Universidad de las Ciencias Informáticas

Facultad 10

Título: Integración de servicios telemáticos enfocado a la migración hacia una plataforma en Software Libre. .

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

Autor: Erick Salabarría Aquino.

Tutor: Ing. Juan Manuel Pernía Rodríguez.

Co-tutor: Ing. Eder Despaigne Herrera.

Julio de 2007

DECLARACION DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 2 días del mes de julio del año 2007

Autor: Erick Salabarría Aquino

Tutor: Ing**.** Juan Manuel Pernía Rodríguez

Firma del Autor

Firma del Tutor

DATOS DE CONTACTO

Tutor

Nombre: Juan Manuel Pernía Rodríguez

Fecha de nacimiento: 21-Noviembre-1979.

Lugar de nacimiento: Manzanillo, Granma. Cuba

Empleo actual:

- Desde 2003: Universidad de las Ciencias Informáticas (UCI), Ciudad de la Habana.

Dirección: Facultad 10, Departamento de la Especialidad, UCI. Carretera a San Antonio de los Baños Km. 2 ½ Municipio. La Lisa, Ciudad de la Habana, Cuba.

Email: jpernia@uci.cu

Nacionalidad: Cubano

Estado civil: Soltero

Grado científico: Ingeniero en Telecomunicaciones y Electrónica.

Trabajos de Investigación:

1. Investigación de la tecnología VoIP (2002).
2. Investigación del Estándar Digital de TV de Alta Definición (HDTV) desarrollado por la Gran Alianza, (2002-2003)
3. Desarrollo de versión 1.0 sistema “Analizador de dinámica de poblaciones, ANDI v1.0”, 2003-2004
4. Estudio de las aplicaciones (LDAP, Samba, DNS, DHCP) de software libre, para la migración de los servicios telemáticos en la UCI a esta plataforma (2005-2007).
5. Actualmente está trabajando en la investigación de la Calidad de Servicio (QoS) enfocada a la seguridad de la red, como tema de tesis de la maestría en Telemática.

Eventos:

1. Durante el período de estudio de la carrera participé en los siguientes eventos:
   - Forum de Matemáticas de Base en segundo año. (2do Lugar).
   - Forum de la Universidad de Oriente con el trabajo EVL-2.1 software de instrumentación virtual. (2do Lugar).
   - Telec -2002, Santiago de Cuba.
   - Forum de la Facultad de Ingeniería Eléctrica con el trabajo Sistema de Tele conferencia. (Relevante).
2. Participación en el evento CITEL-2004, Habana.
3. Participación en el evento CITEL-2006, Habana.
4. Participación en el evento Informática 2007, Habana.

Postgrados:

1. Cursos de Formación del Profesor de la Educación Superior.
2. Certificación UCI como Administrador de Redes.
3. Ha cursado todas las asignaturas de los dos primeros módulos y cursa actualmente las asignaturas del tercer módulo de la maestría de Telemática en su 8va Edición, en la CUJAE.

Curso Impartido:

1. Diplomado en Administración de Redes.
2. Integración de Samba, LDAP y Windows; impartido en la República Bolivariana de Venezuela.
3. Curso de administración de servicios telemáticos en software libre a personal de los Ministerios de Cultura y de la Fuerzas Armadas.

Idiomas:

1. Español.
2. Inglés.

DATOS DE CONTACTO

Co-tutor

Nombre: Eder Despaigne Herrera

Fecha de nacimiento: 7-Noviembre-1977.

Lugar de nacimiento: Palma Soriano, Santiago de Cuba. Cuba

Empleo actual:

- Desde 2003: Universidad de las Ciencias Informáticas (UCI), Ciudad de la Habana.

Dirección: Facultad 10, Departamento de la Especialidad, UCI. Carretera a San Antonio de los Baños Km. 2 ½ Municipio. La Lisa, Ciudad de la Habana, Cuba.

Email: ederdh@uci.cu

Nacionalidad: Cubano

Estado civil: Soltero

Grado científico: Ingeniero en Telecomunicaciones y Electrónica.

Postgrados:

1. Cursos de Formación del Profesor de la Educación Superior.

Curso Impartido:

1. Servicios Web y Correo electrónico; impartido en la República Bolivariana de Venezuela.
2. Curso de administración de servicios telemáticos en software libre a personal de los Ministerios de Cultura y de la Fuerzas Armadas.

Idiomas:

1. Español.
2. Inglés.

"Las obras de conocimiento deben ser libres,

no hay excusas para que no sea así"

Richard Stallman

Agradecimientos

- A mi familia por su perdurable, insaciable y valiosa ayuda. Por permitirme llegar hasta aquí. Gracias a ustedes puedo ser hoy quien soy. Los quiero a todos. Al fin mami y papi, dejé de ser un chaval.
- A mi madre y a mi padre en especial, por sus profundas leyendas, instrucciones y enseñanzas, sus constantes preocupaciones, su confianza, su inigualable apoyo y cariño.
- A mi hermano mayor que siempre me ha guiado por el buen camino, me ha aconsejado mucho y ha puesto en mi toda seguridad y confianza.
- A mi otra madre, Barbarita gracias por brindarme tu mano en los bueno y malos momentos.
- A mis buenos amigos Osmel (basuco), Geykel (dorotille), Ionian (el jabao), Yan Pavel, Leonardo, Noel, Yoel, Randy y Bombino que han estado en los malos y buenos momentos.
- A Leyanis por el gran apoyo y ayuda brindada.
- En general a todos mis compañeros de cuarto y de grupo por compartir una parte de mi vida.
- A mi tutor Juan Manuel Pernía Rodríguez por sus buenos consejos, ayuda y revisiones.
- A mi otra fuente de ayuda el Ing. Eder Despaigne Herrera que me ha ayudado cada vez que he necesitado de su colaboración.
- A Dionner Polanco por su gran cooperación.
- A los compañeros del laboratorio, por las recomendaciones y ayuda brindada.
- A Hany mi pirañita por su enorme paciencia. Dentro de poco tendremos más tiempo para nosotros.
- A los amigos de la cocina y a los custodios, Juan Carlos, Fabio, Osniel, German, el Papa, Mijail.
- A mis suegros que a pesar de sus locuras me han ayudado mucho.
- A la Revolución Cubana por darme la oportunidad de estudiar en una universidad de excelencia.
- A todos los que sin querer olvido, muchas gracias.

DEDICATORIA

A mis padres.

A mi esposa.

A mi futuro bebé.

A mis hermanos.

A mi sobrino.

A mis tíos.

A mis primos.

A mi abuela.

A Barbarita

A mis amigos.

Al Movimiento de Software Libre.

Resumen

El desarrollo del software libre a nivel mundial se ha convertido en la principal alternativa para muchos países, principalmente los que se encuentran en vías de desarrollo. Cuba es uno de estos y la migración de los sistemas informáticos en nuestra sociedad a Software Libre constituye un gran reto. Con este trabajo se pretende integrar los servicios telemáticos que están presentan en una red de datos utilizando Software Libre.

Para su elaboración se realizó un estudio sobre las principales implementaciones de los diferentes servicios que se brindarán y se escogieron las más adecuadas para realizar la integración de los mismos.

Para lograr dicha integración se utilizó OpenLDAP, este es una implementación libre del Protocolo de Acceso Ligero a Directorio (LDAP) la cual permite simplificar el proceso de mantenimiento y gestión de servicios. Esto es posible debido a que los datos referentes a los usuarios de los servicios ofrecidos se encuentran en este directorio, quedando así la información centralizada.

Como resultado final se obtiene un manual de instalación donde se recogen los pasos necesarios para instalar y configurar los diferentes servicios que están presentes en una red de datos.

Índice

Introducción

El gran desarrollo de la industria del software ha llevado consigo el desarrollo de la informática a nivel mundial. Esta industria se ha convertido en una de las más importantes de las últimas décadas. Mediante la producción de software numerosas empresas adquieren extraordinarias ganancias, principalmente las grandes compañías.

Estas son las que van a determinar cuales tecnologías utilizar guiando así el mercado mundial, mientras que los países con menos desarrollo van a depender de estos proveedores, lo que hace que haya un aumento de la dependencia hacia los mismos y una permanente desigualdad en términos de equilibrios económicos.

Cuba se encuentra entre los países que no tienen un alto desarrollo tecnológico debido al cruel bloqueo impuesto por los Estados Unidos desde hace ya más de 48 años, por tanto el país no tiene acceso a muchos software de forma legal, lo que hasta ahora tiene es por otras vías y aunque Cuba no estuviera bloqueada no podría acceder a estos recursos debido a que los costos por concepto de licencias y patentes son bastante elevados, lo cual para el país es prácticamente imposible por su situación económica.

Sin embargo el Software Libre se ha convertido en la principal alternativa. Su utilización nos llevará a la no dependencia, la creación, la innovación y por sí solo al desarrollo individual y colectivo, una opción ideal para Cuba. Sin duda alguna, el movimiento del Software Libre es una consecuencia de la necesidad de establecer nuevos modelos emergentes dentro de la industria del software.

Una de las características del Software Libre es que se desarrolla en comunidades, esto implica redes de desarrolladores trabajando desde distintas localidades, compartiendo códigos para el mejoramiento del software. Esta forma de trabajo es completamente distinta al modelo de desarrollo tradicional de software, que mantiene el control sobre el código y el proceso de desarrollo.

En la actualidad podemos encontrar una gran heterogeneidad en las redes informáticas en las cuales existen múltiples clientes, cada uno de ellos puede tener un sistema operativo distinto, sobre el cual puedan operar muchos usuarios lo cual es bastante complejo.

La necesidad de nuestro país de migrar todos sus sistemas a una plataforma libre demanda una guía que sirva de base para montar los servicios de una red de datos, estos actualmente, en su mayoría, están montados en software propietario. Por lo que se hace necesario buscar un método utilizando Software Libre que facilite, la labor de administración y que permita una integración de los servicios telemáticos contra un servicio de directorio centralizado que almacene los datos de usuarios, lo que constituye el **problema científico** a resolver.

El **objeto de estudio** está orientado a servicios telemáticos que están presentes en una red de datos, delimitando así el **Campo de acción** al proceso de integración de estos con un servicio de directorio.

Objetivos de investigación

Elaborar un documento que sirva de guía en la implantación de los servicios telemáticos en una red de datos.

Las **preguntas científicas** que sustentan la investigación son:

¿Cómo Integrar servicios telemáticos contra un directorio LDAP utilizando Software Libre?
¿Cómo montar un Controlador Primario de Dominio (PDC) con Software Libre?

Tareas de investigación.

1- Estudio de los servicios telemáticos LDAP (Protocolo de Acceso Ligero a Directorio), FTP (Protocolo de Transferencia de Ficheros), DNS (Sistema de Nombres de Dominio), HTTP (Protocolo de Transferencia de Hipertexto), Samba (SMB y NMB).
2- Integración de los servicios telemáticos contra el servicio de directorio con LDAP
3- Montaje de un PDC con Software Libre.

Los **métodos teóricos** que son utilizados para dar cumplimiento a estas tareas son:

Analítico-Sintético: Pues centrándose en el análisis de las teorías y documentos; permitieron la extracción de los elementos más importantes que se relacionan con el objeto de estudio para lograr su comprensión.

Análisis Histórico-Lógico: Para el estudio de la evolución del Software Libre ,el servicio de directorio LDAP y el resto de los servicios que están presentes en este trabajo, pues este método nos permite estudiar de forma analítica la trayectoria histórica real de los fenómenos, su evolución y desarrollo.

Se utiliza además el **método empírico** de la **observación** al realizar el manual de instalación plasmando en él lo referente al proceso de instalación y configuración de los servicios.

Este trabajo se desarrolla en 3 capítulos:

Capítulo 1. Fundamentación Teórica

En este capítulo se abordarán los temas relacionados con el Software Libre y su utilización para Cuba. Se hará un estudio de los distintos servicios que están presentes en una red de datos mediante una breve explicación de los distintos protocolos que tienen relación con dichos servicios así como las implementaciones de los mismos que se utilizarán en este trabajo.

Capítulo 2. Conociendo a LDAP

En este capítulo se abordarán los aspectos relacionados con el servicio de directorio LDAP. Se realiza un estudio en cuanto a su funcionamiento, ventajas y algunas características importantes pues este es fundamental para lograr la integración de los diferentes servicios que se utilizarán en el desarrollo de este trabajo.

Capítulo 3. Integración de Servicios

Este capítulo está dedicado a la integración de los servicios telemáticos presentes en una red. La misma se hará sobre un directorio LDAP utilizando Software Libre. Todos los procedimientos realizados se hacen sobre la distribución **Debian GNU/Linux** versión **4**.

Capitulo 1: Fundamentación Teórica

En este capítulo se abordarán temas relacionados con Software Libre. Dentro de estos podemos encontrar su definición, la importancia de la utilización del Software Libre para Cuba, así como distintas clasificaciones de sus licencias y una breve explicación de algunas de estas. Otro aspecto que se dará a conocer son los distintos servicios que están presentes en una red de datos, mediante una breve explicación de los distintos protocolos que tienen relación con dichos servicios

1.1 Conceptualizando el Software Libre.

Software Libre es la clasificación que se le da a todo software que puede ser usado, copiado, estudiado, modificado, mejorado y redistribuido libremente, Software Libre es un asunto de libertad, no de precio.
Para entender el concepto, se debe pensar en libre como en libertad de expresión. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito (libertad 0).

- La libertad de estudiar como funciona el programa, y adaptarlo a tus necesidades (Libertad 1). El acceso al código fuente es una condición previa para esto.

- La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad2).

- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie (libertad 3). El acceso al código fuente es un requisito previo para esto.(LIBRE: 2006b)

Cuando se habla de Software Libre, es mejor evitar términos como regalar o gratis, porque esos términos implican que lo importante es el precio, y no la libertad.

El Software Libre debe tener la libertad de hacerle modificaciones y utilizarlas de manera privada en tu trabajo, sin ni siquiera tener que anunciar que dichas modificaciones existen. Si publicas tus cambios, no tienes por qué avisar a nadie en particular, ni de ninguna manera en específico.

1.2 Antecedentes del Software Libre.

Hace años, cuando la realidad tecnológica era distinta, la mayoría del software era producido por los precursores de la llamada tercera revolución tecnológica, quienes tenían la posibilidad de cooperar entre ellos, y eventualmente así lo hacían. Alrededor de 1980, la gran mayoría del software ya era propiedad intelectual de alguien.

Los propietarios de estas tecnologías por seguridad decidieron bloquear los códigos y prohibir a los programadores hablar con gente externa. De esta forma mantenían en secreto sus creaciones. Indudablemente estas decisiones fueron limitando gradualmente la cooperación entre los programadores y cercando el desarrollo del software dentro de las necesidades del mercado.

Debido a esto Richard Stallman hace a un lado sus ocupaciones en el laboratorio de inteligencia artificial del Instituto de tecnología de Massachussets (Massachussets Institute of Tecnology) considerando que no solo dejaba de estimular a los usuarios comunes a profundizar sus aprendizajes sobre el desarrollo de software, convirtiéndolos en esclavos de un soporte técnico, caro y deficiente en el mejor de los casos sino que al limitar las colaboraciones entre los programadores limitaba el desarrollo de software .

Es así que en 1986 cuando le piden Stallman firmar un acuerdo de no divulgación, este decide renunciar, y publica el manifiesto GNU (GNU is not UNIX).

Este manifiesto, daba inicio a un proyecto que estaría encaminado a la construcción de un sistema operativo compatible con UNIX pero con la diferencia que este sería totalmente gratuito y abierto a la posibilidad de ser modificado de acuerdo con las necesidades especificas de los usuarios, garantizado por su código abierto.

El proyecto empezó a crecer y a fortalecerse, muchos programadores participaron entusiastas con la idea de desarrollar el software gratuito. Después de varias sesiones de trabajo, habían desarrollado las ideas básicas de un sistema operativo: un compilador, un editor de texto e intérprete de lenguajes y herramientas para el trabajo en red. Pero faltaba un componente esencial en el desarrollo del Software Libre y que propició la aparición de Linux, el Kernel.

En 1990 Linux Torvalds siendo estudiante de la Universidad de Helsinki, Finlandia, decide mejorar un sistema operativo llamado MINIX que explotaba al máximo las capacidades de los recién llegados 80386 y es así cuando surgió el Kernel que hoy conocemos como Linux.

Con el desarrollo de Linux, es cuando Stallman y sus colaboradores encuentran en aquel Kernel el elemento que hacia falta en su sistema operativo, como resultado de la unión de estos dos proyectos surge lo que hoy conocemos como GNU/Linux, que no es ni GNU ni Linux, sino una mezcla de los dos. (GARCÍA 2004)

A partir de este momento el desarrollo del Software Libre siguió evolucionando y con esto se hizo necesario algún mecanismo que garantizara que el software que fuera producido bajo este concepto no pasara a ser propiedad de alguna persona o institución que no fuera su creador, por esta razón surgen las licencias de Software Libre.

1.3 Licencias de Software Libre.

Una licencia de software es aquella autorización formal con carácter contractual que el autor de un software da a un interesado para ejercer "actos de explotación legales". Pueden existir tantas licencias como acuerdos concretos se den entre el autor y el licenciatario. Desde el punto de vista del Software Libre, existen distintas variantes del concepto o grupos de licencias.

1.3.1 Clasificación de una licencia según ciertas cuestiones claves.

- Si puede ser considerada una licencia de Software Libre.
- Si es una licencia de tipo copyleft.
- Si es compatible con la GNU GPL (esto significa que se puede combinar un módulo que fue distribuido bajo esa licencia con otro cubierto por la GPL).
- Si causa cualquier otro problema

Una licencia es considerada de Software Libre cuando cumple con las cuatro libertades establecidas en la definición de Software Libre.(LIBRE: 2006a)

Copyleft describe un grupo de derechos aplicados a una diversidad de trabajos como programas informáticos, arte, cultura y ciencia, es decir prácticamente casi cualquier tipo de producción creativa. (*Copyleft* 2007)

Una licencia es de tipo copyleft cuando el producto que está licenciando no puede tener restricciones de segundos. Sus partidarios la proponen como alternativa a las restricciones de derechos para hacer y redistribuir copias de una obra determinada.

Se pretende garantizar así una mayor libertad, cada persona receptora de una copia o una versión derivada de un trabajo pueda a su vez, usar, modificar y redistribuir tanto el propio trabajo como las versiones derivadas del mismo. En un entorno no legal, puede considerarse como opuesto al copyright o derechos de autor tradicionales.

Las libertades definidas en el concepto de Software Libre están protegidas por licencias de Software Libre, una de las más utilizadas es la Licencia Pública General GNU (GPL). El autor conserva los derechos de autor “copyright”, y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que no sea imposible crear un producto con partes no licenciadas GPL.

1.3.2 Algunas licencias de Software Libre compatibles con GPL.

- **La Licencia Pública General Reducida de GNU**.

Es una licencia de Software Libre, pero no tiene un copyleft fuerte, porque permite que el software se enlace con módulos no libres. Entre la versión 2 y la 2.1, la GNU LGPL cambió su nombre de "Licencia Pública General para Bibliotecas de GNU" a "Licencia Pública General Reducida de GNU", pues no es sólo para bibliotecas.

- **Licencias estilo BSD.**

Licencias estilo BSD, llamadas así porque se utilizan en gran cantidad de software distribuido junto a los sistemas operativos BSD. El autor, bajo estas licencias, mantiene la protección de “copyright” únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas, tanto que son fácilmente absorbidas al ser mezcladas con la licencia GNU GPL con quienes son compatibles.

1.3.3 Licencias de Software Libre incompatibles con la GPL.

- **La Licencia Pública de Mozilla (MPL).**

Es una licencia de Software Libre, pero no tiene un copyleft fuerte. Tiene algunas restricciones complejas que la hacen incompatible con la GNU GPL. De hecho, no se puede, legalmente, enlazar un módulo cubierto por la GPL con un módulo cubierto por la MPL.

Tiene un gran valor porque fue el instrumento que empleó Netscape Communications Corp, para liberar su Netscape Communicator 4.0 y empezar ese proyecto tan importante para el mundo del Software Libre, Mozilla. Esta licencia se utiliza en gran cantidad de productos de Software Libre de uso cotidiano en todo tipo de sistemas operativos.(LIBRE: 2006a)

- **Licencia OpenLDAP.**

En este subepígrafe se tratará la licencia OpenLDAP debido al estrecho vínculo que esta tiene con el servicio de directorio LDAP, el cual tiene un papel importante en el desarrollo de este trabajo ya que el resto de los servicios tratados de una forma u otra se integrarán a él.

OpenLDAP es una licencia de Software Libre que tiene algunos permisos pero no llega a ser copyleft, en la versión 2.5 incorpora algunas sesiones que la hacen incompatible con GNU GPL.

Es recomendable no usar la antigua licencia de OpenLDAP para el software que escriba. Sin embargo no hay razón para no usar programas que hayan sido distribuidos bajo esta licencia.

La versión 2.7 es permisiva y sin copyleft pero compatible con la GNU GPL. Esta versión no incluye las secciones 4 y 5 de la anterior, las cuales la hacían incompatible con la GNU GPL, quedando entonces con las siguientes secciones:

1. Las redistribuciones en forma fuente deben conservar declaraciones y avisos del copyright.
2. Las redistribuciones en forma binaria deben reproducir declaraciones aplicables del copyright y los avisos.
3. Las redistribuciones deben contener una copia in extenso de este documento. La fundación de OpenLDAP puede revisar esta licencia de vez en cuando. Cada revisión es distinguida por un número de versión. (LIBRE: 2006a)

1.4 Cuba y el Software Libre.

El uso de la informática en Cuba ha tenido soporte generalmente en los sistemas operativos de Microsoft Windows. Nuestro país desde mayo del 2005 ha estado organizando la migración de sus aplicaciones hacia una plataforma libre. La estrategia es utilizar el sistema operativo GNU/Linux como sistema base para reemplazar al sistema actualmente en uso, Microsoft Windows.

Desde sus inicios se trazó una estrategia que comprende acciones de organización, técnicas, diseño y desarrollo de un marco legal, así como la capacitación de personal y el cambio paulatino de un sistema a otro.

El proceso de migración en nuestro país esta protagonizado por el Instituto Superior Politécnico "José Antonio Echeverría"(ISPJAE) y la Universidad de las Ciencias Informáticas (UCI), en la misma existe una facultad cuyo perfil es el Software Libre y es donde primero se empezó a trabajar este concepto dentro de la universidad, actualmente en otras facultades se le da seguimiento a las acciones relacionadas con la migración, un ejemplo de esto es la facultad cuyo perfil es bioinformática. En estas se desarrollan importantes proyectos productivos tanto nacionales como internacionales utilizando Software Libre. Un aspecto esencial es que no solo se pretende aplicar esta estrategia a una facultad sino a toda la Universidad y paulatinamente al resto del país.

Si se lleva a cabo la migración en la Universidad de las Ciencias Informáticas el país tendría una parte importante de este proceso vencido, debido que esta es una de las instituciones del país con mayor desarrollo tecnológico y por consiguiente es donde mayor esfuerzo y empeño hay que realizar para lograr dicha migración pues hay una gran cantidad de recursos informáticos prestando servicios. Con esta misión cumplida Cuba contaría con una base sólida para llevar acabo este proceso y entonces solo quedaría aplicar estos conceptos, aunque en mayor tiempo, a otros sectores e instituciones de nuestro país donde el desarrollo tecnológico es menor.

1.5 Significación del uso del Software Libre para Cuba.

El Software Libre representa la no utilización de productos informáticos que demanden la autorización de sus propietarios para su explotación. La mayoría de las instituciones cubanas se encuentran a merced de la empresa norteamericana Microsoft, que tiene la capacidad legal de reclamar a Cuba la no utilización del sistema operativo de su propiedad, basada en leyes de propiedad industrial. Esto provocaría una interrupción inmediata del programa de informatización de la sociedad que como parte de la batalla de ideas se está desarrollando en el país, además pudiera implementarse una campaña de descrédito a la isla, abogando el uso de la piratería informática por parte de las instituciones estatales cubanas. Sin embargo con la utilización del Software Libre no habría que preocuparse por todas estas cuestiones legales.

Cuando un producto de Software Libre empieza a circular, rápidamente está disponible a un costo muy bajo o sin costo alguno. Al mismo tiempo, su utilidad no decrece, esto significa que el Software Libre se puede caracterizar como un bien público en lugar de un bien privado. Este software es desarrollado de forma colectiva y cooperativa, tanto en su creación como en su desarrollo, tanto cuantitativa como cualitativamente mostrando su carácter público y sus objetivos de beneficiar a toda la comunidad.

Su utilización no implica gastos adicionales por concepto de cambio de plataforma de software, por lo cual es operable en el mismo soporte de hardware con que cuenta el país. La adquisición de cualquiera de sus distribuciones puede hacerse de forma gratuita, descargándolas directamente de Internet o en algunos casos a muy bajos precios, se garantiza su explotación con un mínimo de recursos, por tanto no hay que pagar absolutamente nada por su utilización ,no requiere de licencia de uso, las cuales son generalmente muy caras, distribución y modificación.

Permite su adaptación a los contextos de aplicación, al contar con su código fuente, lo cual garantiza un mayor por ciento de efectividad, además la corrección de sus errores de programación y obtención de las actualizaciones y nuevas versiones.(ESPINOSA)

1.6 Conociendo algunos servicios.

Al abordar los temas de este subepígrafe el autor pretende dar una breve panorámica de los distintos servicios que pueden estar presentes en una red de datos mediante diferentes software.

¿Qué es un Servicio?

Un servicio es un conjunto de actividades que buscan responder a una o más necesidades de un cliente. Se define un marco en donde las actividades se desarrollarán con la idea de fijar una expectativa en el resultado de éstas. (*Servicios* 2007)

En el campo de la informática estos servicios se pueden ver como software o aplicaciones que satisfacen las necesidades de determinados clientes (software o usuarios) en correspondencia con tareas específicas que estos deben cumplir**.**

1.6.1 Servicio DNS.

En los inicios de Internet, todos los nombres existentes estaban en un archivo centralizado conocido como HOSTS.TXT. Este archivo contenía los nombres de todos los dominios conocidos. El crecimiento acelerado de la red causó que el sistema de nombres centralizado en el archivo HOSTS no resultara práctico y en 1983, Paul Mockapetris publicó lo que hoy en día ha evolucionado al DNS moderno.

El Domain Name System (DNS) es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio. Aunque como base de datos el DNS es capaz de asociar distintos tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y viceversa, la localización de los servidores de correo electrónico de cada dominio.(MIGUEL)

La asignación de nombres a direcciones IP es ciertamente la función más conocida de los protocolos DNS. Por ejemplo, si la dirección IP del sitio del correo de yahoo.com es 216.109.112.135, la mayoría de las personas llegan a este sitio especificando mail.yahoo.com y no a través de la dirección IP, pues el

nombre de el sitio es más fácil de recordar y más fiable. La dirección IP podría cambiar por muchas razones, y el nombre sigue siendo el mismo.

En este trabajo se utilizará BIND como implementación del protocolo DNS, este es uno de los primeros servidores DNS creados. Las primeras implementaciones del servidor BIND mostraban una gran cantidad de vulnerabilidades, la versión 9 del producto ya no presenta estas complicaciones. Fue escrita desde cero para superar las dificultades técnicas de antiguos desarrollos, dicha versión fue impulsada por proveedores UNIX.

BIND 9 ofrece un servidor de nombres de dominio, una biblioteca de resolución de sistemas de nombres de dominio y un paquete de herramientas para monitorizar el correcto funcionamiento de todo el sistema. Entre sus principales características se incluyen los protocolos de seguridad y el soporte de IPv6, actualizaciones dinámicas, notificación DNS, vistas y procesamiento en paralelo. Gracias a su arquitectura mejorada se ha conseguido una mejor portabilidad entre sistemas. Por estas razones se escogió esta versión para brindar el servicio DNS. (TEMPUS 2007)

1.6.2 Servicio FTP.

FTP (File Transfer Protocol) es un protocolo de transferencia de ficheros entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor, de manera que desde un equipo cliente nos podemos conectar a un servidor para descargar ficheros desde él o para enviarle nuestros propios archivos independientemente del sistema operativo utilizado en cada equipo. (*File Transfer Protocol* 2007)

El servicio de FTP es uno de los más antiguos y utilizados para la transferencia de archivos, los Servidores Web lo utilizan para actualizar sus páginas estáticas, subir imágenes o para permitir la descarga de archivos a los visitantes.

El Servicio FTP es ofrecido por la capa de aplicación del modelo de capas de red **TCP/IP** al usuario, utilizando normalmente el puerto de red 20 y el 21. Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, ya que todo el intercambio de

información, desde la autenticación del usuario en el servidor hasta la transferencia de cualquier fichero, se realiza en texto plano sin ningún tipo de cifrado, por lo que para un posible atacante sería muy fácil capturar este tráfico, acceder al servidor, o apropiarse de los ficheros transferidos. Existen protocolos de comunicación FTP para que los datos se transmitan cifrados, como el SFTP.

Los servidores FTP generalmente brindan 3 tipos de accesos:

- El acceso al directorio personal del usuario.
- El acceso dentro de una jerarquía de directorios o archivos destinada al FTP.
- El acceso anónimo.(HINOSTROZA 2007)

Existen varios servidores FTP en GNU/Linux:

- Proftpd
- Vsftpd
- Pureftpd
- wu-ftpd
- twoftpd

En este trabajo se decidió escoger a Proftpd porque es un servidor FTP rápido, de fácil instalación y flexible configuración, con un esquema similar a la configuración de Apache. Además permite diferentes métodos de autenticación: mediante usuarios del sistema, base de datos Mysql o directorio LDAP.

1.6.3 Servicio HTTP.

El protocolo de transferencia de hipertexto (HTTP) es el protocolo usado en cada transacción de la Web. El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceso a una página y la respuesta con el contenido. También sirve el protocolo para enviar información adicional en ambos sentidos, como formularios con campos de texto. (*Hypertext Transfer Protocol* 2007)

HTTP es un protocolo sin estado, esto significa que no guarda ninguna información sobre conexiones anteriores. Al finalizar la transacción todos los datos se pierden. Debido a esto fueron creados pequeños ficheros en la propia computadora llamados cookies, estos guardan información referente a las visitas a un determinado sitio, gracias a esto el sitio web puede almacenar gran número de información sobre cada visitante, ofreciéndole así un mejor servicio.

Se utilizará en este trabajo el Servidor Web Apache como implementación del protocolo HTTP debido que es un Servidor Web muy potente de código abierto, altamente configurable, puede usarse en distintas plataformas como BSD, GNU/Linux, Windows. Existen muchos módulos que permiten la integración de este servidor con otras aplicaciones. Apache presenta una gran eficiencia gestionando las tareas de forma más óptima y rápida. Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Fundation. Actualmente es el Servidor Web más usado en el mundo. Se instalará la versión 2.2.3-4.

1.6.4 SMB

Este es un protocolo que permite compartir entra otras cosas archivos e impresoras entre nodos de una red. Fue originalmente inventado por IBM, pero la versión más común hoy en día es la modificada por Microsoft. Esta compañía renombró SMB a Common Internet File System (CIFS) en 1998 y añadió más utilidades. Para lo sistemas GNU/Linux y otros UNIX existe una implementación libre de dicho protocolo llamada Samba. (*Server Message Block* 2007)

1.6.5 Servicio de Directorio.

Un directorio es una base de datos optimizada para lectura, navegación y búsqueda. Los directorios tienden a contener información descriptiva basada en atributos y tienen capacidades de filtrado muy sofisticada.

Los directorios generalmente no soportan transacciones complicadas ni esquemas de vuelta atrás como los que se encuentran en los sistemas de bases de datos diseñados para manejar grandes y complejos volúmenes de actualizaciones.

Las actualizaciones de los directorios son normalmente cambios simples, o todo o nada, siempre y cuando estén permitidos. Los directorios están afinados para dar una rápida respuesta a grandes volúmenes de búsquedas. Estos tienen la capacidad de replicar la información para incrementar la disponibilidad y la fiabilidad, al tiempo que reducen los tiempos de respuesta. Cuando la información de un directorio se replica, se pueden producir inconsistencias temporales entre las réplicas mientras esta se está sincronizando.

Hay muchas formas de proveer un servicio de directorio. Diferentes métodos permiten almacenar distintos tipos de información en el directorio, tener distintos requisitos sobre como la información ha de ser referenciada, consultada, actualizada y como es protegida de los accesos no autorizados.(GONZÁLEZ 2004)

LDAP.

El Protocolo de Acceso Ligero a Directorio (LDAP), está basado en el estándar X.500, pero significativamente más simple y adaptado para satisfacer las necesidades del usuario. A diferencia de X.500, LDAP soporta TCP/IP (Transport Control Protocol and Internet Protocol), que es necesario para el acceso a Internet.

LDAP aparece con el estándar de los directorios de servicios X.500. La versión original fue desarrollada por la Universidad de Michigan. Se usó inicialmente como un Front-end o interfaz final para dicho estándar pero también puede usarse con servidores de directorio únicos y con otros tipos de servidores de directorio.

Las características de un servidor LDAP son:

- **Operaciones de lectura muy rápidas.** Debido a la naturaleza de los datos almacenados en los directorios las lecturas son más comunes que las escrituras.
- **Datos relativamente estáticos.** Los datos almacenados en los directorios no suelen actualizarse con mucha frecuencia.
- **Entorno distribuido, fácil replicación.**
- **Estructura jerárquica.** Los directorios almacenan la información de forma jerárquica.

- **Orientadas a objetos.** El directorio representa a elementos y a objetos. Los objetos son creados como entradas, que representan una colección de atributos.
- **Esquema Standard.** Los directorios utilizan un sistema estándar que pueden usar fácilmente diversas aplicaciones.
- **Atributos multi-valor.** Los atributos pueden almacenar un valor único o varios.
- **Replicación multi-master.** Muchos de los servidores LDAP permiten que se realicen escrituras o actualizaciones en múltiples servidores.(*Manual de OpenLDAP en español* 2004)

Lo que está ocurriendo en la actualidad con el servicio de directorio LDAP es novedoso. Su utilización en cualquier empresa o institución puede facilitar la obtención de información del directorio LDAP desde cualquier aplicación, ejecutándose en cualquier plataforma de computación. Puede ser utilizado para almacenar un amplio rango de datos: dirección de correo electrónico e información de encaminamiento de correo, claves publicas de seguridad, listas de contactos, y mucho más.

En la actualidad existen diversas implementaciones y aplicaciones reales del protocolo LDAP. Todas estas implementaciones tienen algunas diferencias según quienes las hayan desarrollado pero en concreto todas basan su funcionamiento en el protocolo LDAP.

Implementaciones de LDAP.

- Active Directory para Microsoft Windows
- Novell Directory Services para Novell
- Red Hat Directory Server para Red Hat
- OpenLDAP

OpenLDAP es una implementación libre del protocolo, soporta múltiples esquemas por lo que puede utilizarse para conectarse a cualquier otro LDAP. Es la versión que se utilizará en este trabajo debido a las ventajas que tiene y que lo hacen ideal de acuerdo a la política de migración que nuestro país esta llevando a cabo. Debido a que es completamente libre permite implementar nuevos módulos y adaptarlo a las necesidades existentes. Es independiente de la plataforma por lo que varias distribuciones como Linux, BSD, Solaris lo utilizan, incluso el sistema operativo Windows (2000/XP).

1.7 Controlador de Dominio.

El tema de controlador de dominio es muy importante en el desarrollo y comprensión de este trabajo. Los controladores de dominio tienen una serie de responsabilidades, una de ellas es la autenticación. Este es el proceso de permitir o denegar a un usuario el acceso a recursos compartidos o a otra máquina de la red, normalmente a través del uso de una contraseña.

Cada controlador de dominio usa un security account manager (SAM) para mantener una lista de combinaciones nombre usuario-contraseña. El controlador de dominio entonces forma una central repositoria de contraseñas que están enlazadas a nombres de usuarios, una contraseña por usuario, lo cual es más eficiente que mantener en cada máquina cliente centenares de contraseñas para cada recurso de red disponible.

En un dominio Windows, cuando un cliente solicita acceso a los recursos compartidos de un servidor, el servidor actúa y pregunta al controlador de dominio si ese usuario está autentificado. Si lo está, el servidor establecerá una conexión de sesión con los derechos de acceso correspondientes para ese servicio y usuario, si no lo está la conexión es denegada. Una vez que un usuario es autentificado por el controlador de dominio, una ficha especial de autentificación será retornada al cliente, de manera que el usuario no necesitará reloguearse a otros recursos en ese dominio. En éste punto, el usuario se considera logueado en el dominio.

El controlador de dominio que está actualmente activo sobre un dominio es denominado como el Controlador Primario de Dominio (PDC). Además pueden existir uno o más Controladores de Dominio de Seguridad (BDCs) en el dominio, los cuales actuarán en caso de que el controlador primario falle o se vuelva inaccesible. Los BDCs frecuentemente sincronizan sus datos SAM con el controlador primario de dominio, de manera que si este tiene que cumplir su función la realizaría de forma transparente sin provocar ningún tipo de impacto en los clientes. Los BDCs, sin embargo, sólo tienen copias de sólo lectura del SAM; pueden actualizar sus datos sólo mediante la sincronización con un PDC.(ROBERT ECKSTEIN 2001)

En este trabajo el autor hará uso de Samba como Controlador Primario de Dominio, este software en su versión 3.0.24-6 sólo puede actuar para procesos de autentificación. Debido a la privacidad del protocolo usado por Microsoft para sincronizar datos SAM, Samba actualmente no puede servir como controlador de dominio de seguridad. Con el desarrollo de versiones posteriores se espera que Samba pueda realizar esta funcionalidad.

Hasta aquí se trataron los temas relacionados con Software Libre y la importancia que tiene el mismo para Cuba. Se plasmaron los principales conceptos relacionados con el tema y se realizó un estudio sobre los diferentes servicios que están presentes en una red de datos, definiendo cuáles serán utilizados para realizar la integración de dichos servicios.

Capítulo2: Conociendo a LDAP.

Este capítulo está destinado a abordar los principales aspectos del servicio de directorio LDAP. Se hará referencia a su funcionamiento, sus ventajas, algunas características importantes, así como a otros aspectos que ayudarán a comprender al lector todo lo relacionado con este servicio, ofreciéndose algunos ejemplos que facilitarán la comprensión del contenido que aquí se trata.

2.1 Funcionamiento de LDAP.

El servicio de directorio LDAP se basa en un modelo cliente-servidor. Uno o más servidores LDAP contienen los datos que conforman el árbol de directorio LDAP, el cliente LDAP se conecta con el servidor LDAP y le hace una consulta. El servidor contesta con la respuesta correspondiente, o bien con una indicación de donde puede el cliente hallar más información. No importa con que servidor LDAP se conecte el cliente ya que siempre observará la misma vista del directorio.

2.2 Información en el directorio.

El modelo de información de LDAP está basado en entradas. Una entrada es una colección de atributos que tienen un único y global Nombre Distinguido (**DN**). El DN se utiliza para referirse a una entrada sin ambigüedades. Cada atributo de una entrada posee un tipo y uno o más valores. Los tipos son normalmente palabras nemotécnicas, como "**cn**" para nombre común, o "**mail**" para una dirección de correo. (GONZÁLEZ 2004)

La sintaxis de los atributos depende del tipo de atributo. Por ejemplo, un atributo **cn** puede contener el valor "Roberto González". Un atributo **mail** puede contener un valor "roberto@debian.uci.cu". El atributo *jpegPhoto* ha de contener una fotografía en formato JPEG.

2.3 ¿Cómo es referenciada la información?

En un directorio LDAP el nivel superior es la base, denominada Nombre Distinguido Base. Existen varios tipos de formato para denominar la base de un directorio. En este trabajo se utilizará el formato derivado de los componentes de dominio DNS, para especificar el DN base, el mismo está separado por componentes de dominio, **dc**. Actualmente es muy utilizado en Internet. A continuación se ilustrará un ejemplo aplicado a la Universidad de las Ciencia Informáticas (UCI).

Ejemplo1:

Dominio	**DN base**
uci.cu	dc=uci, dc=cu

Seguramente por debajo de la base querrás crear contenedores que separen los datos de una forma lógica. Por razones históricas la mayoría de los directorios los configuran como entradas **ou**, "Unidades Organizacionales", que en **X.500** eran utilizadas para indicar la organización funcional dentro de la institución, sedes, departamento, etcétera.

Actualmente las implementaciones de LDAP han mantenido la convención del nombre **ou**, pero separa las cosas por categorías amplias, ejemplo ou=grupos, ou=facultad, ou=usuarios. Después de estas categorías se pueden crear más entradas, estas lógicamente tendrán un nivel jerárquico menor. Ver Anexo 1

Todas las entradas almacenadas en un directorio LDAP tienen un único Nombre Distinguido. Este está compuesto por dos partes: el nombre de la propia entrada llamado *Nombre Relativo Distinguido* **(RDN)** y la concatenación de los nombres de las entradas que le anteceden. El **RDN** es la porción del DN que no está relacionada con la estructura del árbol de directorio.

La mayoría de los elementos que se almacenan en un directorio LDAP tendrán un nombre, y el nombre es almacenado en el atributo **cn** (Nombre Común). De esto se deriva que casi todos los objetos que se almacenarán en LDAP utilizarán su valor **cn** como base para su RDN.

En LDAP el RDN esta formado por un atributo, este va a depender del valor del objectclass que tenga la entrada.

Ejemplos de **objectclass**:

Person
PosixAccount
PosixGroup
ShadowAccount

Si el **objectClass** es **Person** el RDN será el **cn** de la entrada.
Si el **objectClass** es **posixAccount** el RDN será el **uid** (identificador de usuario).Ver Anexo 2

Atributos.
LDAP permite controlar que atributos son requeridos o permitidos en una entrada gracias al uso del atributo denominado **objectClass**

Los atributos requeridos en una entrada son aquellos que no pueden faltar en esta según el **objectClass** que tenga la misma. Los atributos permitidos en una entrada son aquellos que el **objectClass** de la misma tiene predeterminado según su tipo. A continuación se mostrarán ejemplos de **objectClass** con algunos de sus atributos.

En el **objectClass posixAccount** los atributos **uid** (identificador de usuario), **cn** (nombre común) y **sn** (segundo nombre) son requeridos mientras que el atributo **userPassword** (contraseña del usuario) y **telephoneNumber** (número de teléfono) son opcionales.

Ejemplo de una entrada con **objectClass posixAccount** en formato LDIF.

dn: uid=esalabarria,ou=usuarios,dc=uci,dc=cu
uid:esalabarria
cn: Erick
sn: Salabarria Aquino
loginShell: /bin/false
uidNumber: 1002
gidNumber: 1001
homeDirectory: /home/esalabarria
Objectclass: top
Objectclass: person
ObjectClass: posixAccount

En el **objectClass person**, se requieren los atributos **cn** y **sn**. Los atributos **description** (descripción), **telephoneNumber** (número de teléfono) y **userpassword** (contraseña del usuario) se permiten pero no son obligatorios.

Ejemplo de una entrada utilizando **objectClass person** en formato LDIF.

dn: cn=vmail,ou grupos,dc=uci,dc=cu
cn: vmail
sn: correo
loginShell: /bin/false
uidNumber: 1001
gidNumber: 1001
homeDirectory: /home/vmail
Objectclass: top
ObjectClass: person

Como se muestra en los ejemplos anteriores las entradas pueden pertenecer a más de un **objectClass**. La estructura de estos determina la lista total de atributos requeridos y permitidos para una entrada específica.

2.4 ¿Cómo se almacena la información?

En LDAP, las entradas están organizadas en una estructura jerárquica en árbol. Esto es un aspecto muy importante dentro de un directorio debido que nos proporciona ciertas ventajas a la hora de tratar los datos.

Si tenemos un árbol de directorio, como se muestra en el Anexo 3, quizás se desee dar acceso a los profesores que visiten el área docencia para poder poner y modificar las notas de los estudiantes y no a otras áreas.

También la jerarquía permite a la hora de consultar cierta información en el directorio optimizar el tiempo y así el ancho de banda, este es un recurso muy importante, de esta forma puedes darle un mejor uso a tus servicios en la red.

El Anexo 4 muestra un ejemplo de un árbol de directorio **LDAP** haciendo uso del **nombramiento tradicional**.
Tradicionalmente, esta estructura reflejaba los límites geográficos y organizacionales. Las entradas que representan países aparecen en la parte superior del árbol. Debajo de ellos, están las entradas que representan las provincias y las organizaciones nacionales. Debajo de estás, pueden estar las entradas que representan las unidades organizacionales, empleados, grupos, documentos o todo aquello que pueda imaginarse.

El árbol también se puede organizar basándose en los **nombres de dominio de Internet**. Este tipo de nombramiento se está utilizando mucho, ya que permite localizar un servicio de directorio haciendo uso de los **DNS**. El Anexo 5 muestra un ejemplo de un directorio **LDAP** que hace uso de los nombres basados en dominios.

2.5 Tratamiento de la información.

LDAP define operaciones para interrogar y actualizar el directorio, añadir y borrar entradas, modificar una entrada existente y cambiar el nombre de una entrada. Sin embargo la mayor parte del tiempo LDAP se utiliza para buscar información almacenada en el directorio. Las operaciones de búsqueda de LDAP permiten buscar entradas que concuerdan con algún criterio especificado por un filtro de búsqueda. La información puede ser solicitada desde cada entrada que concuerda con dicho criterio.

2.6 ¿Cómo se protege la información de los accesos no autorizados?

Algunos servicios de directorio no proveen protección, permitiendo a cualquier persona acceder a la información. LDAP provee un mecanismo de autenticación para los clientes, o la confirmación de identidad en un servidor de directorio, facilitando el camino para un control de acceso que proteja la información que el servidor posee. LDAP también soporta los servicios de privacidad e integridad.(GONZÁLEZ 2004)

2.7 Ventajas de los directorios LDAP.

- La popularidad de LDAP es la unión de varios factores. A continuación se muestran algunas razones básicas, demostrando la potencialidad de este servicio de directorio.

- Quizás la mayor ventaja de LDAP es que permite acceder al directorio LDAP desde casi cualquier plataforma.

- Es también fácil personalizar tus aplicaciones para añadirles soporte LDAP.

- El protocolo LDAP es utilizable por distintas plataformas y esta basado en estándares, de ese modo las aplicaciones no necesitan preocuparse por el tipo de servidor en que se hospeda el directorio. LDAP esta encontrando mucha más amplia aceptación a causa de esto como estándar de Internet.

- Es muy rápido en la lectura de registros.

- A diferencia de las bases de datos relacionales, no tienes que pagar por cada conexión de software cliente o por licencia.

- La mayoría de los servidores LDAP son simples de instalar, fácilmente mantenibles y optimizables.

- Los servidores LDAP pueden replicar algunos de sus datos o todos a través de métodos de envío y recepción, lo que permite enviar datos a oficinas remotas, incrementar la seguridad y demás. La tecnología de replicación está incorporada y es fácil de configurar.

- LDAP permite delegar con seguridad la lectura y modificación basada en autorizaciones según las necesidades utilizando **ACLs** (Lista de Control de Acceso).

- Las ACLs pueden controlar el acceso dependiendo de quien está solicitando los datos, que datos son solicitados, dónde se encuentran los datos almacenados, y otros aspectos del registro que está siendo solicitado. Todo esto es hecho directamente a través del directorio LDAP, así que no es necesario preocuparse de hacer comprobaciones de seguridad en el nivel de aplicación de usuario.

- LDAP es particularmente utilizable para almacenar información que se desee leer desde muchas localizaciones, pero que no sea actualizada frecuentemente. (GONZÁLEZ 2004)

2.8 Usos prácticos de LDAP.

Según las características de este servicio de directorio sus usos más comunes son:

1. Servidores de certificados públicos y llaves de seguridad.
2. Libretas de direcciones compartidas.
3. Sistemas de alojamiento de páginas web y FTP, con el repositorio de datos de usuario compartido.
4. Autenticación para la personalización de aplicaciones.
5. Sistemas de correo electrónico. Grandes sistemas formados por más de un servidor que accedan a un repositorio de datos común.
6. Directorios de información.(*Manual de OpenLDAP en español* 2004)

2.9 Utilización de LDAP.

Si se lee detenidamente las ventajas mencionadas anteriormente no será difícil darse cuenta cuando es recomendable usar un servicio de directorio LDAP. A continuación se darán algunas razones mas especificas.

Utiliza LDAP si:

- Se desea que los datos estén disponibles a través de varias plataformas
- Se necesita acceso a estos datos desde diversos ordenadores o aplicaciones.
- Se almacena información que se desee leer desde muchas localizaciones, pero que no sea actualizada frecuentemente.
- Se desea controlar el acceso dependiendo de quien está solicitando los datos, que datos son solicitados y dónde se encuentran almacenados los datos.

2.10 Características de la versión 3 de LDAP (LDAPv3).

LDAPv3 fue desarrollado en los años 90 para reemplazar a LDAPv2, la versión 3 incorpora las siguientes características a LDAP:

- Autenticación fuerte haciendo uso de SASL
- Protección de integridad y confidencialidad haciendo uso de TLS (SSL)
- Descubrimiento de esquemas
- Internacionalización gracias al uso de Unicode.
- Extensibilidad (controles, operaciones extendidas)

Como LDAPv2 difiere significativamente de LDAPv3, la interacción entre ambas versiones puede ser un poco problemática, por lo que no es recomendable utilizar las dos versiones al unísono, en la implementación de OpenLDAP viene deshabilitado por defecto. (GONZÁLEZ 2004)

2.11 Slapd.

Slapd es la versión de OpenLDAP para la distribución Debian, esta es la utilizada en este trabajo.
Algunas de las características más interesantes de slapd son:

- Implementa la versión 3 de LDAP. Slapd soporta LDAP sobre IPv4, IPv6 y Unix IPC.

- Tiene soporte de autentificación fuerte gracias al uso de SASL. La implementación SASL de slapd hace uso del software Cyrus SASL el cual soporta un gran número de mecanismos de autentificación, como: DIGEST-MD5, EXTERNAL, y GSSAPI.

- Provee protecciones de privacidad e integridad gracias al uso de TLS o (SSL). La implementación TLS de slapd hace uso del software OpenSSL.

- Provee facilidades de control de acceso muy potentes, permitiéndole controlar el acceso a la información de sus bases de datos. Puede controlar el acceso a las entradas basándose en la información de autorización de LDAP, en la dirección IP, en los nombres de dominio y otros criterios.
- Viene con una serie de backends para diferentes bases de datos

- Puede ser configurado para servir a múltiples bases de datos al mismo tiempo. Esto significa que un único servidor slapd puede responder a peticiones de diferentes porciones lógicas del árbol de LDAP, haciendo uso del mismo o distintos *backends* de bases de datos.

- Si necesita más personalización, slapd le permite escribir sus propios módulos fácilmente. Slapd consiste en dos partes diferentes: un frontend que maneja las comunicaciones protocolares con los clientes LDAP y módulos que manejan tareas específicas como las operaciones con las bases de datos.
- Puede ser configurado como un servicio Proxy de caché LDAP.

- Altamente configurable a través de un único archivo de configuración, que permite modificar todo aquello que se necesite cambiar.

2.11 Slurpd.

Slurpd es un demonio que, con la ayuda de slapd, provee el servicio de replicación. Es el responsable de distribuir los cambios realizados en la base de datos slapd principal hacia las distintas réplicas slapd. Este demonio libera a slapd de preocuparse por el estado de las réplicas. Slurpd maneja automáticamente el reenvío de las peticiones fallidas. Slapd y slurpd se comunican a través de un simple archivo de texto, que es utilizado para almacenar los cambios ocurridos.(GONZÁLEZ 2004)

En este capítulo se trataron los principales aspectos del servicio de directorio LDAP. Se hizo referencia a su funcionamiento, sus ventajas, algunas características importantes.

Capitulo 3: Integración de Servicios.

Este capítulo está dedicado a la instalación y configuración de los servicios telemáticos presentes en una red. La integración de los mismos se hará sobre un directorio LDAP utilizando Software Libre, dicho directorio almacena la información de todos los usuarios que harán uso de los servicios brindados en la red. Al finalizar este capítulo el sistema debe estar listo para autenticar usuarios contra el servicio de directorio, según se avance en la instalación se agregarán funcionalidades necesarias que permitirán cumplir el objetivo fundamental del trabajo. Todos los procedimientos expuestos a continuación fueron hechos sobre la distribución **Debian GNU/Linux** versión **4**.

3.1 Instalación y configuración de OpenLDAP.

Para comenzar con la instalación de OpenLDAP instale los siguientes paquetes **slapd** y **ldap-utils**.

Nota: Para ver la información relativa a los paquetes que se instalarán ejecute.

apt-cache show slapd ldap-utils

Paso para la instalación

apt-get install slapd ldap-utils

Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Se instalarán los siguientes paquetes NUEVOS:
ldap-utils slapd
0 actualizados, 2 se instalarán, 0 para eliminar y 1 no actualizados.
Se necesita descargar 0B/1042kB de archivos.
Se utilizarán 2884kB de espacio de disco adicional después de desempaquetar.
Preconfiguring packages...
Seleccionando el paquete ldap-utils previamente no seleccionado.
(Leyendo la base de datos...
252690 ficheros y directorios instalados actualmente.)
Desempaquetando ldap-utils (de.../ldap-utils_2.3.30-5_i386.deb)...
Seleccionando el paquete slapd previamente no seleccionado.

Desempaquetando slapd (de.../slapd_2.3.30-5_i386.deb)...
Configurando ldap-utils (2.3.30-5)...
Configurando slapd (2.3.30-5)...
Creating initial slapd configuration... done
Creating initial LDAP directory... done
Starting OpenLDAP: slapd.

Para reconfigurar el servidor ejecute el siguiente comando.

dpkg-reconfigure --priority=low slapd

3.1.1 Comprobaciones iniciales de la instalación

Hasta este momento ya se dispone de un servidor instalado y ejecutándose aunque todavía no está ajustado a las necesidades requeridas en este epígrafe. A continuación se verificará que el demonio slapd está ejecutándose correctamente.

En la imagen se mostrará el procedimiento para comprobar que **slapd** esté en la lista de procesos actuales del sistema.

```
debian-euler:/etc/default# ps auxf | /bin/grep slapd
root      2568  0.0  1.1   2028   740 pts/1    S+   13:34   0:00                  \_ /bin/grep slapd
root      2558  0.0  4.4  14520  2732 ?        Ssl  13:30   0:00 /usr/sbin/slapd
debian-euler:/etc/default#
```

Figura 6: Proceso slapd.

Seguidamente se comprobará si slapd está escuchando de la red.

```
debian-euler:/etc/default# netstat -puta
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 *:ldap                  *:*                     LISTEN      2558/slapd
tcp        0      0 localhost.localdoma:907 *:*                     LISTEN      2427/famd
tcp        0      0 *:sunrpc                *:*                     LISTEN      1884/portmap
tcp        0      0 debian.uci.cu:domain    *:*                     LISTEN      2185/named
tcp        0      0 localhost.locald:domain *:*                     LISTEN      2185/named
tcp        0      0 *:smtp                  *:*                     LISTEN      2375/master
tcp        0      0 localhost.localdoma:953 *:*                     LISTEN      2185/named
tcp6       0      0 *:ldap                  *:*                     LISTEN      2558/slapd
tcp6       0      0 *:pop3                  *:*                     LISTEN      2306/couriertcpd
tcp6       0      0 *:imap2                 *:*                     LISTEN      2297/couriertcpd
tcp6       0      0 *:www                   *:*                     LISTEN      2464/apache2
```

Figura 7: Escuchando de la red.

La Figura 7 muestra varios procesos que se encuentran escuchando de la red dentro de estos está slapd

Ahora se verifica que el servidor permite conexión, para esto se hará una búsqueda sencilla con el comando **ldapsearch**.

```
debian-euler:/home/erick# ldapsearch -x -b '' -s base '(objectclass=*)' namingCo
ntexts
# extended LDIF
#
# LDAPv3
# base <> with scope baseObject
# filter: (objectclass=*)
# requesting: namingContexts
#

#
dn:
namingContexts: dc=debian,dc=uci,dc=cu

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
debian-euler:/home/erick#
```

Figura 8: Búsqueda sencilla

Para que el servidor se encuentre en funcionamiento las comprobaciones realizadas anteriormente deben dar los resultados esperados, o sea que el servidor esté en lista de procesos que se están ejecutando en el sistema, que se encuentre escuchando de la red, y que permita hacer búsquedas.

3.1.2 Especificación de las interfaces donde escuchar.

Para especificar las interfaces de red donde escuchará el servidor así como el protocolo que utilizará se debe modificar el valor de la variable **SLAPD_SERVICES** en el fichero **/etc/default/slapd.**

Para este documento queda de la siguiente forma.

SLAPD_SERVICES=" ldap://debian.uci.cu:389/ ldaps: // debian.uci.cu:636/"

El protocolo **ldap** específica las interfaces y los puertos donde escuchará **slapd,** con la particularidad que con el protocolo **ldaps** toda la comunicación entre el cliente y el servidor será encriptada.

En estos momentos ya el servidor se encuentra en condiciones de iniciar con las nuevas configuraciones.

/etc/init.d/slapd restart

Verificando que el servidor se esté ejecutando con el usuario y grupo así como con las interfaces y protocolos de red especificados.

.

```
debian-euler:/etc/default# ps auxf | /bin/grep slapd
root      7280  0.0  1.1   2024   740 pts/3    S+   00:13   0:00                          \_ /bin/grep slapd
openldap  7274  0.1  4.8  16780  2976 ?        Ssl  00:12   0:00 /usr/sbin/slapd -h ldap://debian.uci.cu:389/  ldaps://debian
.uci.cu:636/ -g openldap -u openldap
debian-euler:/etc/default#
```

Figura 9: Procesos actuales

Verificando que el servidor esté escuchando con los protocolos y interfaces de red específicos.

```
debian-euler:/etc/default# netstat -puta
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 localhost:929           *:*                     LISTEN      2449/famd
tcp        0      0 debian.uci.cu:ldap      *:*                     LISTEN      7274/slapd
tcp        0      0 *:sunrpc                *:*                     LISTEN      1903/portmap
tcp        0      0 debian.uci.cu:domain    *:*                     LISTEN      7001/named
tcp        0      0 localhost:domain        *:*                     LISTEN      7001/named
tcp        0      0 localhost:953           *:*                     LISTEN      7001/named
tcp        0      0 *:smtp                  *:*                     LISTEN      2394/master
tcp        0      0 debian.uci.cu:ldaps     *:*                     LISTEN      7274/slapd
```

Figura 10: Escuchando de la red2.

El fichero /etc/ldap/ldap.conf se encuentra la configuración del cliente. La única modificación que se realizará sobre este archivo es un cambio de permisos, de forma que todo el mundo tenga permisos de lectura.

chmod -v 644 /etc/ldap/ldap.conf

Otro fichero importante es /etc/ldap/slapd.conf, la única modificación que se ha de realizar sobre este archivo, de momento, es un cambio de permisos, de forma que solo el propietario tenga permisos de lectura y escritura.

chmod -v 600 /etc/ldap/slapd.conf

De esta forma concluye la parte de la instalación y configuración del servidor slapd, más adelante se tratarán otros aspectos de forma que el directorio tenga más funcionalidades.

3.2 Autenticación de usuarios a través de OpenLDAP.

PAM (Pluggable Authentication Module) es una biblioteca de autentificación genérica que cualquier aplicación puede utilizar para validar usuarios, utilizando por debajo múltiples esquemas de autenticación alternativos como ficheros locales, LDAP, etc. Esta biblioteca es utilizada por el proceso de "login" para averiguar si las credenciales tecleadas por el usuario son correctas.

En esta sección se configura una máquina para que sus usuarios se autentiquen a través de un servidor LDAP. Para logra esto se han de modificar dos aspectos del comportamiento del sistema.

1. El mapeado entre los números de identificación de los usuarios y sus nombres o la localización del directorio home. La búsqueda de este tipo de información es responsabilidad del servicio de nombres, el archivo de configuración de este es **/etc/nsswitch.conf.**

2. La autenticación es responsabilidad del subsistema PAM, cuya configuración se hace sobre el directorio **/etc/pam.d/**

Para autenticar a los usuarios a través de un servidor LDAP, es necesario instalar el paquete **libpam-ldap**.

Instalación

apt-get install libpam-ldap

Para configurar algunos aspectos del paquete libpam-ldap ejecute:

dpkg-reconfigure --priority=low libpam-ldap

3.2.1 Configuración de PAM

PAM permite configurar el método de autenticación que van a utilizar las aplicaciones que hagan uso de él. Gracias a esto se pueden añadir fácilmente distintas opciones de autenticación, como el uso de una base de datos LDAP.

Archivos que se deben modificar para lograr la autenticación a través de LDAP.

/etc/pam.d/common-account

account required pam_unix.so
account sufficient pam_ldap.so

/etc/pam.d/common-auth

auth sufficient pam_unix.so
auth sufficient pam_ldap.so try_first_pass
auth required pam_env.so
auth required pam_securetty.so
auth required pam_unix_auth.so
auth required pam_warn.so
auth required pam_deny.so

/etc/pam.d/common-session

session required pam_mkhomedir.so skel=/etc/skel/ umask=0022
session required pam_limits.so
session required pam_unix.so
session optional pam_ldap.so

La primera línea esta destinada a crear los home de los usuarios, cuando se añaden al directorio LDAP.

/etc/pam.d/common-passwd

password required pam_cracklib.so retry=3 minlen=8 difok=4
password sufficient pam_unix.so use_authtok md5 shadow
password sufficient pam_ldap.so use_authtok
password required pam_warn.so
password required pam_deny.so

Se debe tener instalada la librería libpam-cracklib.
apt-get install libpam-cracklib

3.2.2 Instalación y configuración nss-ldap.

Las librerías **nss-ldap** permiten a un servidor LDAP actuar como un servidor de nombres, o sea que dicho servidor 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 del sistema.

El paquete que provee esta funcionalidad en Debian GNU/Linux es **libnss-ldap.**

Instalación

apt-get install libnss-ldap

El fichero **/etc/nsswitch.conf** es el fichero de configuración de las bases de datos del sistema y del sistema de conmutación de los Servicios de Nombres, indica el orden y el procedimiento a seguir para la búsqueda de la información requerida.

Para configurar este archivo primero se especifica la base de datos sujeta a la búsqueda y después el procedimiento que se va a emplear para realizar una búsqueda sobre la misma. En este trabajo el procedimiento de búsqueda hará uso en algún momento de LDAP.

Ejemplo de un fichero **nsswitch.conf:**

passwd: compat ldap
group: compat ldap
shadow: compat ldap
hosts: files ldap dns

En este ejemplo primero se busca en los archivos locales, después en LDAP y en un caso en el DNS. Es muy importante que no se elimine el uso de los ficheros locales ya que algunos usuarios y grupos de usuarios permanecerán de forma local. Si el sistema no utiliza las entradas locales (compat) y el servidor LDAP se cae, nadie ni siquiera root, podrá entrar al sistema.

Se puede comprobar que todo funciona ejecutando el comando **getent** seguido de la base de datos a consultar

Ejemplo:

getent passwd

Nota: Esto debe mostrar la base de datos consultada en este caso passwd

3.2.3 Comprobando la autenticación contra LDAP

Para comprobar que el sistema está listo para autenticar usuarios contra LDAP se utilizará el comando **pamtest** este lo proporciona el paquete **libpam-dotfile**, **pamtest** acepta dos parámetros: el primero es el nombre del servicio al cual se va a conectar para realizar la autenticación, el segundo es el nombre del usuario que se va a autenticar sobre dicho servicio.

Ejemplo: ***pamtest passwd erick***
En estos momentos el sistema ya se encuentra preparado para autenticar a los usuarios a través de LDAP.

3.3 Uso de la conexión segura (OpenSSL)

En esta sección se aborda el tema de cómo añadir una capa de seguridad a la comunicación cliente ldap-servidor ldap. Para esto es necesario crear el certificado para el servidor LDAP, así como añadir algunas líneas al fichero de configuración del mismo. Primeramente se procede a instalar **OpenSSL,** esta implementación libre de los protocolos **SSL** y **TLS** nos proporciona las herramientas necesarias para generar el certificado y la llave para el servidor.

Instalación de OpenSSL

apt-get install openssl

Después se debe crear un directorio dentro de **/etc/ldap/** para alojar en él la información relacionada con el certificado.

mkdir /etc/ldap/ssl

Nos paramos dentro del directorio creado
cd /etc/ldap/ssl

3.3.1 Creando el certificado y la llave

Durante el proceso de generación del certificado nos aparecerán varias preguntas, relacionadas con el lugar físico en el que está el servidor, la organización a la que pertenece, el correo electrónico del responsable del mismo, etc. Las mismas deben ser respondidas de acuerdo a las particularidades de cada servidor. Para este caso queda de la siguiente forma

openssl req -new -x509 -nodes -out servercrt.pem -keyout serverkey.pem -days 365

Country Name (2 letter code) [AU]: ***CU***
State or Province Name (full name) [Some-State]: ***UCI***
Locality Name (eg, city) []: ***UCI***
Organization Name (eg, company) [Internet Widgits Pty Ltd]: ***UCI***

Organizational Unit Name (eg, section) []: ***UCI***

Common Name (eg, YOUR name) []: ***ldap. debian.uci.cu***

*Email Address []:****erick@debian.uci.cu***

3.3.2 Configurar el servidor ldap

Se deben agregar las siguientes líneas en el fichero de configuración del servidor LDAP.

TLSCipherSuite HIGH: MEDIUM: +SSLv2
TLSCertificateFile /etc/ldap/ssl/servercrt.pem
TLSCertificateKeyFile /etc/ldap/ssl/serverkey.pem
TLSVerifyClient never

3.3.3 Configurar samba

Para que samba utilice ssl en la comunicación con el servidor ldap es muy sencillo, tan sólo hay que añadir ***ldap ssl = On*** en la sección global de samba.

3.3.4 Configurar los clientes ldap

Para que los clientes utilicen un canal seguro para comunicarse con el servidor LDAP tenemos que añadir tanto en el fichero de **libpam-ldap** como en el de **libnss-ldap** las siguientes líneas.

/etc/pam_ldap.conf
ssl start_tls
tls_checkpeer no

/etc/libnss_ldap.conf
ssl start_tls
tls_checkpeer no

3.3.5 Configuración de la herramienta smbldap-tools

En el fichero de configuración de la herramienta **smbldap-tools** se deben especificar los siguientes datos para que la comunicación entre esta y el servidor sea por un canal seguro.

ldapTLS="1"
verify="require"
cafile="/etc/ldap/ssl/servercrt.pem"
clientcert="/etc/ldap/ssl/servercrt.pem"
clientkey="/etc/ldap/ssl/serverkey.pem"

Al culminar esta sección se ha logrado que toda la comunicación entre el servidor LDAP y los clientes se realice de forma segura. De esta forma se consigue que la información que se trasmite mantenga su integridad.

3.4 Instalación y Configuración DNS

En este subepígrafe se realizará la instalación y configuración del servidor DNS utilizando BIND 9.

Instalación

apt-get install bind9 dns-utils

En este trabajo solo se utilizó en Servidor Primario, pero es recomendable utilizar también un Servidor Secundario o Esclavo. A continuación se muestra la declaración de las zonas utilizadas, así como los ficheros de cada una de ellas.

```
 zone “debian.uci.cu"{
       type master;
       file “/etc/bind/db.masterdebian";
};


zone “13.33.10.in-addr.arpa"{
     type master;
     file "/etc/bind/db.13.33.10";
};
```

Fichero de la zona directa

```
@       IN      SOA     wiki02.debian.uci.cu.       wiki02.debian.uci.cu. (
                            1           ; Número de serie
                            604800      ; Tiempo de refresco
                            86400       ; Tiempo entre reintentos de consulta
                            2419200     ; Tiempo de expiración de zona
                            604800)     ; TTL Negativo
;

@               IN      NS      wiki02.debian.uci.cu.
@               IN      A       10.33.13.31
wiki02          IN      A       10.33.13.31
blinblineo      IN      A       10.8.112.202
ldap            IN      CNAME   wiki02
ftp             IN      CNAME   wiki02
prueba          IN      A       10.33.13.190
```

Fichero de la zona inversa

```
@       IN      SOA     wiki02.debian.uci.cu.       wiki02.debian.uci.cu. (
                          1           ; Número de serie
                          604800      ; Tiempo de refresco
                          86400       ; Tiempo entre reintentos de consulta
                          2419200     ; Tiempo de expiración de zona
                         604800)      ; TTL Negativo
;

@           IN      NS      wiki02.debian.uci.cu
1           IN      PTR     wiki02.debian.uci.cu
31          IN      PTR     wiki02.debian.uci.cu
202         IN      PTR     blinblineo
190          IN     PTR      prueba
```

A (Address) - Registro de dirección que resuelve un nombre de un anfitrión hacia una dirección IPv4 de 32 bits.

CNAME (Canonical Name) - Registro de nombre canónico que hace que un nombre sea alias de otro.

PTR (Pointer) - Registro de apuntador que resuelve direcciones IPv4 hacia el nombre anfitriones. Es decir, hace lo contrario al registro A. Se utiliza en zonas de Resolución Inversa.

NS (Name Server) - Registro de servidor de nombres que sirve para definir una lista de servidores de nombres con autoridad para un dominio.

SOA (Start of Authority) - Registro de inicio de autoridad que especifica el Servidor DNS Maestro (o Primario) que proporcionará la información con autoridad acerca de un dominio de Internet, dirección de correo electrónico del administrador, número de serie del dominio y parámetros de tiempo para la zona

Una opción importante a tener en cuenta es la de **fordwarders** (redireccionadores), que se usará para suministrar al servidor DNS las direcciones IP de los redireccionadores encargados de consultar ciertas direcciones a otros servidores DNS, cuando éstas no estén disponibles de forma local. Esta opción esta en el fichero **named.conf.options** que se encuentra en el directorio **/etc/bind**. Para este caso quedó de la siguiente manera.

```
forwarders {
        10.0.0.3;
        10.0.0.4;
    };
```

3.5 Instalación y configuración de Apache.

A continuación se realizará el proceso de instalación del Servidor Web Apache en su versión 2.2.3-4.

Instalación:

apt-get install apache2

Sus ficheros principales son

/etc/apache2/apache2.conf configuración principal de Apache

/etc/apache2/ports.conf puertos e interfaces donde escuchar peticiones http

Aspectos importantes en **/etc/apache2/apache2.conf**

- **ServerRoot** <path> Especifica el directorio de máximo nivel del Servidor.(no de las paginas Web)
- **PidFile** <path> Fija la ruta del archivo PID.
- **DocumentRoot** <path> Indica el directorio de máximo nivel donde estarán las páginas Web.

3.5.1 Directivas en contenedores de directorios

Hay que tener en cuenta que el alcance de estas directivas está limitado sólo al directorio donde están, incluidos los subdirectorios. Sólo se deberán usar las directivas permitidas para el contexto de directorios.

Ejemplo1:

```
<Directory “/var/www/datos”>
   Option indexes FollowSymLinks
   AllowOverride None
   DirectoryIndex index.html welcome.html
   Order Allow, Deny
   Allow from all
   Deny 10.8.112.0   255.255.255.0
</Directory>
```

Estas directivas son muy importantes, mediante ellas se puede tener controlado el acceso al directorio que este compartido, esto se hace mediante diferentes criterios por ejemplo de que red o subred se puede acceder, quienes pueden acceder , que privilegios tienen los que pueden acceder, entre otras cosas. En el ejemplo anterior se puede acceder desde todas partes excepto de la subred 10.8.112.0

Otras directivas.

ServerTokens Minimal | OS | Full | ProductOnly Dependiendo del valor que tenga asignado le dice al cliente que servidor esta ejecutando. Esto no es recomendable por razones de seguridad.

ServerSignature Si está en **On** le muestra al usuario diferentes características del servidor, es recomendable que esté en **Off**.

3.5.2 Autenticación de Apache contra LDAP

En esta sección se aborda el tema de la autenticación del servidor Apache contra el directorio LDAP, para esto es necesario instalar el módulo de Apache que permite esta funcionalidad (***libapache-mod-ldap),*** así como activar el mismo.

Ejemplo de un virtualhost con autenticación contra LDAP.

```
NameVirtualHost 10.33.13.31:80
  <VirtualHost 10.33.13.31:80>
      ServerName prueba.debian.uci.cu
      ServerAdmin root@debian.uci.cu
      DocumentRoot /var/www/prueba

     <Directory /var/www/prueba>
          Options Indexes FollowSymLinks
          AllowOverride None
```

```
            Order allow, deny
            Allow from all
            AuthType Basic
            AuthName "Zona Segura"
            AuthBasicProvider ldap
            AuthLDAPURL ldap://10.33.13.31:389/dc=debian,dc=uci,dc=cu
            require ldap-user leo erick
      </Directory>
</VirtualHost>
```

Lo que a continuación se muestra es un ejemplo de un hosts virtual con la particularidad que está sobre ssl. Para esto se debe haber creado previamente el certificado para dicho hosts virtual. El proceso de creación del certificado se muestra en el Manual de Instalación.

```
NameVirtualHost 10.33.13.31:443
  <VirtualHost 10.33.13.31:443>
      ServerName prueba.debian.uci.cu
      ServerAdmin root@debian.uci.cu
      DocumentRoot /var/www/prueba
      SSLEngine On
      SSLCertificateFile /etc/apache2/ssl/server.pem
      SSLProtocol all
      SSLCipherSuite HIGH: MEDIUM

      <Directory /var/www/prueba>
            Options Indexes FollowSymLinks
            AllowOverride None
            Order allow, deny
            Allow from all
```

```
        AuthType Basic
        AuthName "Zona Segura"
        AuthBasicProvider ldap
        AuthLDAPURL ldap://10.33.13.31:389/dc=debian,dc=uci,dc=cu
        require ldap-user leo erick
    </Directory>
</VirtualHost>
```

Se debe tener en cuenta que un certificado sirve para una sola dirección **IP,** y una dirección **IP** sólo puede tener un certificado. Es decir, que si tenemos dos virtualhosts que apuntan a una misma dirección **IP** y creamos un certificado para cada virtualhosts, no serviría, sólo valida el primer certificado, el segundo se ignora. Para tener un certificado en cada virtualhosts, deben apuntar a diferentes direcciones **IP.**

3.6 Instalación y Configuración de FTP.

En este subepígrafe se configura **proftpd** para que autentique sus usuarios contra un directorio **LDAP.** Para instalar el servidor proftpd ejecute.

apt - get install proftpd-ldap

Estas líneas se deben agregar al fichero de configuración **/etc/proftpd/proftpd.conf** para que el servidor FTP autentique contra el servidor LDAP.

AuthOrder mod_ldap.c
AuthPAM on
LDAPServer 10.33.13.31
LDAPDoAuth on "ou=usuarios, dc=debian, dc=uci, dc=cu"

Para iniciar el servidor, detenerlo y reiniciarlo se hace de la siguiente manera:
/etc/init.d/proftpd start | stop | restart

De esta forma ya el servidor FTP está preparado para autenticar los usuarios contra el servidor LDAP.

3.7 Instalación y Configuración de Samba.

Se instalará el servidor **Samba** para que actúe como PDC de la red en la que esté presente. La información de las cuentas de los usuarios se almacenará en un directorio LDAP, además permitirá compartir directorios para los usuarios

Una vez incluida la estructura en el directorio LDAP, los usuarios que ahí se almacenen tendrán la posibilidad de autenticarse en cualquier sistema GNU/Linux o Windows que haga uso del servidor LDAP para la autenticación de usuarios. La particularidad es que tendrán la misma cuenta de acceso para todos sistemas, tanto en GNU/Linux como en Windows, en toda la red.

Para ver la información relativa al paquete samba que se va a instalar ejecute el siguiente comando

apt-cache show samba

Unas de las dependencias de samba es el paquete ***samba-common*** aun así será tarea del administrador la elección de su instalación.

apt-get install samba samba-common

Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Se instalarán los siguientes paquetes extras:
samba-common
Se instalarán los siguientes paquetes NUEVOS:
samba samba-common
0 actualizados, 2 se instalarán, 0 para eliminar y 3 no actualizados.

Se necesita descargar 0B/3988kB de archivos.

Se utilizarán 9839kB de espacio de disco adicional después de desempaquetar.

¿Desea continuar? [S/n] S

Hay dos paquetes importantes para un cliente Samba: **smbclient** y **smbfs**

Instalación

apt-get install smbclient smbfs

Herramientas suministradas por ***smbclient*** y ***smbfs.***

dpkg -L smbclient | /bin/grep bin

/usr/bin
/usr/bin/smbclient
/usr/bin/smbtar
/usr/bin/rpcclient
/usr/bin/smbspool
/usr/bin/smbtree
/usr/bin/smbcacls
/usr/bin/smbcquotas

dpkg -L smbfs | /bin/grep bin
/sbin
/usr/bin
/usr/bin/smbmount
/usr/bin/smbumount
/usr/bin/smbmnt
/sbin/mount.smbfs
/sbin/mount.smb

3.7.1 Ajustes en la configuración de OpenLDAP.

Antes de continuar con la configuración de Samba, es necesario realizar algunas modificaciones y ajustes en la configuración de OpenLDAP para que soporte las características de **Samba**.

Para esto hay que copiar el esquema de **Samba** al directorio de esquemas de OpenLDAP. El esquema se encuentra en el paquete samba-doc por lo que hay que instalarlo.
apt-get install samba-doc

Copiando del esquema de **Samba** al directorio de esquemas de OpenLDAP.
cp -v /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz /etc/ldap/schema

Descompactando el esquema
gunzip -v /etc/ldap/schema/samba.schema.gz

Cambiando el propietario y el grupo
chown -v openldap.openldap /etc/ldap/schema/samba.schema

Cambiando los permisos.
chmod -v 644 /etc/ldap/schema/samba.schema

Sólo queda añadir el nuevo esquema en el archivo de configuración de **slapd** y reiniciar el demonio. Para ello se debe editar el archivo **/etc/ldap/slapd.conf.**

include /etc/ldap/schema/samba.schema

Reiniciando el servidor **slapd.**

/etc/init.d/slapd restart

3.7.2 Configuración de Samba

La configuración de **Samba** se almacena en el archivo **smb.conf**, que en el sistema Debian GNU/Linux se encuentra en el directorio /etc/samba/

Un archivo de configuración de **Samba** como controlador de dominio podría quedar de la siguiente forma.

[Global]
Workgroup = Debian
NetBIOS name = wiki02
server string = SAMBA-LDAP PDC server

[Global]- Autenticación
security = user
encrypt passwords = true
passdb backend = ldapsam:ldap://debian.uci.cu
guest account = guest
invalid users = root
unix password sync = yes
passwd program = /usr/sbin/smbldap-passwd -o %u

passwd chat = *Enter\snew\sUNIX\spassword:*%n\n*Retype\snew\sUNIX\spassword:* %n\n
add user script = /usr/sbin/smbldap-useradd -m "%u"
delete user script = /usr/sbin/smbldap-userdel "%u"
add machine script = /usr/sbin/smbldap-useradd -t 0 -w "%u"
add group script = /usr/sbin/smbldap-groupadd -p "%g"
delete group script = /usr/sbin/smbldap-groupdel "%g"
add user to group script = /usr/sbin/smbldap-groupmod -m "%u" "%g"
delete user from group script = /usr/sbin/smbldap-groupmod -x "%u" "%g"
set primary group script = /usr/sbin/smbldap-usermod -g '%g' '%u'

[Global] - LDAP

ldap admin dn = cn=admin,dc=debian,dc=uci,dc=cu
; ldap server = debian.uci.cu
; ldap port = 389
ldap ssl = off
dap delete dn = no
ldap filter = (&(uid=%u)(objectclass=sambaSamAccount))
ldap suffix = ou=usuarios,dc=debian,dc=uci,dc=cu
ldap user suffix = ou=usuarios
ldap group suffix = ou=grupos
ldap machine suffix = ou=maquinas

[Global] - Impresión

load printers = yes
printing = cups printcap name = cups
printer admin = @domainadmins

[Global] - Controlador de dominio

os level = 80
preferred master = yes
domain master = yes
local master = yes
domain logons = yes
logon path = \\%L\profiles\%u
logon drive = H:
logon home = \\%L\%u\.profile
logon script =
; domain admin group = @domainadmins

[Global] - Misceláneo

socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
idmap uid = 10000-20000
idmap gid = 10000-20000
template shell = /bin/bash

[Homes] - Directorios personales

browseable = yes
writeable = yes
create mask = 0700
directory mask = 0700

Netlogon

El recurso compartido *NETLOGON* juega un papel fundamental en el soporte de inicio de sesión en un dominio y Miembro de Dominio. Este recurso compartido se provee en todos los Controladores de Dominio de Microsoft. Se utiliza para proveer de scripts de inicio de sesión.

[Netlogon]

path = /home/samba/netlogon
writeable = no
write list = @domainadmins

[Profiles] - Perfiles móviles

path = /home/samba/profiles
writeable = yes
browseable = no
create mask = 0600
directory mask = 0700

[Printers] – Impresoras

browseable = no
path = /tmp
printable = yes
guest ok = no
writable = no
create mask = 0700

[tmp] - Directorio temporal

comment = Temporal
writeable = yes
path = /tmp
guest ok = no

[cdrom] - CDROM

comment = Samba server's CD-ROM
writable = no
locking = no
path = /cdrom
guest ok = yes

3.7.3 Creando la estructura de directorios en el home

En el fichero de configuración de **Samba** se definieron varios directorios destinados a realizar diferentes tareas, dichos directorios deben de estar en el sistema, sino existen, hay que crearlos.

Creación de los directorios.

mkdir -vpm 755 /home/samba/

mkdir -vpm 755 /home/samba/netlogon /home/samba/users

mkdir -vpm 1757 /home/samba/profiles

Especificando la clave del administrador de LDAP en Samba

smbpasswd -w clave

NOTA: clave se debe sustituir por la contraseña de administración de LDAP.

Comprobación

Ahora se procede a comprobar que todo funcione correctamente para esto ejecute en la consola el comando ***testparm*****,** se mostrará en la pantalla de la siguiente manera.

Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[netlogon]"
Processing section "[profiles]"
Processing section "[printers]"
Processing section "[print$]"
Processing section "[tmp]"
Processing section "[cdrom]"
Loaded services file OK.
Server role: ROLE_DOMAIN_PDC
Press enter to see a dump of your service definitions

Se debe reiniciar el Samba
/etc/init.d/samba reload

Por último se instalará la herramienta **smbldap-tools** la cual brinda varias funcionalidades para trabajar con el directorio donde están los datos. Dicha herramienta tiene dos ficheros de configuración, **smbldap.conf** y **smbldap_bind.conf**. Estos ficheros deben ser configurados de acuerdo a como se tengan los datos en el directorio. De esta forma **Samba** ya está preparado para autenticar sus usuarios contra LDAP y como PDC.

En este capítulo se realizó la integración de los servicios telemáticos presentes en una red, utilizando un directorio LDAP para almacenar la información de todos los usuarios que harán uso de los servicios brindados. Los procedimientos se hicieron sobre la distribución **Debian GNU/Linux** versión **4**. En el Manual de instalación, resultado de este trabajo, se ofrece de forma más detallada el proceso de integración.

Conclusiones

La implantación de Software Libre en países en vía de desarrollo como el nuestro, representa un avance en la rama de la informática. Mediante el uso del Software Libre se pueden integrar los servicios telemáticos utilizando LDAP como servicio de directorio, permitiendo que la información esté centralizada.

Analizando las características jerárquicas de LDAP así como la capacidad de distribución de la información y su gran disponibilidad, se llega a la conclusión que este protocolo facilita el establecimiento de políticas, respalda una gran cantidad de información y ofrece mayor cantidad de servicios íntegros, debido que hay muchas aplicaciones que tienen soporte para LDAP. La personalización de los servicios existentes en la red se logra mediante diferentes herramientas, por ejemplo phpldapadmin, permitiendo una fácil gestión y administración de los usuarios.

Al culminar este trabajo se le dio cumplimiento a los objetivos propuestos y a las preguntas científicas planteadas, obteniendo como resultado un manual de instalación donde se integran los servicios sobre LDAP con el uso del Software Libre.

Recomendaciones

- Aplicar el resultado de esta investigación en la universidad y en otras instituciones del país como parte del proceso de migración a Software Libre que se lleva a cabo.
- Estudiar y documentar la integración del resto de los servicios telemáticos.
- Utilizar la replicación de servicios, para lograr una mayor disponibilidad de estos.
- Profundizar en el estudio de las funcionalidades de los diferentes servicios implementados.

Referencias Bibliográficas

1. *Copyleft*. 2007. [Disponible en: http://es.wikipedia.org/wiki/Copyleft#Historia
2. ESPINOSA, R. A. H. *Reflexiones sobre el uso del Software Libre en Cuba. Ventajas*.
3. *File Transfer Protocol*. 2007. [Disponible en: http://es.wikipedia.org/wiki/FTP
4. GARCÍA, C. *Un poco de historia del software libre*, 2004. [Disponible en: http://espora.org/revueltas/article.php3?id_article=23
5. GONZÁLEZ, S. G. *Integración de redes con OpenLDAP, Samba, CUPS y PyKota*, 2004. [Disponible en: http://es.tldp.org/Tutoriales/doc-openldap-samba-cups-python/htmls/
6. HINOSTROZA, R. R. *Configurar ProFTPD*, 2007. [Disponible en: http://www.linuxcentro.net/linux/staticpages/index.php?page=ServidorFTPCompleto
7. *Hypertext Transfer Protocol*. 2007. [Disponible en: http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Primeros_Servidores
8. LIBRE:, P. G. Y. F. D. S. *Diversas Licencias y comentarios sobre ellas*, 2006a. [Disponible en: http://www.gnu.org/licenses/license-list.es.html#TOCIntroduction
9. *La Definición de Software Libre*, 2006b. [Disponible en: http://www.gnu.org/philosophy/free-sw.es.html
10. *Manual de OpenLDAP en español*. 2004. [Disponible en: http://www.ldap-es.org/node/20
11. MIGUEL, C. *Servicios TCP/IP*. Disponible en: http://www.monografias.com/trabajos15/servicios-tcp-ip/servicios-tcp-ip.shtml#DNS
12. ROBERT ECKSTEIN, D. C.-B., PETER KELLY. *Usando Samba*. 2001. 27-28 p.
13. *Server Message Block*. 2007. [Disponible en:http://es.wikipedia.org/wiki/Server_Message_Block
14. *Servicios*. 2007. [Disponible en: http://es.wikipedia.org/wiki/Servicios
15. TEMPUS, L. *Instalación de un servidor DNS con Bind*, 2007. [Disponible en: http://www.liberaliatempus.com/articulos/linux/instalacion-de-un-servidor-dns-con-bind.html

Bibliografía

1. *Cuba otro país listo para el Software Libre*. 2005. [Disponible en: http://www.softwarelibre.cl/drupal//?q=node/38

2. *Cuba utilizará software libre*. 2005. [Disponible en: http://es.wikinews.org/wiki/Cuba_utilizar%C3%A1_software_libre

3. DONNELLY, M. *Introducción a LDAP*, 2000. [Disponible en: http://www.benavent.org/recetas/articles/intro.htm

4. FOUNDATION, A. S., 2006. [Disponible en: http://httpd.apache.org/docs/2.0/es/

5. FRANCO, J. R. *Instalación y configuración de OpenLDAP*, 2002. [Disponible en: http://bulmalug.net/body.phtml?nIdNoticia=1343

6. FREDRIKSSON, T. *OpenLDAP, OpenSSL, SASL and Kerberos HOWTO*, 2005. [Disponible en: http://www.bayour.com/LDAPv3-HOWTO.html

7. JIMENEZ, A. V. *Mini Howto Proftpd + LDAP*, 2005. [Disponible en: http://www.primates.cl/public/imagenes/proftpdldap.pdf

8. LANGFELDT, N. *DNS COMO*, 1997. [Disponible en: http://www.insflug.org/COMOs/DNS-Como/DNS-Como.html

9. MEDINA, J. A. *El COMO de SAMBA LDAP en Linux*, 2006. [Disponible en: http://www.tuxjm.net/docs/samba+ldap-como/

10. MUQUIT, M. A. *LDAP authentication module for Apache 2.x*, 2003. [Disponible en: http://www.muquit.com/muquit/software/mod_auth_ldap/mod_auth_ldap.html

11. RED HAT, I. *Red Hat Enterprise Linux 4: Manual de referencia.Cap 15*, 2005. [Disponible en: http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ftp.html

12. ROBERT ECKSTEIN, D. C.-B., PETER KELLY Usando Samba, 2001.

13. TALLMAN, R. *Porqué "Software Libre" es mejor que software de "Código Fuente Abierto*. Disponible en: http://www.gnu.org/philosophy/free-software-for-freedom.es.html.

14. WIKILEARNING. *ftp*, 2007. [Disponible en: http://www.wikilearning.com/ftp-wkccp-9634-21.htm

15. P*am*, 2007. [Disponible en: http://www.wikilearning.com/pam-wkccp-9634-6.htm
16. WOOD, D. *Samba COMO*, 1996. [Disponible en: http://www.insflug.org/COMOs/Samba-Como/Samba-Como.html

Glosario de términos

LDIF: Es el formato que utilizan los servidores de directorios LDAP para importar, exportar información o para describir una serie de cambios que han de aplicarse en el mismo. Un fichero LDIF almacena información en jerarquías de entradas orientadas a objeto. Todos los servidores LDAP incluyen una utilidad para convertir ficheros LDIF a formato orientadas a objeto.

Frontend y Backend

El **front-end** es la parte del software que interactúa con el usuario y el **back-end** es la parte que procesa la entrada desde el front-end. La separación del sistema en "**front-ends**" y "**back-ends**" es un tipo de abstracción que ayuda a mantener las diferentes partes del sistema separadas. La idea general es que el front-end es el responsable de recolectar los datos de entrada del usuario, que pueden ser de muchas y variadas formas y procesarlas de una manera conforme a la especificación que el back-end pueda usar. La conexión del front-end y el back-end es un tipo de interfaz.

Secure Sockets Layer (**SSL**) y **Transport Layer Security** (**TLS**)

Son protocolos criptográficos que proporcionan comunicaciones seguras.

WWW: World Wide Web o la "Web" es un sistema de documentos de hipertexto enlazados y accesibles a través de Internet. Con un navegador Web, un usuario visualiza páginas Web que pueden contener texto, imágenes u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces.

DBMS: Sistema manejador de bases de datos.

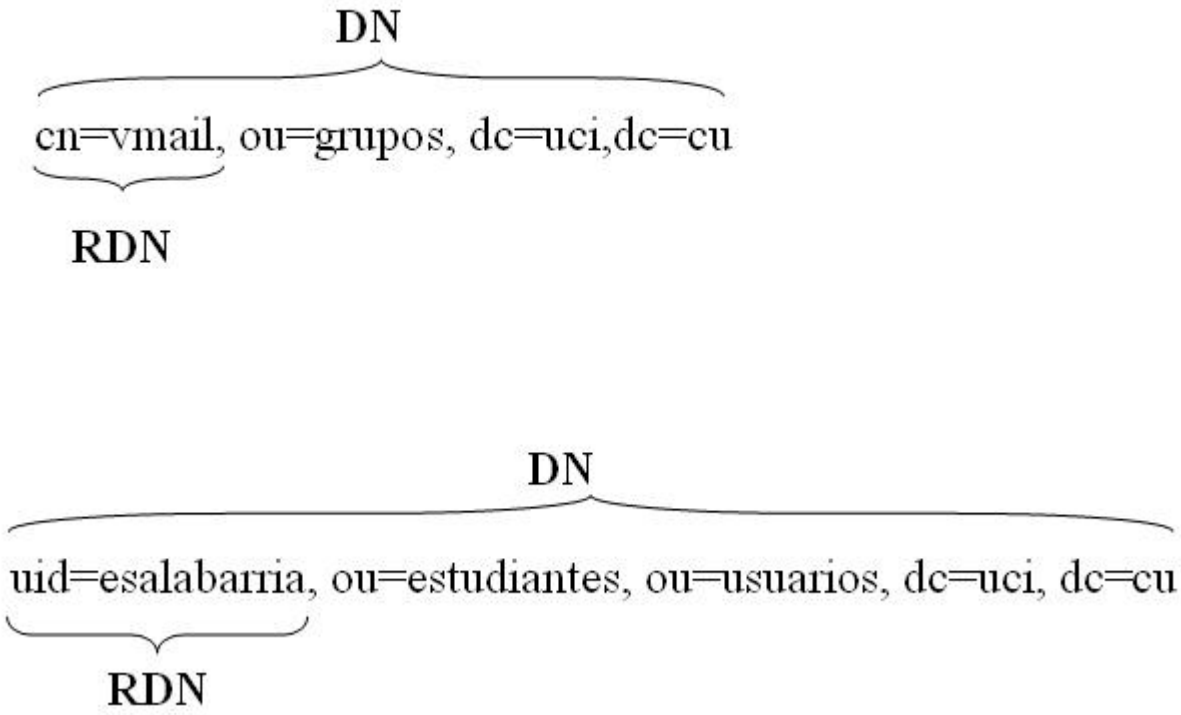
TCP Wrappers: Es una herramienta simple que sirve para monitorear y controlar el tráfico que llega por la red. Esta ha sido utilizada exitosamente en la protección de sistemas y la detección de actividades ilícitas. Es una herramienta de seguridad libre y muy útil.

IPv4, IPv6: Diferentes versiones del protocolo IP**.**

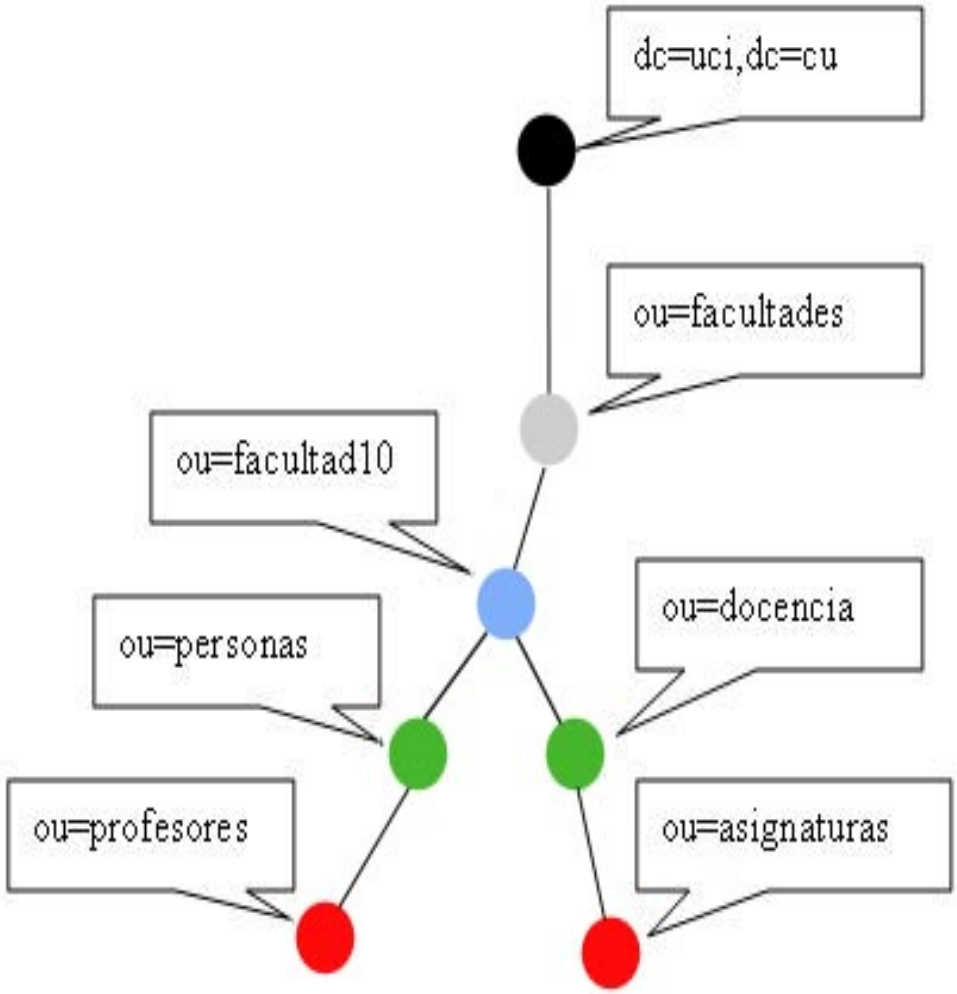
X.500: Es un conjunto de estándares de redes de ordenadores sobre servicios de directorio.

Anexos

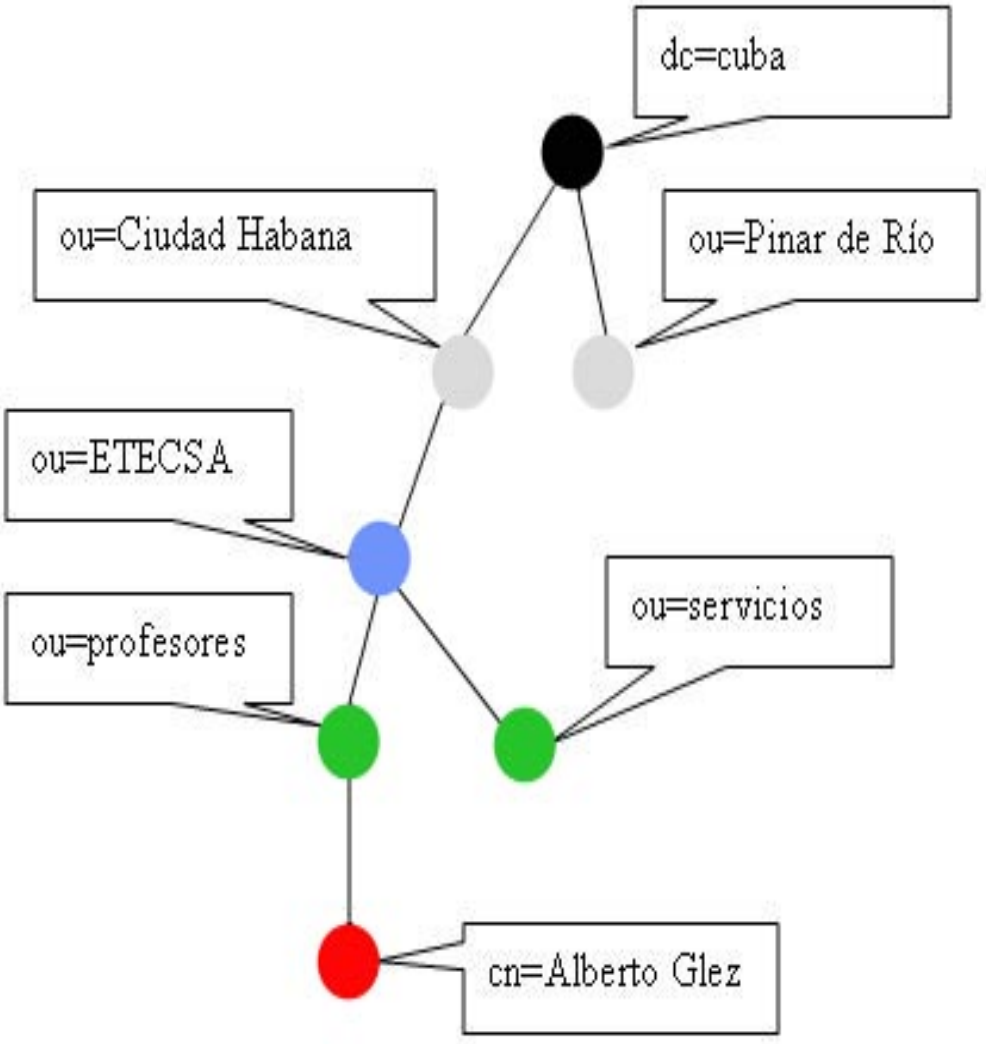
Anexo 1. Directorio



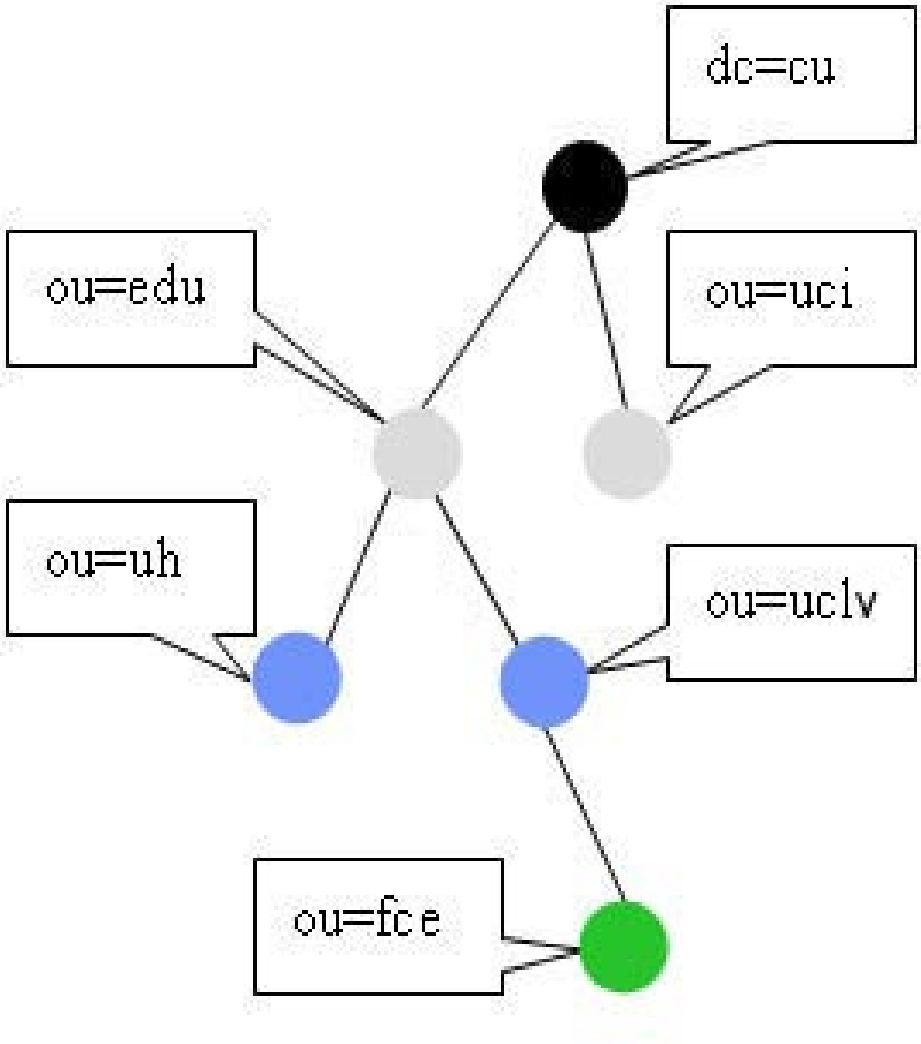
Anexo 2. RDN y DN



Anexo 3. Árbol de Directorio



Anexo 4. Nombramiento tradicional



Anexo 5. Nombres de dominios