene desarrollado según el sitio oficial del Webmin [4].

Esta herramienta brinda la posibilidad de configurar sus módulos, permitiendo cambiar el acceso a los ficheros con los que trabaja, los cuales al ser modificados pueden ocasionar que el módulo no funcione correctamente. Lo mismo sucede con los archivos de configuración de los servicios, que también pueden ser editados.

La ayuda que se ofrece a los usuarios no está completa, pues en algunos casos como el módulo de SSH se explica solo los términos que son pocos usados, representando estos un bajo por ciento respecto al número de configuraciones que se muestra en la interfaz. En otros módulos la ayuda no está implementada de igual forma, pues se ofrece la del servicio que se esté trabajando y no cómo trabajar en la ventana que se encuentra situado, siendo engorroso el trabajo con el mismo.

Los autores consideran que el Webmin no es adaptable a las PYMES cubanas, debido a que, el trabajo con el mismo es algo complicado por la cantidad de configuraciones que tiene en cada interfaz de usuario. Además los módulos desarrollados hace varios años no han sido actualizados más, por lo que puede causar problemas cuando se va a administrar un servicio que en Webmin tiene una versión y los repositorios tienen otra. Esta herramienta tiene en cuenta cada detalle de configuración de los servicios de redes, por lo que para usar el mismo es necesario que se tengan conocimientos avanzados respecto a la administración de servidores. Tampoco está escrita la ayuda sobre el trabajo en todos los módulos.

Además, como se había mencionado anteriormente, el trabajo con este programa se realiza vía web, conectándose al servidor por el puerto 10000, mediante el protocolo SSL. En Cuba, los mecanismos que usan SSL mediante la generación de certificados en un servidor, tienen la desventaja de no contar con los servicios de una entidad certificadora, de prestigio internacional, que genere el certificado necesario para establecer conexiones seguras, por lo que pueden ser víctimas de ataques de suplantación de identidad.

Otro aspecto importante a tener en cuenta es que deben instalarse en cada servidor de la red, lo cual es innecesario cuando solo se quiere tener un servidor dedicado, como un Proxy o un DNS. También requiere que las conexiones a los mismos se realicen mediante el usuario root del sistema y no otro.

1.1.2 YAST.

YaST (acrónimo de **Y**et **a**nother **S**etup **T**ool, cuya traducción es "Otra Herramienta de Configuración Más"), es la herramienta de instalación y configuración de openSUSE y SUSE Linux Enterprise. Es popular por su facilidad de uso y atractiva interfaz gráfica, así como por la capacidad de configurar el sistema rápidamente durante la instalación y/o una vez terminado esta. Es un proyecto de código abierto, el cual está bajo la licencia GPL⁴, patrocinado y desarrollado activamente por Novell⁵.

YaST es la herramienta con la que interactúa durante la instalación de openSUSE. Se utiliza también para configurar sus dispositivos, el entorno de red, los servicios del sistema, la seguridad y muchas otras cosas. Todas estas tareas pueden realizarse desde el Centro de Control de YaST.

YaST2 es la interfaz desarrollada en QT, para la gestión de los servicios telemáticos que ofrece dicho sistema operativo. El mismo está conformado de forma modular, pero en este caso los módulos no se activan y desactivan como en otras herramientas similares, sino que hay que instalar y desinstalar cada uno a medida que se necesiten. La instalación de un módulo no significa que se instale el servicio, pues esa opción la brinda cuando se quiere configurar el mismo.

Entre los módulos para la administración de servidores (según información publicada en el sitio oficial de openSuse⁶) se pueden encontrar los siguientes:

- DNS y Nombre del Host: Configurar el Nombre del Host (Hostname) y nombre de servidores.

⁴ **GPL:** General Public License

⁵ **Novell:** es líder global de software, ofreciendo soluciones que hacen que los ambientes de trabajo más productivo, seguro y manejable.

⁶ **OpenSuse:** <http://www.opensuse.org/es/>

- Servidor DHCP: Configurar un Servidor DHCP.
- Servidor HTTP: Configurar un Servidor Web.
- Hostnames: Asignar nombres de host y alias para las direcciones IP.
- Cliente Kerberos: Configurar un cliente Kerberos.
- Cliente LDAP: Configurar el cliente LDAP.
- Agente de transferencia de correo: Configurar correo.
- Servidor NFS: Configurar un servidor NFS.
- Servidor NIS: Configurar un servidor NIS.
- Cliente NTP: Configurar un cliente NTP.
- Servicios de Red (xinetd): Configurar servicios de Red (xinetd).
- Proxy: Configurar un servidor Proxy.
- Administración remota: Instalar la configuración remota.
- Encaminamiento o Routing: Configurar un Routing.
- Servidor Samba: Configurar el servidor Samba.
- Servidor TFTP: Configurar un servidor TFTP.

Cuando se ejecuta algún módulo de los antes expuestos, lo primero que hace es verificar si está instalado y leer los archivos de configuración para comprobar los cambios realizados desde la última vez que se ejecutó. En caso de detectar alguno, informa al usuario donde se encuentra la modificación y advierte la posibilidad de que pueden ser borrados.

La ayuda en los módulos antes mencionados está conformada por pequeños diálogos, donde se describen las opciones que se muestran en la ventana que esté activa en el momento que se seleccionó la opción Ayuda. Aunque siempre brinda la posibilidad de ver la ayuda, esta no suele ser lo más explicativa posible, sino que, muestra lo mismo que se tiene en pantalla sin decir para qué se puede usar cada opción de configuración mostrada.

Los autores consideran que esta herramienta no se puede adaptar a las características que tienen las

PYMES cubanas ya que solo está desarrollado para las diferentes versiones de openSUSE, por lo tanto, habría que instalar este sistema operativo en los servidores y no la distribución cubana Nova Server⁷ que es la propuesta para estas empresas. Además YaST fue creado principalmente para la administración completa del sistema operativo desde una sola aplicación, no para la administración de servidores.

1.1.3 Zentyal.

El desarrollo de Zentyal se inició en el año 2004 con el nombre de eBox Platform⁸ y actualmente es una solución consolidada de reconocido prestigio que integra más de 30 herramientas de código abierto para la administración de sistemas y redes en una sola tecnología. Está incluido en Ubuntu⁹ desde el año 2007, en la actualidad tiene más de 1.000 descargas diarias y dispone de una comunidad activa de más de 5.000 miembros [7].

Zentyal se desarrolló con el objetivo de acercar GNU/Linux a las PYMES y permitirles aprovechar su potencial como servidor de empresa. Es la alternativa en código abierto a *Windows Small Business Server*, basado en la distribución Ubuntu. Este sistema permite a los informáticos de una institución, administrar todos los servicios telemáticos necesarios en su infraestructura de servidores, tales como el acceso a Internet, la seguridad de la red, la compartición de recursos, las comunicaciones, entre otros, de forma sencilla y a través de una única plataforma.

Durante su desarrollo se hizo énfasis en la usabilidad, creando una interfaz intuitiva¹⁰ que incluye únicamente aquellas funcionalidades de uso más frecuente, aunque también dispone de los medios necesarios para realizar toda clase de configuraciones más complejas.

Otra de las características más importantes de este sistema es que todas sus funcionalidades, construidas

⁷ Nova Server: Sistema operativo libre cubano, creado por el proyecto Nova de la UCI.

⁸ **eBox Platform:** Servidor de linux para PYMES, creado en el 2004 por eBox Technologies S.L.

⁹ **Ubuntu:** es una distribución de *Linux* desarrollada por *Canonical* y la comunidad orientada a ordenadores portátiles, de sobremesa y servidores: <http://www.ubuntu.com/>.

¹⁰ **Interfaz Intuitiva:** Es aquella interfaz fácil de trabajar, donde el usuario final sabe que tiene que hacer en cada opción que se muestre, pues generalmente son comunes a todos los usuarios.

a partir de una serie de aplicaciones en principio independientes, están estrechamente integradas entre sí, automatizando la mayoría de las tareas y ahorrando tiempo en la administración de sistemas. Teniendo en cuenta que el 42 % de los fallos de seguridad y el 80 % de los cortes de servicio en una empresa se deben a errores humanos en la configuración y administración de los mismos. [7] El resultado es una solución no solo más sencilla de manejar, sino también más segura y fiable que mejora la seguridad y disponibilidad de los servicios en una empresa.

Este producto informático tiene un conjunto de funcionalidades entre los cuales se destacan los siguientes:

- Gestión de red: cortafuego, servidor DHCP, servidor NTP, servidor DNS, soporte para VPN, proxy HTTP, entre otros.
- Servidor de correo: POP3 e IMAP con SSL/TLS, filtro antispam y antivirus, webmail, etc.
- Comunicaciones: FTP, servidor VoIP, Voicemail y servidor Jabber de mensajería instantánea.
- Compartición de recursos y trabajo en grupo: servidor de archivos, servidor de impresión y groupware (agenda, contactos, etc.).
- Gestión centralizada de usuarios mediante LDAP, sincronización con Directorio de Windows, etc.
- Reportes y monitoreo del sistema, además del envío de notificaciones vía correo, RSS o Jabber.
- Respaldos de configuraciones y datos de manera remota.
- Autoridad de certificación.

La gestión de Zentyal se realiza por medio de un navegador web, por lo tanto, resulta accesible desde cualquier equipo que se disponga en la red. Al ser una solución modular resulta lo bastante flexible para cubrir cualquier necesidad que se tenga en la empresa, ya sea para crear una infraestructura más segura o un servidor de impresión dedicado.

La fuente de ayuda principal de este programa se encuentra publicada en Español y en Inglés en su sitio oficial¹¹. Esta documentación explica bien lo que se debe hacer en cada interfaz, tiene imágenes que

¹¹ **Sitio de oficial de Zentyal:** <http://doc.zentyal.org/es/>

ayudan a comprender mejor el trabajo con Zentyal en cada servicio. También cuenta con una ayuda sencilla que se muestra en la parte superior izquierda de la mayoría de sus módulos, pero esta solo describe, de forma muy general, para qué sirve el módulo en el que se encuentra y no qué se puede hacer en cada opción de configuración necesaria.

Los autores de este trabajo consideran que este sistema no se puede adaptar a las necesidades de las PYMES cubanas, pues a diferencia del Webmin que tenía muchas configuraciones, este tiene muy pocas con respecto a las que se necesitan. Ejemplo de esto es cuando se adiciona un *host* virtual, no permite cambiar la dirección donde va a estar lo que se quiere compartir. Otro ejemplo que se puede mencionar es en la configuración del servidor FTP, el cual solo permite compartir el *home*¹² de los usuarios y no otro directorio.

También hay que destacar que este sistema no reconoce los cambios que se realicen directamente en los ficheros, por lo que todo lo que se necesite hacer tiene que ser por medio de su interfaz. En caso de escribir directamente en los ficheros, Zentyal lo borra cuando guarda sus configuraciones. Esto puede causar problemas ya que muchos administradores tienen ficheros de configuración guardados y cuando quieren restablecerlos no pueden, haciendo que los mismos sean completamente dependientes de este sistema para realizar cualquier cambio.

Zentyal además de ser local, por defecto instala un servidor web (Apache) y uno de base de datos (Postgres), esto es innecesario cuando se quiere tener un servidor dedicado a otro servicio. Ejemplo de esto es cuando se tiene un servidor de bajas prestaciones que cumpla la función de DNS, pues este no necesita tener instalado un servidor web ni uno de base de datos. Tampoco puede ser modificado el servidor LDAP que instala por defecto para gestionar sus servicios, por lo que, si se quiere hacer un controlador de dominio primario (PDC) tendría que usarse otro software.

Este sistema no permite gestionar correctamente un servidor DHCP porque bloquea algunas opciones que

¹² **home:** la carpeta `/home/` es donde se guardan todos los datos personales y preferencias de los usuarios en *Gnu/Linux*.

en muchas ocasiones son necesarias. Por ejemplo, no se puede adicionar una nueva subred que no esté enmarcada en la que se instaló el servidor. Solo permite las actualizaciones automáticas del DHCP con el servidor DNS que él instala y no con otro.

En caso de querer realizar actualizaciones con garantía de calidad de Zentyal o de uno de sus componentes, hay que adquirir una de las versiones empresariales o profesionales, pues los que ofrece la comunidad de Zentyal no son aprobados por esta organización, teniendo que pagar por la actualización y soporte del mismo.

El acceso para la administración del servidor se puede realizar vía web, mediante una conexión SSL, utilizando un certificado digital¹³ auto firmado por el mismo, no siendo emitidos por Autoridades Certificadoras¹⁴ de prestigio internacional, ya que en Cuba no existe ninguna empresa que brinde esos servicios. Esto hace que las conexiones no sean fiables, ya que pueden ser víctimas de ataques de suplantación de identidad.

A continuación se muestra una tabla en la que se comparan las herramientas antes mencionadas ubicadas en las columnas y en las filas se encuentran los aspectos a evaluar.

	Webmin	YaST	Zentyal
Sistema operativo	GNU/Linux	openSUSE	Ubuntu
Licencia	BSD	GPL	GPL
Facilidad de uso	Baja	Media	Alta
Modo de ejecución	Local	Local	Local
Funcionalidades	Alta	Media	Media
Seguridad	Media	Alta	Alta

¹³ **Certificado digital:** Un certificado digital permite identificar como válido a un emisor de información.

¹⁴ **Autoridades Certificadoras:** tienen como misión verificar que la compañía en cuestión sea confiable y se encuentre legalmente constituida.

Migración	No	No	No
------------------	----	----	----

Es importante destacar que estos sistemas no cuentan con la funcionalidad de migrar un servidor que se encuentre instalado con sistemas operativos privativos a sistemas GNU/Linux. Por lo tanto, se hace necesario crear un sistema que automatice este proceso, ahorrando a los administradores el trabajo de escribir todos los datos que tenían en un servidor con Windows para el nuevo servidor que se estará ejecutando sobre tecnologías libres.

1.2 Herramientas para la migración de servidores.

En el año 2004 fue publicado el libro Windows To Linux Migration Toolkit por los autores David Allen, Herbert Lewis, Christian Lahti, John Streeton Stile, James Stanger, Andrew Taylor Scott y Timothy Tuck, en el cual exponen soluciones para realizar la migración a tecnologías libres [8]. Las mismas no dan solución a los problemas planteados en esta investigación, pues abordan soluciones para migrar únicamente desde Windows Server 2000, siendo enfocada a la migración hacia el sistema operativo Red Hat Enterprise Linux, el cual no es libre y no hacia otras distribuciones como Debian o Ubuntu, que sí cumplen con las cuatro libertades del Software Libre.

A raíz de este libro fueron creados varios *script* escritos en Perl, bajo el nombre de w2lmt, los cuales facilitan el trabajo de la migración de los servicios, pero aún no dan solución debido a que son ejecutados manualmente, teniendo que configurar los archivos previamente y de forma directa en el código.

También existen aplicaciones que realizan la migración de forma más sencilla como es el directorio activo que se instala en Zentyal, este puede configurarse para que se actualice automáticamente con el Directorio Activo (DA) de un servidor de Windows. Para esto se requiere que Zentyal sea configurado como “esclavo” del DA privativo, una vez que se conecten y si están bien configurados, el DA realizará la transferencia de la información al esclavo, entre ellos, los datos de los usuarios y las contraseñas. Cuando el OpenLdap de Zentyal tenga toda la información requerida entonces podrá apagarse el DA,

convirtiéndose el esclavo en máster, aunque esto ya funciona no da solución a todos los servicios pues solo es para el directorio activo.

En la bibliografía consultada por los autores, solo se encontraron esas dos propuestas que no dan solución al problema de la migración de los servidores en las PYMES cubanas.

1.3 Lenguaje de programación.

Un lenguaje de programación es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen un lenguaje informático [9]. Permite al programador especificar de manera precisa sobre qué datos debe operar una computadora, cómo deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias [10].

Los lenguajes de programación y los entornos de desarrollo integrados tienen suma importancia para cualquier proyecto de software pues con estos se han creado innumerables programas y herramientas que han ayudado al hombre a controlar de una manera más sencilla a los ordenadores, así como realizar múltiples tareas y actividades. En la actualidad, existe un numeroso conjunto de lenguajes que se utilizan en el desarrollo de aplicaciones de escritorio, algunos de estos son mencionados a continuación.

Python.

Python es un lenguaje interpretado de programación, multiplataforma, flexible y su implementación está bajo la licencia de código abierto Python Software Foundation License.

Python, al ser un lenguaje interpretado, implica ahorro de tiempo durante el desarrollo de un programa ya que no necesita ser compilado. El intérprete puede usarse de modo interactivo, el cual hace fácil, el experimentar con las características del lenguaje para probar funciones durante la etapa inicial de desarrollo [14].

Permite escribir programas muy compactos y legibles, permitiéndole ser más pequeños que sus contrapartes en C u otros lenguajes por las siguientes razones:

- **Lenguaje Interpretado o de Script:** es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje de máquina.
- **Funciones y bibliotecas:** dispone de un gran cúmulo de funciones incorporadas en el propio lenguaje, permitiendo el tratamiento de cadenas de caracteres, números, archivos, etc. Además, existen librerías que son importadas con facilidad en los programas para tratar temas específicos.
- **Tipado Dinámico:** se refiere a que no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que se determinará en tiempo de ejecución, según el tipo de valor al que se asigne, y el tipo de esta variable puede cambiar si se le asigna otro valor.
- **Fuertemente Tipado:** no permite tratar a una variable sin tener en cuenta el tipo de dato que almacena, es necesario convertir de forma explícita dicha variable al nuevo tipo previamente. Es decir, si se tiene una variable que contiene un texto (variable de tipo cadena o string) se podrá tratar como un número (sumar "9" y el número 8).
- **Sintaxis clara:** posee una sintaxis visual gracias a la notación indentada (con márgenes) de obligado cumplimiento. Además, para distanciar las porciones de código se aconseja tabular, colocando un margen al código que irá dentro de una función o un bucle. Esto beneficia a que todos los programadores adopten las mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.
- **Multiplataforma:** el intérprete de Python está disponible en multitud de plataformas (Solaris, GNU/Linux, DOS, Window, OS/2, Mac OS) por lo que si no se utilizan bibliotecas específicas de cada plataforma el programa podrá correr en todos estos sistemas sin grandes cambios.

- **Orientado a Objetos:** la orientación a objetos es un paradigma de programación en el que los conceptos relevantes del mundo real para los problemas se trasladan a clases y objetos del programa. La ejecución del programa consiste en una serie de interacciones entre los objetos.

Se selecciona el lenguaje de programación Python para el desarrollo de la aplicación dada su característica de ser orientado a objetos, su alto grado de legibilidad y su tipado dinámico. Posee un desarrollo abierto, por lo que está en un proceso de continuo de mejoras, realizado por una gran comunidad de desarrolladores. Aproximadamente cada seis meses se hace pública una nueva versión de este lenguaje, el cual posee una variada documentación comenzando por la expuesta en su sitio web, así como sus librerías, todos disponibles en la web de manera gratuita. Se integra con la librería Augeas, destinada a facilitar el trabajo con los archivos de configuración de los sistemas GNU/Linux y de la mayoría de los servicios. También es seleccionado por ser el que dominan los autores, lo que implica ahorro de tiempo y recursos.

1.4 Herramientas para el desarrollo.

A continuación se caracterizarán algunas de las herramientas existentes para trabajar con el lenguaje de programación seleccionado (Python).

1.4.1 Bibliotecas.

A continuación se describen las bibliotecas utilizadas para el desarrollo de la aplicación.

- **Python-augeas.**

Es el módulo de Python que permite trabajar con la biblioteca de Augeas, la cual se encarga de manipular los archivos de configuración con seguridad, más seguras que las técnicas ad-hoc usadas generalmente con grep, sed, el awk y los mecanismos similares en idiomas scripting [15]. Analiza los archivos en sus formatos nativos y los transforma en un árbol. Los cambios necesarios se realizan mediante la

manipulación de este árbol y se guarda nuevamente en los archivos de configuración originales. Es de código abierto y puede ser descargado de su sitio oficial (<http://augeas.net/>).

- **Python-paramiko.**

Es un módulo para Python 2.2 (o superior) que implementa el protocolo SSH2 (cifrado y autenticado) para conexiones remotas seguras. A diferencia de SSL (TLS alias), el protocolo SSH2 no requiere certificados jerárquicos firmado por una autoridad central. SSH es el protocolo que sustituyó a Telnet y RSH para el acceso remoto seguro a consolas. Este protocolo también incluye la posibilidad de abrir canales arbitrarios a los servicios remotos a través de túneles encriptados. Está escrito completamente en Python y fue liberado bajo la licencia GNU LGPL [16].

- **PyQt4.**

PyQt4 es un conjunto de enlaces Python para Qt 4, está doblemente licenciado bajo la licencia GPL (versión 2 y 3, con excepciones de las licencias adicionales) y una licencia comercial. PyQt4 expone gran parte de la funcionalidad de Qt 4 a Python, incluyendo:

- Amplio conjunto de widgets.
- Características estándares de interfaz gráfica de usuario para las aplicaciones (menús, barras de herramientas, la base de las ventanas).
- Fácil comunicación entre componentes de la aplicación (señales y slots).
- Internacionalización (i18n) de apoyo e integración con la herramienta de traducción de Qt Linguist.
- Incluido el apoyo para la exportación a PDF (Qt 4.1 y posteriores).
- Entrada / salida de una conexión en red.
- Qt 4.3 y posterior admite comunicaciones por Secure Sockets Layer.
- Soporte para las tecnologías de la consulta XML, como XQuery, XPath y XSLT.

1.4.2 Herramientas de programación.

Las herramientas que se plantean a continuación son las principales y las más usadas en la programación con el lenguaje seleccionado.

- **Eclipse.**

Eclipse es un IDE¹⁵ (*Integrated Development Environment*, entorno integrado de desarrollo) para Java¹⁶ muy potente. Es libre y fue creado originalmente por IBM. Se está convirtiendo en el estándar de los entornos de desarrollo para Java. Otros IDE comerciales como JBuilder han anunciado que su próxima versión se basará en Eclipse. Este no es tan solo un IDE, se trata de un marco de trabajo modular ampliable mediante complementos [17].

Una de las ventajas es la existencia de complementos que permiten usar Eclipse para programar en PHP, Perl, **Python**, C/C++, etc.

Eclipse ofrece servicios que permiten el trabajo entre desarrolladores de forma más eficiente y sincronizada, pues cuenta con las tareas, que posibilitan una comunicación directa entre los miembros del equipo, permitiendo dar órdenes de trabajo de manera *online* en forma de correo electrónico. Esta opción hace de Eclipse un IDE ideal para el trabajo en equipo [18].

1.4.3 Herramientas CASE.

Las herramientas CASE¹⁷ son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas

¹⁵ **IDE:** Entorno de Desarrollo Integrado, definido al principio de este capítulo.

¹⁶ **Java** es un lenguaje de programación orientado a objetos.

¹⁷ Case: Computer Aided Software Engineering, en español (Ingeniería de Software Asistida por Computadoras).

como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras.

Estas herramientas tienen como objetivo:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Mejorar el tiempo, coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto.
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- Automatizar el desarrollo del software como: documentación, generación de código, pruebas de errores y gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación.
- Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

Visual Paradigm.

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Esta herramienta también proporciona abundantes tutoriales de UML, demostraciones interactivas y proyectos [19].

El Modelado Visual, es el proceso que permite representar gráficamente el sistema, permitiendo resaltar los detalles más importantes. Un buen modelo:

- Identifica requisitos y comunica información.
- Se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos.
- Permite ver las relaciones entre los componentes del diseño.
- Mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico.

Visual Paradigm es una Herramienta CASE que da soporte al modelado visual con UML 2.0 como mínimo y entre sus principales características se destacan las siguientes:

- Entorno de creación de diagramas para UML 2.0 o superior.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse con los principales IDEs.
- Disponibilidad en múltiples plataformas.

1.5 Herramientas auxiliares.

Además de las herramientas antes mencionadas, son utilizadas otras para la realización de pequeñas pruebas al sistema, cambio de código, ejecución de métodos independientes entre otras.

- **Qt Designer.**

Qt Designer es la herramienta de Qt para el diseño y la creación gráfica de interfaces de usuario (GUI) de los componentes de Qt, es multiplataforma y entre sus características principales cuenta con extensa documentación de sus componentes, permite incluirle eventos a los botones, lo que ahorra tiempo al programador. También brinda la posibilidad de validar la entrada de texto en algunos componentes como el LineEdit, lo cual es de mucha ayudada en la aplicación, sobre todo para validar criterios de búsqueda. Permite además ejecutar la interfaz dando una vista previa de la misma y permitiendo determinar errores en la ubicación de los componentes. Con el comando pyuic4 se puede generar el código Python de las interfaces. Por las características anteriormente descritas, se decide utilizar Qt Designer para el diseño de las interfaces del software.

- **RapidSVN.**

RapidSVN es un cliente de interfaz gráfica para la comunicación con servidores Subversion¹⁸. Está escrito en C++ y distribuido bajo licencia GPL. Este facilita el mantenimiento de las diferentes versiones de ficheros, desde una interfaz sencilla e intuitiva, y se encuentra disponible para plataformas Windows, Linux, MAC OS X y Solaris. Es una herramienta rápida y eficiente.

1.6 Metodología ágil utilizada.

Las metodologías ágiles intentan evitar los engorrosos caminos usados en las metodologías tradicionales, enfocándose en las personas y los resultados. Se basan en promover iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto, logrando que se minimicen los riesgos desarrollando software en cortos lapsos de tiempo.

El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar poco tiempo. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño,

¹⁸ **Subversión:** Servidor para el control de versiones, muy usado en el desarrollo de software, permite el trabajo colaborativo.

codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, pero la meta es tener un demo (sin errores) al final de cada iteración, donde el equipo vuelve a evaluar las prioridades del proyecto.

Por las grandes ventajas que proporcionan estas metodologías se propone para el desarrollo de esta investigación el uso de la metodología ágil SXP. Además, está orientado por la dirección del centro, el uso de esta metodología en los trabajos que pertenezcan a él.

Metodología SXP.

SXP es una metodología de desarrollo de software compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo.

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo [21].

Conclusiones parciales del capítulo

En el desarrollo de este capítulo fueron analizadas las principales herramientas de administración y migración de servidores mediante, estudio que permitió obtener las principales funcionalidades de cada uno para adicionarlas al producto final. En las descripciones realizadas a las mismas, los autores llegaron a la conclusión de que ninguna de las aplicaciones estudiadas pueden adaptarse a las condiciones de las PYMES cubanas. La más que se asemeja al software necesitado es Zentyal, pero por las deficiencias

mencionadas, no se garantiza su correcto funcionamiento en los servidores de bajas prestaciones existentes en las empresas cubanas. Además, no existen aplicaciones que automaticen el proceso de migración de los diferentes servicios telemáticos a código abierto.

Con la aplicación de la metodología SXP se logra gestionar un equipo para que trabaje eficientemente y tener siempre medidos los progresos en el desarrollo del software.

Capítulo 2: Análisis y Diseño.

En este capítulo se abordan temas referentes a las herramientas que se utilizarán en la gestión de los módulos y se describe la propuesta del software a implementar. También se exponen las funcionalidades que debe cumplir el sistema, las cuales están descritas mediante las Historias de Usuario. Además, se expone la arquitectura y los patrones de diseño que se utilizarán y la estructura que deben cumplir sus módulos.

2.1 Propuesta del sistema a desarrollar.

Después de realizar un estudio de las principales herramientas existentes en Cuba y el mundo para la migración y administración de servidores, se concluye en la necesidad de implementar una aplicación libre, que permita administrar y migrar los servicios telemáticos desde las diferentes versiones de Windows Server a plataformas libres.

Para darle solución al problema planteado, el presente trabajo de diploma propone desarrollar una herramienta compuesta por un sistema base que permita administrar los servidores, en el cual se tengan las funcionalidades necesarias para administrar los usuarios, las tareas programadas y los servicios, entre otros. Además, se desarrollarán dos módulos, uno para administrar el servicio SSH y otro para el DHCP, permitiendo agilizar considerablemente el proceso de migración de este último.

Buscando la flexibilidad en la configuración del sistema que permite la Migración y Administración de Servidores (HMAS), se propone el diseño de una arquitectura de software distribuida. La propuesta, es la utilización de una arquitectura en capas, lo cual simplifica la comprensión y organización del sistema, reduciendo las dependencias de forma que las capas más bajas no sean conscientes de ningún detalle o interfaz de las superiores, además fomenta la reutilización. La arquitectura propuesta permitirá la interoperabilidad en entornos distribuidos con un nivel de abstracción superior, lograr una interfaz de usuario más flexible y permitir que la aplicación sea más simple y escalable.

2.2 Propuesta de módulos para el sistema.

A continuación se muestra la propuesta de software que serán utilizados para la construcción de los módulos de la aplicación, estos fueron extraídos de la Guía Cubana para la Migración a Código Abierto [22]. Es importante destacar, que estos no son los únicos servicios que se tendrán en cuenta, ya que la arquitectura de la aplicación permitirá adicionar otros módulos que se deseen implementar.

Para la primera versión del software, solo se desarrollarán los módulos para administrar los servicios SSH y DHCP, permitiendo migrar este último. Los programas utilizados son openssh-server y dhcp3-server respectivamente.

2.3 Roles.

Los roles que a continuación se mencionan fueron los definidos en el documento Concepción del Sistema. Es responsabilidad de los miembros del equipo, cumplir con el rol asignado para que se obtenga un producto con la calidad requerida y en el tiempo establecido. Ver Anexo 4: Roles.

2.4 Requerimientos funcionales.

Las condiciones que el sistema debe cumplir o capacidad que debe tener con el objetivo de establecer un entendimiento común entre el usuario y el proyecto de software son los requerimientos. El propósito de su gestión es establecer un entendimiento común entre el usuario y el desarrollador. Los requerimientos se clasifican en requerimientos funcionales y no funcionales [23].

Lista de reserva del producto (LRP)

Los requerimientos funcionales son la determinación exacta de qué debe ser capaz de hacer el sistema, estas se corresponden con opciones que ejecutará el software, operaciones realizadas de forma oculta o condiciones extremas a determinar por el sistema.

Durante la captura de requisitos se confecciona la LRP, en la cual, se definen las funcionalidades que tendrá el producto en forma de requisitos técnicos y de negocio. Es una lista priorizada y garantiza la organización de los requisitos funcionales y no funcionales, a partir de la prioridad que tengan para el desarrollo del sistema, de igual forma facilita el trabajo en la confección de las Historias de Usuario.

Para el desarrollo de esta investigación se planifican dos iteraciones, siendo la mayoría de los requisitos identificados de prioridad muy alta y alta, pues el desarrollo de otros módulos por parte del proyecto depende de la implementación de las funcionalidades que se exponen a continuación:

Prioridad	Ítem	Descripción	Estimación (Semanas)	Estimado por
Muy Alta				
	1	Autenticar el sistema.	0,1	Reidiel Castillo
	2	Mostrar nombre del sistema.	0,1	
	3	Mostrar información de los discos duros.	0,1	
	4	Mostrar información de la memoria RAM.	0,1	
	5	Mostrar información de la memoria de intercambio.	0,1	
	6	Mostrar la arquitectura del sistema.	0,1	
	7	Mostrar el nombre del sistema operativo que tenga instalado.	0,1	
	8	Mostrar fecha y hora del sistema operativo.	0,1	
	9	Mostrar configuración de red.	0,1	
	10	Modificar configuración de red	0,3	
	11	Eliminar configuración de red.	0,1	
	12	Mostrar usuarios del sistema.	0,1	
	13	Mostrar información de usuario.	0,1	
	14	Adicionar usuario al sistema.	0,1	
	15	Editar usuarios del sistema.	0,1	
	16	Eliminar usuarios del sistema.	0,1	
	17	Mostrar grupos del sistema.	0,1	
	18	Adicionar grupo al sistema.	0,1	
	19	Editar grupo del sistema.	0,1	

	20	Eliminar grupo del sistema.	0,1	
	21	Mostrar tareas programadas.	0,1	
	22	Adicionar tarea programada.	0,1	
	23	Modificar tarea programada.	0,2	
	24	Eliminar tarea programada.	0,1	
	25	Mostrar servicios instalados.	0,2	
	26	Instalar servicio en el sistema.	0,2	
	27	Desinstalar servicio del sistema.	0,2	
	28	Conectar a un servidor.	0,3	
	29	Cerrar la conexión con un servidor.	0,1	
	30	Instalar software necesario para la conexión un servidor.	0,3	
	31	Cargar módulos	0,3	
	32	Modificar nivel de los Logs del SSH.	0,05	
	33	Modificar el tiempo máximo para autenticar por SSH.	0,05	
	34	Establecer el puerto de conexión del servidor SSH.	0,05	
	35	Permitir conexiones a root por SSH.	0,05	
	36	Denegar conexiones a root por SSH.	0,05	
	37	Permitir exportación de X11.	0,05	
	38	Denegar la exportación de X11.	0,05	
	39	Mostrar usuarios permitidos del SSH.	0,05	
	40	Adicionar usuarios al SSH.	0,1	
	41	Eliminar usuarios del SSH.	0,1	
	42	Mostrar grupos permitidos del SSH.	0,1	
	43	Adicionar grupo al SSH.	0,1	
	44	Eliminar grupo del SSH.	0,1	
	45	Iniciar servicio SSH.	0,05	
	46	Reiniciar servicio SSH.	0,05	
	47	Detener servicio SSH.	0,05	
	48	Modificar configuraciones generales del DHCP.	0,2	
	49	Mostrar ámbitos de red en el DHCP.	0,1	
	50	Crear ámbito de red en el DHCP.	0,1	
	51	Modificar ámbito de red en el DHCP.	0,1	
	52	Eliminar ámbito de red en el DHCP.	0,1	
	53	Mostrar asignaciones de IP dadas.	0,2	
	54	Iniciar servidor DHCP.	0,05	

	55	Reiniciar servidor DHCP.	0,05	
	56	Detener servidor DHCP.	0,05	
	57	Migrar ámbitos del servidor DHCP de Windows para GNU/Linux.	0,5	
Alta				
	1	Adicionar IP del servidor al que se ha conectado.	0,2	
	2	Mostrar historial de los servidores a los que se ha conectado.	0,2	
	3	Borrar historial de los servidores a los que se ha conectado.	0,1	
	4	Cambiar nombre del sistema.	0,1	
RNF (Requisitos No Funcionales)				
	1	Crear interfaz sencilla e intuitiva desde la cual se acceda a todas las funcionalidades del sistema.		
	2	Usar como lenguaje de programación Python.		
	3	La herramienta de desarrollo a utilizar será Eclipse + pydev.		
	4	Disponer para el SO GNU/Linux de los siguientes paquetes: <ul style="list-style-type: none"> • python-augeas • python-paramiko • nmap • openssh • gksu2 		
	5	Utilizar QT para la interfaz de la aplicación.		
	6	Utilizar QT Designer para el diseño de la interfaz.		
	7	Las conexiones con los servidores deben ser mediante SSH2.		
	8	La aplicación se ejecutará en sistemas operativos libres.		
	9	El sistema contará con una ayuda que explicará cómo trabajar con el sistema.		

	10	La aplicación deberá permitir la adición de módulos.		
	11	Registrar acciones del sistema.		

2.5 Historias de Usuario (HU).

Las HU constituyen la técnica utilizada en SXP para especificar los requisitos del software, lo que equivaldría a los casos de uso en el proceso unificado. Las HU guían la construcción de las pruebas de aceptación y son utilizadas para estimar tiempos de desarrollo. En este sentido, proveen los detalles suficientes para hacer una estimación razonable del tiempo que llevará implementarlas. Además, se deben detallar a través de la comunicación con el cliente, pues constituyen una base para las pruebas funcionales.

A continuación se exponen las Historias de Usuarios más importantes para el funcionamiento del software, las demás HU pueden verse en el Anexo 5 del documento.

HU: Autenticar sistema.

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Autenticar sistema.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Pablo Soria Acosta	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 0,5 semana
Riesgo en Desarrollo: Medio	Puntos Reales: 0,1 semana
Descripción: Permitirá la autenticación en el sistema.	
Observaciones: Para que la aplicación se ejecute es necesario verificar que el usuario tenga permisos de administración (debe ser sudo).	
Prototipo de interfaz: No	

HU: Mostrar información del sistema.

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: Mostrar información del sistema.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Pablo Soria Acosta	Iteración Asignada: 1

Prioridad en Negocio: Muy Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Medio	Puntos Reales: 1 semana
Descripción: Mostrará información básica del sistema.	
Observaciones:	
Debe ser la primera ventana que se muestra cuando se ejecuta la aplicación.	
Prototipo de interfaz: Ver Anexo 1, Figura 1.	

HU: Gestionar configuración de red.

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Gestionar configuración de red.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reidiel Castillo Arbelo	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Alto	Puntos Reales: 0,5 semana
Descripción: El software mostrará la información relacionada con la configuración de la red del sistema. También debe permitir modificar o eliminar dichas configuraciones.	
Observaciones:	
Se implementarán las siguientes funcionalidades:	
<ul style="list-style-type: none"> • Mostrar Información de la red: Lo primero que se mostrará al usuario son las configuraciones de las interfaces de red que tiene el sistema. • Cambiar Información de la red: Cambiará el método por el cual se obtiene la configuración de red (estática, dinámica por DHCP y no configurada). • Eliminar Información de red: Borrará la configuración de la interfaz seleccionada. 	
Prototipo de interfaz: Ver Anexo 1, Figura 2	

HU: Gestionar usuarios del sistema.

Historia de Usuario	
Número: 4	Nombre Historia de Usuario: Gestionar usuarios del sistema.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Pablo Soria	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Alto	Puntos Reales: 0,5 semana

<p>Descripción: Mostrará en un listado los usuarios que tiene creado el sistema. Permite que se adicionen, se modifiquen o eliminen estos.</p>
<p>Observaciones:</p> <p>Se implementan las siguientes funcionalidades.</p> <ul style="list-style-type: none"> • Mostrar usuarios del sistema. • Mostrar información de usuarios. • Adicionar usuario al sistema. • Editar datos de usuarios del sistema. • Eliminar usuario del sistema.
<p>Prototipo de interfaz: Ver Anexo 1, Imagen 3 e Imagen 4.</p>

HU: Gestionar tareas programadas.

Historia de Usuario	
Número: 6	Nombre Historia de Usuario: Gestionar tareas programadas.
Modificación de Historia de Usuario Número: <i>Ninguna</i>	
Usuario: Reidiel Castillo Arbelo.	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Alto	Puntos Reales: 0,5 semana
<p>Descripción: Se mostrarán las tareas programadas del root. También debe permitir que se adicionen nuevas tareas y se modifiquen o eliminen las existentes.</p>	
<p>Observaciones:</p> <p>La tabla donde se muestran las tareas programadas deberá actualizarse cada vez que se adicione, modifique o elimine una tarea. Se implementan las siguientes funcionalidades.</p> <ul style="list-style-type: none"> • Mostrar tareas programadas. • Adicionar tareas programadas. • Editar tareas programadas. • Eliminar tareas programadas. 	
<p>Prototipo de interfaz: Ver Anexo 1, Imágenes 8 y 9.</p>	

HU: Conectar a un servidor.

Historia de Usuario	
Número: 8	Nombre Historia de Usuario: Conectar a un servidor.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Pablo Soria Acosta	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Alto	Puntos Reales: 1 semana
Descripción: Permitirá conectarse a un servidor.	
Observaciones: Las conexiones a los servidores se realizarán mediante SSH, usando un usuario y contraseña del servidor al cual se va a conectar. El envío de datos entre la máquina cliente y el servidor deberá ser segura, por lo que, la conexión deberá realizarse mediante Socke SSH2, haciendo uso de la librería python-paramiko. Además, permitirá cerrar la conexión en cualquier momento que el usuario desee.	
Prototipo de interfaz: Ver Anexo 1, Imagen 11.	

HU: Administrar servicio SSH.

Historia de Usuario	
Número: 13	Nombre Historia de Usuario: Administrar servicio SSH.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Pablo Soria Acosta.	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 2 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 2 semanas
Descripción: Permitirá administrar el servicio SSH.	
Observaciones: Todos los cambios que se realicen en la interfaz de administración de este servicio no serán guardados hasta que se seleccione el botón Aplicar Cambios. Cuando se realice esta acción, se debe reiniciar el servicio para que se apliquen las nuevas configuraciones. El archivo de configuración a modificar será "sshd_config" que encuentra en "/etc/ssh/". Se	

implementan las siguientes funcionalidades:

- Modificar nivel de los Logs.
- Modificar el tiempo máximo para autenticar por SSH.
- Establecer el puerto por el cual se va a conectar el SSH.
- Permitir conexiones al root.
- Denegar las conexiones al root.
- Permitir exportación de las X.
- Denegar exportación de las X.
- Mostrar usuarios permitidos del SSH.
- Adicionar usuarios al SSH.
- Eliminar usuarios del SSH.
- Mostrar grupos permitidos del SSH.
- Adicionar grupo al SSH.
- Eliminar grupo del SSH.
- Iniciar servicio SSH.
- Reiniciar servicio SSH.
- Detener servicio SSH.

Prototipo de interfaz: Ver Anexo 1, Imagen 14.

HU: Administrar servicio DHCP.

Historia de Usuario	
Número: 14	Nombre Historia de Usuario: Administrar servidor DHCP.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reidiel Castillo Arbelo	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta	Puntos Estimados: 2 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 2 semanas
Descripción: Permitirá administrar el servicio DHCP.	

Observaciones: En las interfaces de las opciones generales y de los ámbitos deben existir dos botones, uno para cancelar y regresar al inicio sin realizar cambios y otro para realizarlos. Todos los cambios que se realicen en la interfaz no serán guardados hasta que se seleccione el botón Aplicar Cambios, el cual aplicará cualquier modificación que se haya realizado, escribiéndolo en el fichero de configuración del servidor. Se implementan las siguientes funcionalidades:

- Modificar configuraciones generales del DHCP.
- Crear ámbito de red en el DHCP.
- Modificar ámbito de red en el DHCP.
- Eliminar ámbito de red en el DHCP.
- Mostrar asignaciones de IP.
- Iniciar servicio DHCP.
- Reiniciar servicio DHCP.
- Detener servicio DHCP.

Prototipo de interfaz: Ver Anexo 1, imágenes 15, 16, 17 y 18.

HU: Migrar ámbitos del DHCP de Windows para GNU/Linux.

Historia de Usuario	
Número: 15	Nombre Historia de Usuario: Migrar ámbitos del DHCP de Windows para GNU/Linux.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Pablo Soria Acosta.	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Alto	Puntos Reales: 1 semana
Descripción: Permitirá migrar las configuraciones de los ámbitos que estén definidos en un servidor de Windows para GNU/Linux.	
Observaciones:	

Se realizará a partir de un fichero, este puede generarse con el siguiente comando NETSH DHCP SERVER DUMP>c\salida.txt.

Prototipo de interfaz: no

2.6 Estructuración de los componentes.

En esta sección se describe la arquitectura de software del sistema, explicando el patrón o patrones arquitectónicos a utilizar, la estructuración de los componentes y la interacción entre ellos, así como el modo de integración de los mismos.

2.6.1 Fundamentación de los Patrones de Arquitectura.

La descripción de la arquitectura es importante y a la vez necesaria para el desarrollo de un sistema de software, pues constituye una expresión del sistema y de su evolución, permitiendo una mejor comunicación entre los clientes y los desarrolladores, así como la evaluación y comparación de arquitecturas de forma consistente. Facilita además, la planificación, gestión y ejecución de las actividades del desarrollo del sistema, ya que es una forma de verificar la correcta implementación de acuerdo con la descripción arquitectural.

Arquitectura en 3 capas.

La arquitectura en 3 capas o programación en 3 capas consiste en separar un proyecto en **Capa de Presentación, Capa de Negocio y Capa de Datos**. Esto permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre ellos [25]. Ver Anexo 1, Imagen 19.

Ventajas de esta arquitectura.

- El desarrollo se puede llevar a cabo en varios niveles.
- Desarrollos paralelos (en cada capa).
- Aplicaciones más robustas debido al encapsulamiento.

- En caso de que sobrevenga algún cambio, solo se ataca al nivel requerido sin tener que revisar entre código mezclado.
- Mantenimiento y soporte más sencillo (es más sencillo cambiar un componente que modificar una aplicación).
- Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad).
- Alta escalabilidad. La principal ventaja de una aplicación distribuida bien diseñada es su buen escalado, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir esta escalabilidad.

La calidad tan especial de este estilo arquitectónico consiste en separar la lógica de la aplicación y convertirla en una capa intermedia bien definida. Esta separación entre la lógica de aplicación de la interfaz de usuario añade una enorme flexibilidad al diseño de la aplicación. Pueden construirse y desplegarse múltiples interfaces de usuario sin cambiar en absoluto la lógica de aplicación siempre que esté presente una interfaz claramente definida a la capa de presentación.

Las ventajas del estilo en capas son obvias. En primer lugar, el estilo soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite la partición de un problema complejo en una secuencia de pasos incrementales. En segundo lugar, admite muy naturalmente optimizaciones y refinamientos. En tercer lugar, proporciona amplia reutilización. Al igual que los tipos de datos abstractos, se pueden utilizar diferentes implementaciones o versiones de una misma capa en la medida que soporten las interfaces relacionadas con las capas adyacentes. Esto conduce a la posibilidad de definir interfaces de capa estándar, a partir de las cuales se pueden construir extensiones o prestaciones específicas [26].

Para la Herramienta de Migración y Administración de Servidores las tres capas que se definen son: Presentación (Interfaz de usuario), Lógica de Negocio o Dominio (tareas y reglas que rigen el proceso),

Acceso a Datos o Servicios (Mecanismos de acceso a los datos). Para una mejor comprensión se puede auxiliar del diagrama de componentes. (Anexo 1, Imagen 20) A continuación se explican cada una de las capas de este modelo.

Capa de Presentación: Es la que interactúa directamente con el usuario, captura la información entrada por este y hace las peticiones a la capa inferior mostrando al usuario la respuesta proveniente de esta. Esta capa se comunica únicamente con la capa de negocio llevando y trayendo los datos o registros necesarios, es la interfaz gráfica del programa y debe ser lo más amigable posible para una mejor comunicación con el usuario. Ver Anexo 1, Imagen 21.

Capa de Lógica de Negocio: Está conformada por los paquetes que integran el sistema, los cuales se ajustan a las historias de usuarios arquitectónicamente significativas y a los requisitos. Desde el punto de vista del diseño, esta capa es contenedora de las clases entidades y controladoras. Se comunica con la Capa de Presentación donde obtiene las solicitudes de los usuarios, también se comunica con la Capa de Datos para la extracción de la información necesaria que es procesada y mostrada al usuario. En esta capa quedan definidos los siguientes componentes que conforman el sistema HMAS. Ver Anexo 1, Imagen 22.

Capa de Acceso a Datos: Contiene las clases encargadas de obtener la información del sistema operativo y los servicios, realizando todas las operaciones de forma transparente a la Capa de Negocio. Los datos que obtiene no los almacena de forma persistente sino que los guarda en como objetos cuando son ejecutadas las clases correspondientes. Ver Anexo 1, Imagen 23.

Estilos arquitectónicos utilizados.

Los estilos arquitecturales son la herramienta básica de un arquitecto a la hora de dar forma a la arquitectura de una aplicación. Un estilo arquitectural se puede entender como un conjunto de principios que definen a alto nivel un aspecto de la aplicación, viene definido por un conjunto de componentes, un

conjunto de conexiones entre dichos componentes y un conjunto de restricciones sobre cómo se comunican dos componentes cualesquiera conectados. Los estilos arquitecturales pueden organizarse en torno al aspecto de la aplicación sobre el que se centran. Los principales aspectos son: comunicaciones, despliegue, dominio, interacción y estructura [27].

Estilo arquitectónico basado en componentes.

El estilo de arquitectura basada en componentes describe un acercamiento al diseño de sistemas como un conjunto de componentes que exponen interfaces bien definidas y que colaboran entre sí para resolver el problema [28]. Ver Anexo 1, Figura 24.

Características:

- Es un estilo para diseñar aplicaciones a partir de componentes individuales.
- Enfatiza la descomposición del sistema en componentes con interfaces muy bien definidas.
- Define una aproximación al diseño a través de componentes que se comunican mediante interfaces que exponen métodos, eventos y propiedades.

Beneficios que se obtienen en la aplicación de este estilo:

- Fácil despliegue: puede sustituir un componente por su nueva versión sin afectar a otros componentes o al sistema.
- Reducción de costes: pueden usar componentes de terceros para abaratar los costes de desarrollo y mantenimiento.
- Reusables: son independientes del contexto se pueden emplear en otras aplicaciones y sistemas.
- Reducción de la complejidad gracias al uso de contenedores de componentes que realizan la activación, gestión del ciclo de vida, etc.

En el desarrollo de esta investigación es utilizado el estilo arquitectónico basado en componente, el cual permite que a la aplicación se le desarrollen módulos para la administración de los diferentes servicios que se requieran en las PYMES Cubanas. La implementación de estos módulos se puede realizar por

diferentes desarrolladores, en diferentes lugares y de forma simultánea o no, pues solo importa que cumpla con el formato requerido y que tenga una interfaz bien definida que le permita integrarse al sistema.

2.6.2 Arquitectura de los módulos.

Los módulos que se adicionarán a la aplicación tendrán una arquitectura en 3 capas como se describió anteriormente para el sistema base. En cada módulo que se desarrolle debe existir una clase Config, la cual debe tener las configuraciones necesarias para que el módulo sea leído cuando se ejecute la aplicación. La estructura de esta clase debe ser la siguiente:

```
__Clases__ = "modulo_dhcp"  
__Descripción__ = "Este es el módulo del servidor DHCP"  
__Servidor__ = "dhcpd"  
__Modulo__ = "dhcp"
```

2.7 Reutilización de componentes.

Todos los componentes cumplen una tarea importante dentro del sistema en general; cada uno de ellos se relacionan entre sí de acuerdo con su propósito final en el sistema. Para lograr la integración de los diferentes componentes de un sistema, se debe tener en cuenta los que son creados por el equipo de desarrollo y los ya existentes.

Cuando se comenzó la implementación de la aplicación no se contaba con módulos o clases que se pudieran utilizar, por tanto, solo serán reutilizables aquellos componentes que realicen los autores de este trabajo. En el desarrollo futuro de módulos para el sistema que permitan administrar los diferentes servicios de los servidores, podrán utilizarse las clases Servidor Remoto y Servidor Local, las cuales tienen implementado los métodos necesarios para la conexión con los servidores, envío de datos y comando, verificación de las conexiones y cancelación de las mismas entre otras.

Capítulo 3 Implementación y Pruebas.

En este capítulo se describe el proceso realizado para la implementación del sistema, en el cual se expone como queda conformado el mismo mediante el Diagrama de Componentes. También se expone el Diagrama de Despliegue, donde se puede observar la distribución física del sistema. Además, se plasman las Tareas de Ingeniería propuestas para las Historias de Usuario más significativas del software, destacándose los responsables de las mismas y el tiempo de duración de cada una. Por último, en la etapa de implementación se detalla como es el tratamiento de los errores que pueden ocurrir en producto.

En el flujo de trabajo de prueba, se verificará el resultado de la implementación mediante la ejecución de Pruebas de Aceptación, donde se comprueba el funcionamiento de cada funcionalidad definida en la concepción del sistema.

3.1 Implementación.

En el flujo de trabajo implementación se empieza con el resultado del diseño y se desarrolla el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de códigos binarios, ejecutables y similares.

3.1.1 Diagrama de Componentes.

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

Se representa como un grafo de componentes de software, unidos por medio de relaciones de dependencia (compilación, ejecución, etc.), mostrándose las interfaces que estos soporten. Se utiliza para modelar la vista estática de un sistema. Además, muestra la organización y las dependencias lógicas entre un conjunto de componentes software. Ver Anexo 1, Figura 22.

3.1.2 Diagrama de Despliegue.

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Contiene los nodos que forman la topología de hardware sobre la que se ejecuta el sistema y la distribución de las partes en ellos [30].

En la siguiente figura se muestra dos nodos, el primero representa la “PC Gestora” (computadora en la cual se va a ejecutar el sistema desarrollado) y otro que representa el servidor (Servidores que serán administrados de forma remota). La interacción entre estos nodos se realizará a través de conexiones TCP/IP, mediante el protocolo SSH2 usado para realizar conexiones seguras con los servidores.

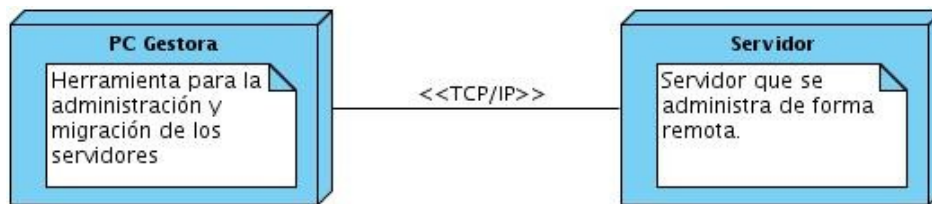


Figura 1: Diagrama de Despliegue.

3.1.3 Tareas de Ingeniería

Las Tareas de Ingeniería se efectúan para detallar mejor las HU, facilitando con ello, el entendimiento en el proceso de implementación. Cada HU puede contener una o más tareas en caso de necesitarla, explicando paso a paso las acciones que se realizan en la misma. Los puntos de estimación asignados a cada tarea son medidos por días según las características y la complejidad que posea la misma.

Definen cada una de las actividades asociadas a las HU y permiten organizar el proceso de implementación, así como conocer el grado de complejidad de cada HU, teniendo en cuenta la cantidad de tareas asociadas.

A continuación se exponen las Tareas de Ingeniería pertenecientes a las Historias de Usuario más significativas del proyecto. Las demás pueden verse en el Anexo 5.

Tareas de Ingeniería para la HU Autenticar Sistema.

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 1
Nombre Tarea: Estudio de la librería gksu2.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Pablo Soria Acosta	
Descripción: La librería gksu2 es la que permite que se ejecute una aplicación con permisos de administración del sistema. Se instala la librería y se prueban las funcionalidades que tiene.	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: 1
Nombre Tarea: Desarrollo de una clase para autenticarse en el sistema.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Pablo Soria Acosta.	
Descripción: Se desarrolla una clase que importe la librería gksu2 y mediante la funcionalidad gksu2.sudo (python Main.py) se ejecuta la aplicación.	

Tareas de Ingeniería para la HU Gestionar configuración de red.

Tarea de Ingeniería	
Número Tarea: 7	Número Historia de Usuario: 3
Nombre Tarea: Diseño de una interfaz para gestionar la red del sistema.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Reidiel Castillo Arbelo.	
Descripción: Se diseña una interfaz que permita configurar la red y se habilita un botón para salvar los cambios realizados. Es utilizado el componente QComboBox para mostrar las interfaces de red, el método por el cual se le asigna el IP y las máscaras de red. Para los demás elementos se hace uso de QlineEdit.	

Tarea de Ingeniería	
Número Tarea: 9	Número Historia de Usuario: 3

Nombre Tarea: Obtención de los datos de la red.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Reidiel Castillo Arbelo.	
Descripción: Se crea la clase Red, con los métodos necesarios para configurar el fichero “/etc/resolv.conf” y el fichero “/etc/network/interface”. Mediante los manejadores que tiene la librería Augeas para la red, se obtiene la información necesaria y se muestra en la interfaz. Además, se implementan métodos para verificar si hay cambios realizados y para salvarlos.	

Tareas de Ingeniería para la HU Gestionar Usuarios del Sistema.

Tarea de Ingeniería	
Número Tarea: 10	Número Historia de Usuario: 4
Nombre Tarea: Diseño de interfaz para gestionar usuarios.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0,2
Fecha Inicio:	Fecha Fin:
Programador Responsable: Pablo Soria Acosta.	
Descripción: Se confecciona una interfaz para la gestión de los usuarios, en la cual	

Tarea de Ingeniería	
Número Tarea: 12	Número Historia de Usuario: 4
Nombre Tarea: Obtención de los datos de los usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Pablo Soria Acosta	
Descripción: Se crea la clase Usuario para almacenar los datos, esta cuenta con los siguientes atributos: nombre, id, id del grupo, home y shell. Además, en la clase principal, se desarrollan los métodos para la obtención de los valores mediante la lectura del fichero “/etc/passwd”.	

Tareas de Ingeniería para la HU Gestionar Tareas Programadas.

Tarea de Ingeniería	
Número Tarea:	Número Historia de Usuario: 6

Nombre Tarea: Diseño de interfaz para gestionar tareas programadas.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Reidiel Castillo Arbelo.	
Descripción: Se diseña una interfaz que cuenta con el componente QtableWidged para mostrar las tareas programadas.	

Tarea de Ingeniería	
Número Tarea:	Número Historia de Usuario: 6
Nombre Tarea: Obtención de las tareas programadas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Reidiel Castillo Arbelo.	
Descripción: Se crea una clase para almacenar los datos de las tareas programadas del sistema	

Tareas de Ingeniería para la HU Conectar a un Servidor.

Tarea de Ingeniería	
Número Tarea:	Número Historia de Usuario: 8
Nombre Tarea: Estudiar protocolo SSH2.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Pablo Soria Acosta.	
Descripción: Se ejecutan las funcionalidades del protocolo SSH para las conexiones remotas y se analizan los resultados.	

Tarea de Ingeniería	
Número Tarea:	Número Historia de Usuario: 8
Nombre Tarea: Estudio de la librería python-paramiko.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Pablo Soria Acosta.	
Descripción: Se instala la librería y analizan las funcionalidades que tiene para las conexiones remotas.	

Tarea de Ingeniería	
Número Tarea:	Número Historia de Usuario: 8
Nombre Tarea: Diseño de interfaz para las conexiones con otros servidores.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Pablo Soria Acosta.	
Descripción: Se diseña una interfaz que cuenta con tres QlineEdit para permitir la introducción del IP al cual se va a conectar, el usuario y la contraseña.	

Tareas de Ingeniería para la HU Administrar Servicio SSH.

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 13
Nombre Tarea: Estudiar servicio SSH.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Pablo Soria Acosta.	
Descripción: Se instala un servidor SSH (openssh-server) y se realizan diferentes configuraciones comprobando su funcionamiento.	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: 13
Nombre Tarea: Estudiar librería Augeas para el servicio SSH.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Pablo Soria Acosta.	
Descripción: Se instala un servidor SSH y se prueban los manejadores de Augeas para administrarlo.	

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: 13
Nombre Tarea: Diseño de una interfaz para administrar el servicio SSH.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Pablo Soria Acosta.	
Descripción: Se diseña una interfaz que permita administrar este servicio.	

Tareas de Ingeniería para la HU Administrar Servicio DHCP.

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: 14
Nombre Tarea: Estudio del servicio DHCP.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Pablo Soria Acosta.	
Descripción: Se instala "dhcp3-server" en un servidor y se modifican las opciones de configuración comprobando su funcionamiento. Además, se ponen máquinas virtuales a coger la red por DHCP para probar las configuraciones realizadas.	

Tarea de Ingeniería	
Número Tarea: 6	Número Historia de Usuario: 14
Nombre Tarea: Estudio de los manejadores de Augeas para el servicio DHCP.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Pablo Soria Acosta.	
Descripción: Se prueban las opciones de Augeas que permiten configurar un servidor DHCP para probar su correcto funcionamiento.	

Tarea de Ingeniería	
Número Tarea: 7	Número Historia de Usuario: 14
Nombre Tarea: Diseño de una interfaz para administrar el servicio DHCP.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Pablo Soria Acosta.	
Descripción: Se crea una interfaz que permite administrar este servicio.	

Tareas de Ingeniería para la HU Migrar ámbitos del DHCP de Windows.

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 15
Nombre Tarea: Estudio del servicio DHCP en Windows.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Pablo Soria Acosta.	

Descripción: Se instalan servidores Windows en sus diferentes versiones y se configura el servicio DHCP creando diferentes ámbitos en los mismos, con diversas configuraciones.

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: 15
Nombre Tarea: Obtención de las configuraciones del servidor DHCP de Windows.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Pablo Soria Acosta.	
Descripción: En los servidores de Windows se ejecuta el comando NETSH DHCP SERVER DUMP>c:\salida.txt para obtener las configuraciones del mismo y se estudia la estructura del fichero en cada uno de ellos, con el objetivo de lograr un estándar en los mismos y que todos puedan migrarse.	

3.2 Prueba.

Uno de los pilares de SXP es el proceso de pruebas de aceptación. Esta metodología propone realizar pruebas constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados, y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También aumenta la seguridad de evitar efectos adyacentes no deseados a la hora de realizar modificaciones y refactorizaciones¹⁹.

A continuación se muestran las pruebas realizadas a las Historias de Usuarios más significativas para el producto. Debido al extenso número de pruebas realizadas, solo se exponen aquellas en las que todos los datos son correctos. Las demás pruebas pueden verse en el Anexo 6, donde también se encuentran las realizadas a las HU menos significativas.

Pruebas a la HU Autenticar Sistema.

¹⁹ **Refactorización:** es una técnica de la ingeniería de software para reestructurar un código fuente.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HMAS-01-01	Nombre Historia de Usuario: Autenticar Sistema.
Nombre de la persona que realiza la prueba: Reidiel Castillo Arbelo.	
Descripción de la Prueba: Se prueba el caso en que la contraseña es correcta.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • La aplicación debe estar instalada correctamente. • Debe existir un usuario con permisos de administración en el sistema. 	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Ejecutar la aplicación. 2. Introducir la contraseña. 	
Resultado Esperado: Se abre la interfaz de la aplicación.	
Evaluación de la Prueba: Satisfactoria.	

Pruebas a la HU Gestionar Configuración de Red.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HMAS-03-01	Nombre Historia de Usuario: Gestionar Configuración de Red.
Nombre de la persona que realiza la prueba: Pablo Soria Acosta	
Descripción de la Prueba: Probar que se muestra la configuración de la red.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Tener instalado python-augeas. 	
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Seleccionar el módulo Datos de la Red. • Seleccionar la opción Configuración. 	
Resultado Esperado: Se muestra la información de la red.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: HMAS-03-02	Nombre Historia de Usuario: Gestionar Configuración de Red.
Nombre de la persona que realiza la prueba: Pablo Soria Acosta	
Descripción de la Prueba: Probar la opción de cambiar la configuración de red	

cuando los datos son correctos.
Condiciones de Ejecución:
<ul style="list-style-type: none"> • Tener instalado python-augeas.
Entrada / Pasos de ejecución:
<ol style="list-style-type: none"> 1. Seleccionar el módulo Datos de la red. 2. Llenar la información de la red en los campos que se muestran. 3. Seleccionar el botón Salvar Cambios.
Resultado Esperado: Se cambia la configuración de la red.
Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Nombre Historia de Usuario: Gestionar
HMAS-03-07	Configuración de Red.
Nombre de la persona que realiza la prueba: Pablo Soria Acosta	
Descripción de la Prueba: Cambiar configuración de la red, cuando se pone el método DHCP.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Tener instalada la librería Python-augeas. • Tener seleccionado el módulo Datos de la Red. 	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar el método Automático DHCP. 2. Seleccionar Aplicar Cambios. 	
Resultado Esperado: Se desactivan los demás campos para que no sean editados, ya que no son necesarios, y la interfaz de red es configurada para que obtenga la red por DHCP.	
Evaluación de la Prueba: Satisfactoria.	

Pruebas a la HU Gestionar Usuarios del Sistema.

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Nombre Historia de Usuario:
HMAS-04-02	Gestionar Usuarios del Sistema.
Nombre de la persona que realiza la prueba: Reidiel Castillo Arbelo	

Descripción de la Prueba: Probar la funcionalidad: mostrar información de los usuarios.
Condiciones de Ejecución: <ul style="list-style-type: none"> • Tener el módulo de Gestión de Grupos y Usuarios abierto.
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar un usuario de la lista mostrada.
Resultado Esperado: Se muestra en una tabla la información del usuario seleccionado.
Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HMAS-04-03	Nombre Historia de Usuario: Gestionar Usuarios del Sistema.
Nombre de la persona que realiza la prueba: Reidiel Castillo Arbelo	
Descripción de la Prueba: Probar que se adiciona un usuario, cuando todos los datos son correctos.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Tener el módulo de Gestión de Grupos y Usuarios abierto. 	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar la opción Adicionar Usuario. 2. Llenar los campos solicitados. 3. Seleccionar la opción Adicionar. 	
Resultado Esperado: Se adiciona un usuario al sistema y se actualiza la interfaz mostrando el nuevo usuario en la lista.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: HMAS-04-06	Nombre Historia de Usuario: Gestionar Usuarios del Sistema.
Nombre de la persona que realiza la prueba: Reidiel Castillo Arbelo.	
Descripción de la Prueba: Probar la funcionalidad de editar usuario cuando se cambia el nombre.	

<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> • Tener el módulo de Gestión de Grupos y Usuarios abierto.
<p>Entrada / Pasos de ejecución:</p> <ul style="list-style-type: none"> • Seleccionar un usuario de la lista mostrada. • Seleccionar el botón Editar Usuario. • Escribir en nuevo nombre: castillo • Seleccionar el botón Aplicar.
<p>Resultado Esperado: Se cambia el usuario y se actualiza la interfaz.</p>
<p>Evaluación de la Prueba: Satisfactoria.</p>

Caso de Prueba de Aceptación	
<p>Código Caso de Prueba:</p> <p>HMAS-04-10</p>	<p>Nombre Historia de Usuario:</p> <p>Gestionar Usuarios del Sistema.</p>
<p>Nombre de la persona que realiza la prueba: Reidiel Castillo Arbelo.</p>	
<p>Descripción de la Prueba: Probar la funcionalidad de eliminar usuario.</p>	
<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> • Tener el módulo de Gestión de Grupos y Usuarios abierto. 	
<p>Entrada / Pasos de ejecución:</p> <ul style="list-style-type: none"> • Seleccionar un usuario de la lista mostrada. • Seleccionar la opción Eliminar Usuario. 	
<p>Resultado Esperado: Se elimina el usuario del sistema y se actualiza la interfaz.</p>	
<p>Evaluación de la Prueba: Satisfactoria.</p>	

Pruebas a la HU Gestionar Tareas Programadas.

Caso de Prueba de Aceptación	
<p>Código Caso de Prueba:</p> <p>HMAS-06-02</p>	<p>Nombre Historia de Usuario:</p> <p>Gestionar Tareas Programadas.</p>
<p>Nombre de la persona que realiza la prueba: Pablo Soria Acosta</p>	
<p>Descripción de la Prueba: Probar que se adiciona una tarea cuando los datos son correctos.</p>	
<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> • Tener seleccionado el módulo Tareas Programadas. 	

<p>Entrada / Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Seleccionar la opción Adicionar Tarea. 2. Llenar los campos solicitados. 3. Seleccionar la opción Adicionar. 4. Seleccionar el botón Salvar Cambios.
<p>Resultado Esperado:</p> <p>Se adiciona la tarea al sistema y se actualiza la interfaz.</p>
<p>Evaluación de la Prueba: Satisfactoria.</p>

Caso de Prueba de Aceptación	
<p>Código Caso de Prueba:</p> <p>HMAS-06-04</p>	<p>Nombre Historia de Usuario:</p> <p>Gestionar Tareas Programadas.</p>
<p>Nombre de la persona que realiza la prueba: Pablo Soria Acosta</p>	
<p>Descripción de la Prueba: Probar la funcionalidad de eliminar tarea programada.</p>	
<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> • Tener seleccionado el módulo Tareas Programadas. • Existir al menos una tarea programada. 	
<p>Entrada / Pasos de ejecución:</p> <ul style="list-style-type: none"> • Seleccionar la tarea a eliminar. • Seleccionar el botón Eliminar Tarea. • Seleccionar el botón Salvar Cambios. 	
<p>Resultado Esperado:</p> <p>Se elimina la tarea del sistema y se actualiza la interfaz.</p>	
<p>Evaluación de la Prueba: Satisfactoria</p>	

Caso de Prueba de Aceptación	
<p>Código Caso de Prueba:</p> <p>HMAS-06-10</p>	<p>Nombre Historia de Usuario:</p> <p>Gestionar Tareas Programadas.</p>
<p>Nombre de la persona que realiza la prueba: Pablo Soria Acosta</p>	
<p>Descripción de la Prueba: Probar la funcionalidad de editar tareas programadas, cuando todos los datos son correctos.</p>	
<p>Condiciones de Ejecución:</p>	

<ul style="list-style-type: none"> • Tener seleccionado el módulo Tareas Programadas. • Debe existir al menos una tarea.
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar la tarea a editar. 2. Seleccionar el botón Editar Tarea. 3. Modificar los datos. 4. Seleccionar la opción Editar. 5. Seleccionar la opción Salvar Cambios.
Resultado Esperado: Se modifica la tarea y se actualiza la interfaz, mostrando la tarea con los cambios realizados.
Evaluación de la Prueba: Satisfactoria.

Pruebas a la HU Conectar a un Servidor.

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Nombre Historia de Usuario:
HMAS-08-01	Conectar a un Servidor.
Nombre de la persona que realiza la prueba: Reidiel Castillo Arbelo.	
Descripción de la Prueba: Conectarse a un servidor cuando los datos son correctos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • La máquina remota debe tener instalado un servidor SSH y la librería Python-augeas. • En la configuración del Firewall de la máquina remota debe estar permitida las conexión mediante SSH. • La configuración de la red en ambas máquinas tiene que ser correcta. 	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar el módulo Conectar a un Servidor. 2. Escribir los datos para la conexión. 3. Seleccionar el botón Conectar. 	

Resultado Esperado: Se conecta al servidor y se abre una nueva pestaña con la información del mismo.
Evaluación de la Prueba: Satisfactoria.

Pruebas a la HU Administrar Servicio SSH.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HMAS-13-01	Nombre Historia de Usuario: Administrar servicio SSH.
Nombre de la persona que realiza la prueba: Reidiel Castillo Arbelo	
Descripción de la Prueba: Se prueba la funcionalidad de modificar el nivel de los logs del SSH.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Tener instalado openssh y python-augeas.• Tener seleccionado el módulo Servidor SSH.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none">1. Seleccionar en el campo Nivel de los Logs, la opción contraria a la que aparezca por defecto.2. Aplicar los cambios.	
Resultado Esperado: Se modifica el fichero de configuración del SSH, quedando en la línea LogLevel la opción elegida.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: HMAS-13-02	Nombre Historia de Usuario: Administrar servicio SSH.
Nombre de la persona que realiza la prueba: Reidiel Castillo Arbelo	
Descripción de la Prueba: Se prueba la funcionalidad de modificar el Tiempo Máximo para Autenticar por SSH.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Tener instalado openssh y python-augeas.• Tener seleccionado el módulo Servidor SSH.	

<p>Entrada / Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Seleccionar en el campo Tiempo Máximo para Autenticar, el tiempo que se desee establecer. 2. Aplicar los cambios.
<p>Resultado Esperado: Se modifica el fichero de configuración del SSH, quedando en la línea LoginGraceTime el número seleccionado.</p>
<p>Evaluación de la Prueba: Satisfactoria.</p>

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Nombre Historia de Usuario:
HMAS-13-04	Administrar servicio SSH.
Nombre de la persona que realiza la prueba: Reidiel Castillo Arbelo	
Descripción de la Prueba: Se prueba la funcionalidad de modificar el Puerto de Conexión del SSH.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Tener instalado openssh y python-augeas. • Tener seleccionado el módulo Servidor SSH. 	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar en el campo Puerto de Conexión el valor deseado. 2. Seleccionar el botón Aplicar Cambios. 	
Resultado Esperado: Se modifica el fichero de configuración del SSH, quedando en la línea Port el número seleccionado.	
Evaluación de la Prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Nombre Historia de Usuario:
HMAS-13-06	Administrar servicio SSH.
Nombre de la persona que realiza la prueba: Reidiel Castillo Arbelo	
Descripción de la Prueba: Se prueba la funcionalidad Permitir conexiones al root por SSH.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Tener instalado openssh y python-augeas. 	

<ul style="list-style-type: none"> • Tener seleccionado el módulo Servidor SSH. • Debe estar denegado el acceso del root.
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar el campo Permitir conexiones al root. 2. Aplicar los cambios.
Resultado Esperado: Se modifica el fichero de configuración del SSH, quedando en la línea PermitRootLogin el valor "yes".
Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Nombre Historia de Usuario:
HMAS-13-08	Administrar servicio SSH.
Nombre de la persona que realiza la prueba: Reidiel Castillo Arbelo	
Descripción de la Prueba: Se prueba la funcionalidad Permitir la exportación de las X.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Tener instalado openssh y python-augeas. • Tener seleccionado el módulo Servidor SSH. • Debe estar denegado la exportación de las X. 	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar el campo Permitir exportar las X. 2. Aplicar los cambios. 	
Resultado Esperado: Se modifica el fichero de configuración del SSH, quedando en la línea X11Forwarding el valor "yes".	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Nombre Historia de Usuario:
HMAS-13-11	Administrar servicio SSH.
Nombre de la persona que realiza la prueba: Reidiel Castillo Arbelo	
Descripción de la Prueba: Se prueba la funcionalidad Adicionar Usuarios al SSH.	
Condiciones de Ejecución:	

<ul style="list-style-type: none"> • Tener instalado openssh y python-augeas. • Tener seleccionado el módulo Servidor SSH.
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar el usuario de la lista. 2. Seleccionar el botón Adicionar. 3. Seleccionar la opción Salvar Cambios.
Resultado Esperado: Se actualiza la interfaz y se modifica el fichero de configuración del SSH adicionando en la línea AllowUsers el usuario marcado.
Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Nombre Historia de Usuario:
HMAS-13-12	Administrar servicio SSH.
Nombre de la persona que realiza la prueba: Reidiel Castillo Arbelo.	
Descripción de la Prueba: Se prueba la funcionalidad Eliminar Usuarios del SSH.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Tener instalado openssh y python-augeas. • Tener seleccionado el módulo Servidor SSH. 	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar el usuario de la lista. 2. Seleccionar el botón Eliminar. 3. Seleccionar la opción Salvar Cambios. 	
Resultado Esperado: Se actualiza la interfaz y se modifica el fichero del SSH eliminando en la línea AllowUsers el usuario.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Nombre Historia de Usuario:
HMAS-13-14	Administrar servicio SSH.
Nombre de la persona que realiza la prueba: Reidiel Castillo Arbelo.	
Descripción de la Prueba: Se prueba la funcionalidad Adicionar Grupos al SSH.	
Condiciones de Ejecución:	

<ul style="list-style-type: none"> • Tener instalado openssh y python-augeas. • Tener seleccionado el módulo Servidor SSH.
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar el grupo de la lista. 2. Seleccionar el botón Adicionar. 3. Seleccionar la opción Salvar Cambios.
Resultado Esperado: Se actualiza la interfaz y se modifica el fichero del SSH adicionado en la línea AllowGroup el grupo seleccionado.
Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HMAS-13-15	Nombre Historia de Usuario: Administrar servicio SSH.
Nombre de la persona que realiza la prueba: Reidiel Castillo Arbelo.	
Descripción de la Prueba: Se prueba la funcionalidad Eliminar Grupos del SSH.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Tener instalado openssh y python-augeas. • Tener seleccionado el módulo Servidor SSH. 	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar el grupo de la lista. 2. Seleccionar el botón Eliminar. 3. Seleccionar la opción Salvar Cambios. 	
Resultado Esperado: Se actualiza la interfaz y se modifica el fichero del SSH eliminando en la línea AllowGroup el grupo seleccionado.	
Evaluación de la Prueba: Satisfactoria.	

Pruebas a la HU Administrar Servicio DHCP.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HMAS-14-01	Nombre Historia de Usuario: Administrar Servicio DHCP.
Nombre de la persona que realiza la prueba: Pablo Soria Acosta	

Descripción de la Prueba: Probar que se modifican las opciones generales cuando los datos son correctos.
Condiciones de Ejecución: <ul style="list-style-type: none"> • Tener un servidor DHCP instalado. • Tener seleccionado el módulo DHCP.
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar una de las primeras cuatro líneas mostradas en la interfaz. 2. Seleccionar el botón Editar Opción Seleccionada. 3. Escribir datos correctos. 4. Seleccionar el botón Aplicar. 5. Seleccionar la opción Salvar Cambios.
Resultado Esperado: Se actualiza la interfaz y se modifica el fichero de configuración del DHCP que se encuentra en /etc/dhcp3/dhcpd.conf.
Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HMAS-14-02	Nombre Historia de Usuario: Administrar Servicio DHCP.
Nombre de la persona que realiza la prueba: Pablo Soria Acosta	
Descripción de la Prueba: Probar que se adiciona un ámbito dinámico cuando los datos introducidos son correctos.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Tener un servidor DHCP instalado. • Tener seleccionado el módulo DHCP. 	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar el botón Adicionar Ámbito. 2. Llenar los campos. 3. Seleccionar la opción Adicionar Rango. 4. Seleccionar la opción Aplicar Cambios. 	

Resultado Esperado:
Se actualiza la interfaz y en el fichero de configuración del DHCP que se encuentra en /etc/dhcp3/dhcpd.conf se escriben los cambios.
Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Nombre Historia de Usuario:
HMAS-14-03	Administrar Servicio DHCP.
Nombre de la persona que realiza la prueba: Pablo Soria Acosta	
Descripción de la Prueba: Probar que se adiciona un ámbito dinámico cuando los datos son correctos y se activan las opciones del ámbito.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Tener un servidor DHCP instalado. • Tener seleccionado el módulo DHCP. 	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar el botón Adicionar Ámbito. 2. Llenar los campos que se muestran. 3. Seleccionar la opción Adicionar Rango. 4. Marcar el checkbox Opciones del Ámbito. 5. Escribir los valores. 6. Seleccionar la opción Salvar Cambios. 	
Resultado Esperado:	
Se actualiza la interfaz y se escriben los cambios en el fichero de configuración del DHCP que se encuentra en /etc/dhcp3/dhcpd.conf.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Nombre Historia de Usuario:
HMAS-14-04	Administrar Servicio DHCP.
Nombre de la persona que realiza la prueba: Pablo Soria Acosta	
Descripción de la Prueba: Probar que se adiciona un ámbito estático cuando los	

datos son correctos.
Condiciones de Ejecución: <ul style="list-style-type: none"> • Tener un servidor DHCP instalado. • Tener seleccionado el módulo DHCP.
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar el botón Adicionar Ámbito. 2. Seleccionar la opción Asignación Estática. 3. Llenar los campos. 4. Seleccionar el botón Aplicar. 5. Seleccionar la opción Salvar Cambios.
Resultado Esperado: Se actualiza la interfaz y se escriben los cambios en el fichero de configuración del DHCP que se encuentra en /etc/dhcp3/dhcpd.conf.
Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Nombre Historia de Usuario:
HMAS-14-05	Administrar Servicio DHCP
Nombre de la persona que realiza la prueba: Pablo Soria Acosta	
Descripción de la Prueba: Probar que se edita un ámbito estático cuando los datos son correctos.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Tener un servidor DHCP instalado. • Tener seleccionado el módulo DHCP. • Debe existir un ámbito en el servidor. 	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar el ámbito a editar. 2. Seleccionar el botón Adicionar Ámbito. 3. Seleccionar la opción Asignación Estática. 4. Llenar los campos. 	

5. Seleccionar el botón Aplicar.
6. Seleccionar la opción Salvar Cambios.
Resultado Esperado: Se actualiza la interfaz y se escriben los cambios en el fichero de configuración del DHCP que se encuentra en /etc/dhcp3/dhcpd.conf.
Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación	
Código Caso de Prueba: HMAS-14-06	Nombre Historia de Usuario: Administrar Servicio DHCP
Nombre de la persona que realiza la prueba: Pablo Soria Acosta	
Descripción de la Prueba: Probar que se edita un ámbito dinámico cuando los datos son correctos.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Tener un servidor DHCP instalado. • Tener seleccionado el módulo DHCP. • Debe existir un ámbito en el servidor. 	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar el ámbito que se desea editar. 2. Seleccionar el botón Editar Ámbito. 3. Llenar los campos. 4. Seleccionar la opción Adicionar Rango. 5. Seleccionar el botón Aplicar. 6. Seleccionar el botón Salvar Cambios. 	
Resultado Esperado: Se actualiza la interfaz y se escriben los cambios en el fichero de configuración del DHCP que se encuentra en /etc/dhcp3/dhcpd.conf.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Nombre Historia de Usuario:

HMAS-14-07	Administrar Servicio DHCP.
Nombre de la persona que realiza la prueba: Pablo Soria Acosta.	
Descripción de la Prueba: Probar que se edita un ámbito dinámico cuando los datos son correctos y se activan las opciones del ámbito.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Tener un servidor DHCP instalado. • Tener seleccionado el módulo DHCP. • Debe existir un ámbito en el servidor. 	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar el ámbito que se desea editar. 2. Seleccionar el botón Editar Ámbito. 3. Llenar los campos. 4. Seleccionar la opción Adicionar Rango. 5. Marcar el checkbox Opciones del Ámbito. 6. Se llenan los campos: 7. Seleccionar el botón Aplicar. 8. Seleccionar el botón Salvar Cambios. 	
Resultado Esperado:	
Se actualiza la interfaz y en el fichero de configuración del DHCP que se encuentra en /etc/dhcp3/dhcpd.conf se edita el ámbito.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Nombre Historia de Usuario:
HMAS-14-08	Administrar Servicio DHCP.
Nombre de la persona que realiza la prueba: Pablo Soria Acosta	
Descripción de la Prueba: Probar la funcionalidad de eliminar ámbitos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Tener un servidor DHCP instalado. • Tener seleccionado el módulo DHCP. 	

<ul style="list-style-type: none"> • Debe existir un ámbito en el servidor.
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar el ámbito que se desea eliminar. 2. Seleccionar el botón Eliminar Ámbito. 3. Seleccionar el botón Salvar Cambios.
Resultado Esperado: Se actualiza la interfaz y en el fichero de configuración del DHCP que se encuentra en /etc/dhcp3/dhcpd.conf se elimina el ámbito.
Evaluación de la Prueba: Satisfactoria.

Pruebas a la HU Migrar ámbitos del DHCP de Windows para GNU/Linux.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HMAS-15-1	Nombre Historia de Usuario: Migrar ámbitos del DHCP de Windows para GNU/Linux.
Nombre de la persona que realiza la prueba: Pablo Soria Acosta	
Descripción de la Prueba: Probar que se adicionan los ámbitos cuando el fichero esta con el formato correcto.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Tener un servidor DHCP instalado. • Tener seleccionado el módulo DHCP. • Tener copiado en el sistema, el fichero con la configuración. 	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar el botón Migrar Servidor. 2. Buscar el fichero deseado. 3. Seleccionar la opción Salvar Cambios. 	
Resultado Esperado: Se crean los ámbitos en el servidor y se actualiza la interfaz.	
Evaluación de la Prueba: Satisfactoria.	

Conclusiones

Con la culminación del presente trabajo se dio cumplimiento a cada uno de los objetivos trazados, destacándose de manera general los siguientes aspectos:

- Fueron analizados diferentes sistemas existentes para la administración y migración de servidores, estudio que permitió identificar las características y funcionalidades que se ajustan a las necesidades de las PYMES cubanas para integrarlas al producto.
- Con el uso de las herramientas, tecnologías y lenguajes de programación que soportaron el proceso de desarrollo guiado por la metodología SXP, se obtiene un producto formado por un sistema base para la administración del servidor y dos módulos, uno para la administración del servicio SSH y otro para el DHCP, permitiendo agilizar considerablemente la migración de este último.
- El desarrollo de una interfaz gráfica, intuitiva y fácil de utilizar, hace que el sistema se adecue a las necesidades de los administradores cubanos de redes, facilitando el trabajo de los mismos.
- El diseño y ejecución de un conjunto de pruebas funcionales realizadas al culminar cada nueva funcionalidad aseguraron la calidad del producto final.

Recomendaciones

Con la realización del presente trabajo se desarrolló una aplicación para la administración de servidores y migración de los mismos a sistemas GNU/Linux. Para mejor funcionamiento de este se recomienda:

- Agregar módulos que permitan administrar y migrar los demás servicios telemáticos utilizados en las empresas cubanas como son: correo, servidor web, proxy, samba, ftp y firewall.
- Implementar la autenticación mediante llaves con los servidores, para evitar que se tenga que escribir el usuario y contraseña cada vez que se vaya a conectar a ellos.

Referencia Bibliográfica

- [1] *Desarrollo de la Industria Regional de Software en América Latina y el Caribe: Consideraciones y propuestas*. Secretaría Permanente del SELA, Caracas, Venezuela. Sección: Resumen Ejecutivo, página 3.
- [2] Fidel Castro. *Acto por el aniversario 15 del Palacio Central de Computación*. Marzo 2006. [8 Febrero 2012]. Disponible en la Web: <http://www.cubaminrex.cu/archivo/Presidente/2006/FC_070306.htm>.
- [3] León Polanco. *Windows Small Business Server*. [8 Marzo 2012]. Disponible en la Web: <http://es.wikipedia.org/wiki/Windows_Small_Business_Server>.
- [4] Anon. *Modules < Webmin < TWiki*. [25 Noviembre 2011]. Disponible en la Web: <<http://doxfer.webmin.com/Webmin/Modules>>.
- [5] Anon. *YaST/About - openSUSE*. [25 Noviembre 2011]. Disponible en la Web: <<http://es.opensuse.org/YaST/About>>.
- [6] Stefan Schubert. *openSUSE Lizards*. Febrero 2009. [16 Febrero 2012]. Disponible en la Web: <<http://lizards.opensuse.org/2009/02/03/yast-web/>>.
- [7] Anon. *Las PYMES y las TICs*. [28 Noviembre 2011]. Disponible en la Web: <<http://doc.zentyal.org/es/presentation.html>>.
- [8] David Allen, Herbert Lewis, Christian Lahti, John Streeton Stile, James Stanger, Andrew Taylor Scott y Timothy Tuck. *Windows To Linux Migration Toolkit*. [United States of American], 2004 [8 Enero 2012].
- [9] Marcos A. Polanco Velasco. *Externalización de servicios seguros*. [28 November 2011]. Disponible en la Web: <<http://enise.inteco.es/images/stories/Ponencias/T25/marcos%20polanco.pdf>>.

- [10] Melodysoft . Melodysoft. *LOS LENGUAJES DE PROGRAMACION*. [1 Diciembre 2011]. Disponible en la Web: <<http://boards5.melodysoft.com/2007AISC0107/los-lenguajes-de-programacion-15.html>>.
- [11] Slideshare. *Conceptos Básicos de Programación*. [1 Diciembre 2011]. Disponible en la Web: <<http://www.slideshare.net/mandre55/conceptos-basicos-programacion>>.
- [12] Anón. *Objective-C | MacProgramadores*. [15 Mayo 2012]. Disponible en la Web: <<http://macprogramadores.org/?q=content/objective-c>>.
- [13] Dr. Juan Segura Salazar. *Características de JAVA*. [17 Febrero 2012]. Disponible en la Web: <<http://tikal.cifn.unam.mx/~jsegura/LCGII/java3.htm>>.
- [14] Anon. *Lenguajes de programación: Python*. [29 Noviembre 2011]. Disponible en la Web: <<http://hosting.udlap.mx/profesores/miguela.mendez/alephzero/archivo/historico/az26/python.html>>.
- [15] Anon. *Augeas-Main*. [20 Febrero 2012]. Disponible en la Web: <<http://augeas.net/index.html>>.
- [16] Anon. *paramiko: ssh2 protocol for python*. Mayo 2011. [2 Marzo 2012]. Disponible en la Web: <<http://www.lag.net/paramiko/>>.
- [17] Anon. *IDE Eclipse, Breve Guía*. [29 Noviembre 2011]. Disponible en la Web: <<http://www.slideshare.net/Benedeti/ide-eclipse-breve-gua-201399>>.
- [18] Omar Moreno Suárez, Omar Moreno Suárez. *Subsistema de Gestión de Servidores y VLAN para la Producción en la Universidad de las Ciencias Informáticas*. June 2010. [29 Noviembre 2011].
- [19] Anon. *Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)) por Visual Paradigm International Ltd*. [27 Abril 2012]. Disponible en la Web: <http://www.freedownloadmanager.org/es/downloads/%20Paradigma_Visual_para_UML_%28M%C3%8D%29%20_14720_p/>.

[20] IBM. [En línea] <http://www-01.ibm.com/software/awdtools/developer/rose/>.

[21] Gladys Marsi Peñalver Romero, Sergio Jesús García de la Puente, Abel Meneses Abad. *SXP, metodología de desarrollo de software*. 2010.

[22] Paumier Samón, “Ramón. “Guía de Migración a Software Libre”. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.[Mayo, 2007].

[23] Ivar Jacobson. *El proceso unificado de desarrollo de software*. Habana: Félix Varela. G. B 2004.

[24] Cesar de la Torre, Miguel Ramos, Javier Calvarro, Unai Zorrilla. *Guía de Arquitectura N-Capas orientadas al Dominio con .NET*. Beta [España], 2010. Capítulo 3: Arquitectura Marzo N-Capas, página 42.

[25] Anón. *Arquitectura 3 Capas*. [12 Marzo 2012]. Disponible en la Web: <<http://kernelerror.net/programacion/php/arquitectura-3-capas/>>.

[26] Carlos Reynoso, Nicolás Kicillof. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. 0.1*. Disponible en la Web: <http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/style.asp>.

[27] Cesar de la Torre, Miguel Ramos, Javier Calvarro, Unai Zorrilla. *Guía de Arquitectura N-Capas orientadas al Dominio con .NET*. Beta [España], 2010. Capítulo 2: Estilos Arquitecturales, página 33.

[28] Cesar de la Torre, Miguel Ramos, Javier Calvarro, Unai Zorrilla. *Guía de Arquitectura N-Capas orientadas al Dominio con .NET*. Beta [España], 2010. Capítulo 2: Estilos Arquitecturales, página 12.

[29] Joaquín Gracia. *Patrones de diseño. Análisis y Diseño. Ingeniería del Software*. [12 Abril 2012]. Disponible en la Web: <<http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>>.

[30] Ídem 23.

Glosario de Términos

1. **Dirección IP:** Dirección de un ordenador dentro de una red con protocolo TCP/IP.
2. **Domain Name System (DNS):** es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada. Este sistema asocia información variada con nombres de dominios asignado a cada uno de los participantes. Su función más importante, es traducir (resolver) nombres inteligibles para los humanos en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.
3. **Dynamic Host Configuration Protocol (DHCP):** Protocolo de Configuración de Hosts Dinámicos. Es un protocolo para asignar direcciones de IP dinámicas en una red.
4. **Framework:** es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.
5. **Gateway:** Dispositivo dedicado a intercomunicar sistemas con protocolos incompatibles. Puerta de enlace, acceso, pasarela. Nodo en una red informática que sirve de punto de acceso a otra red.
6. **GPL:** es una licencia creada por la Free Software Foundation en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

7. **Historias de Usuario (HU):** Secuencias de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de las secuencias.
8. **HU significativas para el negocio:** HU que representa procesos de gran importancia en la línea del negocio.
9. **IDE:** Entorno de desarrollo integrado o en inglés Integrated Development Environment ('IDE'). Es un programa compuesto por un conjunto de herramientas para un programador desde el que se pueden editar programas, compilarlos y depurarlos.
10. **Instalar:** Incorporar a la computadora un programa o dispositivo para ser utilizado.
11. **IPV4:** Dirección IP versión 4, direcciones de 32 bits empleadas por las computadoras para el acceso a la red.
12. **Kernel:** Núcleo. Parte esencial de un sistema operativo que provee los servicios más básicos del sistema. Se encarga de gestionar los recursos como el acceso seguro al hardware de la computadora, determina qué programa accederá a un determinado hardware, si dos o más programas quieren usarlo al mismo tiempo, entre otras.
13. **Logs:** Registro oficial de eventos durante un rango de tiempo en particular.
14. **MAC:** Es un identificador hexadecimal de 48 bits que se corresponde de forma única con una tarjeta o interfaz de red.
15. **Máscara de red:** es una combinación de bits que sirve para delimitar el ámbito de una red de computadoras. Su función es indicar a los dispositivos qué parte de la dirección IP es el número de la red, incluyendo la subred, y qué parte es la correspondiente al host.

16. **Partición:** División lógica en un disco duro.
17. **Privativo:** El software no libre (también llamado software propietario, software privativo, software privado, software con propietario o software de propiedad) se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo, cuyo código fuente no está disponible.
18. **RAM:** Se utiliza como memoria de trabajo para el sistema operativo, los programas y la mayoría del software. Es allí donde se cargan todas las instrucciones que ejecutan el procesador y otras unidades de cómputo.
19. **Servidor:** Computadora central de un sistema de red que provee servicios y recursos (programas, comunicaciones, archivos, etc.) a otras computadoras (clientes) conectadas a ella.
20. **SSH:** Es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos si se tiene un servidor X (en sistemas Unix y Windows) corriendo.
21. **SSL:** Secure Sockets Layer. Protocolo de Capa de Conexión Segura. Protocolos criptográficos que proporcionan comunicaciones seguras por una red.
22. **SXP:** Metodología compuesta por las metodologías SCRUM y XP, especialmente indicada para proyectos pequeños, rápido cambio de requisitos donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad.
23. **TCP/IP:** Transfer Control Protocol / Internet Protocol. Protocolos que se utiliza en Internet para transmitir datos. El TCP está orientado a la conexión que establece una línea de diálogo entre el emisor y el receptor antes de que se transfieran los datos.

Anexos.

Los anexos se encuentran en la versión extendida de este documento, al mismo puede accederse mediante la siguiente dirección del Sistema de Control de Versiones perteneciente al departamento SIMAYS. https://repositorio.geitel.prod.uci.cu/svn/simays/resultados/tesis/2011-2012/Pablo_Reidiel/