

Universidad de las Ciencias Informáticas

Facultad 10



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

Título:

***Integración de servicios a través de un Sistema de Perfiles
Centralizados.***

Autores: Javier Menéndez Rizo

Abel Ernesto Cuadrado Sospedra

Tutores: Ing. Roberto Rodríguez Montoya.

Ing. Dayron Pérez Roldan.

Ciudad de La Habana, Cuba.

Mayo, 2010

“Año 52 de la Revolución”



DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma con carácter exclusivo.

Para que así conste firmo la presente a los__ días del mes de__ del 2010.

Javier Menéndez Rizo.

Firma del Autor

Abel Ernesto Cuadrado Sospedra.

Firma del Autor

Roberto Rodríguez Montoya.

Firma del Tutor

Dayron Pérez Roldán.

Firma del Tutor

Daynel Mármol Lacal.

Firma del Oponente



Agradecimientos.

Agradezco de todo corazón:

A mi mamá por ser la mejor madre del mundo, por quererme tanto, por ser padre y madre a la misma vez, por guiarme por el buen camino y sobre todo por creer siempre en mí.

A mi hermano por preocuparse siempre por mí, por cuidarme y ayudarme no como hermano mayor sino como un padre también.

A tío Normando por preocuparse por nosotros y ayudarnos en todo momento.

A tío Noel, tía Estrella, a mi primo Noelito, Amarilis, Alnedis y Zandra por preocuparse por mi.

A tía Gladys, por cuidarme desde que era pequeño.

A mi prima Nayibi y Alain por ser cariñosos conmigo.

A todos mis tíos y mis tías, mis primos y primas y a toda mi familia en general.

A mis amigos Orlando, Angel, Frangel, Eduardo, Dariel que aunque no nos volvamos a ver por un buen tiempo,

siempre los tendré presente.

A Javier, mi compañero de tesis por ser responsable y eficiente.

A mis compañeros de aula, a mis compañeros de cuarto, por haberme soportado en estos 5 años.

Abel Ernesto Cuadrado Sospedra.

A mi mamá por ser mi vida y mi inspiración.

A mi papá, por siempre estar pendiente de mí y por ser mi ejemplo a seguir.

A mi abuela Gloria, por todo el cariño y la dedicación que siempre ha tenido conmigo.

A mi hermano, por ser el mejor de los hermanos y quererme tanto.

A mi prima Yane, que más que una prima es una hermana para mí.

A mi tío Pepe y mi tía Xiomara, por siempre ayudarnos cuando lo necesitamos.

A mi abuela Verania, por sus consejos de las noches.

A mis tíos Juan Carlos, Gaspar, Ramona y también a Milurde por su preocupación.

A mis primos Maikel, Gaspar, Osmar, Sarai y Carlos.

En fin a toda mi familia, gracias a ellos he llegado hasta aquí.

A mi amigo Yidian, por estar siempre en las buenas y en las malas.

A mi novia Alibech, por darme cariño y apoyo en los momentos más difíciles de mi vida.

A mi compañero de tesis Abel, por enseñarme a ser combativo.

A mis amigos Orlando, Manuel, Carlos Rafael, Adnier y Angel, que aunque no los vea por un tiempo nunca los olvidaré.

Javier Menéndez Rizo.



Dedicatoria.

De Abel:

A Marina mi mamá por ser mi vida y mi inspiración.

A mi hermano mayor por ser el mejor hermano del mundo.

A la memoria de Enrique mi padre y Emilio mi abuelo porque sé que estarían muy orgullosos de mi.

De Javier.

A mi mamá, que aunque ya no está entre nosotros, la tengo presente cada segundo.

A mi papá, que siempre ha sido mi ejemplo a seguir.

A mi hermano Chucho el mejor hermano menor que se puede tener.



“Si hay un secreto del éxito, reside en la capacidad para apreciar el punto de vista del prójimo y ver las cosas desde ese punto de vista así como del propio”.

Dale Carnegie



Resumen.

En este trabajo se realiza un análisis de los diferentes conceptos relacionados con el Sistema de Perfiles Centralizados, así como un estudio sobre las tendencias y tecnologías existentes a nivel nacional e internacional. Se explica el funcionamiento de la solución propuesta y una descripción detallada de todo el proceso investigativo y de desarrollo. Se propone como solución una aplicación que permite configurar y administrar los servicios que brinda el Sistema de Perfiles Centralizados. Este documento se estructura en cuatro capítulos donde se recoge la fundamentación teórica, la descripción y análisis de la solución propuesta, el desarrollo ágil del Sistema de Perfiles Centralizados y por último la validación de la solución propuesta.



Índice de Contenido

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 SISTEMAS DE PERFILES CENTRALIZADOS.	5
1.2 SISTEMAS SIMILARES EN LA ACTUALIDAD.	6
1.2.1 <i>Active Directory</i>	7
1.2.2 <i>Novell eDirectory</i>	8
1.2.3 <i>Red Hat Directory Server</i>	8
1.2.4 <i>OpenLDAP</i>	9
1.2.5 <i>Apache Directory Server</i>	9
1.2.6 <i>OpenDS</i>	10
1.3 SISTEMAS DE ARCHIVOS DISTRIBUIDOS.	10
1.4 NECESIDAD DE CREAR UN SISTEMA DE PERFILES CENTRALIZADOS.	11
1.5 MODELO CLIENTE/SERVIDOR.	12
1.5.1 <i>Ventajas y desventajas del Modelo Cliente/Servidor</i>	13
1.6 HERRAMIENTAS, LENGUAJES Y TECNOLOGÍAS.	14
1.6.1 <i>Zend Studio</i>	14
1.6.2 <i>Quanta Plus</i>	14
1.6.3 <i>PHP</i>	15
1.6.4 <i>Bash</i>	15
1.6.5 <i>Apache</i>	16
1.6.6 <i>Python</i>	16
1.6.7 <i>NFS</i>	17
1.6.8 <i>LDAP</i>	18
1.6.9 <i>DNS</i>	18
1.6.10 <i>QUOTA</i>	19
1.6.11 <i>PhpLDAPadmin</i>	19
1.6.12 <i>XML</i>	20
1.6.13 <i>Javascript</i>	21
1.6.14 <i>BIND</i>	22
1.7 METODOLOGÍAS DE DESARROLLO DE SOFTWARE.	22
CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.	25
2.1 DESCRIPCIÓN DEL SISTEMA DE PERFILES CENTRALIZADOS.	25
2.2 ESTRATEGIA DE INTEGRACIÓN.....	28
2.3 FUNCIONAMIENTO DEL SPC.....	29



2.4 FUNCIONAMIENTO DEL SPCADMIN.....	30
2.5 BREVE DEFINICIÓN DE LA ARQUITECTURA DEL SISTEMA.	33
2.5.1 Arquitectura del SPC.....	33
2.5.2 Arquitectura del SPCAdmin.....	34
2.6 DIAGRAMA DE CLASES DEL SPCADMIN.	35
2.7 DIAGRAMA DE COMPONENTES DEL SPCADMIN.....	40
2.8 DIAGRAMA DE DESPLIEGUE DEL SPCADMIN.	43
CAPÍTULO 3. DESARROLLO ÁGIL DEL SISTEMA DE PERFILES CENTRALIZADOS.....	45
3.1 PLANIFICACIÓN DEL PROYECTO POR ROLES.....	45
3.2 LISTA DE RESERVA DEL PRODUCTO (LRP).....	46
3.3 HISTORIAS DE USUARIO.	53
3.4 TAREAS DE INGENIERÍA.....	64
3.5 PLAN DE RELEASE.....	76
CAPÍTULO 4. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	78
4.1 CASOS DE PRUEBAS DE ACEPTACIÓN.	78
4.2 RESULTADOS OBTENIDOS.	94
4.3 ACERCA DE LAS FUNCIONALIDADES OBTENIDAS.....	94
CONCLUSIONES GENERALES.	95
RECOMENDACIONES.	96
REFERENCIAS BIBLIOGRÁFICAS.	97
BIBLIOGRAFÍA.....	99
ANEXOS.	100
GLOSARIO DE TÉRMINOS.....	101



INTRODUCCIÓN

En la actualidad existe un amplio consenso acerca del impacto que tienen las Tecnologías de la Información y las Comunicaciones (TICs) en varios niveles de la sociedad. Las TICs emergen como instrumentos novedosos, generándose una amplia gama de utilidades para estas, siendo una de sus principales condiciones la de funcionar como eficaces herramientas para el desarrollo económico, político y social de un país. La relación que existe entre las tecnologías de la información y desarrollo ha sido ampliamente estudiada e investigada y, por lo general, se considera que existe una relación positiva entre ambas, de modo que la inversión en TICs se considera importante en la consecución exitosa de proyectos de desarrollo.

Con el auge de la industria del software a nivel mundial ligado a diferentes indicadores como las ventas, el empleo, la exportación, el desarrollo y la independencia tecnológica, el país ha dirigido parte de sus esfuerzos en lograr un impacto significativo en el campo de las ciencias de la información basada en la investigación, desarrollo e integración de soluciones informáticas dirigidas a varias ramas de la economía, creando como parte de los programas de la Revolución a la Universidad de las Ciencias Informáticas (UCI) con el objetivo de convertirla en la principal portadora del desarrollo tecnológico para Cuba.

La UCI es un centro de estudios universitarios que radica en Ciudad de La Habana, Cuba. Nacido como un proyecto de la Revolución Cubana, esta universidad tiene dos objetivos fundamentales: informatizar el país y desarrollar la industria del software para contribuir al desarrollo económico del mismo además de ser la primera universidad cubana creada bajo los propósitos de la Batalla de Ideas.

La UCI dedica sus esfuerzos y recursos a la producción de software y a la formación de profesionales de la informática, la misma está compuesta por facultades donde cada facultad se relaciona con un perfil determinado. Los estudiantes de esta universidad se forman bajo el principio del proceso docente educativo que es la formación desde la producción, junto a la docencia, se vinculan directamente a los diferentes proyectos productivos, por lo que la producción es un factor importante en su formación como ingenieros informáticos.

Actualmente existe la problemática de que en cada proyecto productivo la asignación de computadoras se lleva a cabo teniendo en cuenta la cantidad de personas que lo integran, lo pueden conformar estudiantes de cualquier año de la carrera dependiendo de las necesidades del proyecto y las cualidades de cada uno de ellos, es importante señalar que generalmente en la UCI esta



distribución es de dos o más estudiantes por cada computadora. Todos los integrantes configuran las máquinas que les fueron asignadas según la distribución acordada por el líder del proyecto, ejemplo de esto es una máquina con tres estudiantes asignados, estos crean tres perfiles para el trabajo individual de cada uno de ellos y se define entre los mismos un tiempo de máquina, teniendo en cuenta el horario docente y de producción asignado para cada uno de ellos según el año de la carrera que estén cursando en ese momento. Este proceso trae como consecuencia que un estudiante que desee trabajar en un horario fuera de su tiempo de máquina, se vea imposibilitado a hacerlo, aunque existan máquinas disponibles en ese momento, debido a que solo tiene su perfil en la computadora que tiene asignada. Por lo que se plantea el siguiente **problema científico**, ¿Cómo lograr la integración de diferentes servicios para la gestión de proyectos en una solución de perfiles centralizados para la UCI?

La **idea a defender** se basa en que el desarrollo de un sistema de perfiles centralizados permitirá la mejoría de la productividad y la gestión para los proyectos productivos en la UCI.

Con este trabajo se pretende desarrollar un sistema de perfiles centralizados para ponerlo en práctica en la UCI. Por tanto el **objeto de investigación** se enfocará en los sistemas de perfiles centralizados que existen y su funcionamiento. El **campo de acción** será los sistemas de perfiles centralizados para los proyectos productivos que utilicen GNU/Linux como plataforma de desarrollo de software en la UCI.

El **objetivo** del trabajo consiste en desarrollar un sistema cliente/servidor que permita la integración de diferentes servicios para la gestión de proyectos en una solución de perfiles centralizados para la UCI. Para lograr esto se tienen los siguientes **objetivos específicos**:

- Caracterizar las diferentes soluciones sobre el desarrollo de perfiles centralizados en Cuba y el mundo.
- Desarrollar una aplicación web que permita, de manera eficiente y sencilla, administrar los servicios que se integran en la solución propuesta.
- Desarrollar un instalador para el cliente y el servidor que permita facilitar las labores de configuración al administrador del sistema.

Para darle cumplimiento a estos objetivos se plantean las siguientes **tareas de investigación**:

- Investigación de los sistemas de perfiles centralizados existentes en Cuba y el mundo, su funcionamiento, características, ventajas y desventajas.



- Estudio del funcionamiento de cada uno de los servicios que se integrarán al sistema de perfiles centralizados.
- Estudio del principio de funcionamiento de la creación de paquetes de software para Debian (.deb).
- Diseño y realización de todas las pruebas necesarias para detectar y corregir los posibles errores que se puedan presentar en el ciclo de vida del producto.

Para llevar a cabo las tareas propuestas y arribar satisfactoriamente al resultado final de la investigación se utilizan los siguientes **métodos teóricos**:

Analítico-sintético: en el análisis de diversas metodologías y documentos relacionados con el objeto de estudio de los cuales se extrajeron los rasgos distintivos más importantes relacionados sobre el desarrollo de un Sistema de Perfiles Centralizados.

Modelación: este método permitió modelar el funcionamiento de la solución propuesta, debido a que se utilizó para la esquematización de todo el proceso investigativo.

Histórico-Lógico: este método fue utilizado para hacer un estudio de la evolución de las herramientas similares en Cuba y el mundo.

Este trabajo está estructurado en 4 capítulos y anexos, que incluye todo lo relacionado con el trabajo investigativo acerca de los sistemas de perfiles centralizados, así como la implementación y pruebas de la solución propuesta. A continuación se muestra una breve descripción de cada uno de los capítulos:

Capítulo 1. Fundamentación teórica: Son analizados los principales conceptos que se utilizarán en el trabajo. Se realizará un análisis de las diferentes características de los sistemas de perfiles centralizados, así como de las tecnologías actuales vinculadas a estos sistemas. Además se hace una breve descripción de las tecnologías utilizadas para el desarrollo de la solución propuesta.

Capítulo 2. Descripción y análisis de la solución propuesta: Se especifica la utilización de herramientas ya existentes que formarán parte de la solución, además de hacer un análisis de la solución propuesta. También se describe el funcionamiento que posee el sistema desarrollado y los elementos que lo integran.



Capítulo 3: Desarrollo ágil del Sistema de Perfiles Centralizados: Utilizando como metodología de desarrollo SXP se realiza el desarrollo ágil del software a través de modelos auxiliares, historias de usuarios y prototipos de interfaz de usuario, se hace una descripción de la planificación del proyecto por roles y se explica el funcionamiento del software.

Capítulo 4. Validación de la solución propuesta: Se exponen los casos de prueba realizados a la aplicación en cada una de las iteraciones, se analizan las funcionalidades alcanzadas y se muestran los resultados obtenidos.



CAPÍTULO I: Fundamentación Teórica.

En el mundo informático de hoy, las compañías, empresas, consorcios e instituciones están cada vez más dirigidas a la satisfacción de las comodidades y de las necesidades de cada profesional, para así lograr una excelente armonía entre trabajo a realizar y la satisfacción que se tiene a la hora de efectuar cualquier actividad productiva, así como también las de mejorar sus condiciones de trabajo, conllevando a lograr mejores resultados productivos y elevando la calidad del producto. Para lograr estos objetivos en este documento se propone una solución empresarial para entornos de desarrollo de software que utilizan GNU/Linux como plataforma de desarrollo.

1.1 Sistemas de Perfiles Centralizados.

Antes de plantear el concepto de Sistema de Perfiles Centralizados (SPC), se deben dominar una serie de conceptos relacionados con el tema en cuestión, como son perfiles de usuarios, perfil móvil y perfil obligatorio:

Perfiles de usuarios: Carpetas propias de un usuario, que incluyen básicamente sus configuraciones personalizadas del entorno de trabajo y los documentos por él creados. [1]

Perfil Móvil: Perfil de usuario que se descarga desde el servidor en el equipo donde el usuario en cuestión haya iniciado sesión; cuando dicho usuario cierre sesión en el cliente, los cambios en el perfil se almacenarán en el servidor en el espacio destinado para tal fin para dicho usuario. [1]

Perfil Obligatorio: Perfil de usuario que se descarga desde el servidor en el equipo donde el usuario en cuestión haya iniciado sesión; cuando dicho usuario cierre sesión en el cliente, los cambios en el perfil **NO** se almacenarán en el servidor, de modo que el usuario siempre dispondrá del mismo perfil. [1]

Un **Sistema de Perfiles Centralizados** no es más que la integración de diferentes servicios que permiten a los usuarios iniciar sesión en los equipos de un dominio y conservar al mismo tiempo su configuración de perfil de usuario. En el contexto de la informática, un **usuario** es aquella persona que utiliza e interactúa con un sistema informático, para obtener seguridad, acceso al sistema y administración de recursos. Estos poseen una cuenta de usuario, en la mayoría de los casos asociados a una contraseña para tener acceso al sistema. Cuando estos usuarios se autentican, acceden al sistema automáticamente, y se inician las distintas configuraciones personales de cada uno de ellos, como por ejemplo el fondo de Escritorio, la configuración de su navegador web, documentos personales, accesos directos a los diferentes programas, además de otras



configuraciones de aplicaciones para el trabajo de escritorio, que es a lo que se le conoce como **perfil de usuario**.

Los perfiles de usuario se almacenan en una ubicación del servidor especificada por el administrador. Una vez que el usuario se ha autenticado en el servicio de directorio y haya iniciado una sesión, toda su configuración personal se inicia en el equipo local, que no es más que el perfil de usuario almacenado en el servidor (incluidos los documentos y las distintas configuraciones del usuario), es decir los cambios que se realicen en la terminal cliente serán guardados en el servidor.

1.2 Sistemas similares en la actualidad.

A nivel mundial y con el aumento del número de ordenadores conectados entre sí, se han desarrollado aplicaciones que permiten la administración de los recursos disponibles en una red (tales como usuarios, ordenadores, impresoras, etc.). Dentro de las soluciones que almacenan los perfiles de usuarios en un servidor existen varias alternativas:

Para compartir ficheros en la red en algunas soluciones se utiliza Sistema de Ficheros de Red (NFS) o Samba. Para lograr un sistema de autenticación distribuida se tienen Sistema de Información de Red (NIS) y Protocolo de Acceso Ligerero a Directorios (LDAP). Las practicas comunes son las de combinar estos servicios para logran centralizar la autenticación de los usuarios, configuraciones y datos personales. Las diferentes soluciones más difundidas son la combinación de Samba-LDAP o NFS-LDAP. Pero hasta este momento no existe un sistema que integre estos servicios, automatice parte de las configuraciones y además que brinde una interfaz de administración para gestionar estos servicios.

En el contexto de las redes de ordenadores, se denomina *directorio* a una base de datos especializada que almacena información sobre los recursos, u objetos presentes en la red y que pone dicha información a disposición de los usuarios. Entre las implementaciones del servicio de directorio (LDAP) que se utilizan para gestionar recursos en la red están:

Nombre del Sistema	Sistemas Operativos	Licencia
Active Directory	Windows Server 2000/2003/2008	Privativo
Novell eDirectory	Windows 2000/2003, HP-UX,	GPL



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

	Sun Solaris, IBM-AIX	
Red Hat Directory Server	HP-UX, Solaris, GNU/Linux	GPL
OpenLDAP 2.4.22	BSD, AIX, HP-UX, Mac OS X, Solaris, Microsoft Windows /NT/ 2000/ XP Vista, y z/OS.	OpenLDAP Public License
Apache Directory Server 1.5.7	Multi-Plataforma.	Apache License 2.0
Open DS 2.2	GNU/Linux, Windows, Mac OS X	CDDL(Common Development and Distribution License)

A continuación se describirán algunas de ellas con sus características principales:

1.2.1 Active Directory.

Active Directory es la implementación de servicio de directorio de Microsoft en una red distribuida de computadoras, utiliza protocolos como: LDAP, DNS, DHCP, Kerberos entre otros. El servicio Active Directory proporciona la capacidad de establecer un único inicio de sesión y un repositorio central de información para toda su infraestructura, lo que simplifica ampliamente la administración de usuarios y equipos, proporcionando además la obtención de un acceso mejorado a los recursos en red.

Active Directory posee numerosas ventajas, no sólo el poder manejar instalaciones de cualquier tamaño, desde un único servidor con unos cientos de objetos hasta miles de servidores con millones de objetos. Active Directory también simplifica enormemente el proceso de localizar recursos a lo largo de una gran red. La Interfaz de Servicios de Active Directory (ADSI) permite a los desarrolladores hacer que sus aplicaciones soporten el directorio, proporcionando a los usuarios una única forma de acceder a múltiples directorios, ya estén basados en LDAP, NDS o en los Servicios de directorio de NT (NTDS). También soporta directamente el Protocolo de Transferencia de Hipertexto (HTTP). Todo objeto de Active Directory se puede mostrar como una página en Lenguaje de Marcas de Hipertexto (HTML) en un explorador Web. El directorio soporta extensiones para que el Servicio de Información de Internet (IISI) de Microsoft traduzca las peticiones HTTP para objetos del directorio en páginas HTML para mostrarlas en cualquier cliente HTML. Entre sus principales desventajas están



que: este software no es libre, es decir, para poder utilizarlo es necesario comprar el producto y sólo soporta las plataformas Windows Server 2000/2003/2008.[2]

1.2.2 Novell eDirectory.

Novell eDirectory: (antes conocido como Servicios de directorio de Novell, a veces denominado Netware Directory Services) es un x.500 compatible con el servicio de productos de directorio de software publicado inicialmente en 1993 por Novell para la gestión centralizada de acceso a los recursos en los servidores múltiples y los equipos dentro de una red dada. Es una base de datos jerárquica y orientada a objetos para representar determinados activos de una organización en un árbol lógico, incluidas las personas, puestos, servidores, estaciones de trabajo, aplicaciones, impresoras, servicios, y los grupos. [3]

En el 2006 adoptó la Licencia Pública General (GPL), lo que le permitió soportar una amplia gama de plataformas entre las que se encuentran: Windows 2000/2003, Solaris, IBM AIX, HP-UX y Novell NetWare.

1.2.3 Red Hat Directory Server.

Red Hat Directory Server es un servidor basado en LDAP que centraliza la configuración de las aplicaciones, perfiles de usuario, grupos de datos, políticas, y la información de control de acceso en un sistema operativo independiente, basados en la red del Registro. La formación de un repositorio central de una infraestructura de gestión de identidad, Red Hat Directory Server simplifica la administración de usuarios, eliminando la redundancia de datos y la automatización de datos de mantenimiento. También mejora la seguridad, mediante el almacenamiento de las políticas y la información de control de acceso.

Características:

- Centraliza la gestión de personas y sus perfiles, la reducción de los costes administrativos.
- Actúa como un repositorio central para los perfiles y preferencias del usuario.
- Permite replicación de datos a través de la empresa, proporcionando un sistema centralizado, proporcionando una fuente de datos consistente, disponible para las aplicaciones de la empresa.
- Proporciona escalabilidad de grandes cantidades de usuarios que contiene el control de la información necesaria para el desarrollo de aplicaciones de extranet.
- Proporciona soporte completo para 64-bits de HP-UX, Solaris y Red Hat Enterprise Linux



plataformas.

- Proporciona la base para el certificado de autenticación fuerte cuando se utiliza junto a un sistema Red Hat Certificate. [4]

1.2.4 OpenLDAP.

El proyecto OpenLDAP se inició en 1998 por Kurt Zeilenga, comenzó como un clon de la implementación LDAP de la Universidad de Michigan, entidad donde se desarrolló originalmente el protocolo LDAP. Es una implementación libre y de código abierto del protocolo Lightweight Directory Access Protocol (LDAP) desarrollada por el proyecto OpenLDAP. Está liberada bajo su propia licencia OpenLDAP Public License. Muchas distribuciones GNU/Linux incluyen el software OpenLDAP para el soporte LDAP. Este software también corre en plataformas BSD, AIX, HP-UX, Mac OS X, Solaris, Microsoft Windows (NT y derivados, incluyendo 2000, XP, Vista), y z/OS. [5]

Componentes:

Básicamente, OpenLDAP posee tres componentes principales:

- slapd - Dominio de servidor y herramientas.
- Bibliotecas que implementan el protocolo LDAP
- Programas cliente: ldapsearch, ldapadd, ldapdelete, entre otros.

Adicionalmente, el proyecto OpenLDAP es anfitrión de los subproyectos:

- JLDAP - biblioteca de clases LDAP para Java.
- JDBC-LDAP - controlador Java JDBC - LDAP.
- ldapc++ - biblioteca de clases LDAP para C++.

1.2.5 Apache Directory Server.

El Apache Directory Server (ApacheDS) es un servidor LDAP integrable implementado en Java puro. Tiene varias características que lo hacen único entre los servidores LDAP. Algunas de estas son las siguientes:

- Diseñado como una plataforma de LDAP y X.500, los componentes enchufables y los subsistemas hacen a ApacheDS muy modular e ideal para los experimentos con diversos aspectos del protocolo LDAP.
- El servidor de interfaz es totalmente separable de su backend y viceversa por lo que es muy flexible para la implementación de los directorios virtuales, servidores proxy y puertas de acceso a directorios X.500.



- El servidor expone los aspectos de la administración a través de un sistema especial de backend. LDAP puede ser usado para manejar estos problemas a través del sistema de nombres en el contexto del sistema como `ou=system`.
- Tanto el subsistema de backend y frontend son separables e integrables de forma independiente.
- El servidor contiene un servidor LDAP JNDI proveedor como la fachada del subsistema de backend completa. Operaciones de JNDI se traduce directamente por este proveedor en las operaciones de denunciar el vínculo y las particiones de almacenamiento de entradas de servidor de destino.
- El código de red del servidor, Infraestructura Multipropósito de Aplicaciones de Red (MINA) fue diseñado para los proveedores de protocolo conectable, de todo tipo y no sólo LDAP. MINA ApacheDS da la capacidad de manejar grandes cantidades de concurrencia.
- El servidor utiliza el Twix herramientas y APIs para ASN.1 BER codificación y decodificación. Estas herramientas están diseñadas para una codificación y decodificación muy pequeña así como para su uso en servidores no-bloqueo.
- Compatible con LDAPv3, certificado por el OpenGroup. [6]

1.2.6 OpenDS.

OpenDS es un servicio de directorio de código abierto, escrito en Java, desarrollado como parte del proyecto OpenDS. Es totalmente compatible con LDAPv3, apoya la mayoría de los estándares y esquemas LDAP. También varios maestros de replicación modelo que garantiza la alta disponibilidad de los datos para todas las operaciones, búsquedas o actualizaciones. OpenDS es liberado bajo la Licencia Común de Desarrollo y Distribución (CDDL). El software se desarrolla y funciona en cualquier sistema con soporte del lenguaje de programación Java, incluyendo GNU/Linux y las distribuciones comunes de UNIX, Microsoft Windows, Mac OS X. [7]

Características:

- Cuenta con un panel de control gráfico para realizar la configuración en línea, así como el esquema y los datos de gestión.
- Dispone de una gran herramienta de línea de comandos para llevar a cabo todas las tareas administrativas, tanto en línea interactiva o guión.
- Posee una puerta de enlace DSML Servet.

1.3 Sistemas de Archivos Distribuidos.

Un sistema de archivos distribuidos permite a los procesos el acceso transparente y eficiente de archivos que permanecen en servidores remotos. Estos están encargados de la organización,



almacenamiento, recuperación, nominación, comportamiento y protección de los archivos. También proporcionan una interfaz de programación que oculta a los desarrolladores los detalles de la localización y asignación del almacenamiento.

La gestión de un sistema de archivos distribuido se sustenta a través de dos funciones a menudo bien diferenciadas, una es el **servicio de nombres** o directorios y la otra es el **servicio de ficheros**. Es de suma importancia proporcionar un rendimiento aceptable, por lo que hay que alcanzar el compromiso entre la disponibilidad local de la información para disminuir los costos de comunicación y la consistencia, que se refleja en la semántica que muestran los accesos compartidos. [8]

Los sistemas de archivos distribuidos se organizan en dos tipos, los de Arquitecturas Cliente-Servidor y los basados en Clúster.

Un sistema de ficheros se caracteriza por un conjunto de propiedades generales:

- Proporciona almacenamiento de información permanente.
- Identifica los ficheros en un espacio de nombres (normalmente estructurado).
- Es posible el acceso concurrente desde varios procesos.
- En sistemas multiusuario proporciona protección de accesos.

1.4 Necesidad de crear un sistema de perfiles centralizados.

El Sistema de Perfiles Centralizados es de gran importancia porque permite optimizar el uso de los recursos asignados al proyecto y mejorar la gestión del tiempo de desarrollo para el cumplimiento de las tareas productivas.

De manera general el Sistema de Perfiles Centralizados permite:

Mayor independencia tecnológica:

- Los integrantes del proyecto, en el caso específico de los entornos productivos de la UCI, tendrán una mayor independencia hacia las computadoras, ya que es posible trabajar en cualquiera de éstas para realizar el trabajo asignado.

Centralizar la información:

- Permite centralizar la información relacionada con el proyecto y posibilita realizar salvadas automáticas de esta información.

Gestión de perfiles:



- Permite administrar usuarios, conceder privilegios y administrar cuotas de manera automatizada.

Trabajo Colaborativo:

- Desarrollar una cultura de trabajo basada en el desarrollo colaborativo, debido a que brinda la posibilidad a usuarios que no estén en el proyecto de poder colaborar con el mismo.

Desventajas del Sistema de Perfiles Centralizados:

- Las direcciones IP duplicadas.
- El cambio de subred afecta el buen funcionamiento del sistema de perfiles centralizados. Puede conllevar a un trabajo adicional por parte del administrador de sistema en lo referido a configuraciones en la parte del cliente y en la parte del servidor.

1.5 Modelo Cliente/Servidor.

Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Este mecanismo se puede aplicar a programas que se ejecutan sobre una sola computadora y realmente posee más ventajas en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En este tipo de arquitectura la capacidad de procesamiento está distribuida entre los clientes y los servidores, aunque las ventajas de tipo organizativo son más importantes debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y el servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los diferentes tipos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma. Esta arquitectura sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico. [9]

Las características del cliente son:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación (dispositivo **maestro** o **amo**).



- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

Las características del servidor son:

- Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación.
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No es frecuente que interactúen directamente con los usuarios finales.

1.5.1 Ventajas y desventajas del Modelo Cliente/Servidor.

Ventajas:

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la administración de los servicios.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de cliente-servidor que aseguran la seguridad en las transacciones, una interfaz amigable para el usuario y la facilidad de empleo. [10]

Desventajas:

- La congestión del tráfico trae consigo afectaciones a la arquitectura cliente-servidor. Cuando una gran cantidad de clientes envían peticiones simultáneas al mismo servidor, puede ser que



cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor).

- El paradigma de cliente-servidor clásico no tiene suficiente fortaleza. Cuando un servidor está caído, las peticiones de los clientes no pueden ser satisfechas.
- El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de una computadora personal puede no poder servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo y con esto aumentará el costo.
- El cliente no dispone de los recursos que puedan existir en el servidor. [10]

1.6 Herramientas, lenguajes y tecnologías.

1.6.1 Zend Studio.

Zend Studio es uno de los ambientes de desarrollo integrado o Integrated Development Environment (IDE), orientado a desarrollar aplicaciones web en lenguaje PHP. Además de servir de editor para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, GNU/Linux y MacOS. [11]

1.6.2 Quanta Plus.

Quanta Plus (Quanta+) es una herramienta libre de desarrollo de páginas web, diseñado para el proyecto KDE que rápidamente se ha convertido en un editor maduro que cuenta con varias funcionalidades. Su versión actual es la 3.5 y proporciona un interfaz de múltiples documentos poderosos e intuitivos para los desarrolladores web. Puede incrementar exponencialmente la productividad, a través del uso de acciones personalizadas, guiones y barras de herramientas, con lo que se puede automatizar casi cualquier tarea. [12]

Características:

- Usa KIO para FTP y SSH, además de soportar otros protocolos.
- Tiene asistentes para creación de tablas, enlaces y páginas en blanco.
- Posee la sintaxis de HTML, Javascript, CSS y varios lenguajes más.



- Contiene un analizador que informa acerca de la correcta creación de nuestras páginas.
- Soporta plugins a través de KParts.
- Los documentos pueden ser previsualizados dentro de la aplicación usando el motor KHTML.
- Es posible preprocesar los documentos a través de un servidor web antes de previsualizar.

1.6.3 PHP.

Es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor; se incluye entre etiquetas especiales de comienzo y final que permiten entrar y salir del modo PHP. Permite procesar la información de formularios, generar páginas con contenidos dinámicos, y el trabajo con cookies.

PHP es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles, funciones. El resultado es normalmente una página HTML. Permite la posibilidad de ejecutarlo en diferentes tipos de servidores web, entre ellos Apache.

Una de las características más importante que puede tener PHP es su soporte para una gran cantidad de bases de datos. Escribir una interfaz vía web para una base de datos es una tarea simple con PHP, permite además incorporar un gran número de extensiones, aumentando las potencialidades del lenguaje. Permite la integración con varias bibliotecas externas, dándole al programador la posibilidad de realizar cualquier tarea, desde generar documentos en Portable Document Format (PDF) y hasta analizar código XML. [13]

Es multiplataforma, es funcional para sistemas basados en Unix (con Apache) y también para Windows con Servicio de Información de Internet (IIS), de forma que el código que se haya creado para una de ellas no tiene porque modificarse al pasar a la otra. La sintaxis de PHP está inspirada en C, modificada en algo para adaptarlo al entorno en el que trabaja. Su librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de Active Server Pages (ASP). Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparen rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades del mismo.

1.6.4 Bash.



Bash es el intérprete de comandos por defecto en la mayoría de las distribuciones de GNU/Linux. Su nombre es un acrónimo de Bourne-Again Shell, es un programa cuya función consiste en interpretar órdenes. Ofrece mejoras funcionales en sh tanto para la programación y uso interactivo, los cuales incluyen la edición de línea de comandos, sin límite de tamaño histórico de comandos, control de trabajo, las funciones de depósito y los alias, matrices indexadas de tamaño ilimitado, y la aritmética de enteros en cualquier base de dos a sesenta y cuatro.

1.6.5 Apache.

Apache es un servidor de páginas web que permite acceder a páginas web alojadas en un ordenador. Es el más utilizado seguido de Servicio de Información de Internet (IISI) [14]. El servidor Apache es usado por múltiples razones como disponibilidad, facilidad de instalación, pocos recursos de hardware necesarios, precio, disponibilidad del código fuente, entre otras. Existen otros aunque suelen estar especializados en nichos concretos de mercado. Corre en una multitud de sistemas operativos, lo que lo hace una herramienta multiplataforma.

Características:

- Apache es un software gratuito y libre. Esto le da una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor, lo podemos saber.
- Es altamente configurable y de diseño modular. Es muy sencillo ampliar sus capacidades. En el día de hoy existen muchos módulos y están disponibles para toda la comunidad.
- Soporta gran cantidad de lenguajes como Perl, PHP, Java, entre otros.
- Nos permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- Es altamente configurable para la creación y gestión de logs. También permite la creación de ficheros de log a la medida del administrador, de este modo podemos tener un mayor control sobre lo que sucede en el servidor.[14]

1.6.6 Python.

Python es un lenguaje de programación interpretado creado por Guido van Rossum en el año 1991. Se compara habitualmente con Tcl, Perl, Scheme, Java y Ruby. En la actualidad Python se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation. La última versión estable del lenguaje es la 3.1.1.



Permite dividir el programa en módulos reutilizables desde otros programas Python. Viene con una gran colección de módulos estándar que se pueden utilizar como base de los programas (o como ejemplos para empezar a aprender Python). También hay módulos incluidos que proporcionan E/S de ficheros, llamadas al sistema y hasta interfaces a GUI (interfaz gráfica con el usuario) como Tk, GTK, Qt entre otros. [15]

Se utiliza como lenguaje de programación interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa.

1.6.7 NFS.

El Network File System (NFS) por sus siglas en inglés, conocido también como Sistema de Archivos de Red es un protocolo de nivel de aplicación del modelo de Interconexión de Sistemas Abiertos (OSI), dado que brinda servicios de red a aplicaciones. Se utiliza para sistemas de archivo distribuido en un entorno de red de área local. Permite que varios sistemas conectados a una misma red accedan a ficheros remotos como si estuvieran dentro de la misma computadora. Fue desarrollado en el año 1984 por Sun Microsystems, con el objetivo de que fuera independiente de la máquina, el sistema operativo y el protocolo de transporte TCP o UDP, esto fue posible porque esta implementado sobre los protocolos XDR de nivel de presentación y ONC RPC de nivel de sesión respectivamente. El protocolo NFS se encuentra por defecto en los sistemas operativos UNIX y la mayoría de las distribuciones GNU/Linux basadas en UNIX. [16]

Características:

- El sistema NFS está constituido en dos partes fundamentales: un servidor y uno o más clientes. Los clientes acceden de forma remota a los datos que se encuentra almacenados en el servidor.
- Las PCs de trabajo locales utilizan menos espacio de disco debido a que los datos se encuentran centralizados en un único lugar pero pueden ser accedidos y modificados por varios usuarios, de tal forma que no es necesario replicar la información.
- Los clientes en el caso específico del directorio donde se almacenan los usuarios no necesitan disponer de un directorio “/home” en cada una de las máquinas de la organización. Los directorios “/home” pueden crearse en el servidor de NFS para posteriormente poder acceder a ellos desde cualquier máquina a través de la infraestructura de red.



- También se pueden compartir a través de la red dispositivos de almacenamiento como disquetes, CD-ROM y unidades ZIP. Esto puede reducir la inversión en dichos dispositivos y mejorar el aprovechamiento del hardware existente en la organización.
- Todas las operaciones sobre ficheros son síncronas. Esto significa que la operación sólo retorna cuando el servidor ha completado todo el trabajo asociado para esa operación. En caso de una solicitud de escritura, el servidor escribirá físicamente los datos en el disco, y si es necesario, actualizará la estructura de directorios, antes de devolver una respuesta al cliente. Esto garantiza la integridad de los datos almacenados.

1.6.8 LDAP.

Protocolo Ligero de Acceso a Directorios (LDAP) es un protocolo de tipo cliente-servidor para acceder a un servicio de directorio. Se usó inicialmente como un Frontend o interfaz final para x.500, pero también puede usarse con servidores de directorio únicos y con otros tipos de servidores de directorio. Es considerado una base de datos a la que pueden realizarse consultas denominada directorio LDAP para tener acceso a información referente a usuarios, contraseñas y otras entidades en un entorno de red, ofreciendo una amplia capacidad de filtrado sobre la información solicitada.

Ventajas en el uso de LDAP:

Un directorio LDAP destaca sobre los demás tipos de bases de datos por las siguientes características: [17]

- Es muy rápido en la lectura de registros.
- Permite replicar el servidor de forma muy sencilla y económica.
- Muchas aplicaciones de todo tipo tienen interfaces de conexión a LDAP y se pueden integrar fácilmente.
- Dispone de un modelo de nombres globales que asegura que todas las entradas son únicas.
- Usa un sistema jerárquico de almacenamiento de información.
- Permite múltiples directorios independientes.
- Funciona sobre TCP/IP y SSL.
- La mayoría de aplicaciones disponen de soporte para LDAP.
- La mayoría de servidores LDAP son fáciles de instalar, mantener y optimizar.

1.6.9 DNS.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Domain Name System (DNS) traducido al español Sistema de Nombres de Dominio es una base de datos distribuida y jerárquica que almacena la información necesaria para los nombre de dominio. Sus usos principales son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico correspondientes para cada dominio. El DNS nació de la necesidad de facilitar a los seres humanos el acceso hacia los servidores disponibles a través de Internet permitiendo hacerlo por un nombre, algo más fácil de recordar que una dirección IP, además de ser más fácil de recordar, el nombre es más fiable. La dirección numérica podría cambiar por muchas razones, sin que tenga que cambiar el nombre.

Inicialmente, el DNS nació de la necesidad de recordar fácilmente los nombres de todos los servidores conectados a Internet. En un inicio, SRI (ahora SRI International) alojaba un archivo llamado HOSTS que contenía todos los nombres de dominio conocidos (técnicamente, este archivo aún existe, la mayoría de los sistemas operativos actuales todavía pueden ser configurados para revisar su archivo hosts). El crecimiento explosivo de la red causó que el sistema de nombres centralizado en el archivo hosts no resultara práctico y en 1983, Paul Mockapetris publicó los RFCs 882 y 883 definiendo lo que hoy en día ha evolucionado hacia el DNS moderno. (Estos RFCs han quedado obsoletos por la publicación en 1987 de los RFCs 1034 y 1035). [18]

Los Servidores DNS utilizan TCP y UDP en el puerto 53 para responder las consultas. Casi todas las consultas consisten de una sola solicitud UDP desde un Cliente DNS seguida por una sola respuesta UDP del servidor. TCP interviene cuando el tamaño de los datos de la respuesta excede los 512 bytes, tal como ocurre con tareas como transferencia de zonas.

1.6.10 QUOTA.

La utilidad quota permite especificar dos tipos de límites de espacio dentro de los sistemas de archivos, por usuario y/o por grupos de usuarios. Y dentro de estos dos tipos, por un lado el número máximo de inodos, y por otro lado el número máximo de bloques de disco. Esta característica permitirá controlar el espacio consumido (Web, FTP, etc.) de cada uno de los usuarios del sistema.

1.6.11 PhpLDAPAdmin.

PhpLDAPAdmin también conocido como PLA, es una herramienta para la administración de servidores LDAP escrito en PHP, basado en interfaz web. Trabaja en varias plataformas, permitiéndole acceder al servidor LDAP desde cualquier lugar en internet usando un navegador web. Se encuentra disponible bajo licencia GPL.



Posee una vista jerárquica basada en árbol en donde se puede navegar por toda la estructura de directorio. Permite ver los esquemas LDAP, realizar búsquedas, crear, borrar, copiar y editar entradas LDAP, incluso copiar entradas entre servidores LDAP. [19]

Características:

- Árbol jerárquico de navegación LDAP.
- Edición de entradas basadas en Plantillas.
- Copia entradas LDAP (incluso entre servidores distintos).
- Copia recursivamente un árbol entero.
- Borra entradas LDAP.
- Borra recursivamente un árbol.
- Ver y editar atributos de imágenes.
- Navegador avanzado de esquema LDAP.
- Búsquedas LDAP (simple y avanzada).
- Exporta LDIF y DSML.
- importa LDIF.
- Renombra entradas LDAP.
- Administración de hash de password (sha, crypt, md5, blowfish, md5crypt).
- Automáticamente determina el DN root server.
- Traducción de atributos en una forma más amigables (p.e muestra "Fax" en vez de "facsimileTelephoneNumber").
- Soporta atributos Binarios.
- Automáticamente incrementa los números de UID.
- Disponible en 10 idiomas.

1.6.12 XML



Extensible Markup Language (XML) no es más que un conjunto de reglas para definir etiquetas semánticas que nos organizan un documento en diferentes partes. XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados.

Como parte de su estructura el metalenguaje XML consta de cuatro especificaciones (el propio XML sienta las bases sintácticas y el alcance de su implementación):

-Document Type Definition (DTD): Definición del tipo de documento. Es, en general, un archivo/s que encierra una definición formal de un tipo de documento y, a la vez, especifica la estructura lógica de cada documento. Define tanto los elementos de una página como sus atributos. El DTD del XML es opcional. En tareas sencillas no es necesario construir una DTD, entonces se trataría de un documento "bien formado"(well-formed) y si lleva DTD será un documento "validado" (valid). [20]

-eXtensible Stylesheet Language (XSL): Define o implementa el lenguaje de estilo de los documentos escritos para XML. Permite modificar el aspecto de un documento. Se pueden lograr múltiples columnas, texto girado, orden de visualización de los datos de una tabla, múltiples tipos de letra con amplia variedad en los tamaños. Por otra parte, XSL puede ser utilizado donde se requiera más potencia de diseño como documentos XML que encierran datos estructurados (tablas, organigramas, etc.).

-eXtensible Linking Language (XLL): Define el modo de enlace entre diferentes enlaces. Este lenguaje de enlaces extensible tiene dos importantes componentes: Xlink y el Xpointer. Va más allá de los enlaces simples que sólo soporta el HTML.

-XML User Agent (XUA): Estandarización de navegadores XML. Todavía está en proceso de creación de borradores de trabajo. Se aplicará a los navegadores para que compartan todas las especificaciones XML.

1.6.13 Javascript.

JavaScript es un lenguaje de scripting basado en objetos, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. JavaScript es un dialecto de ECMAScript y se caracteriza por ser un lenguaje basado en prototipos, con entrada dinámica y con funciones de primera clase. JavaScript ha tenido influencia de múltiples lenguajes y se diseñó con una sintaxis similar al lenguaje de programación Java, aunque más fácil de utilizar para personas que no programan. [21]



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web y para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model ó Modelo de Objetos del Documento en español (DOM).

Para evitar incompatibilidades, el World Wide Web Consortium diseñó el estándar DOM, que incorporan Konqueror, las versiones 6 de Internet Explorer y Netscape Navigator, Opera la versión 7, y Mozilla Application Suite, Mozilla desde su primera versión. [22]

JavaScript puede incluirse en cualquier documento y es compatible con HTML en el navegador del cliente, ya sea PHP, Active Server Pages, ASP, JSP y SVG. Incluir código directamente en una estructura HTML es una práctica invasiva y no recomendada. El método correcto que define la W3C es incluir javascript como un archivo externo, tanto por cuestiones de accesibilidad, como práctica y velocidad en la navegación. [23]

1.6.14 BIND.

BIND es una implementación del protocolo DNS. El nombre BIND significa “Berkeley Internet Name Domain”, ya que el programa se creó en la década de 1980 en la Universidad de California en Berkeley. Una nueva versión de BIND (BIND 9) fue escrita desde cero en parte para superar las dificultades arquitectónicas presentes anteriormente para auditar el código en las primeras versiones de BIND, y también para incorporar DNS Security Extensions (DNSSEC). BIND 9 incluye entre otras características importantes: TSIG, notificación DNS, nsupdate, IPv6, rndc flush, vistas, procesamiento en paralelo, y una arquitectura mejorada en cuanto a portabilidad. Es comúnmente usado en sistemas GNU/Linux. [24]

BIND es el servidor de nombres de dominio más popular en Internet, que trabaja en todas las plataformas informáticas principales y se caracteriza por su flexibilidad y seguridad. [25]

1.7 Metodologías de Desarrollo de software.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando que personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Especifican la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla. [26]

Metodología SXP.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

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 propiciando 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.

SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que sepamos por dónde andamos. XP más bien es una metodología encaminada para el desarrollo; consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar el éxito del proyecto.

Consta de 4 fases principales:

- Planificación-Definición donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- Desarrollo, es donde se realiza la implementación del sistema hasta que esté listo para ser entregado.
- Entrega, puesta en marcha.
- Mantenimiento, donde se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, se prioriza la Lista de Reserva del Producto, definición de las Historias de Usuario, diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, lo que nos permite mejorar el diseño cada vez que se le añada una nueva funcionalidad. [26]

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, que posean requisitos 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 precisa el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo.

En este capítulo se exponen las principales características de los sistemas de perfiles centralizados, su importancia, ventajas y desventajas, además se realiza un estudio de las principales tecnologías que implementan sistemas de perfiles centralizados en la actualidad, analizando de estas sus



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

ventajas y desventajas. Se describen las herramientas, lenguajes y tecnologías a utilizar en el desarrollo futuro de la solución propuesta y además la metodología de desarrollo del software seleccionada.



CAPÍTULO 2. Descripción y Análisis de la solución propuesta.

El Sistema de Perfiles Centralizados (SPC) es una solución viable para entornos de desarrollo de software que utilizan GNU/Linux como plataforma de desarrollo, como la mayoría de los proyectos productivos de la UCI.

El **SPC** permite integrar un conjunto de aplicaciones basadas en sistemas GNU/Linux con el objetivo de exportar desde un servidor central el perfil de cada usuario (entiéndase configuraciones personales de las diferentes aplicaciones con las cuales trabaja el usuario y datos almacenados en el escritorio), también posee una interfaz administrativa que permite configurar el sistema de una manera fácil y sencilla con el objetivo de simplificar todo el trabajo del administrador que va a interactuar con el sistema.

2.1 Descripción del Sistema de Perfiles Centralizados.

Después de haber realizado un estudio de las diferentes tecnologías que implementan las características que posee un sistema de perfiles centralizados se puede llegar a la conclusión de que para los entornos de desarrollo productivos y en el caso específico de la UCI estas aplicaciones no se encuentran fusionadas de manera que faciliten el trabajo del administrador del sistema en una solución integral que proporcione personalización, escalabilidad y fiabilidad.

El sistema está compuesto por un conjunto de aplicaciones que son utilizadas en la PC cliente y otro conjunto de aplicaciones que son usadas por el servidor, en la que se incluye una aplicación web para administrar los diferentes servicios que brinda la solución.

Las aplicaciones que componen la parte cliente son las que permiten intercambiar con el servidor a través de peticiones que se le envían, las cuales están relacionadas en la siguiente tabla:

Nombre del paquete	Descripción
nfs-common	Este paquete se utiliza en cualquier máquina que usa NFS, ya sea como cliente o servidor. Programas incluidos: nfsstat, lockd, statd, showmount, gssdy, idmapd.
Portmap	Este paquete proporciona un servidor que convierte los números del programa RPC (llamada a procedimiento remoto) a números de puerto del protocolo de DARPA. Se debe estar ejecutando para hacer llamadas RPC.



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

	Entre los servicios que usan RPC se incluyen NFS y NIS.
libnss-ldap	Este paquete permite a un servidor LDAP actuar como un servidor de nombres. Esto significa que provee la información de las cuentas de usuario, los IDs de los grupos, la información de la máquina, los alias, los grupos de red y básicamente cualquier cosa que normalmente se obtiene desde los archivos almacenados bajo /etc o desde un servidor NIS.
libpam-ldap	Este paquete proporciona una interfaz entre un servidor LDAP y PAM para la autenticación de usuarios (comprobación de claves) a aquellas aplicaciones que utilicen PAM. Su uso junto con libnss-ldap permite al LDAP sustituir completamente otros métodos de búsqueda (como NIS o archivo plano) para cuadros de cuentas del sistema.
nscd	Este paquete básicamente es un demonio que se encarga de los ficheros passwd, group y búsquedas de acogida para los programas en ejecución y almacena en caché los resultados para las próximas consultas. Usted debe instalar este paquete sólo si se utiliza servicios lentos como LDAP, NIS o NIS+.

Las aplicaciones que componen la parte del servidor son las que permiten intercambiar con los clientes respondiendo a las solicitudes que estos realizan y que se encuentran relacionadas en la siguiente tabla:

Nombre del paquete	Descripción
nfs-common	Este paquete se utiliza en cualquier máquina que usa NFS, ya sea como cliente o servidor. Programas incluidos: nfsstat, lockd, statd, showmount, gssdy, idmapd.
portmap	Este paquete proporciona un servidor que convierte los números del programa RPC (llamada a procedimiento remoto) a números de puerto del protocolo de DARPA. Se debe estar ejecutando para hacer llamadas RPC.



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

	<p>Entre los servicios que usan RPC se incluyen NFS y NIS.</p>
nfs-kernel-server	<p>El servidor NFS es actualmente el núcleo del servidor NFS se recomienda su uso con GNU/Linux, con características como NFSv3 y NFSv4, Kerberos apoyo a través de SGS, y mucho más. También es significativamente más rápido y suele ser más fiable que el NFS servidores de espacio de usuario (de la unfs3 y paquetes nfs-server por el usuario). Este paquete contiene el apoyo a los usuarios del espacio necesario para el aprovechamiento de la Servidor NFS del núcleo. Son utilizados por mayoría de los administradores que deseen configurar un servidor NFS.</p>
ldap-utils	<p>Este paquete proporciona utilidades a la implementación del servidor OpenLDAP. Estas utilidades pueden acceder a un servidor LDAP local o remoto y contener todos los programas de cliente necesario para acceder a los servidores LDAP.</p>
libnss-ldap	<p>Este paquete permite a un servidor LDAP actuar como un servidor de nombres. Esto significa que provee la información de las cuentas de usuario, los IDs de los grupos, la información de la máquina, los alias, los grupos de red y básicamente cualquier cosa que normalmente se obtiene desde los archivos almacenados bajo /etc o desde un servidor NIS.</p>
libpam-ldap	<p>Este paquete proporciona una interfaz entre un servidor LDAP y PAM para la autenticación de usuarios (comprobación de claves) a aquellas aplicaciones que utilicen PAM. Su uso junto con libnss-ldap permite al LDAP sustituir completamente otros métodos de búsqueda (como NIS o archivo plano) para cuadros de cuentas del sistema.</p>
	<p>Este paquete básicamente es un demonio que se encarga de los ficheros passwd, group y búsquedas de acogida para los programas en ejecución y almacena en caché los resultados para las próximas consultas. Usted</p>



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

nscd	debe instalar este paquete sólo si se utiliza servicios lentos como LDAP, NIS o NIS+.
slapd	Es el servidor LDAP y puede ser utilizado para proporcionar un directorio independiente de servicios. Dominio de servidor y herramientas
quota	Este paquete permite especificar dos tipos de límites de espacio dentro de los sistemas de archivos, por usuario y/o por grupos de usuarios. Y dentro de estos dos tipos, por un lado el número máximo de inodos, y por otro lado el número máximo de bloques de disco.
apache2	Cuenta con soporte para HTTPS, hosting virtual, CGI, SSI, Ipv6 y la integración de bases de datos, solicitud/respuesta de filtrado, muchos esquemas flexibles de autenticación, y mucho más.

2.2 Estrategia de integración.

La reutilización de componentes resulta sumamente importante porque nos proporciona beneficios significativos, uno de ellos es el de ser primordial a la hora de reducir los tiempos y costos del desarrollo además permite realizar aplicaciones eficientes, robustas y de gran fiabilidad.

Para dar solución al sistema propuesto en el presente trabajo es necesario hacer uso de otros componentes ya existentes que son de vital importancia para dar cumplimiento a los requerimientos del sistema.

Primeramente se habla de la tecnología a utilizar para poder exportar los perfiles de cada usuario hacia cada PC cliente para esto se escogió el NFS (ver epígrafe 1.6.7), traducido al español Sistema de Archivos de Red, protocolo de nivel de aplicación del modelo de Interconexión de Sistemas Abiertos (OSI) que permite que varios sistemas conectados en un entorno de red de área local accedan a ficheros remotos como si fueran locales dentro de la misma PC.

Con utilización de la implementación de este protocolo se logra montar el directorio /home del servidor que posee los perfiles de usuario de cada integrante del proyecto en el cliente, con esto se



logra una de las principales funcionalidades de la solución que es poder almacenar y centralizar todas las configuraciones de todos los usuarios del sistema.

Mediante la implementación de cuotas de disco denominada QUOTA (ver epígrafe 1.6.10) se administra las cuotas de todos los usuarios teniendo en cuenta la cantidad de espacio de almacenamiento con que cuenta el servidor, administrando la cuotas de los usuarios según las necesidades de cada uno de ellos dependiendo del rol que desempeñan en el proyecto.

Para que el SPC funcione correctamente en la configuración de la parte cliente se necesita la dirección IP del servidor y con el objetivo de que el cliente no tenga que configurarse de nuevo si la dirección IP del servidor cambia, se utiliza la implementación libre del Sistema de Nombre de Dominio o DNS por sus siglas en inglés denominado BIND (ver epígrafe 1.6.14) comúnmente usado en Internet para 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 a estos equipos.

Con el objetivo de gestionar y almacenar la información de autenticación (usuario y contraseña), datos de contacto del usuario (nombre, apellidos, UID), permisos y grupos a los que pertenece se hace uso de **OpenLDAP** una implementación libre del protocolo LDAP.

2.3 Funcionamiento del SPC.

Con el objetivo de definir y describir el funcionamiento del SPC, de debe primeramente identificar los componentes físicos que van a pertenecer al sistema, unido a los requerimientos previos que se necesitan para que funcione correctamente.

Para el servidor del SPC se necesita una computadora personal con al menos 256 MB de memoria RAM, un dispositivo de almacenamiento o disco duro con capacidad 80 GB como mínimo, como sistema operativo Debian GNU/Linux u otra distribución basada en esta, y tener instalado el paquete de aplicaciones correspondiente al servidor del SPC denominado ServidorSPC. Se debe destacar que la unidad de almacenamiento debe estar particionada de la siguiente manera:

Unidad de almacenamiento o disco duro perteneciente al servidor:

- Primera partición: destinada para la SWAP del sistema operativo, es una zona o partición del disco duro cuyo objetivo es el de guardar datos que no han de mantenerse en memoria física.
- Segunda partición: destinada específicamente para almacenar los perfiles de usuarios del SPC.
- Tercera partición: destinada para almacenar el directorio raíz del sistema operativo.



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Para el cliente del SPC se necesita una computadora personal con al menos un dispositivo de almacenamiento o disco duro, como sistema operativo Debian GNU/Linux o alguna otra distribución basada en esta, y tener instalado el paquete de aplicaciones correspondiente al cliente del SPC denominado ClienteSPC. Se debe destacar que la unidad de almacenamiento debe estar particionada de la siguiente forma:

Unidad de almacenamiento o disco duro perteneciente al cliente:

- Primera partición: destinada para la SWAP del sistema operativo, es una zona del disco duro de la computadora cuyo objetivo es el de guardar datos que no han de mantenerse en memoria física.
- Segunda partición: destinada para almacenar el directorio raíz del sistema operativo.

Concluida la instalación se procede a la configuración de los mismos. En las computadoras donde se encuentra el conjunto de aplicaciones cliente se debe especificar en el archivo de configuración que coordina el orden de consultas de los servidores DNS asociados al cliente, la dirección IP del servidor del SPC, se debe señalar que este archivo se encuentra en (/etc/resolv.conf). Para el caso del servidor, el administrador del sistema haciendo uso de la interfaz web administrativa SPCAdmin concluye todas las configuraciones correspondientes al servidor.

Posteriormente cuando concluye la configuración del SPC el usuario inicia el sistema operativo donde instaló el paquete cliente en este caso Debian GNU/Linux, en el transcurso del arranque primeramente se solicita que se resuelva una dirección IP a partir de un nombre de dominio, en este caso es la dirección IP del servidor del SPC enviando una petición al servidor DNS el cual responde con la dirección IP correspondiente al nombre de dominio. Posteriormente el cliente del servidor NFS solicita que se monte el directorio personal de los usuarios que se encuentra en el servidor SPC, en el directorio de los usuarios de la PC cliente, después cuando concluye el inicio del sistema operativo, el directorio que contiene los perfiles de los usuarios que pertenecen al SPC y que están almacenados en el servidor queda montado en la PC cliente como si se tratase de un directorio local. Seguidamente el usuario del sistema procede a iniciar sesión introduciendo sus datos para el proceso de autenticación y autorización, esto se realiza utilizando LDAP como mecanismo centralizado de autenticación, combinado con los Módulos de Autenticación Insertables o PAM por sus siglas en inglés que poseen actualmente los sistemas GNU/Linux, los datos entrados se comprueban si se encuentran almacenados en el directorio LDAP y en caso de autorización el usuario inicia sesión satisfactoriamente.

2.4 Funcionamiento del SPCAdmin.



Como parte de las herramientas que componen la parte del servidor se encuentra el Administrador del Sistema de Perfiles Centralizados (**SPCAAdmin**), que es una aplicación que posee una interfaz web que permite administrar de manera fácil e intuitiva todos los servicios que brinda la solución propuesta. En la siguiente figura se describe su funcionamiento de manera general:

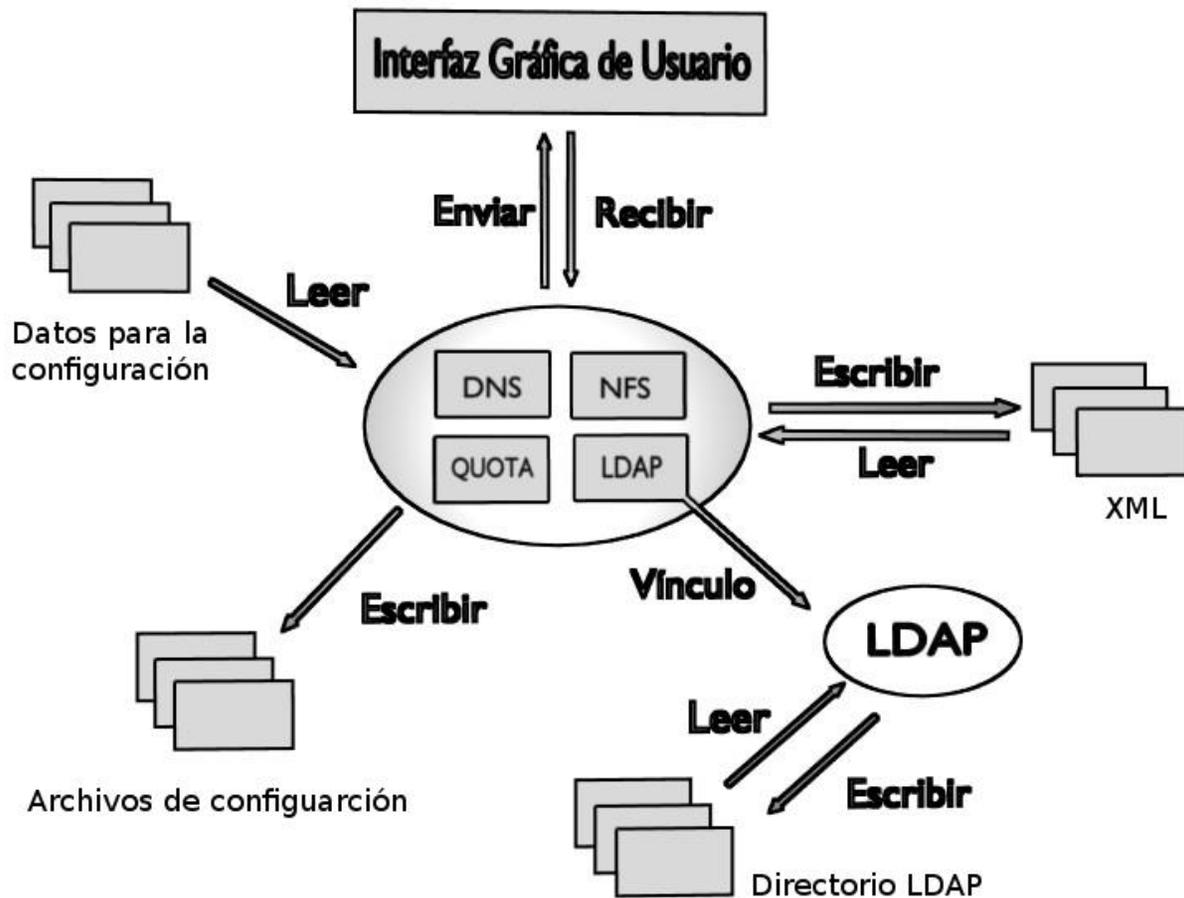


Figura 1. Descripción del SPCAdmin.

Como se puede apreciar en la figura el **SPCAAdmin** cuenta primeramente con una interfaz gráfica de usuario que se encarga de interactuar con el administrador de sistema, proporcionarle datos al módulo principal y mostrar la información que recibe de este para ser mostrada posteriormente al usuario, el módulo principal está compuesto por cuatro módulos que a continuación se describen en la siguiente tabla:

Nombre del módulo	Descripción
-------------------	-------------



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

NFS	Este módulo gestiona los archivos de configuración que permite administrar un servidor NFS que se encarga de exportar el directorio /home de cada usuario que se encuentra en el servidor hacia cada cliente NFS, y a través de las opciones de administración se modifican la lista de control de acceso para los directorios específicos del sistemas de archivos (perfil de cada usuario) que van a ser exportados a cada cliente NFS, además de exportarlo con los permisos que permiten a los usuarios leer y escribir sobre su perfil.
DNS	Este módulo gestiona los archivos de configuración que permite administrar un servidor DNS o servidor de nombres de dominio destinado específicamente para una red local, es preciso destacar que este módulo no pretende administrar todas las opciones que posee cualquier implementación de DNS, sólo se encarga de una configuración estricta para que funcione correctamente la solución propuesta.
QUOTA	Este módulo gestiona toda la configuración que permite especificar dos tipos de límites de espacio dentro de los sistemas de archivos, por usuario y/o por grupos de usuarios. Y dentro de estos dos tipos, por un lado el número máximo de inodos, y por otro lado el número máximo de bloques de disco. Esta característica permitirá controlar el espacio consumido de cada uno de los usuarios del sistema teniendo en cuenta las características del servidor.
LDAP	Este módulo gestiona toda la configuración que permite administrar el servicio de directorio LDAP, a través de phpldapadmin un administrador web destinado para configurar un servidor LDAP.

El módulo principal interactúa con la estructura de ficheros que poseen los datos que son utilizados para la configuración de los diferentes servicios, obtiene de estos la información que necesita para luego ser procesada, y posteriormente ser escrita en los ficheros de configuración, estos son utilizados por las diferentes implementaciones integradas al sistema para configurar sus servicios, también esta información es escrita en los archivos de configuración del SPCAdmin que utilizan XML



para almacenar esta información, la persistencia de los datos es fundamental para proporcionarle al usuario final información actualizada de los cambios realizados, esto se logra mediante la lectura de los archivos de configuración del SPCAdmin y es visualizada a través de la interfaz gráfica de usuario. Por último el módulo LDAP se gestiona mediante un vínculo que se realiza a un sistema de administración en el caso específico de los archivos del directorio LDAP.

2.5 Breve Definición de la Arquitectura del Sistema.

La arquitectura de software nos proporciona una visión abstracta de alto nivel de un sistema formado por sus componentes, sin especificar detalladamente a cada uno de los módulos que lo componen, representa la organización de cada componente que pertenece a un determinado sistema con el objetivo de satisfacer sus funcionalidades, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad.

La arquitectura de software establece los fundamentos para que analistas, diseñadores, probadores, programadores, y el resto de los integrantes del equipo o grupo de desarrollo trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades.

2.5.1 Arquitectura del SPC.

La arquitectura base del **SPC** está conformada por la arquitectura Cliente-Servidor, esto se debe a que el **SPC** está conformado por un conjunto de aplicaciones que están basados en esta arquitectura. La capacidad operacional de procesamiento se encuentra distribuida entre las terminales clientes y el servidor, aunque se debe señalar que esta presenta ventajas más bien de tipo organizativa debido a la centralización de la gestión de la información y la asignación de las responsabilidades, esto posibilita que se facilite el trabajo y que se realice de una forma más clara el diseño del sistema, además el acceso a los recursos y la integridad de los datos son gestionados por el servidor, de esta forma no se puede acceder a la información almacenada de forma no autorizada y permite actualizar los datos y otros recursos.

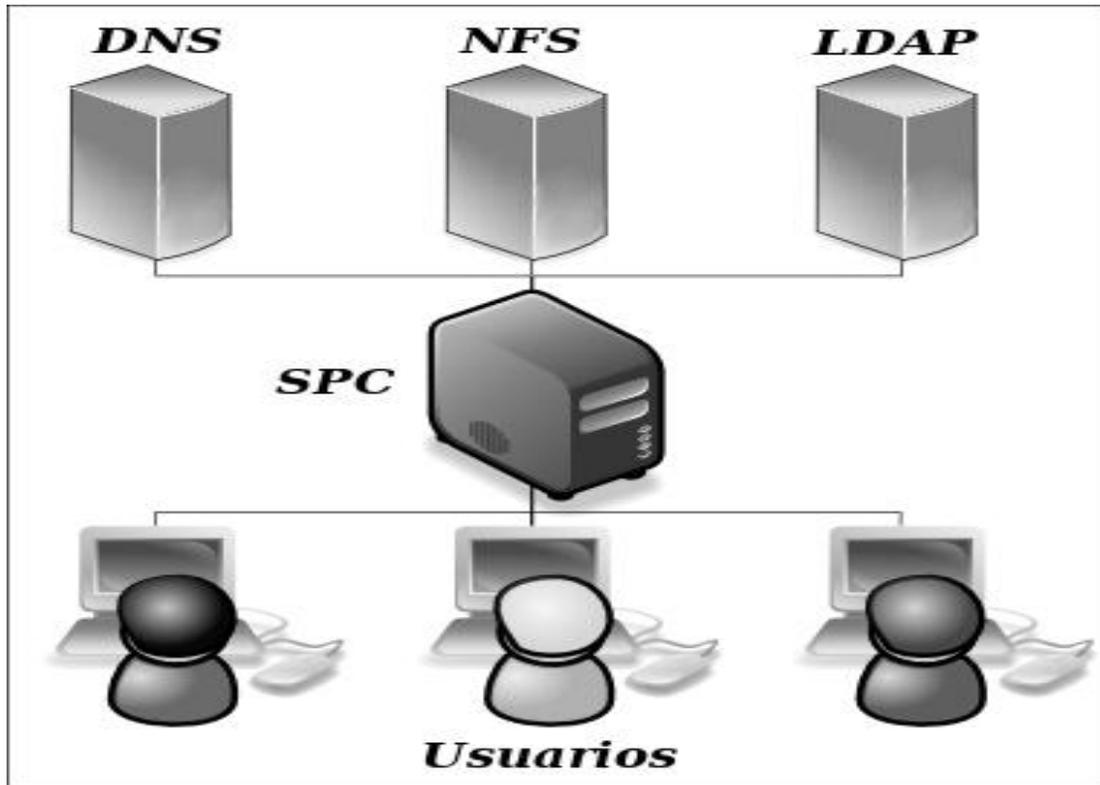


Figura 2. Descripción del SPC.

2.5.2 Arquitectura del SPCAdmin.

La arquitectura en capas define una organización jerárquica, lo que posibilita un diseño basado en niveles de abstracción creciente, posibilitando a los desarrolladores, fragmentar un problema en una secuencia de pasos incrementales. Este estilo de desarrollo en varios niveles, facilita que en caso que ocurra algún cambio, solo se tendrían que realizar las correcciones necesarias en el nivel requerido sin tener que revisar código de otros niveles.

La programación por capas es un estilo de programación en la que el objetivo primordial es dividir, fraccionar y llegar a separar la Presentación (donde muestras y obtienes datos), de la Lógica de Negocio (donde realizas operaciones) y con todo esto se obtiene independencia, por lo que si hay cambios de arquitectura se puede acceder a los datos o cambiar el negocio o modificar la presentación y solo sería en esa fracción de la capa. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).



La arquitectura base del **SPCAdmin** está conformada por una arquitectura de 3 capas debido a las ventajas que presenta este patrón particularmente.

A continuación se presenta un esquema donde se ve aplicado este patrón:

- Capa de presentación: la cual está centrada en la presentación al usuario que va a usar la aplicación, conformada por una interfaz de usuario desarrollada con una interfaz web.
- Capa de lógica de negocio: contiene el núcleo de la aplicación con sus principales módulos y funcionalidades.
- Capa de Abstracción de datos: la cual está conformada por las diferentes estructuras de archivos en las cuales se almacenan los datos.

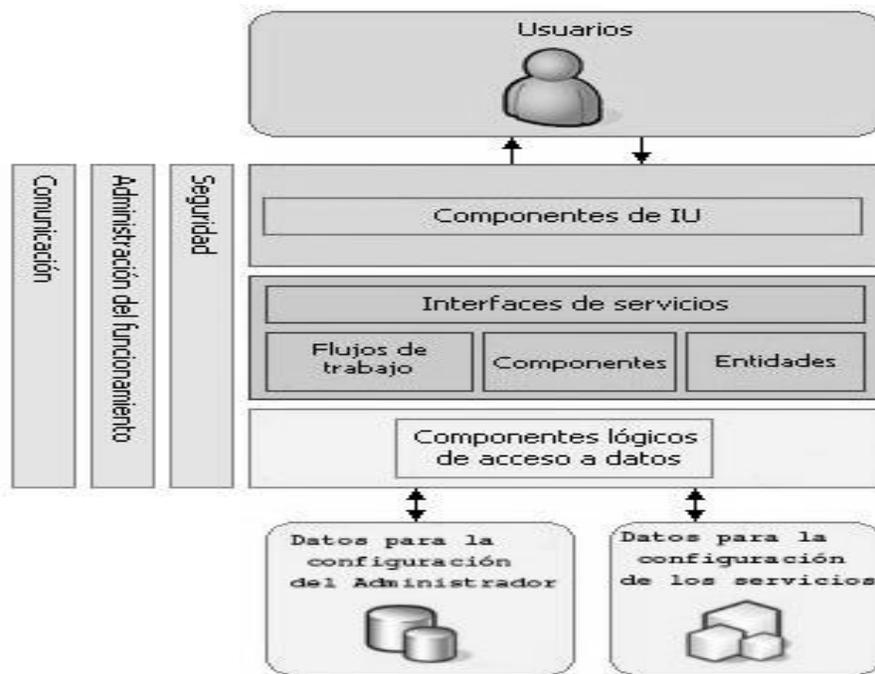


Figura 3.Descripció de Arquitectura tres capas.

2.6 Diagrama de clases del SPCAdmin.

El **diagrama de clases** describe de manera estructural como está formado un sistema mostrando sus clases, atributos y las relaciones entre ellos. Son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro. El diagrama de clases incluye mucha más información como la relación entre un objeto y otro, la herencia de



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

propiedades de otro objeto, conjuntos de operaciones/propiedades que son implementadas para una interfaz. A continuación se relaciona en la siguiente figura el diagrama de clases del SPCAdmin y posteriormente una breve descripción de cada clase de diseño.

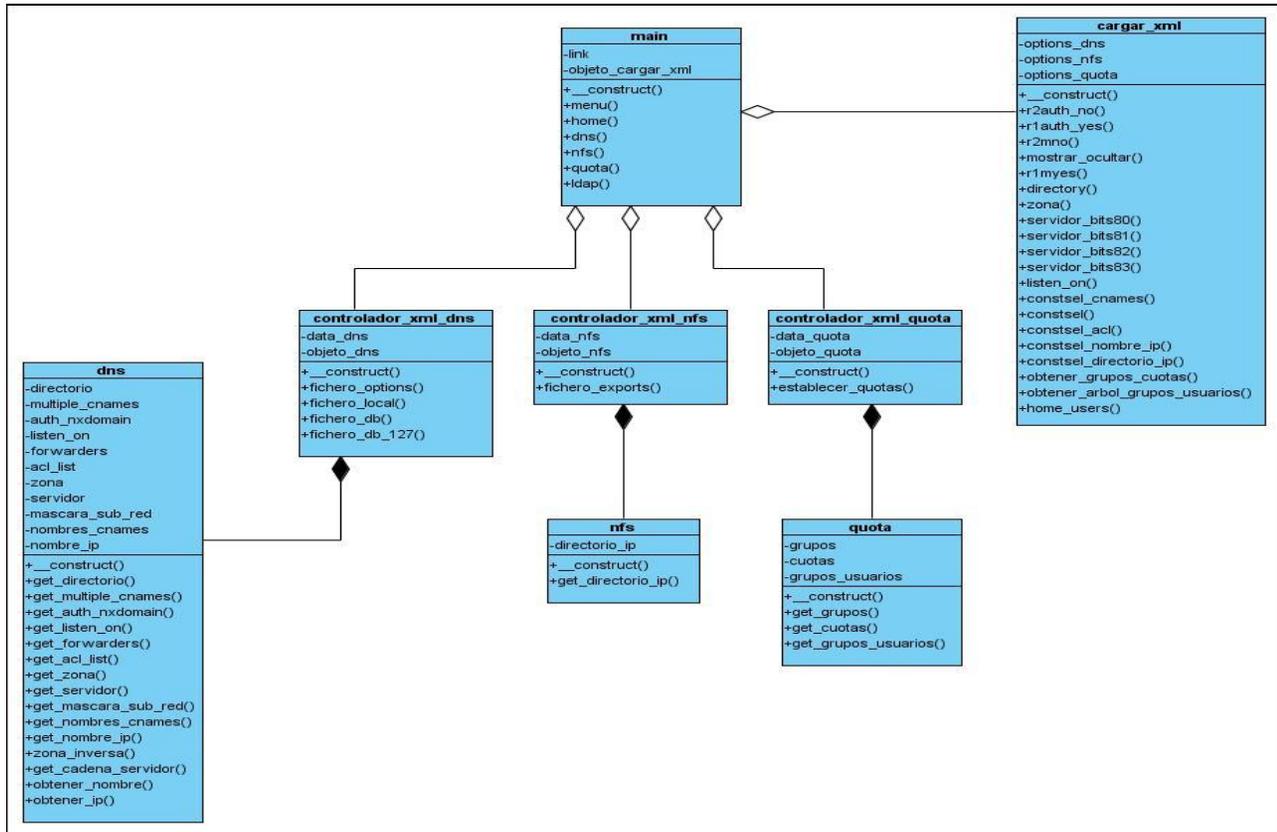


Figura 4. Diagrama de Clases del SPCAdmin.

Nombre de la clase: main		Módulo: Ninguno
Descripción: Según el valor que tenga atributo link esta clase construye los formularios que contienen los datos por cada módulo.		
Atributo	Tipo	Descripción
link	int	Almacena el identificador para cada módulo.
objeto_cargar_xml	cargar_xml	Lee los datos almacenados en el archivo xml.



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Nombre de la clase: controlador_xml_dns		Módulo: DNS
Descripción: Esta clase es la encargada de construir el archivo data_dns.xml y modifica los archivos de configuración de la implementación del servidor DNS.		
Atributo	Tipo	Descripción
data_dns	SimpleXMLElement	Construye el archivo data_dns.xml y crea la estructura de los datos que se van a almacenar.
objeto_dns	dns	Contiene los datos del fichero de configuración de la implementación del servidor DNS.

Nombre de la clase: controlador_xml_nfs		Módulo: NFS
Descripción: Esta clase es la encargada de construir el archivo data_nfs.xml y modifica los archivos de configuración de la implementación del servidor NFS.		
Atributo	Tipo	Descripción
data_nfs	SimpleXMLElement	Construye el archivo data_nfs.xml y crea la estructura de los datos que se van a almacenar
objeto_nfs	nfs	Contiene los datos del fichero de configuración de la implementación del servidor NFS.

Nombre de la clase: controlador_xml_quota		Módulo: QUOTA
Descripción: : Esta clase es la encargada de construir el archivo data_quota.xml		
Atributo	Tipo	Descripción
data_quota	SimpleXMLElement	Construye el archivo data_nfs.xml y crea la estructura de los datos que se van a almacenar
objeto_quota	quota	Contiene los datos del fichero de configuración de la implementación de las cuotas de disco en sistemas GNU/Linux.

Nombre de la clase: cargar_xml		Módulo: Ninguno
---------------------------------------	--	------------------------



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Descripción: Esta clase se encarga de leer los datos contenidos en los archivos data_dns.xml, data_nfs.xml, data_quota.xml y almacenarlos en la memoria principal para ser procesados posteriormente.

Atributo	Tipo	Descripción
options_dns	SimpleXMLElement	Lee los datos almacenados en el archivo data_dns.xml.
options_nfs	SimpleXMLElement	Lee los datos almacenados en el archivo data_nfs.xml.
options_quota	SimpleXMLElement	Lee los datos almacenados en el archivo data_quota.xml.

Nombre de la clase: dns

Módulo: DNS

Descripción: Esta clase se encarga de almacenar los datos que se utilizarán posteriormente para modificar las opciones de configuración del servidor DNS.

Atributo	Tipo	Descripción
directorio	string	Se encarga de almacenar la estructura del directorio.
multiple_cnames	string	Se encarga de almacenar el valor de la opción multiple_cnames (yes, no)
auth_nxdomain	string	Se encarga de almacenar el valor de la opción auth_nxdomain (yes, no)
listen_on	string	Se encarga de almacenar el nombre de la lista de control de acceso
forwarders	string array	Se encarga de almacenar la dirección IP de los servidores pertenecientes a la jerarquía de reenvío
acl_list	string array	Se encarga de almacenar la dirección IP que pertenecen a la lista de control de acceso
zona	string	Se encarga de almacenar el nombre de la zona



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

		de dominio
servidor	string array	Se encarga de almacenar la dirección IP del servidor
maskara_sub_red	string array	Se encarga de almacenar la máscara de subred del servidor
nombres_cnames	string array	Se encarga de almacenar los alias del servidor DNS
nombre_ip	string array	Se encarga de almacenar el nombre y su respectiva dirección IP a resolver por el servidor DNS.

Nombre de la clase: nfs		Módulo: NFS
Descripción: Esta clase se encarga de almacenar los datos que se utilizarán posteriormente para modificar las opciones de configuración del servidor NFS.		
Atributo	Tipo	Descripción
directorio_ip	string array	Se encarga de almacenar la dirección del directorio a compartir y la dirección IP o sintaxis Máscara de Subred de Longitud Variable (VLSM) para una subred.
permisos	string array	Se encarga de almacenar las opciones que configuran los permisos sobre los directorios compartidos a través de NFS.

Nombre de la clase: quota		Módulo:
Descripción: Esta clase se encarga de almacenar los datos que se utilizarán posteriormente para modificar las opciones de configuración de las cuotas de disco en sistemas GNU/Linux.		
Atributo	Tipo	Descripción
grupos	string array	Se encarga de almacenar el nombre de los grupos de usuarios.



cuotas	string array	Se encarga de almacenar las cuotas por grupos de usuarios.
grupos_usuarios	string array	Se encarga de almacenar los usuarios por grupos.

2.7 Diagrama de componentes del SPCAdmin.

Para que el lector tenga una idea de los diagramas que se presentan a continuación se puede decir que los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación.

Es un diagrama que muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones.

Es un grafo de componentes unidos a través de relaciones que pueden ser de compilación o de ejecución, y además se pueden representar las interfaces de esos componentes. Interfaces se refiere a la descripción de las operaciones que se realizan en esos componentes y en esos subsistemas.



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

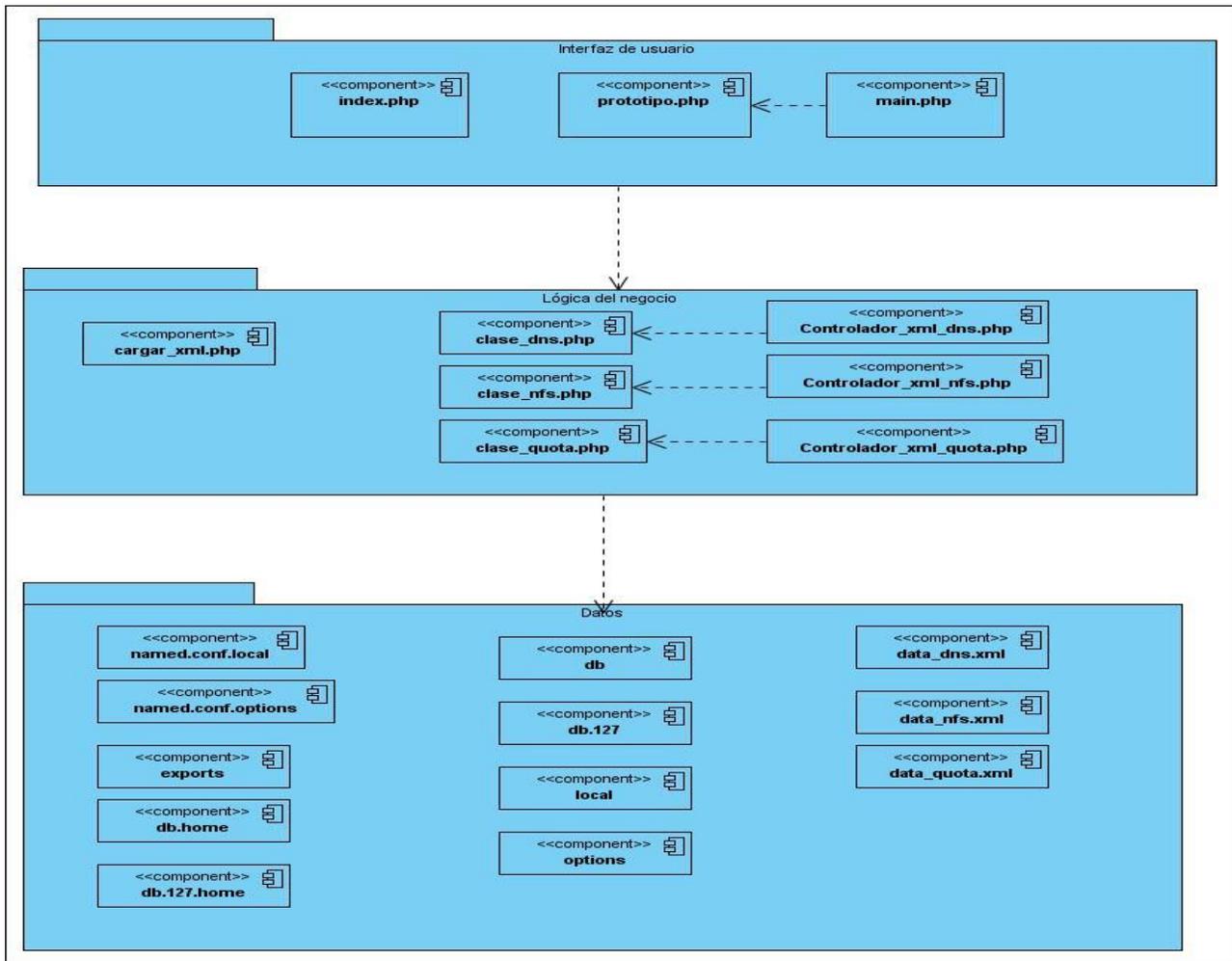


Figura 5. Diagrama de Componentes del SPCAdmin.

A continuación una tabla que describe cada uno de los componentes representados en el diagrama anterior:

Componente	Propósito
index.php	Este componente se utiliza para mostrar el formulario para autenticarse en el sistema.
prototipo.php	Este componente muestra la interfaz cuando el usuario accede al sitio después de autenticarse.



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

main.php	Este componente se encarga de construir las diferentes páginas que conforman la interfaz de usuario.
cargar_xml.php	Este componente se encarga de leer el contenido que se encuentra almacenado en los archivos xml para ser procesados posteriormente.
clase_dns.php	Este componente se encarga de almacenar la información referente a la configuración del servidor DNS.
clase_nfs.php	Este componente se encarga de almacenar la información referente a la configuración del servidor NFS.
clase_quota.php	Este componente se encarga de almacenar la información referente a la configuración del servidor de las cuotas de disco pertenecientes a cada grupo de usuario.
controlador_xml_dns.php	Este componente se encarga de crear el componente data_dns.xml y los archivos de configuración para el servidor DNS.
controlador_xml_nfs.php	Este componente se encarga de crear el componente data_nfs.xml y los archivos de configuración para el servidor NFS.
controlador_xml_quota.php	Este componente se encarga de crear el componente data_quota.xml.
named.conf.local	Este componente se encarga de almacenar la información que el servidor DNS lee para configurar



	sus servicios.
named.conf.options	Este componente se encarga de almacenar la información que el servidor DNS lee para configurar sus servicios.
exports	Este componente almacena la información de la zona directa que es utilizada por el servidor DNS para resolver direcciones IP.
db.home	Este componente almacena la información de la zona directa que es utilizada por el servidor DNS para resolver direcciones IP.
db.127.home	Este componente almacena la información de la zona inversa que es utilizada por el servidor DNS para resolver nombres de dominio.
data_dns.xml	Este componente se encarga de almacenar la información relacionada con el servidor DNS que se utiliza para mostrar información actualizada al usuario.
data_nfs.xml	Este componente se encarga de almacenar la información relacionada con el servidor NFS que se utiliza para mostrar información actualizada al usuario.
data_quota.xml	Este componente se encarga de almacenar la información sobre las cuotas de disco que se utiliza para mostrar información actualizada al usuario.

2.8 Diagrama de despliegue del SPCAdmin.

El diagrama 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. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar. Muestra la configuración de los componentes hardware, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución. En este tipo de diagramas intervienen nodos, asociaciones de comunicación, componentes dentro de los nodos y objetos que se encuentran a su vez dentro de los componentes. Un nodo es un objeto físico en tiempo de ejecución, es decir una máquina que se compone habitualmente de, por lo menos, memoria y capacidad de procesamiento, a su vez puede estar



formada por otros componentes. El diagrama de despliegue muestra la topología del hardware sobre el que se ejecuta el sistema.

En el diagrama de despliegue se sitúa el software en el hardware que lo contiene. En este caso el **SPCAdmin** se encuentra en un servidor web en este caso Apache se comunica con las PC clientes mediante conexión HTTP.

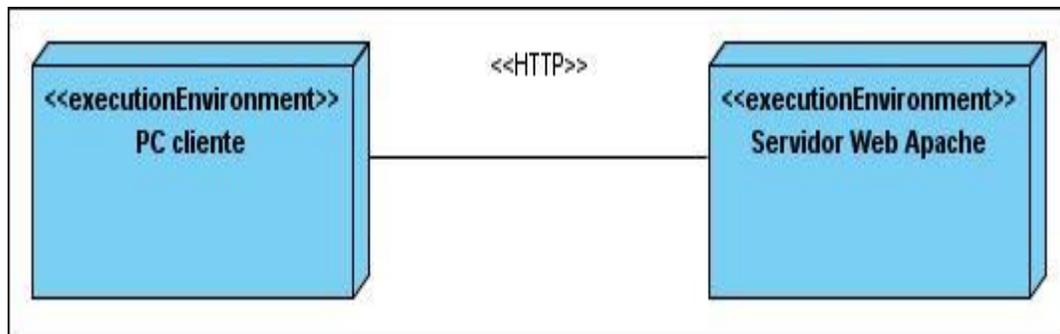


Figura 6. Diagrama de Despliegue del SPCAdmin.

En este capítulo se mostró al lector una descripción detallada de la solución propuesta, la base de su funcionamiento y los componentes que lo integran, para brindar un servicio óptimo y fiable. Así como una breve definición de la arquitectura del sistema propuesto y el análisis y diseño de la aplicación web de administración del Sistema de Perfiles Centralizados.



CAPÍTULO 3. Desarrollo Ágil del Sistema de Perfiles Centralizados.

En este capítulo se realiza el desarrollo ágil del Sistema de Perfiles Centralizados. Se explica toda la dinámica del proyecto en forma de historias de usuarios, prototipos de interfaz de usuario, tareas de ingeniería y algunos modelos auxiliares.

3.1 Planificación del proyecto por roles.

Rol	Responsabilidad	Miembro
Gerente (Management)	Es el responsable de tomar las decisiones finales, acerca de estándares y convenciones a seguir durante el proyecto.	Javier Menéndez Rizo Abel Ernesto Cuadrado Sospedra
Cliente (Costumer)	El cliente participa en las tareas que involucran la lista de reserva del producto.	Laboratorios de producción (UCI)
Programadores (Programmers)	Es el encargado de producir el código y escribir las pruebas unitarias. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.	Javier Menéndez Rizo Abel Ernesto Cuadrado Sospedra
Analista (Analyst)	Es el encargado de escribir las historias de usuario y las pruebas funcionales para validar su implementación.	Javier Menéndez Rizo Abel Ernesto Cuadrado Sospedra
Diseñadores (Designers)	Encargados del diseño del sistema; así como el de los prototipos de interfaces, máximos responsables de la realización del diseño de las metáforas y supervisan el proceso de construcción.	Javier Menéndez Rizo Abel Ernesto Cuadrado Sospedra



Encargado de Pruebas (Tester)	Es el encargado de ayudar al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.	Javier Menéndez Rizo Abel Ernesto Cuadrado Sospedra
Arquitecto (Architec)	Se vincula directamente con el analista y el diseñador debido a que su trabajo tiene que ver con la estructura y el diseño en grande del sistema. Ayuda en el diseño de las metáforas.	Javier Menéndez Rizo Abel Ernesto Cuadrado Sospedra

3.2 Lista de Reserva del Producto (LRP).

Una de las actividades más importantes definidas en la metodología SXP es la Lista de Reserva del Producto (LRP), en la cual se recoge en una lista priorizada todo el trabajo a desarrollar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto. Sin embargo, suelen surgir los más importantes que casi siempre son más que suficientes para una iteración.

Esta lista puede crecer y modificarse a medida que se obtienen más conocimientos acerca del producto y del cliente. Con la restricción de que sólo puede cambiarse entre iteraciones. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto. Esta lista puede estar conformada por requerimientos técnicos y del negocio, funciones, errores a reparar, defectos, mejoras y actualizaciones tecnológicas requeridas.

Se debe tener en cuenta que los **requisitos funcionales** son capacidades o necesidades desde el punto de vista del usuario que debe cumplir el sistema y que están fuertemente ligados a las opciones del software.



CAPÍTULO 3: DESARROLLO ÁGIL DEL SISTEMA DE PERFILES CENTRALIZADOS

Y los **requisitos no funcionales** son características que describen alguna forma o restricción para la realización de algún requerimiento (funcionalidad), conjunto de ellos o todos. Se consideran los atributos del sistema, propiedades o cualidades que debe tener el producto.

Asignado a:	Ítem *	Descripción	Estimación	Estimado por
Prioridad			Muy Alta	
Javier	1	Crear lista de control de acceso para el servidor de DNS.	2 días	Abel
Javier	2	Modificar el nombre de la lista de control de acceso del servidor DNS.	2 días	Abel
Javier	3	Adicionar dirección IP o subred a la lista de control de acceso del servidor DNS.	2 días	Abel
Javier	4	Eliminar dirección IP o subred de la lista de control de acceso del servidor DNS.	2 días	Abel
Javier	5	Listar los elementos correspondientes a la lista de control de acceso [dirección IP o subred] del servidor DNS.	2 días	Abel
Javier	6	Crear zona de dominio correspondiente para el servidor DNS.	2 días	Abel



CAPÍTULO 3: DESARROLLO ÁGIL DEL SISTEMA DE PERFILES CENTRALIZADOS

Javier	7	Modificar el nombre de la zona de dominio del servidor DNS.	2 días	Abel
Javier	8	Modificar la dirección IP del servidor DNS que se relaciona con la zona de dominio.	2 días	Abel
Javier	9	Modificar la máscara de red que se relaciona con la zona de dominio.	2 días	Abel
Javier	10	Adicionar dirección IP del servidor DNS a la jerarquía de reenvío.	2 días	Abel
Javier	11	Eliminar dirección IP del servidor DNS a la jerarquía de reenvío.	2 días	Abel
Javier	12	Listar dirección IP de los servidores DNS que pertenecen a la jerarquía de reenvío.	2 días	Abel
Javier	13	Guardar todos los cambios de configuración para el servidor DNS.	2 días	Abel
Javier	14	Adicionar dirección IP o subred a la lista de control de acceso	2 días	Abel



CAPÍTULO 3: DESARROLLO ÁGIL DEL SISTEMA DE PERFILES CENTRALIZADOS

		del servidor NFS.		
Javier	15	Eliminar dirección IP o subred de la lista de control de acceso del servidor NFS.	2 días	Abel
Javier	16	Listar los elementos correspondientes a la lista de control de acceso [dirección IP o subred] del servidor NFS.	2 días	Abel
Javier	17	Guardar todos los cambios de configuración para el servidor NFS.	2 días	Abel
Javier	18	Crear grupos de usuarios con una cuota de disco asignada.	2 días	Abel
Javier	19	Eliminar grupos de usuarios con una cuota de disco asignada.	2 días	Abel
Javier	20	Cargar usuarios del directorio LDAP.	2 días	Abel
Javier	21	Adicionar usuario a un grupo.	2 días	Abel
Javier	22	Eliminar usuario de un grupo.	2 días	Abel
Javier	23	Aplicar las cuotas de disco a los usuarios.	2 días	Abel
Prioridad			Alta	



CAPÍTULO 3: DESARROLLO ÁGIL DEL SISTEMA DE PERFILES CENTRALIZADOS

Javier	1	Automatizar la configuración del cliente y su instalación.	2 días	Abel
Javier	2	Automatizar la configuración del servidor y su instalación.	2 días	Abel
Prioridad			Media	
Javier	1	Habilitar o deshabilitar los nombre múltiples al servidor DNS.	2 días	Abel
Javier	2	Adicionar alias a la opción de nombres múltiples del servidor DNS.	2 días	Abel
Javier	3	Eliminar alias a la opción de nombres múltiples del servidor DNS.	2 días	Abel
Javier	4	Listar los alias de la opción de múltiples nombres.	2 días	Abel
Javier	5	Adicionar hosts [nombre del host y su dirección IP] a la zona de dominio.	2 días	Abel
Javier	6	Eliminar hosts [nombre del host y su dirección IP] a la zona de dominio.	2 días	Abel
		Listar hosts [nombre		



CAPÍTULO 3: DESARROLLO ÁGIL DEL SISTEMA DE PERFILES CENTRALIZADOS

Javier	7	del host y su dirección IP] de la zona de dominio.	2 días	Abel
Prioridad			Baja	
Javier	1	Habilitar o deshabilitar el auth_nsdomain al servidor DNS.	2 días	Abel
Javier	2	Modificar el directorio de trabajo del servidor DNS.	2 días	Abel
RNF (Requisitos No Funcionales)				
	1	Interfaz externa	La interfaz no contiene muchas imágenes para no demorar las respuestas al usuario. El diseño de la interfaz es sencillo y claro de usar con reconocimiento visual a través de elementos visibles que identifiquen cada una de sus acciones. Es formal, serio e intuitivo, todo esto teniendo en cuenta el fin con el que se desarrolla la aplicación.	
	2	Usabilidad	El sistema puede ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente web en sentido general.	
	3	Rendimiento	La disponibilidad de trabajo en red contra el servidor es constante. Se garantiza que la respuesta a solicitudes de los usuarios del sistema, sea en un período de tiempo breve (de segundos) para evitar inconformidades en los usuarios. El sistema deberá ser lo más estable y confiable posible.	
	4	Soporte	Se requiere que el producto reciba mantenimiento ante cualquier fallo que ocurra.	



CAPÍTULO 3: DESARROLLO ÁGIL DEL SISTEMA DE PERFILES CENTRALIZADOS

	5	Portabilidad	El producto está destinado para distribuciones basadas en Debian GNU/Linux incluyendo a la misma, el administrador corre sobre una plataforma web, codificada en PHP 5.
	6	Seguridad	El sistema se encarga de controlar los diferentes niveles de acceso y funcionalidad de usuarios al administrador del sistema, de identificar al usuario antes de que pueda realizar cualquier acción sobre el mismo. Está destinado para un entorno de red local.
	7	Restricciones en el diseño y la implementación	En el caso específico del administrador del sistema es una aplicación web desarrollada con la tecnología para creación de páginas web dinámicas PHP 5.
	8	Legales	Las tecnologías escogidas para el desarrollo de la aplicación, están basadas en la licencia GNU/GPL.
	9	Software	Para el funcionamiento del sistema será necesario distribuciones basadas en Debian GNU/Linux incluyendo a esta distribución.
	10	Hardware	Se necesitan como requerimientos mínimos una PC con procesador Pentium III o superior.
	11	Confidencialidad	Toda la información está protegida del acceso no autorizado, los administradores de sistema son los únicos que podrán gestionar cualquier información.
	12	Disponibilidad	Se garantiza a los usuarios del sistema el acceso a la información solicitada en todo momento (si tiene permiso para ello).
	13	Ayuda y documentación	El producto brindará a los usuarios una documentación apropiada para el trabajo con el mismo, principalmente para el caso de los usuarios que administrarán el sistema.



3.3 Historias de Usuario.

Historia de Usuario	
Número: U-SPC-01	Nombre de Historia de Usuario: Gestionar lista de control de acceso del servidor DNS
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Javier Menéndez Rizo	Iteración Asignada: 1
Prioridad en Negocio: Muy alta	Puntos estimados: 1 semanas
Riesgo en Desarrollo: Alta	Puntos Reales: 1 semanas
Descripción: Esta sección garantiza la gestión de la lista de control de acceso. Crear la lista de control de acceso para el servidor DNS. Modificar la lista de control de acceso para el servidor DNS. Adicionar elementos correspondientes a la lista de control de acceso [dirección IP o subred]. Eliminar elementos correspondientes a la lista de control de acceso [dirección IP o subred]. Listar los elementos correspondientes a la lista de control de acceso [dirección IP o subred].	
Observaciones: Los elementos que se corresponden con la lista de control de acceso están representados por una dirección IP o por la sintaxis de VLSM (Máscara de Subred de Longitud Variable) para una subred. Formato de notación para una dirección IP: 10.33.2.200 La sintaxis de VLSM (Máscara de Subred de Longitud Variable) para una subred es: 10.33.2.200/24 Nota: La técnica VLSM es usada para especificar prefijos de red de longitud variable. Una dirección se escribe con un sufijo que indica el número de bits de longitud de prefijo, como por ejemplo 10.33.2.200/24 que indica que la máscara de red tiene 24 bits (es decir, los primeros 24 bits de la máscara son 1 y el resto 0).	



Prototipo:

Lista de Control de Acceso

Lista de Control de Acceso ac

Configurar ACL:

10.8.122.111

10.33.2.111

Adicionar Eliminar

Historia de Usuario

Número: U-SPC-02

Nombre de Historia de Usuario:

Gestionar zona de dominio para el servidor DNS.

Modificación de Historia de Usuario Número: Ninguna

Usuario: Javier Menéndez Rizo

Iteración Asignada: 1

Prioridad en Negocio: Muy alta

Puntos estimados: 1 semanas

Riesgo en Desarrollo: Alta

Puntos Reales: 1 semanas

Descripción:

Esta sección garantiza la gestión de la zona de dominio para el servidor DNS. Crear la zona de dominio para el servidor DNS. Modificar la zona de dominio para el servidor DNS. Modificar la dirección IP del servidor DNS que se relaciona con la zona de dominio. Modificar la máscara de red del servidor DNS que se relaciona con la zona de dominio.



Observaciones:

Los elementos que se corresponden con la dirección IP y la máscara de red del servidor están representados por una dirección IP o por la sintaxis de VLSM para una subred.

Formato de notación para una dirección IP: 10.33.2.200

La sintaxis de VLSM para una subred es: 255.255.255.0

Nota: La técnica VLSM es usada para especificar prefijos de red de longitud variable. Una dirección se escribe con un sufijo que indica el número de bits de longitud de prefijo, como por ejemplo 10.33.2.200/24 que indica que la máscara de red tiene 24 bits (es decir, los primeros 24 bits de la máscara son 1 y el resto 0).

Prototipo:

Zonas

Nombre de la Zona:

IP del servidor: . . .

Máscara de subred . . .

Historia de Usuario

Número: U-SPC-03	Nombre de Historia de Usuario: Gestionar logs y autenticación para el servidor DNS.
-------------------------	--

Modificación de Historia de Usuario Número: Ninguna

Usuario: Javier Menéndez Rizo

Iteración Asignada: 1

Prioridad en Negocio: Muy alta

Puntos estimados: 1 semanas

Riesgo en Desarrollo: Alta

Puntos Reales: 1 semanas

Descripción:

Esta sección garantiza la gestión de logs y autenticación para el servidor DNS. Modificar el



directorio de logs para el servidor DNS. Habilitar o deshabilitar el nx_domain para el servidor DNS.

Observaciones:

Para el caso específico que se corresponde con el directorio de logs se debe escribir la jerarquía de directorios.

Prototipo:

Logs y Autenticación

Directorio:

Auth_nxdomain yes no

Historia de Usuario

Número: U-SPC-04

Nombre de Historia de Usuario:

Gestionar la jerarquía de reenvío para el servidor DNS.

Modificación de Historia de Usuario Número: Ninguna

Usuario: Javier Menéndez Rizo

Iteración Asignada: 1

Prioridad en Negocio: Muy alta

Puntos estimados: 1 semanas

Riesgo en Desarrollo: Alta

Puntos Reales: 1 semanas

Descripción:

Esta sección garantiza la gestión de la jerarquía de reenvío para el servidor DNS. Adicionar dirección IP de los servidores correspondientes a la zona de reenvío. Eliminar dirección IP de los servidores correspondientes a la zona de reenvío. Listar los servidores que se relacionan con la zona de reenvío.

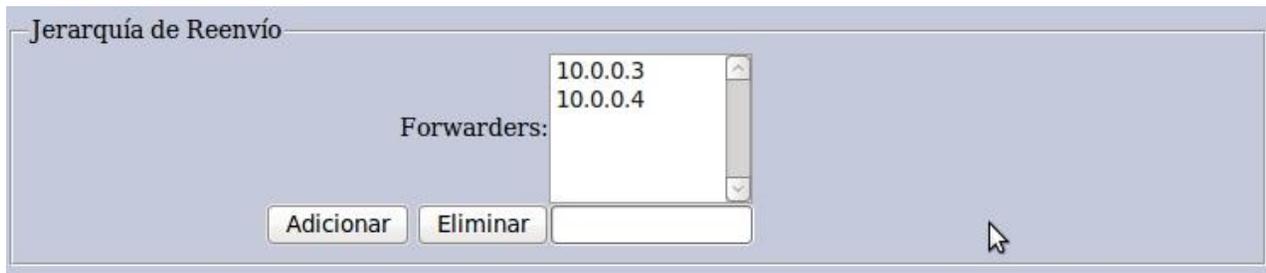


Observaciones:

Los elementos que se corresponden con la lista de servidores de la zona de reenvío para el servidor DNS están representados por una dirección IP.

Formato de notación para una dirección IP: 10.33.2.200

Prototipo:



Historia de Usuario

Número: U-SPC-05

Nombre de Historia de Usuario:

Gestionar nombres múltiples y alias correspondientes al servidor DNS.

Modificación de Historia de Usuario Número: Ninguna

Usuario: Javier Menéndez Rizo

Iteración Asignada: 1

Prioridad en Negocio: Muy alta

Puntos estimados: 1 semanas

Riesgo en Desarrollo: Alta

Puntos Reales: 1 semanas

Descripción:

Esta sección garantiza la gestión de los nombres múltiples y alias correspondientes al servidor DNS. Habilitar o deshabilitar los nombres múltiples correspondientes al servidor DNS. Adicionar alias correspondientes a opción de nombres múltiples del servidor DNS.

Eliminar alias correspondientes a opción de nombres múltiples del servidor DNS.



Observaciones:

Los alias son los nombres a los que puede ser referenciado el servidor de nombres de dominio.

Prototipo:

Historia de Usuario

Número: U-SPC-06

Nombre de Historia de Usuario:

Gestionar la resolución de direcciones IP mediante nombres de dominio del servidor DNS.

Modificación de Historia de Usuario Número: Ninguna

Usuario: Javier Menéndez Rizo

Iteración Asignada: 1

Prioridad en Negocio: Muy alta

Puntos estimados: 1 semanas

Riesgo en Desarrollo: Alta

Puntos Reales: 1 semanas

Descripción:

Esta sección garantiza la gestión de la resolución de direcciones IP mediante nombres de dominio del servidor DNS. Adicionar nombre de dominio a una zona del servidor DNS. Eliminar nombre de dominio a una zona del servidor DNS. Adicionar dirección IP a resolver por el servidor



DNS. Eliminar dirección IP a resolver por el servidor DNS.

Observaciones:

Los elementos que se corresponden con la lista de nombres y dirección a resolver para el servidor DNS están representados por una dirección IP.

Formato de notación para una dirección IP: 10.33.2.200

Prototipo:

Historia de Usuario

Número: U-SPC-07

Nombre de Historia de Usuario:

Gestionar opciones de configuración correspondientes al servidor NFS.

Modificación de Historia de Usuario Número: Ninguna

Usuario: Javier Menéndez Rizo

Iteración Asignada: 1

Prioridad en Negocio: Muy alta

Puntos estimados: 1 semanas

Riesgo en Desarrollo: Alta

Puntos Reales: 1 semanas



Descripción:

Esta sección garantiza la gestión de las opciones de configuración correspondientes al servidor NFS. Adicionar directorio a compartir por el servidor NFS. Eliminar directorio a compartir por el servidor NFS. Adicionar elementos correspondientes al directorio a compartir por el servidor NFS [dirección IP o subred]. Eliminar elementos correspondientes al directorio a compartir por el servidor NFS [dirección IP o subred]. Cambiar los permisos de acceso al directorio que se va a compartir.

Observaciones:

Los elementos que se corresponden con el directorio que será compartido por el servidor NFS están representados por una dirección IP o por el formato de una notación postfija para una subred.

Formato de notación para una dirección IP: 10.33.2.200

Formato de notación postfija para una subred: 10.33.2.200/24

Nota: La técnica VLSM es usada para especificar prefijos de red de longitud variable. Una dirección se escribe con un sufijo que indica el número de bits de longitud de prefijo, como por ejemplo 10.33.2.200/24 que indica que la máscara de red tiene 24 bits (es decir, los primeros 24 bits de la máscara son 1 y el resto 0).

Prototipo:

Configurar NFS

Directorio a Compartir:

IP ó subred:

Sólo lectura(L):

Lectura y Escritura(LE):

Adicionar Eliminar

Listado:

/home 10.33.2.0/24 (LE)



Historia de Usuario	
Número: U-SPC-08	Nombre de Historia de Usuario: Gestionar cuotas de disco a través de grupos y usuarios.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Javier Menéndez Rizo	Iteración Asignada: 1
Prioridad en Negocio: Muy alta	Puntos estimados: 1 semanas
Riesgo en Desarrollo: Alta	Puntos Reales: 1 semanas
Descripción: Esta sección garantiza la gestión de cuotas de disco a través de grupos y usuarios. Adicionar grupo de usuarios. Eliminar grupo de usuarios. Adicionar usuarios a los grupos. Eliminar usuarios de los grupos. Listar usuarios del sistema. Listar grupos con sus cuotas de disco asignadas. Listar grupos y sus usuarios.	
Observaciones: Los alias son los nombres a los que puede ser referenciado el servidor de nombres de dominio.	
Prototipo:	



Gestionar Grupos y Usuarios:

Nombre del Grupo:

Cuota a asignar:

Grupos y cuotas asignadas:

Usuarios:

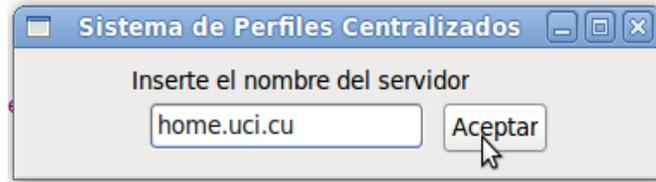
>>

Grupos y Usuarios:

Historia de Usuario	
Número: U-SPC-09	Nombre de Historia de Usuario: Configuración del cliente
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Javier Menéndez Rizo	Iteración Asignada: 1
Prioridad en Negocio: Muy alta	Puntos estimados: 1 semanas
Riesgo en Desarrollo: Alta	Puntos Reales: 1 semanas
Descripción: Esta sección garantiza la gestión de los archivos de configuración para el cliente después de la instalación de las aplicaciones que se utilizan en el cliente.	



Prototipo:



Historia de Usuario

Número: U-SPC-10	Nombre de Historia de Usuario: Configuración del servidor
-------------------------	---

Modificación de Historia de Usuario Número: Ninguna

Usuario: Javier Menéndez Rizo

Iteración Asignada: 1

Prioridad en Negocio: Muy alta

Puntos estimados: 1 semanas

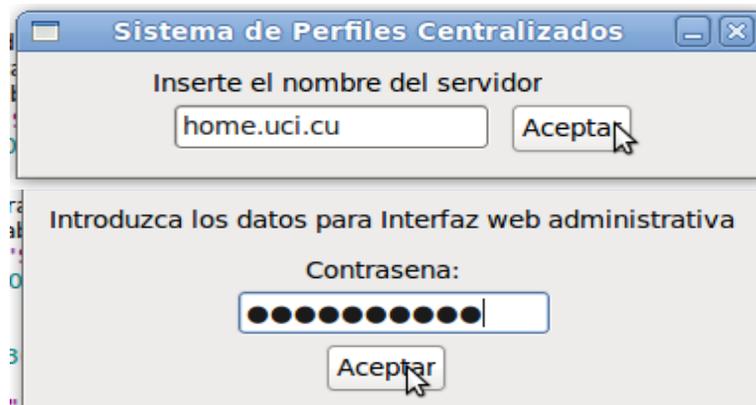
Riesgo en Desarrollo: Alta

Puntos Reales: 1 semanas

Descripción:

Esta sección garantiza la gestión de los archivos de configuración para el servidor después de la instalación de las aplicaciones que son utilizadas en el mismo.

Prototipo:





3.4 Tareas de ingeniería.

A continuación se dan a conocer las distintas tareas de ingeniería que están presentes en el sistema; así como su relación con las historias de usuario y prototipos de interfaz usuario asociadas a cada tarea.

Tareas de ingeniería para la historia de usuario U-SPC-01:

Tarea de Ingeniería	
Número de tarea: 01	Número de Historia de Usuario: 01
Nombre de Tarea: Crear lista de control de acceso	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 01/02/10	Fecha Fin:03/02/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para crear la lista de control de acceso. Se valida que el nombre de la lista de control de acceso contenga caracteres alfanuméricos y que no esté vacío. Se implementa la funcionalidad que se encarga de crear la lista de control de acceso.	

Tarea de Ingeniería	
Número de tarea: 02	Número de Historia de Usuario: 01
Nombre de Tarea: Modificar lista de control de acceso	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 04/02/10	Fecha Fin: 06/02/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para modificar la lista de control de acceso. Se valida que el nombre de la lista de control de acceso contenga caracteres alfanuméricos y que no esté vacío. Se implementa la funcionalidad que se encarga de	



modificar la lista de control de acceso.

Tarea de Ingeniería	
Número de tarea: 03	Número de Historia de Usuario: 01
Nombre de Tarea: Adicionar dirección IP o subred	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 07/02/10	Fecha Fin: 09/02/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para adicionar direcciones IP o la sintáxis VLSM. Se definen los métodos para la validación de los campos, estos verifican que se escriban las direcciones IP correctamente y que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de adicionar estos elementos a la lista de control de acceso.	

Tarea de Ingeniería	
Número de tarea: 04	Número de Historia de Usuario: 01
Nombre de Tarea: Eliminar dirección IP o subred	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 10/02/10	Fecha Fin: 12/02/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para adicionar direcciones IP o la sintáxis VLSM. Se definen los métodos para la validación de los campos, estos verifican que se escriban las direcciones IP correctamente y que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de eliminar estos elementos a la lista de control de acceso.	



Tarea de Ingeniería	
Número de tarea: 05	Número de Historia de Usuario: 01
Nombre de Tarea: Listar dirección IP o subred	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 13/02/10	Fecha Fin: 15/02/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Se implementa la funcionalidad que permite listar direcciones IP o la sintáxis VLSM para una subred que pertenecen a la lista de control de acceso.	

Tareas de ingeniería para la historia de usuario U-SPC-02:

Tarea de Ingeniería	
Número de tarea: 01	Número de Historia de Usuario: 02
Nombre de Tarea: Crear zona de dominio	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 17/02/10	Fecha Fin: 19/02/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para crear la zona de dominio. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de crear la zona de dominio.	

Tarea de Ingeniería	
Número de tarea: 02	Número de Historia de Usuario: 02
Nombre de Tarea: Modificar zona de dominio	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 20/02/10	Fecha Fin: 22/02/10



CAPÍTULO 3: DESARROLLO ÁGIL DEL SISTEMA DE PERFILES CENTRALIZADOS

Programador Responsable: Javier Menéndez Rizo

Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para modificar la zona de dominio. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de modificar la zona de dominio.

Tarea de Ingeniería

Número de tarea: 03

Número de Historia de Usuario: 02

Nombre de Tarea: Modificar dirección IP del servidor

Tipo de Tarea: Desarrollo

Puntos Estimados: 1 semana

Fecha Inicio: 23/02/10

Fecha Fin: 25/02/10

Programador Responsable: Javier Menéndez Rizo

Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para modificar la dirección IP del servidor. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de modificar la dirección IP del servidor.

Tarea de Ingeniería

Número de tarea: 04

Número de Historia de Usuario: 02

Nombre de Tarea: Modificar la máscara de red

Tipo de Tarea: Desarrollo

Puntos Estimados: 1 semana

Fecha Inicio: 26/02/10

Fecha Fin: 28/02/10

Programador Responsable: Javier Menéndez Rizo

Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para modificar la máscara de red. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de modificar la máscara de red.



Tareas de ingeniería para la historia de usuario U-SPC-03:

Tarea de Ingeniería	
Número de tarea: 01	Número de Historia de Usuario: 03
Nombre de Tarea: Modificar el directorio de logs	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 28/02/10	Fecha Fin: 01/03/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para modificar el directorio de logs del servidor DNS. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de modificar el directorio de logs del servidor DNS.	

Tarea de Ingeniería	
Número de tarea: 02	Número de Historia de Usuario: 03
Nombre de Tarea: Habilitar o deshabilitar auth_nxdomain	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 02/03/10	Fecha Fin: 04/03/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para habilitar o deshabilitar la opción auth_nxdomain del servidor DNS. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de habilitar o deshabilitar la opción auth_nxdomain del servidor DNS.	



Tareas de ingeniería para la historia de usuario U-SPC-04:

Tarea de Ingeniería	
Número de tarea: 01	Número de Historia de Usuario: 04
Nombre de Tarea: Adicionar dirección IP de los servidores	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 05/03/10	Fecha Fin: 07/03/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para adicionar direcciones IP. Se definen los métodos para la validación de los campos, estos verifican que se escriban las direcciones IP correctamente y que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de adicionar estos elementos a la jerarquía de reenvío.	

Tarea de Ingeniería	
Número de tarea: 02	Número de Historia de Usuario: 04
Nombre de Tarea: Eliminar dirección IP de los servidores	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 08/03/10	Fecha Fin: 11/03/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para adicionar direcciones IP. Se definen los métodos para la validación de los campos, estos verifican que se escriban las direcciones IP correctamente y que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de eliminar estos elementos a la jerarquía de reenvío.	

Tarea de Ingeniería	
Número de tarea: 03	Número de Historia de Usuario: 04



Nombre de Tarea: Listar dirección IP de los servidores	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 12/03/10	Fecha Fin: 13/03/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Se implementa la funcionalidad que permite listar direcciones IP que pertenecen a la jerarquía de reenvío.	

Tareas de ingeniería para la historia de usuario U-SPC-05:

Tarea de Ingeniería	
Número de tarea: 01	Número de Historia de Usuario: 05
Nombre de Tarea: Habilitar o deshabilitar nombres múltiples	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 14/03/10	Fecha Fin: 17/03/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para habilitar o deshabilitar la opción nombres múltiples del servidor DNS. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de habilitar o deshabilitar la opción nombres múltiples del servidor DNS.	

Tarea de Ingeniería	
Número de tarea: 02	Número de Historia de Usuario: 05
Nombre de Tarea: Adicionar alias	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 18/03/10	Fecha Fin: 20/03/10



CAPÍTULO 3: DESARROLLO ÁGIL DEL SISTEMA DE PERFILES CENTRALIZADOS

Programador Responsable: Javier Menéndez Rizo

Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para adicionar los alias. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de adicionar estos elementos al listado de nombres múltiples.

Tarea de Ingeniería	
Número de tarea: 03	Número de tarea: 03
Nombre de Tarea: Eliminar alias	
Tipo de Tarea: Desarrollo	Tipo de Tarea: Desarrollo
Fecha Inicio: 21/03/10	Fecha Inicio: 21/03/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para eliminar los alias. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de eliminar estos elementos al listado de nombres múltiples.	

Tareas de ingeniería para la historia de usuario U-SPC-06:

Tarea de Ingeniería	
Número de tarea: 01	Número de Historia de Usuario: 06
Nombre de Tarea: Adicionar nombre de dominio y dirección IP a resolver	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 24/03/10	Fecha Fin: 27/03/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para adicionar el nombre con su dirección IP correspondiente a la base de datos del servidor DNS. Se definen los	



métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de adicionar el nombre con su dirección IP correspondiente a la base de datos del servidor DNS.

Tarea de Ingeniería	
Número de tarea: 02	Número de Historia de Usuario: 06
Nombre de Tarea: Eliminar nombre de dominio y dirección IP a resolver	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 28/03/10	Fecha Fin: 30/03/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para eliminar el nombre con su dirección IP correspondiente a la base de datos del servidor DNS. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de eliminar el nombre con su dirección IP correspondiente a la base de datos del servidor DNS.	

Tareas de ingeniería para la historia de usuario U-SPC-07:

Tarea de Ingeniería	
Número de tarea: 01	Número de Historia de Usuario: 07
Nombre de Tarea: Adicionar directorio	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 06/04/10	Fecha Fin: 08/04/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para adicionar la estructura del directorio con la dirección IP o la sintaxis de VLSM correspondiente. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de adicionar la estructura del directorio con la dirección	



IP o la sintaxis de VLISM correspondiente y con permisos de solo lectura o lectura/escritura.

Tarea de Ingeniería	
Número de tarea: 02	Número de Historia de Usuario: 07
Nombre de Tarea: Eliminar directorio	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 09/04/10	Fecha Fin: 11/04/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para eliminar la estructura del directorio con la dirección IP o la sintaxis de VLISM correspondiente. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de eliminar la estructura del directorio con la dirección IP o la sintaxis de VLISM correspondiente.	

Tareas de ingeniería para la historia de usuario U-SPC-08:

Tarea de Ingeniería	
Número de tarea: 01	Número de Historia de Usuario: 08
Nombre de Tarea: Adicionar grupo de usuarios	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 18/04/10	Fecha Fin: 20/04/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para adicionar grupos asignándole una cuota de espacio en disco. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de adicionar grupos asignándole una cuota de espacio en disco.	



Tarea de Ingeniería	
Número de tarea: 02	Número de Historia de Usuario: 08
Nombre de Tarea: Eliminar grupo de usuarios	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 21/04/10	Fecha Fin: 23/04/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para eliminar grupos de usuarios asignándole una cuota de espacio en disco. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de eliminar grupos de usuarios.	

Tarea de Ingeniería	
Número de tarea: 03	Número de Historia de Usuario: 08
Nombre de Tarea: Adicionar usuario a un grupo	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 24/04/10	Fecha Fin: 26/04/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para adicionar usuarios a los grupos de usuarios. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de adicionar usuarios a los grupos de usuarios.	

Tarea de Ingeniería	
Número de tarea: 04	Número de Historia de Usuario: 08



CAPÍTULO 3: DESARROLLO ÁGIL DEL SISTEMA DE PERFILES CENTRALIZADOS

Nombre de Tarea: Eliminar usuario de un grupo	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 26/04/10	Fecha Fin: 28/04/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para eliminar usuarios de los grupos de usuarios. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de eliminar usuarios de los grupos de usuarios.	

Tarea de Ingeniería	
Número de tarea: 05	Número de Historia de Usuario: 08
Nombre de Tarea: Listar usuarios del sistema.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 01/05/10	Fecha Fin: 03/05/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para eliminar usuarios de los grupos de usuarios. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de listar los usuarios que pertenecen al sistema.	

Tareas de ingeniería para la historia de usuario U-SPC-09:

Tarea de Ingeniería	
Número de tarea: 01	Número de Historia de Usuario: 09
Nombre de Tarea: Gestionar la configuración del cliente	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 03/05/10	Fecha Fin: 05/05/10



Programador Responsable: Javier Menéndez Rizo

Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para configurar el cliente. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de escribir en los archivos correspondientes la información necesaria para que el cliente funcione correctamente.

Tareas de ingeniería para la historia de usuario U-SPC-10:

Tarea de Ingeniería	
Número de tarea: 01	Número de Historia de Usuario: 10
Nombre de Tarea: Gestionar la configuración del servidor	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 03/05/10	Fecha Fin: 05/05/10
Programador Responsable: Javier Menéndez Rizo	
Descripción: Personalizar el formulario añadiendo los elementos que se van a utilizar para configurar el servidor. Se definen los métodos para la validación de los campos, estos verifican que los campos no queden vacíos. Se implementa la funcionalidad que se encarga de escribir en los archivos correspondientes la información necesaria para que el servidor funcione correctamente.	

3.5 Plan de Release.

En este epígrafe se define el plan de release e iteraciones para realizar las entregas intermedias y la entrega final. Tiene como entrada la relación de Historias de Usuario definidas previamente. Para colocar una historia en cada iteración se tiene en cuenta la prioridad que se definió para cada Historia de Usuario. Como resultado de darle prioridad a cada Historia de Usuario se llegó a la siguiente planificación:

Release	Historias de Usuario	Tiempo estimado (semanas)
1	1,2,3	8
2	4,5,6	8



CAPÍTULO 3: DESARROLLO ÁGIL DEL SISTEMA DE PERFILES CENTRALIZADOS

3

7,8,9,10

8

En este capítulo se realizó la planificación del proyecto, se establecieron las historias de usuario y las tareas asociadas a las mismas para la primera planificación. También se mostró el desarrollo de las iteraciones teniendo en cuenta la prioridad especificada en la lista de reserva del producto.



CAPÍTULO 4. Validación de la Solución Propuesta.

Para lograr un producto con calidad es necesario ir realizando pruebas al producto desde los inicios de su confección y darle seguimiento a los errores que puedan surgir a lo largo de todo el ciclo de desarrollo de software. En este capítulo se plasman los casos de pruebas de aceptación a las que fue sometida la aplicación en cada una de las iteraciones. Se debe destacar que el cumplimiento de estos casos de pruebas fue el hito para avanzar hacia la próxima iteración.

4.1 Casos de Pruebas de Aceptación.

Las pruebas funcionales se basan en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. Estas se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático.

Este tipo de prueba permite detectar funciones incorrectas o ausentes, como son los errores de interfaz, rendimiento, inicialización y terminación, en estructuras de datos o en acceso a la información almacenada.

A continuación se muestran los casos de prueba para todas las historias de usuario y se dan a conocer las pruebas que se realizaron a cada una de las historias de usuario con las que cuenta el sistema:

Casos de prueba para la U-SPC-01:

Caso de Prueba de Aceptación	
Código Caso de Prueba: SPC-01-01	Nombre de la Historia de Usuario: Gestionar lista de control de acceso del servidor DNS
Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra	
Descripción de la prueba: Se verifica que la lista de control de acceso para el servidor DNS se crea.	
Condiciones de ejecución: El administrador del sistema debe estar autenticado.	
Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente escribe el nombre de la lista de control de acceso en el campo denominado Nombre de la Lista de Control de Acceso, después pulsa el botón guardar todo.	
Resultado Esperado: El sistema actualiza la página mostrando el nombre que se escribió para la	



lista de control de acceso guardando, el nombre en los ficheros de configuración para el servidor DNS y el sistema respectivamente.

Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación

Código Caso de Prueba: SPC-01-02

Nombre de la Historia de Usuario:

Gestionar lista de control de acceso del servidor DNS

Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra

Descripción de la prueba: Se verifica que la lista de control de acceso para el servidor DNS se modifica.

Condiciones de ejecución: El administrador del sistema debe estar autenticado. El campo del nombre de la lista de control de acceso no debe estar vacío.

Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente escribe el nombre de la lista de control de acceso en el campo denominado Nombre de la Lista de Control de Acceso, después pulsa el botón guardar todo.

Resultado Esperado: El sistema actualiza la página mostrando el nombre que se escribió para la lista de control de acceso, guardando el nombre en los ficheros de configuración para el servidor DNS y el sistema respectivamente

Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación

Código Caso de Prueba: SPC-01-03

Nombre de la Historia de Usuario:

Gestionar lista de control de acceso del servidor DNS

Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra

Descripción de la prueba: Se verifica que las direcciones IP o subred de la lista de control de acceso para el servidor DNS se adicionan.

Condiciones de ejecución: El administrador del sistema debe estar autenticado.



Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente escribe la dirección IP o la sintaxis de VLSM para una subred de la lista de control de acceso en el campo denominado Subred/Dirección IP, después pulsa el botón adicionar, y posteriormente en guardar todo.

Resultado Esperado: El sistema actualiza la página mostrando la dirección IP o la sintaxis de VLSM para una subred que se añadió para la lista de control de acceso, guardando los cambios en los ficheros de configuración para el servidor DNS y el sistema respectivamente.

Evaluación de la Prueba: Satisfactoria

Casos de prueba para la U-SPC-02:

Caso de Prueba de Aceptación

Código Caso de Prueba: SPC-02-01

Nombre de la Historia de Usuario:

Gestionar zona de dominio para el servidor DNS.

Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra

Descripción de la prueba: Se verifica que el nombre de la zona de dominio del servidor DNS se crea.

Condiciones de ejecución: El administrador del sistema debe estar autenticado.

Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente escribe el nombre de la zona de dominio en el campo denominado Nombre de la Zona, después pulsa el botón guardar todo.

Resultado Esperado: El sistema actualiza la página mostrando el nombre que se escribió para la zona de dominio, guardando el nombre en los ficheros de configuración para el servidor DNS y el sistema respectivamente.

Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación

Código Caso de Prueba: SPC-02-02

Nombre de la Historia de Usuario:

Gestionar zona de dominio para el servidor DNS.



Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra
Descripción de la prueba: Se verifica que el nombre de la zona de dominio del servidor DNS se modifica. El campo denominado Nombre de la Zona para el servidor DNS no debe estar vacío.
Condiciones de ejecución: El administrador del sistema debe estar autenticado.
Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente escribe el nombre de la zona de dominio en el campo denominado Nombre de la Zona, después pulsa el botón guardar todo.
Resultado Esperado: El sistema actualiza la página mostrando el nombre que se escribió para la zona de dominio, guardando el nombre en los ficheros de configuración para el servidor DNS y el sistema respectivamente.
Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación	
Código Caso de Prueba: SPC-02-03	Nombre de la Historia de Usuario: Gestionar zona de dominio para el servidor DNS.
Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra	
Descripción de la prueba: Se verifica que la dirección IP del servidor DNS para la zona de dominio se almacena.	
Condiciones de ejecución: El administrador del sistema debe estar autenticado.	
Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente escribe la dirección IP en el campo denominado IP del servidor, después pulsa el botón guardar todo.	
Resultado Esperado: El sistema actualiza la página mostrando la dirección IP que se escribió para el servidor, guardando el nombre en los ficheros de configuración para el servidor DNS y el sistema respectivamente.	
Evaluación de la Prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba: SPC-02-04	Nombre de la Historia de Usuario:



CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

	Gestionar zona de dominio para el servidor DNS.
Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra	
Descripción de la prueba: Se verifica que la dirección IP del servidor DNS para la zona de dominio se modifica.	
Condiciones de ejecución: El administrador del sistema debe estar autenticado. El campo IP del servidor no debe estar vacío.	
Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente escribe la dirección IP en el campo denominado IP del servidor, después pulsa el botón guardar todo.	
Resultado Esperado: El sistema actualiza la página mostrando la dirección IP que se escribió para el servidor, guardando el nombre en los ficheros de configuración para el servidor DNS y el sistema respectivamente.	
Evaluación de la Prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba: SPC-02-05	Nombre de la Historia de Usuario: Gestionar zona de dominio para el servidor DNS.
Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra	
Descripción de la prueba: Se verifica que la máscara de subred del servidor DNS para la zona de dominio se almacena.	
Condiciones de ejecución: El administrador del sistema debe estar autenticado.	
Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente escribe la máscara de subred en el campo denominado Máscara de subred, después pulsa el botón guardar todo.	
Resultado Esperado: El sistema actualiza la página mostrando la máscara de subred que se escribió para el servidor, guardando el nombre en los ficheros de configuración para el servidor DNS y el sistema respectivamente.	
Evaluación de la Prueba: Satisfactoria	



Caso de Prueba de Aceptación	
Código Caso de Prueba: SPC-02-06	Nombre de la Historia de Usuario: Gestionar zona de dominio para el servidor DNS.
Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra	
Descripción de la prueba: Se verifica que la máscara de subred del servidor DNS para la zona de dominio se modifica.	
Condiciones de ejecución: El administrador del sistema debe estar autenticado. El campo denominado Máscara de subred no debe estar vacío.	
Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente escribe la máscara de subred en el campo denominado Máscara de subred, después pulsa el botón guardar todo.	
Resultado Esperado: El sistema actualiza la página mostrando la máscara de subred que se escribió para el servidor, guardando el nombre en los ficheros de configuración para el servidor DNS y el sistema respectivamente.	
Evaluación de la Prueba: Satisfactoria	

Casos de prueba para la U-SPC-03:

Caso de Prueba de Aceptación	
Código Caso de Prueba: SPC-03-01	Nombre de la Historia de Usuario: Gestionar lista de control de acceso del servidor DNS
Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra	
Descripción de la prueba: Se verifica que el directorio de logs para el servidor DNS se modifica.	
Condiciones de ejecución: El administrador del sistema debe estar autenticado. El campo denominado Directorio no debe estar vacío.	
Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente escribe la sintaxis de la ruta donde quiere que se almacenen los logs del servidor, después pulsa el botón guardar todo.	



Resultado Esperado: El sistema actualiza la página mostrando la jerarquía de directorio que se escribió para almacenar los logs, guardando los cambios en los ficheros de configuración para el servidor DNS y el sistema respectivamente.

Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación

Código Caso de Prueba: SPC-02-03

Nombre de la Historia de Usuario:

Gestionar zona de dominio para el servidor DNS.

Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra

Descripción de la prueba: Se verifica que la opción auth_nxdomain se configura para el servidor DNS.

Condiciones de ejecución: El administrador del sistema debe estar autenticado.

Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente presiona una de las opciones que se corresponde con el campo denominado auth_nxdomain.

Resultado Esperado: El sistema actualiza la página y guarda los cambios en los ficheros de configuración para el servidor DNS y el sistema respectivamente.

Evaluación de la Prueba: Satisfactoria

Casos de prueba para la U-SPC-04:

Caso de Prueba de Aceptación

Código Caso de Prueba: SPC-04-01

Nombre de Historia de Usuario:

Gestionar la jerarquía de reenvío para el servidor DNS.

Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra

Descripción de la prueba: Se verifica que las direcciones IP de los servidores que pertenecen a la jerarquía de reenvío del servidor DNS se adicionan.

Condiciones de ejecución: El administrador del sistema debe estar autenticado.

Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente escribe la dirección IP que se va a adicionar a la jerarquía de reenvío en el campo denominado



Dirección IP del servidor después pulsa en adicionar y posteriormente pulsa en guardar todo.

Resultado Esperado: El sistema actualiza la página mostrando la dirección IP que se escribió para la jerarquía de reenvío, guardando los cambios realizados en los ficheros de configuración para el servidor DNS y el sistema respectivamente.

Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación

Código Caso de Prueba: SPC-04-02

Nombre de Historia de Usuario:

Gestionar la jerarquía de reenvío para el servidor DNS.

Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra

Descripción de la prueba: Se verifica que las direcciones IP de los servidores que pertenecen a la jerarquía de reenvío del servidor DNS se eliminan.

Condiciones de ejecución: El administrador del sistema debe estar autenticado. El listado de direcciones IP para la jerarquía de reenvío no debe estar vacío.

Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente selecciona la dirección IP del servidor DNS que se va a eliminar de la jerarquía de reenvío en el campo denominado Dirección IP del servidor, después pulsa eliminar y posteriormente el botón guardar todo.

Resultado Esperado: El sistema actualiza la página eliminando la dirección IP que se seleccionó para la jerarquía de reenvío, guardando los cambios realizados en los ficheros de configuración para el servidor DNS y el sistema respectivamente.

Evaluación de la Prueba: Satisfactoria

Casos de prueba para la U-SPC-05:

Caso de Prueba de Aceptación

Código Caso de Prueba: SPC-05-01

Nombre de Historia de Usuario:

Gestionar nombres múltiples y alias correspondientes al



CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

	servidor DNS.
Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra	
Descripción de la prueba: Se verifica que la opción múltiples nombres se configura para el servidor DNS.	
Condiciones de ejecución: El administrador del sistema debe estar autenticado.	
Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente presiona una de las opciones que se corresponde con el campo denominado Nombres Múltiples.	
Resultado Esperado: El sistema actualiza la página y guarda los cambios en los ficheros de configuración para el servidor DNS y el sistema respectivamente.	
Evaluación de la Prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba: SPC-05-02	Nombre de Historia de Usuario: Gestionar nombres múltiples y alias correspondientes al servidor DNS.
Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra	
Descripción de la prueba: Se verifica que los alias para el servidor DNS se adicionan.	
Condiciones de ejecución: El administrador del sistema debe estar autenticado.	
Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente escribe el alias en el campo denominado Alias, después pulsa el botón adicionar.	
Resultado Esperado: El sistema actualiza la página mostrando el alias en el listado, guardando los cambios en los ficheros de configuración para el servidor DNS y el sistema respectivamente.	
Evaluación de la Prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba: SPC-05-03	Nombre de Historia de Usuario: Gestionar nombres múltiples y alias correspondientes al



	servidor DNS.
Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra	
Descripción de la prueba: Se verifica que los alias el servidor DNS se eliminan.	
Condiciones de ejecución: El administrador del sistema debe estar autenticado. El listado de los alias no debe estar vacío.	
Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente selecciona el alias de la listado de alias, después pulsa el botón adicionar.	
Resultado Esperado: El sistema actualiza la página eliminando el alias del listado, guardando los cambios en los ficheros de configuración para el servidor DNS y el sistema respectivamente.	
Evaluación de la Prueba: Satisfactoria	

Casos de prueba para la U-SPC-06:

Caso de Prueba de Aceptación	
Código Caso de Prueba: SPC-06-01	Nombre de Historia de Usuario: Gestionar la resolución de direcciones IP mediante nombres de dominio del servidor DNS.
Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra	
Descripción de la prueba: Se verifica que el nombre y la dirección IP a resolver por el servidor DNS se adicionan.	
Condiciones de ejecución: El administrador del sistema debe estar autenticado.	
Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente escribe el nombre que va a pertenecer a la zona de dominio en el campo denominado Nombres y la dirección IP a resolver en el campo denominado IP a resolver después pulsa en adicionar, y posteriormente pulsa el botón guardar todo.	
Resultado Esperado: El sistema actualiza la página mostrando el nombre con su correspondiente dirección IP, guardando los cambios en realizados en los ficheros de configuración y el sistema respectivamente.	



Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación

Código Caso de Prueba: SPC-06-02

Nombre de Historia de Usuario:

Gestionar la resolución de direcciones IP mediante nombres de dominio del servidor DNS.

Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra

Descripción de la prueba: Se verifica que el nombre y la dirección IP a resolver por el servidor DNS se eliminan.

Condiciones de ejecución: El administrador del sistema debe estar autenticado. El campo del listado de los nombres con sus respectivas direcciones IP no debe estar vacío.

Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo DNS, posteriormente selecciona el nombre y la dirección IP a resolver del listado y después pulsa en adicionar, y posteriormente pulsa el botón guardar todo.

Resultado Esperado: El sistema actualiza la página eliminando del listado el nombre con su correspondiente dirección IP, guardando los cambios en realizados en los ficheros de configuración y el sistema respectivamente.

Evaluación de la Prueba: Satisfactoria

Casos de prueba para la U-SPC-07:

Caso de Prueba de Aceptación

Código Caso de Prueba: SPC-07-01

Nombre de la Historia de Usuario:

Gestionar opciones de configuración correspondientes al servidor NFS.

Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra

Descripción de la prueba: Se verifica que el directorio a compartir y la dirección IP o subred a la cual se va a compartir este directorio se adicionan al listado y a las opciones de configuración del servidor NFS.



Condiciones de ejecución: El administrador del sistema debe estar autenticado.

Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo NFS, posteriormente escribe la estructura del directorio a compartir en el campo denominado Directorio a Compartir y la dirección IP o la sintaxis de VLSM para una subred de la lista de control de acceso en el campo denominado Subred/Dirección IP, después pulsa el botón adicionar, y posteriormente en guardar todo.

Resultado Esperado: : El sistema actualiza la página mostrando la estructura del directorio y la dirección IP o la sintaxis de VLSM para una subred que se añadió al listado, guardando los cambios en los ficheros de configuración para el servidor NFS y el sistema respectivamente.

Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación

Código Caso de Prueba: SPC-07-02

Nombre de la Historia de Usuario:

Gestionar opciones de configuración correspondientes al servidor NFS.

Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra

Descripción de la prueba: Se verifica que el directorio a compartir y la dirección IP o subred a la cual se va a compartir este directorio se eliminan del listado y de las opciones de configuración para el servidor NFS.

Condiciones de ejecución: El administrador del sistema debe estar autenticado. El listado de las estructuras de directorios con sus respectivas direcciones IP o sintaxis VLSM para una subred no está vacío.

Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo NFS, posteriormente selecciona la estructura de directorio con sus respectiva dirección IP o sintaxis VLSM para una subred, después pulsa el botón eliminar, y posteriormente en guardar todo.

Resultado Esperado: El sistema actualiza la página eliminando la estructura del directorio y la dirección IP o la sintaxis de VLSM para una subred que se añadió al listado, guardando los cambios en los ficheros de configuración para el servidor NFS y el sistema respectivamente.

Evaluación de la Prueba: Satisfactoria



Casos de prueba para la U-SPC-08:

Caso de Prueba de Aceptación	
Código Caso de Prueba: SPC-08-01	Nombre de Historia de Usuario: Gestionar cuotas de disco a través de grupos y usuarios.
Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra	
Descripción de la prueba: Se verifica que se crea un grupo de usuarios con su respectiva cuota de disco.	
Condiciones de ejecución: El administrador del sistema debe estar autenticado.	
Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo QUOTA, posteriormente escribe el nombre del grupo en el campo denominado Nombre del Grupo después escribe la cuota a asignar y pulsa en Crear Grupo, posteriormente en guardar todo.	
Resultado Esperado: El sistema actualiza la página mostrando el nombre del grupo con su cuota de disco asignada en el listado denominado Grupos y cuotas asignadas además de adicionarse el nombre en el listado de Grupos y Usuarios, guardando los cambios en los ficheros de configuración del sistema.	
Evaluación de la Prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba: SPC-08-02	Nombre de Historia de Usuario: Gestionar cuotas de disco a través de grupos y usuarios.
Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra	
Descripción de la prueba: Se verifica que un grupo de usuarios con su respectiva cuota de disco se elimina.	
Condiciones de ejecución: El administrador del sistema debe estar autenticado. La lista de Grupos y cuotas asignadas no debe estar vacía.	
Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo QUOTA,	



posteriormente selecciona un grupo en el listado de Grupos y cuotas asignadas y presiona eliminar grupo, posteriormente en guardar todo.

Resultado Esperado: El sistema actualiza la página eliminando el grupo del listado de Grupos y cuotas asignadas, también del listado del listado de Grupos y Usuarios, guardando los cambios en los ficheros de configuración del sistema.

Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación

Código Caso de Prueba: SPC-08-03

Nombre de Historia de Usuario:

Gestionar cuotas de disco a través de grupos y usuarios.

Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra

Descripción de la prueba: Se verifica que se listan los usuarios que pertenecen al Sistema de Perfiles Centralizados.

Condiciones de ejecución: El administrador del sistema debe estar autenticado.

Entrada / Pasos de ejecución: El usuario presiona en el botón denominado usuarios del sistema.

Resultado Esperado: El sistema actualiza la página y en el listado de los usuarios del sistema se adicionan los usuarios que pertenecen al Sistema de Perfiles Centralizados.

Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación

Código Caso de Prueba: SPC-08-04

Nombre de Historia de Usuario:

Gestionar cuotas de disco a través de grupos y usuarios.

Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra

Descripción de la prueba: Se verifica que un usuario se adiciona a un grupo de usuarios.

Condiciones de ejecución: El administrador del sistema debe estar autenticado. El campo



denominado Grupos y cuotas no debe estar vacío, al igual que el campo de Usuarios.

Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo QUOTA, posteriormente selecciona un grupo en el listado de Grupos y cuotas, después selecciona un usuario en el campo denominado Usuarios y se presiona el botón denominado “>>”, posteriormente se presiona el botón guardar todo.

Resultado Esperado: El sistema actualiza la página y adiciona el usuario al grupo seleccionado, mostrándose en el campo de Grupos y Usuarios, guardando los cambios en los ficheros de configuración del sistema, aplicando las cuotas de disco a estos usuarios.

Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación

Código Caso de Prueba: SPC-08-05

Nombre de Historia de Usuario:

Gestionar cuotas de disco a través de grupos y usuarios.

Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra

Descripción de la prueba: Se verifica que un usuario se elimina de un grupo de usuarios.

Condiciones de ejecución: El administrador del sistema debe estar autenticado. El campo denominado Grupos y Usuarios no debe estar vacío.

Entrada / Pasos de ejecución: El usuario pulsa sobre el vínculo del módulo QUOTA, posteriormente selecciona un usuario en el listado de Grupos y Usuarios, después presiona el botón eliminar, posteriormente se presiona el botón guardar todo.

Resultado Esperado: El sistema actualiza la página y elimina el usuario del grupo seleccionado, guardando los cambios en los ficheros de configuración del sistema.

Evaluación de la Prueba: Satisfactoria

Casos de prueba para la U-SPC-09:

Caso de Prueba de Aceptación



CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Código Caso de Prueba: SPC-09-01	Nombre de Historia de Usuario: Configuración del cliente
Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra	
Descripción de la prueba: Se verifica que el cliente se configura correctamente	
Condiciones de ejecución: El administrador del sistema debe estar autenticado como súper usuario.	
Entrada / Pasos de ejecución: El usuario escribe en el campo nombre del servidor y presiona aceptar.	
Resultado Esperado: El sistema de instalación cierra la ventana y posteriormente escribe la información necesaria en los archivos de configuración que permite que el cliente interactúe con el servidor.	
Evaluación de la Prueba: Satisfactoria	

Casos de prueba para la U-SPC-10:

Caso de Prueba de Aceptación	
Código Caso de Prueba: SPC-10-01	Nombre de Historia de Usuario: Configuración del servidor
Nombre de la persona que realiza la prueba: Abel Ernesto Cuadrado Sospedra	
Descripción de la prueba: Se verifica que el servidor se configura correctamente	
Condiciones de ejecución: El administrador del sistema debe estar autenticado como súper usuario.	
Entrada / Pasos de ejecución: El usuario escribe en el campo nombre del servidor y presiona aceptar, posteriormente escribe la contraseña administrativa para el SPCAdmin y presiona aceptar.	
Resultado Esperado: El sistema de instalación cierra las ventanas correspondientes al nombre del servidor y contraseña administrativa para el SPCAdmin respectivamente después de presionar aceptar, posteriormente escribe la información necesaria en los archivos de configuración que permite que el servidor funcione correctamente.	



Evaluación de la Prueba: Satisfactoria

4.2 Resultados Obtenidos.

A continuación se relacionan los resultados obtenidos hasta el momento por el equipo de desarrollo del **SPC**.

Se debe señalar que existen dos metapaquetes uno para ser instalado en la PC cliente y otro para el servidor, los cuales poseen un conjunto de aplicaciones que interactúan entre sí incluyendo el **SPCAdmin** en su versión 0.1 desarrollado específicamente para el usuario que va administrar las funcionalidades del **SPC**.

4.3 Acerca de las funcionalidades obtenidas.

Entre las principales funcionalidades que posee el SPC se pueden mencionar las siguientes:

- Permite centralizar la información de cada uno de los usuarios que pertenecen al SPC.
- Permite mediante una aplicación web integrada al sistema:
 - Gestionar cuotas de disco en el servidor para cada uno de sus usuarios.
 - Configurar un servidor DNS.
 - Configurar un servidor NFS.
- Permite almacenar el perfil de cada usuario perteneciente al SPC.

Se presentaron los casos de pruebas que guiaron la calidad del sistema, y determinaron en cada momento si se estaba o no en condiciones de continuar avanzando con el ciclo de desarrollo, se comprobó el funcionamiento correcto de la aplicación y se mostró un análisis de las funcionalidades obtenidas con el desarrollo de este producto.



Conclusiones Generales.

Con este trabajo se ha realizado un estudio de las diferentes soluciones sobre el desarrollo de perfiles centralizados en Cuba y el mundo, se ha analizado su funcionamiento y características principales. Se desarrolló y probó una aplicación web que permite administrar el Sistema de Perfiles Centralizados de una manera eficiente y sencilla.

También se desarrolló un instalador para el cliente y el servidor que permite gestionar de manera automatizada las configuraciones iniciales necesarias para el buen funcionamiento de la solución propuesta. Por todo lo anteriormente expuesto se concluye que los objetivos propuestos fueron cumplidos satisfactoriamente.



Recomendaciones.

Al concluir la presente investigación, se recomienda:

- Desarrollar un módulo al Administrador del Sistema de Perfiles Centralizados que se encargue de gestionar grupos y cuentas de usuario del servicio de directorio.
- Incorporar herramientas para la gestión de proyecto y control de los recursos humanos, por ejemplo DotProject (GENEST), con el objetivo de lograr una solución integral de gestión para entornos de desarrollo de software.
- Sustituir NFS por NBD (Network Block Device, en español Dispositivo de Bloques de Red), con el objetivo mejorar el rendimiento del sistema.



Referencias Bibliográficas.

[1] Definición de Usuarios, Equipos y Grupos [Citado el: 4 de Febrero del 2010] [Disponible en web: <http://sauce.pntic.mec.es/crer0052/usuarios/definici.htm>].

[2] Freddy Vargas, Santiago Durazno. Active Directory. [Citado el: 10 de marzo del 2010] [Disponible en: <http://www.slideshare.net/shantads/active-directory-1726927>].

[3] Novell eDirectory [Citado el: 24 de enero del 2010] [Disponible en: http://www.worldlingo.com/ma/enwiki/es/Novell_eDirectory/1#Features].

[4] Red Hat Directory Server [Citado el: 26 de enero del 2010] [Disponible en: http://www.redhat.com/directory_server/].

[5] Sitio oficial de OpenLDAP [Citado el: 6 de febrero del 2010] [Disponible en: <http://www.openldap.org>].

[6] Sitio oficial de Apache [Citado el: 10 febrero del 2010] [Disponible en: <http://directory.apache.org>].

[7] Sitio oficial de OpenDS [Citado el 15 de febrero del 2010] [Disponible en: <http://www.opensds.org/>].

[8] Prof. Mariela Curiel. Sistemas Distribuidos. [Citado el: 28 enero del 2010] [Disponible en: <http://74.125.93.132/search?q=cache:fa1Msil-52AJ:www ldc.usb.ve/~yudith/docencia/Telematica/TemasSistDist/Sistemas%2520de%2520Archivos%2520Distribuidos.pdf+Sistema+de+archivos+distribuidos&cd=6&hl=es&ct=clnk&gl=cu>].

[9] Arquitectura Cliente/Servidor. [Citado el: 6 de Febrero del 2010] [Disponible en: <http://www.csae.map.es/csi/silice/Global71.html>].

[10] Modelo cliente-servidor [Citado el: 6 de Febrero del 2010] [Disponible en: <http://es.wikipedia.org/wiki/Cliente-servidor>].

[11] Sitio oficial de la compañía Zend [Citado el: 6 de marzo del 2010] [Disponible en: <http://www.zend.com>].

[12] Daniel M. Maldonado. El CoDiGo K » Quanta Plus, la alternativa perfecta para desarrollar nuestras aplicaciones Web. 2008. [Citado el: 5 de Febrero del 2010] [Disponible en: <http://www.elcodigok.com.ar/2008/04/quanta-plus-la-alternativa-perfecta-para-desarrollas-nuestras-aplicaciones-web/>].



- [13] PHP: Hypertext Preprocessor. 2009. [Citado el: 4 de Febrero del 2010] [Disponible en: <http://www.php.net/>].
- [14] Una Introducción a Apache. 2009. [Citado el: 5 de Febrero del 2010] [Disponible en: http://linux.ciberaula.com/articulo/linux_apache_intro/].
- [15] Sitio oficial de Python [Citado el 20 de febrero del 2010] [Disponible en: <http://www.python.org>].
- [16] Design and Implementation of the Sun Network Filesystem [Citado el: 24 enero del 2010] [Disponible en: <http://web.mit.edu/6.033/2002/wwwdocs/papers/nfs.pdf>].
- [17] LDAP [Citado el: 14 de diciembre del 2009] [Disponible en: <http://es.wikipedia.org/wiki/LDAP>].
- [18] DNS - Domain Name Server [Citado el: 14 de diciembre del 2009] [Disponible en: http://compnetworking.about.com/cs/domainnamesystem/g/bldef_dns.htm].
- [19] About phpLDAPAdmin [Citado el 20 de febrero del 2010] [Disponible en: http://phpldapadmin.sourceforge.net/wiki/index.php/Main_Page].
- [20] María Isabel García Arenas. Curso XML 1ra Edición. [Citado el: 4 de abril del 2010] [Disponible en: <http://geneura.ugr.es/~maribel/xml/introduccion/index.shtml>].
- [21] [Citado el: 3 de abril del 2010] [Disponible en: http://www.blinkx.com/video/douglas-crockford-on-functional-javascript/xscZz8XhfuNQ_aaVuyUB2A].
- [22] Javascript/Jscript [Citado el: 3 de abril del 2010] [Disponible en: http://www.webdevelopersnotes.com/basics/languages_on_the_internet.php3].
- [23] Ingeniería del software. Sistemas Informáticos. Nivel de madurez software. Informática Aplicada a la Gestión Pública. 2005/06-2. Universidad de Murcia. Rafael Barzanallana. 2009. [Citado el: 5 de febrero del 2010] [Disponible en: <http://www.um.es/docencia/barzana/IAGP/lagp2.html>].
- [24] ¿Qué es BIND? [Citado el: 5 de febrero del 2010] [Disponible en: <http://www.isc.org/software/bind/whatis>].
- [25] Servidor DNS [Citado el: 5 de febrero del 2010] [Disponible en: <http://angelserrania.blogspot.es/>].
- [26] Gladys M. Peñalver Romero. MA-GMPR-UR2. Metodología ágil para proyectos de software libre. Junio del 2008.



Bibliografía.

- Programming Directory-Enabled Applications with Lightweight Directory Access Protocol, *por Howes and Smith.*
- Understanding and Deploying LDAP Directory Servers*, por Howes, Smith, and Good.
- Introducción a JavaScript. Javier Eguíluz Pérez. febrero de 2008. [Disponible en: <http://www.librosweb.es/javascript>].
- HTML La Guía Completa. Chuck Musciano y Bill Kennedy. junio 1999.
- Pro PHP XML and Web Services. Robert Richards. [Disponible en: <http://www.springeronline.com>].
- Linux Server Los mejores trucos Bill von Hagen y Brian K. Jones. 2006.
- W3C, "Cascading Style Sheets.", Departamento de Técnicas de Programación, "Introducción a la Programación cliente-servidor. Aplicaciones Web. Funcionamiento de un servidor Web," 2009.
- A.M. Abad and Gladys Marsi Peñalver Romero, "SXP, metodología ágil para proyectos de software libre," Universidad de las Ciencias Informáticas, 2009.
- Ltd. Zend Technologies, "Zend Framework," 2009.
- Kerberos V, OpenLDAP, OpenAFS. Using Debian GNU/Linux, Dr. Wolfgang A. [Disponible en: <http://www.dia.uniroma3.it/~wgehrke/docs/AfsWs2006print.pdf>].
- Testing Samba for Bigger Environments Samba / Linux / OpenLDAP at the german federal parliament [Disponible en: http://www.sambaxp.org/uploads/media/06_-_peter_ganten_-_univention_gmbh_-_testing_samba_for_bigger_environments.pdf].
- Implementation Panel: Active Directory, OpenLDAP, Sun, and Novell. [Disponible en: <http://net.educause.edu/ir/library/pdf/EAF0403.pdf>].
- Kerberos and OpenLDAP. [Disponible en: <http://www.kerberos.org/events/2009conf/chu.pdf>].



Anexos.

Anexo 1: Aplicación SPCAdmin, captura de pantalla.





Glosario de Términos.

Apache: Es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux), Windows y otras, que implementa el protocolo HTTP/1.1.

BSD: Es utilizado para identificar un sistema operativo derivado del sistema Unix.

GNU: Es un proyecto que fue iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre. Es un acrónimo recursivo que significa GNU No es Unix (GNU is Not Unix).

GPL: Es la Licencia pública general de proyecto GNU.

HTTP: Es el protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas web.

HTML: Es un Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

IDE: (Integrated Development Environment, en español: Entorno de Desarrollo Integrado). Es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, poder utilizarse para varios.

Interfaz: Es el de mediación, entre hombre y máquina. La interfaz es lo que "media", lo que facilita la comunicación, la interacción, entre dos sistemas de diferente naturaleza, típicamente el ser humano y una máquina como el computador. Esto implica, además, que se trata de un sistema de traducción, ya que los dos "hablan" lenguajes diferentes: verbo-icónico en el caso del hombre y binario en el caso del procesador electrónico.

Open Source: (Código abierto). Es el término con el que se conoce al software distribuido y desarrollado libremente.

Página web: Es una fuente de información adaptada para la World Wide Web y accesible mediante un navegador de Internet.



PHP: Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts. Con el objetivo de crear aplicaciones potentes.

RPC: (*Remote Procedure Call*). Conjunto de herramientas software desarrolladas por un consorcio de fabricantes y diseñadas para asistir a los diseñadores en la creación de aplicaciones distribuidas.

Software libre: Es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

TCP/IP: (*Transmission Control Protocol/Internet protocol*). Sistema de protocolos, definidos en RFC 793, en los que se basa buena parte de Internet. El primero se encarga de dividir la información en paquetes en origen, para luego recomponerla en destino, mientras que el segundo se responsabiliza de dirigirla adecuadamente a través de la red.

UNIX: Es un sistema operativo multitarea, multiusuario. Gran parte de las características de otros sistemas más conocidos como MS-DOS están basadas en este sistema muy extendido para grandes servidores. Internet no se puede comprender en su totalidad sin conocer el Unix, ya que las comunicaciones son una parte fundamental en Unix.

XML: son las siglas en inglés de eXtensible Markup Language (lenguaje de marcado ampliable o extensible) desarrollado por el World Wide Web Consortium (W3C). Es un estándar para describir datos y crear etiquetas. Las características especiales son la independencia de datos, o de la separación de los contenidos de su presentación. Es un metalenguaje que permite diseñar un lenguaje propio de etiquetas para múltiples clases de documentos. Los documentos XML se componen de unidades de almacenamiento llamadas entidades (entities), que contienen datos analizados (parsed) o sin analizar (unparsed).

Zend: Es una compañía encargada del desarrollo del popular lenguaje de programación PHP. La compañía Zend Technologies, fundada por los creadores de PHP.

Dirección IP: Dirección de un ordenador dentro de una red con protocolo TCP/IP.

Protocolo: Conjunto de normas que rigen un determinado proceso de comunicación.

Network File System (NFS): Sistema de Archivos de Red. Desarrollado por Sun Microsystems, es un protocolo que permite establecer sistemas de archivos distribuidos entre múltiples máquinas.



Servidor: computadora central de un sistema de red que provee servicios y recursos (programas, comunicaciones, archivos, etc.) a otras computadoras (clientes) conectadas a ella.

Cliente: Sistema que establece un intercambio de datos con un servidor.

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 IPs el número de la red, incluyendo la subred, y qué parte es la correspondiente al host.

Domain Name System (DNS): es un servicio de búsqueda de datos de uso general, distribuido y multiplicado. Su utilidad principal es la búsqueda de direcciones IP de sistemas centrales (Hosts) basándose en los nombres de éstos. El estilo de los nombres de hosts utilizado actualmente en Internet es llamado nombre de dominio (com, net, org, es, etc.).

Software Privativo: es 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 (con o sin modificaciones), o cuyo código fuente no está disponible o el acceso a éste se encuentra restringido.

Software Libre: se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.