INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
UNIDAD CULHUACAN

SEMINARIO DE TITULACIÓN
"SEGURIDAD DE LA INFORMACIÓN"

TESINA

"Implementación de un Sistema de Autenticación usando LDAP para control de Acceso a una RED"

QUE PRESENTAN PARA OBTENER EL TÍTULO DE
INGENIERO EN COMUNICACIONES Y ELECTRÓNICA

ALEJANDRO AGUILAR SANTOS
JUAN CARLOS PÉREZ PÉREZ
LUIS ENRIQUE CORNEJO TELLO

Asesora:
ESP. LIDIA PRUDENTE TIXTECO

VIGENCIA: DES/ESIME-CUL-2008/23/3/10

México, D.F., Mayo 2011

IPN
ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
UNIDAD CULHUACAN

TESINA

POR LA OPCIÓN DE TITULACIÓN SEMINARIO EN SEGURIDAD DE LA INFORMACIÓN

QUE PARA OBTENER EL TÍTULO DE INGENIERO EN COMUNICACIONES Y ELECTRÓNICA

DEBERÁN DESARROLLAR:

ALEJANDRO AGUILAR SANTOS
JUAN CARLOS PÉREZ PÉREZ
LUIS ENRIQUE CORNEJO TELLO

"IMPLEMENTACIÓN DE UN SISTEMA DE AUTENTICACIÓN USANDO LDAP PARA CONTROL DE ACCESO A UNA RED"

INTRODUCCIÓN

LA SEGURIDAD INFORMÁTICA ES UNA DISCIPLINA QUE SE RELACIONA CON DIVERSAS TÉCNICAS, APLICACIONES Y DISPOSITIVOS, CUYO OBJETIVO PRINCIPAL ES ASEGURAR LA INTEGRIDAD, CONFIDENCIALIDAD Y DISPONIBILIDAD DE LA INFORMACIÓN DE UN SISTEMA INFORMÁTICO Y SUS USUARIOS, LA HERRAMIENTA OPENLDAP PERMITE CUMPLIR CON LOS MINIMOS MECANISMOS DE SEGURIDAD COMO SON LA AUTENTICACIÓN Y LA AUTORIZACIÓN.

El TRABAJO PRESENTADO CONSISTE EN LA IMPLEMENTACIÓN Y CONFIGURACIÓN DE LA HERRAMIENTA OPENLDAP, CON EL PROPÓSITO DE EMPLEARLO EN UNA EMPRESA CON PROBLEMAS DE SEGURIDAD EN CUANTO A AUTENTICACION, DE ESTA FORMA SE DAN LOS PERMISOS Y ACCESO A SU SERVICIO DE ARCHIVOS.

CAPITULADO

México D.F., Mayo 2011

VIGENCIA: DES/ESIME-CUL-2008/23/3/10

DR. GABRIEL SÁNCHEZ PÉREZ
Coordinador del Seminario

ESP. LIDIA PRUDENTE TIXTECO
Asesora

M. EN C. LUIS CARLOS CASTRO MADRID
Jefe de la carrera de I.C.

Agradecimientos

Alejandro Aguilar Santos

Doy gracias a mi esposa ***Sonia*** *por el apoyo incondicional no solo en esta sino en todas las etapas que hemos vivido juntos, agradezco su paciencia y comprensión para concluir con este ciclo en mi vida.*

Agradezco a mis hijas ***Pamela, Nadia*** *y a mi hijo* ***Alejandro****, quienes son el motor de mi vida, mi alegría, mi inspiración y el motivo para nunca darme por vencido y siempre seguir adelante.*

A mis hermanos y hermanas: ***Guillermo, Fernando, Marco Antonio, Esperanza, Leticia, Javier*** *y* ***Virginia****, quienes de alguna u otra manera siempre estuvieron conmigo durante mis estudios.*

A mi madre ***Felisa*** *y a mi padre* ***Francisco****, quienes fueron los responsables de que yo estuviera aquí en este mundo y que hasta donde quiera que estén, se enteren que me siento orgulloso de ellos.*

A mis compañeros de equipo por su colaboración y participación para la realización de este proyecto.

A mis profesores que con su sabiduría y paciencia transmitieron sus conocimientos para nuestro desarrollo profesional.

Y a todas las personas que de alguna manera me han apoyado, parientes y amigos, gracias.

Los amo a todos.

Agradecimientos

Juan Carlos Pérez Pérez

*A **Sofía**, mi esposa y compañera, quién siempre me brindó su apoyo y me impulsó a dar este paso tan importante en mi vida y mi carrera, agradezco su paciencia en todo este tiempo que pase lejos de ella y además me brindó ánimos para concluir este ciclo, a mi hija **Karla** quién tanto me extrañó todo este tiempo y siempre tenía para mí un gran abrazo y una sonrisa. A ellas las hago parte de este logro y les doy todo mi agradecimiento. Las amo con todo mi corazón.*

*A mis padres **Irene** y **Felipe**, quienes gracias a su esfuerzo y gran apoyo he logrado alcanzar muchas metas a lo largo de mi vida, sin su ayuda no sería lo que ahora soy. A tí Madre por tu enorme entrega y sacrificio, y a tí Padre por tu esfuerzo y compromiso, gracias a los dos por darme lo más valioso que se puede dar a un hijo, mis estudios.*

*A mis Hermanas **Lucía** y **Maricela** quienes desde siempre estuvieron pendientes de mí, gracias por motivarme día con día a ser mejor, quiero que se sientan orgullosas por que también son parte de este triunfo.*

*A todos mis **sobrínos** quiénes de una u otra forma me dieron un motivo para cerrar este ciclo, a ustedes les quiero dedicar este triunfo y decirles que nunca es tarde para alcanzar las metas.*

Agradecimientos

Luis Enrique Cornejo Tello

Gracias a Dios.

A mi esposa, ***Bertha*** *quien siempre me ha dado su apoyo incondicional, agradezco que este junto a mi y me brindo ánimos para concluir este ciclo, así como a mis hijos* ***Monserrat*** *y* ***Daniel*** *quienes se privaron de varias salidas por esperar mi llegada. De ellos es este triunfo y para ellos mi agradecimiento.*

Para mis Papas, ***Sara*** *y* ***Carmelo****, quienes me brindaron el apoyo para estudiar y superarme, ya que sin ellos no hubiera podido llegar a concluir mis estudios.*

A mis Hermanos ***Cesar****,* ***Oscar*** *y* ***María del Carmen*** *quienes siempre me animaron a superarme y decir si se puede.*

A mis ***Abuelos****, quienes me brindaron su apoyo y su cariño cuando lo requerí.*

A mis ***Compañeros*** *de Equipo, quienes aportaron sus ideas para la realización de este trabajo.*

A todas aquellas personas que han sido importantes para mí.

A todos mis maestros que aportaron a mi formación.

ÍNDICE GENERAL

ÍNDICE DE TABLAS Y FIGURAS

OBJETIVO

Implementar un servidor de autenticación configurado en un sistema operativo LINUX utilizando la herramienta Open LDAP para control de acceso a usuarios en una red LAN.

Objetivos Específicos:

- Definir qué es un directorio activo LDAP.
- Nombrar los diferentes tipos de directorios activos.
- Explicar el funcionamiento de un servidor LDAP.
- Señalar las características de operación de la herramienta Open LDAP.
- Configurar un servidor con Open LDAP.

JUSTIFICACIÓN

Un servicio de directorio es muy usado en la actualidad, ya que permite administrar de manera muy fácil los recursos y usuarios de alguna empresa. El protocolo ***LDAP (Lightweight Directory Access Protocol)*** es utilizable por distintas plataformas y basado en estándares, de ese modo las aplicaciones no necesitan preocuparse por el tipo de servidor en que se hospeda el directorio, además de ser independiente de la aplicación.

En algunas organizaciones no se tiene el cuidado de establecer políticas de seguridad en el manejo de la información, esto permite que personas no autorizadas puedan llevar a cabo modificaciones en la ruta y contenidos de los archivos informáticos de las organizaciones. Para este tipo de situaciones, una de las soluciones posibles es implementar un servidor LDAP.

A través del servidor LDAP es posible proporcionar a un administrador de un sistema, la capacidad de manejar una base de datos de usuarios y a su vez asignar permisos de los archivos de trabajo contenidos en el servidor.

La herramienta OpenLDAP permite configurar un servicio de directorios que cuenta con la capacidad de crear un árbol de directorios, en donde se pueden crear ramas que agrupen a cada uno de los departamentos de alguna organización y proporciona además la opción de implementar los permisos hacia los archivos o recursos, con acciones como estas se conseguirá que usuarios de otros departamentos no puedan realizar cambios en los archivos en los cuales no tengan los permisos correspondientes, al no poder modificar los archivos, si no se tienen los privilegios asignados por el administrador, se pueden minimizar los riesgos para poder mantener la integridad de los archivos.

INTRODUCCIÓN

La seguridad informática es una disciplina que se relaciona con diversas técnicas, aplicaciones y dispositivos, cuyo objetivo principal es asegurar la integridad, confidencialidad y disponibilidad de la información de un sistema informático y sus usuarios. Es difícil lograr que un sistema informático sea ciento por ciento seguro, pero al establecer medidas de seguridad se evitan daños y problemas que los intrusos pueden ocasionar.

La posibilidad de interconectarse a través de redes ha abierto nuevos horizontes que permiten explorar más allá de las fronteras de la organización. En la actualidad, la seguridad informática ha adquirido gran auge dadas las cambiantes condiciones y las nuevas plataformas de computación disponibles.

A través de cada uno de los capítulos se estarán validando cada uno de los objetivos específicos, se comenzara a explicar de manera básica que es la seguridad de la información, que son los directorios desde el punto de vista informático, además de mencionar sus ventajas al utilizarlos, se continuara con los directorios de aplicación en este caso LDAP, en donde se abordaran temas del funcionamiento de LDAP, las principales características, su constitución y

nomenclatura, para después dar paso a la herramienta Open LDAP en donde se mostrara la forma de instalar, configurar y al final realizar pruebas de funcionalidad.

En caso de que las pruebas sean exitosas entonces se verá la factibilidad de poner en producción el sistema dentro de una organización, ya que se contara con una herramienta fácil de administrar, debido a que la configuración no es tan complicada, además de que proporciona un método sencillo de autenticación y autorización que permitirá mitigar el riesgo de alguna perdida de información dentro de los directorios, además de que se cuenta con un sistema que puede dar cabida a manejar información a detalle del personal que labora en la organización, como por ejemplo, el directorio de empleados puede contener datos como dirección, teléfono, correo electrónico, grado de estudios, departamento, etc.

CAPÍTULO I. INTRODUCCIÓN AL CONTROL DE ACCESO CON DIRECTORIOS

En este capítulo se presenta de manera concisa, los elementos de los cuáles se compone un sistema básico de control de acceso, que cuenta con la capacidad de validar a un usuario y si el usuario es validado por el sistema, saber los permisos que se le asignan, para lo cual se explica de forma general que es la seguridad, controles de acceso. Una vez entendido el tema de seguridad se comienza a trabajar con los directorios desde el punto de vista informático en donde se muestran los diferentes tipos de directorios y por último se enumeran las ventajas de trabajar con un directorio estándar.

1.1 Seguridad.

La seguridad es el proceso mediante el cual se protegen los recursos de información digitales. Los objetivos de seguridad son:

- Proteger la confidencialidad, se refiere a la protección de los datos ante una revelación no autorizada a terceras partes.
- Mantener la integridad, se refiere a la certeza de que los datos no son destruidos o alterados de una forma no autorizada.
- Asegurar la disponibilidad, se define como el funcionamiento continuo de los sistemas de computo.

La seguridad significa una protección a la información frente a los ataques malintencionados de intrusos. Los tipos de seguridad son los siguientes:

- Seguridad de la información.
- Seguridad informática.
- Seguridad de la red.
- Seguridad de la internet.

1.2 Control de Acceso.

Es la prevención del uso no autorizado de una fuente, desde el punto de vista de tecnologías de información se refiere principalmente al uso de archivos que contienen información vital para el negocio.

Una de las más importantes características del sistema de control de acceso a tener en cuenta es la facilidad de uso de su sistema, esto es para que sea aceptado de manera natural en un entorno.

Una vez que un usuario se autentica ante un sistema de control de acceso, si satisface las políticas del sitio, logra convertirse en un usuario activo del sistema de información a través de una sesión que crea el sistema operativo. A partir de ese momento todas sus acciones están ligadas a su identidad. [10]

1.2.1 Protocolo AAA

En seguridad informática, el acrónimo AAA corresponde a un tipo de protocolos que realizan tres funciones: Autenticación, Autorización y Contabilización (Authentication, Authorization and Accounting por sus siglas en inglés). La expresión protocolo AAA no se refiere pues a un protocolo en particular, sino a una familia de protocolos que ofrecen los tres servicios citados.

1.3 Autenticación.

Es el proceso utilizado en los mecanismos de control de acceso con el objetivo de verificar la identidad de un usuario, dispositivo o sistema mediante la comprobación de credenciales de acceso. La forma básica en cómo se realiza la operación es:

1. El usuario envía al servidor un nombre de usuario y una contraseña.
2. El servidor recibe estos datos y compara la información con su base de datos.
3. Si la información coincide se le permite el acceso al usuario a la red.
4. Si la información no coincide se le negara el acceso al usuario.

1.3.1 Autorización

Autorización se refiere a la concesión de privilegios específicos a una entidad o usuario basándose en su identidad (autenticada), los privilegios que solicita, y el estado actual del sistema. Las autorizaciones pueden también estar basadas en restricciones, tales como restricciones horarias, sobre la localización de la entidad solicitante, la prohibición de realizar accesos múltiples simultáneos del mismo usuario, etc. La mayor parte de las veces el privilegio concedido consiste en el uso de un determinado tipo de servicio.

1.4 ¿Qué es un directorio?

En informática un directorio es un contenedor virtual en el que se almacena una agrupación de archivos de datos.

Técnicamente el directorio almacena la información de los archivos, como los atributos o dónde se encuentran físicamente en el dispositivo o dispositivos de almacenamiento.

En redes de computadoras, un directorio es una colección de usuarios, contraseñas, y usualmente contiene información acerca de los recursos de la red a los que ellos puedan tener acceso.[3]

1.5 Tipos de Directorios.

Existen dos grandes grupos de directorios, conocidos como estáticos y dinámicos, y tal como su nombre lo indica, los estáticos son los que sufren poco cambio y normalmente no se modifican por los usuarios del sistema, pero si por los administradores del mismo, en este grupo de directorios se pueden mencionar, el directorio de los archivos de sistema. Por otro lado están los dinámicos que cambian muy seguido y normalmente son modificados por el usuario, ya sea al aumentar información, al quitar información, inclusive borrar el mismo directorio o por la misma interacción con el equipo, dentro de esta categoría se puede mencionar el directorio temporal, tanto en Windows ® como en Linux. ®. Pero hace algunos años se introdujo un nuevo concepto de directorio, conocido como directorio activo.[1]

1.6 Directorios Activos.

Un directorio es una lista de información acerca de objetos acomodados en algún orden que da detalles de los mismos. Ejemplos comunes son un directorio telefónico y las fichas bibliográficas de los libros. Para un directorio telefónico, los objetos listados son personas, los nombres son ordenados alfabéticamente y los detalles de cada persona son los números telefónicos y direcciones. Los libros de una librería son ordenados por autor o por título y la información adicional como la editorial, el año de publicación, edición, etc., serán los detalles.

En términos “computacionales”, un directorio es una base de datos especializada, también conocida como un repositorio de datos, que almacena y ordena información acerca de objetos. Un directorio particular podría listar información acerca de impresoras (los objetos), por ejemplo, la ubicación, velocidad en páginas por minuto, tipo de impresión soportada, marcas, etc.

Los directorios permiten a los usuarios o aplicaciones encontrar recursos que tengan las características necesarias para una tarea en particular.

1.6.1 Directorio versus Base de Datos.

Un directorio es frecuentemente descrito como una base de datos, pero es una base de datos especializada, que tiene características que difieren de las

bases de datos de propósito general. Una característica especial de los directorios es que son accedidos (leídos) mucho más frecuente de lo que son modificados (escritos). Debido a que los directorios deben de ser capaces de soportar altos volúmenes de peticiones de lectura, estos son típicamente optimizados para acceso de lectura. Los accesos de escritura deben de ser limitados a los administradores o a los propietarios de cada pieza de la información.

A causa de que los directorios son principalmente usados para almacenar información estática y son optimizados para este propósito, no son apropiados para almacenar información que cambia rápidamente. Por ejemplo, el número de trabajos en una cola de impresión debería no ser almacenada en una entrada de directorio para una impresora a causa de que la información tendría que ser actualizada frecuentemente para ser precisa; en lugar de esto la entrada de un directorio para una impresora puede contener la dirección de red de el servidor de impresión, el servidor de impresión puede ser consultado para obtener información acerca de la longitud de la cola de impresión si se desea, la información en el directorio (dirección del servidor de impresión) es estática, mientras que el número de trabajos en la cola de la impresora es dinámica.

Otra importante diferencia entre un directorio y una base de datos relacional de propósito general está en la forma en que la información puede ser leída, la mayoría de las bases de datos soportan un método de acceso estándar llamado SQL (Structured Query Languaje), directorios, tales como LDAP (Lightweight Directory Acces Protocol) usan un protocolo de acceso que puede ser usado en aplicaciones relativamente simples. Debido a que la intención de los directorios no es la de proveer tantas funciones como las bases de datos relacionales de propósito general, estos pueden ser optimizados para que económicamente proporcionen más aplicaciones con rápidos accesos en ambientes distribuidos.

1.6.2 Directorio versus Sistemas de Archivos.

Un directorio hace un sistema de archivos pobre. Los archivos tienen diferentes características en relación con la información contenida en los directorios. Los directorios están optimizados para almacenar pequeños fragmentos de información que puede estructurarse como entradas con diferentes atributos, en cambio, los sistemas de ficheros contienen archivos, a veces de tamaños superiores al gigabyte. Además, los sistemas de ficheros permiten acceder a un fichero y posicionarse dentro de él, sin embargo, los directorios a lo sumo permiten acceder a un atributo, pero no hay forma de

posicionarse dentro de dicho atributo, que por lo tanto debe ser leído por completo.

Los directorios son optimizados para leer más que para escribir. Algunos archivos, por supuesto, se leen con mucha más frecuencia de lo que se escriben. Las aplicaciones binarias son un buen ejemplo de los archivos en esta categoría, pero el tamaño de estos archivos es a menudo tan grande que deben de ser almacenados en un directorio.

Las aplicaciones suelen tener acceso a fracciones de archivos, especialmente si el archivo es grande. Las funciones de sistemas de archivo proporcionan las funciones para este fin, tales como: seek -busqueda, read -lectura y write-escritura, que puede ser utilizado para acceder a sólo una parte de un archivo muy grande. Los directorios no proporcionan soporte para este tipo de acceso aleatorio. En lugar de eso, una entrada de directorio se divide en los elementos de datos llamados atributos.

Los sistemas de archivos no son buenos para almacenar información basada en atributos y típicamente no tienen las capacidades de búsqueda de propósito general que tienen los directorios. [2].

1.7 Ventajas de usar un directorio.

Un directorio de aplicación especifica almacena solo la información necesaria por una aplicación en particular y no es accesible por otras aplicaciones.

A causa de que un servicio de directorio de función completa es complejo para construir; los directorios de aplicación especifica son típicamente muy limitados; estos probablemente almacenan solo un tipo de información específica, no tienen capacidades de búsqueda, no soportan replicación y partición y probablemente no tengan herramientas completas de administración.

Un directorio de aplicación específica podría ser tan simple como unos archivos de texto editables y estos pudieran ser almacenados y accedidos en una manera propietaria e indocumentada. En tal ambiente, cada aplicación crea y maneja su propio directorio de aplicación específica, lo cual rápidamente llegaría a ser un problema administrativo, la misma dirección de correo electrónico almacenada por la aplicación del calendario debería también ser almacenada por una aplicación de correo y por una aplicación que notifique a los operadores de los sistemas de los problemas de los equipos; mantener múltiples copias de información actualizadas y sincronizadas usando este concepto es muy difícil,

especialmente cuando diferentes interfaces de usuarios y diferentes administradores están involucrados.

Entonces surge la necesidad de un directorio común independiente de la aplicación.

Si los desarrolladores de la aplicación pudieran asegurarse de la existencia de un servicio de directorio, entonces los directorios de aplicación especifica no serían necesarios.

Como siempre un directorio común puede presentar los problemas mencionados anteriormente; por lo cual debe de estar basado en un “estándar” abierto que sea soportado por la mayoría de los fabricantes en muchas plataformas, este estándar es denominado API, debe de ser tan extenso para que pueda contener los tipos de datos necesarios para aplicaciones arbitrarias y debe de proveer funcionalidad completa sin requerir recursos excesivos en sistemas más pequeños.

Dado que la mayoría de los usuarios y aplicaciones tendrán acceso al directorio común este debe también de ser robusto, seguro y escalable. Cuando tal infraestructura de directorio esta en sitio, los desarrolladores de la aplicación

entonces pueden dedicar su tiempo a desarrollar aplicaciones en lugar de directorios de aplicación especifica.

Aquí una pequeña guía que puede emplearse al momento de decidir el usar o no un directorio:

- ***Tamaño de la información***: Los Directorios son lo mejor para almacenar relativamente piezas pequeñas de información, no archivos con muchos Mbytes de información Los directorios son buenos para almacenar los punteros a las cosas grandes, pero no las cosas grandes en sí mismos.

- ***Carácter de la información***: Los Directorios típicamente tienen un modelo de información basado en atributos donde la información es dividida en un conjunto de pares nombre-valor, si se puede expresar de manera natural la información en esta forma, un directorio será una buena opción, si no es posible, se deberá de considerar usar una base de datos o un sistema de directorios.

- ***Relación Lectura/Escritura***: Los directorios son mejores para la información que se lee con mucha más frecuencia de lo que se escribe. Si la información se va a escribir con más frecuencia, un sistema de base de datos o de archivos podría ser una opción más apropiada.

- ***Capacidad de Búsqueda***: Los directorios son hechos para buscar la información que contienen. Si la aplicación tiene este requisito, un directorio puede ser una buena opción.

- ***Acceso basado en Estándar***: Si se necesita un acceso basado en estándares para el manejo de la información, un directorio es una buena opción.[3]

CAPÍTULO II. DIRECTORIO ACTIVO LDAP.

En este capítulo se da una introducción al directorio activo conocido como LDAP, se mencionan las ventajas clave de LDAP y se concluye con su estructura, para de esta forma conocer el por qué de su uso tan extendido en la actualidad.

Todo aquél que haya estado involucrado en la escena de las redes de computadoras durante los años pasados, seguramente habrá escuchado acerca del Directorio Activo.

LDAP está basado en el modelo cliente servidor, el éxito de LDAP se ha dado debido a que sus características lo hacen simple de usar e implementar; es apropiado para cualquier tipo de información, en donde sea necesaria una búsqueda rápida y pocas actualizaciones.

2.1 Introducción a LDAP.

El Protocolo Ligero de Acceso a Directorios (LDAP), es un servicio que corre directamente sobre la pila de TCP/IP, es un protocolo abierto y las aplicaciones son independientes de la plataforma del servidor que alberga el directorio, define un método estándar para acceder y actualizar información en un directorio.[8]

El protocolo LDAP no define como trabajan los programas en el lado del cliente o del servidor, más bien define el "lenguaje" usado por los programas del cliente para comunicarse con los servidores y también de los servidores a los servidores, LDAP también define permisos, puestos asignados por el administrador para permitir solo a ciertos usuarios acceder a la base de datos, y opcionalmente mantener ciertos datos privados.

Un servicio de directorio es la colección de software, hardware, políticas y procedimientos administrativos involucrados para hacer que la información en el directorio esté disponible para los usuarios del directorio. El servicio de directorio incluye los siguientes componentes:

- La información contenida en el directorio.
- El software de los servidores sujetando esta información.

- El software de los clientes actuando en nombre de los usuarios u otras entidades para acceder a esta información.
- El hardware en el cual los clientes y servidores operan
- El software de soporte, tales como sistemas operativos y “drivers” de dispositivos.
- La infraestructura de red que conecta a los clientes a los servidores y los servidores entre sí.
- Las políticas y/o permisos de quien puede acceder al directorio, que puede almacenar, que puede modificar, etc.
- Los procedimientos por los cuales el servicio de directorio es mantenido y monitoreado.
- El software usado para mantener y monitorear el servicio de directorio.[6]

Estos componentes son mostrados en la figura 2.1.

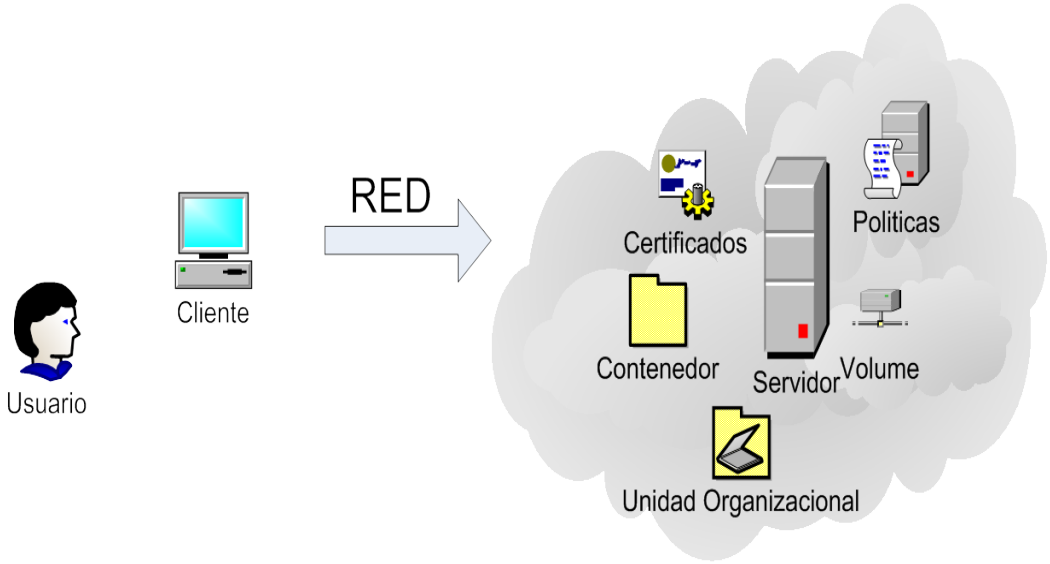

Figura 2.1 Componentes de un sistema de Directorio básico

El servicio de directorio da respuesta a las siguientes preguntas, desde la perspectiva del usuario.

¿Qué información se puede almacenar?

El Directorio Activo almacena toda la información necesaria para el funcionamiento de una red. En el directorio activo se encuentran los usuarios, equipos, impresoras, licencias, ubicación de carpetas compartidas, etc. en forma de objetos accesibles por todos los usuarios o programas de la red.

¿Cómo está organizada la información?

La información es una base de datos jerárquica con estructura de árbol distribuida. Tradicionalmente, esta estructura refleja los límites geográficos y/o de organización.

Las entradas que representan países aparecen al principio del árbol. Debajo de ellos están las entradas que representan estados y las organizaciones nacionales, debajo de ellos deben de estar las entradas que representan unidades organizacionales, gente, impresoras, documentos o cualquier otra cosa que se pueda imaginar. En adición LDAP permite controlar cuales atributos son requeridos y permitidos en una entrada a través del uso de un atributo llamado clase de objeto. Los valores de cada atributo de clase objeto determina las reglas del esquema que la entrada debe obedecer.

La figura 2.2 muestra un ejemplo del directorio LDAP usando nombres tradicionales.

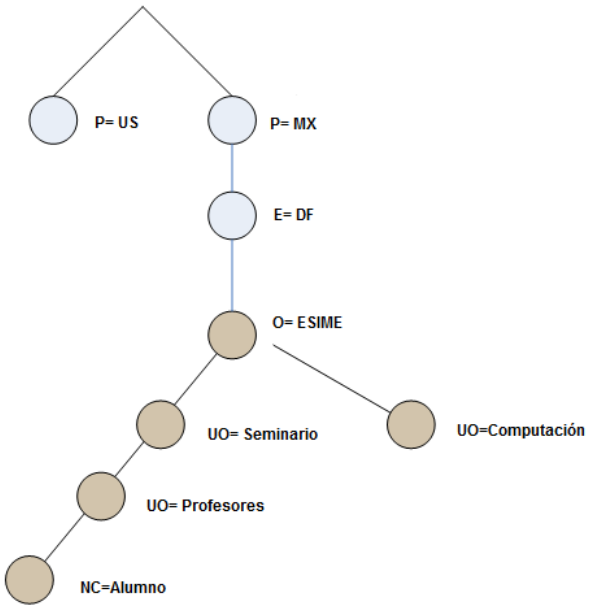

Figura 2.2.Ejemplo de Árbol de Directorio LDAP

¿ Cómo se puede consultar información específica ?

La información se pueda consultar mediante consultas LDAP estándar. La información es utilizada por usuarios y programas para la localización y acceso a todos los elementos de la red.

¿Cómo es protegida la información del acceso no autorizado?

Algunos servicios de directorios no proporcionan el servicio de protección, permitiendo a cualquier persona ver la información, e inclusive algunas veces es posible modificarla.

LDAP proporciona un mecanismo para que un cliente se autentifique y le sean entregados los privilegios de acceso configurados en su perfil. LDAP también soporta los servicios de seguridad para los datos (integridad y confidencialidad). [7].

LDAP ha ganado mucha aceptación como el método de acceso al directorio de la Internet por lo tanto también está adquiriendo un mayor auge en las intranets corporativas, esto está siendo soportado por un número creciente de vendedores de software que están incorporando a un gran número de aplicaciones. Por ejemplo, los dos más populares navegadores Netscape Navigator/Communicator y Microsoft Internet Explorer soportan la funcionalidad LDAP como una función base.

2.1.1 Ventajas Claves de LDAP.

Hoy es claro que LDAP se ha movido hacia delante dentro de los servicios de directorio en línea, y ha generado mucho entusiasmo para el desarrollo y despliegue de directorios LDAP.

LDAP ha emergido del resto de los servicios de directorio y capturado el interés de los profesionales de tecnología de la información.

Una de las ventajas fundamentales del Directorio Activo es que separa la estructura lógica de la organización (dominios) de la estructura física (topología de red). Ello permite, por una parte, independizar la estructuración de dominios de la organización de la topología de la(s) red(es) que interconectan los sistemas; y, por otra parte, permite administrar la estructura física explícitamente cuando es necesario, de forma independiente de la administración de los dominios.

LDAP es universal y simple pero versátil debido a qué soporta una amplia variedad de aplicaciones tipo directorio que tienen diferentes necesidades.

Una buena implementación de LDAP fue desarrollado y distribuido libremente en Internet por los investigadores de la Universidad de Michigan, proporcionando así gran parte del impulso hacia LDAP. Hoy en día, las implementaciones de LDAP están disponibles para cada una de las plataformas de mayor o menor uso.

Los directorios LDAP son baratos y fáciles de entender, las organizaciones que eligen directorios LDAP encuentran que son relativamente baratos para implementar y mantener.

Los sistemas de directorio LDAP son tan sencillos que no es necesario contar con un ejército de consultores para entenderlos e implementarlos.

Los servicios de directorio LDAP simplemente funcionan mejor. La alta confiabilidad, rendimiento y escalabilidad de los productos de directorio LDAP, junto con su diseño de uso general, les permite satisfacer las más importantes necesidades de servicios de directorio. [6]

2.1.2 ¿Cuándo usar LDAP?.

Esta es una muy buena pregunta, en general se debería de usar este servicio de directorio cuando se requiere que los datos se manejen de forma centralizada, se almacenen y se accesen por métodos basados en estándares.

Algunos ejemplos comunes encontrados en la industria, pero no limitados son:

- Autenticación de máquinas
- Autenticación de usuarios
- Libros de direcciones

- Representación de organizaciones
- Seguimiento de activos
- Almacenar información telefónica
- Administración de recursos de usuarios
- Direcciones de correo electrónico
- Almacenamiento de configuraciones de aplicaciones.

En general se puede usar en la mayoría de las veces en la cual la información contenida en el directorio no se deba de modificar de forma muy seguida, ya que el protocolo esta optimizado para realizar búsquedas.[7]

2.1.3 LDAP y la Internacionalización.

Los servicios de directorio, por su propia naturaleza, trascienden las fronteras del lenguaje.

Las empresas multinacionales podrían tener oficinas en decenas de países, cada uno con un lenguaje distinto. Para direccionar estas necesidades, LDAP ha sido diseñado para que pueda soportar múltiples lenguajes.

A causa de que los servidores pueden almacenar texto en varios lenguajes, es muy útil tener una forma de almacenar los atributos por el tipo de lenguaje. Por ejemplo, en una corporación internacional con oficinas en Mexico y en Japón, es deseable guardar muchas representaciones del nombre de los empleados en el directorio, incluyendo una versión en japonés y una versión en español.

El grupo de trabajo de extensión de LDAP dentro de la IETF ha propuesto un método para lograr esto mediante el uso de códigos de idioma. Un código de idioma es una opción en el nombre de un atributo en LDAP, separado desde el nombre del atributo base con un punto y coma, por ejemplo, el tipo de atributo *cn* (nombre canónico); *lang-fr* se refiere a un nombre común en el idioma francés y el tipo de atributo *sn*(nombre simple); *lang-ja* se refiere al apellido en el idioma japonés. Todos los nombres de los idiomas son representados por dos caracteres definidos en el estándar 639, el cual se refiere a los códigos de idiomas.

El cliente de LDAP puede usar códigos de idioma en los filtros de búsqueda y en la lista de atributos, en otras palabras, un cliente de LDAP puede limitar la búsqueda a solo aquellos atributos en el idioma en el que está interesado.

2.2 Unidades Organizativas.

Una Unidad Organizativa (Organizational Unit, OU) es un objeto del Directorio Activo que puede contener a otros objetos del directorio. Es decir, es un contenedor de otros objetos, de forma análoga a una carpeta o directorio en un sistema de archivos tradicional. En concreto, dentro de una unidad de este tipo pueden crearse cuentas de usuario, de grupo, de equipo, de recurso compartido, de impresora compartida, etc., además de otras unidades organizativas.

Los objetos ubicados dentro de una unidad organizativa pueden moverse más tarde a otra, si fuera necesario. Sin embargo, un objeto no puede copiarse: cada objeto es único en el directorio, y su existencia es independiente de la unidad organizativa a la que pertenece.

El objetivo de las unidades organizativas es estructurar u organizar el conjunto de los objetos del directorio, agrupándolos de forma coherente. En el Directorio Activo, las unidades organizativas permiten:

Delegar la administración

Cada unidad organizativa puede administrarse de forma independiente. Esto permite delegar la administración de subconjuntos del dominio a ciertos usuarios que posean el nivel de responsabilidad adecuada.

Establecer de forma centralizada comportamientos distintos a usuarios y equipos

A cada unidad organizativa pueden vincularse políticas de grupo, que aplican comportamientos (generalmente en forma de restricciones) a los usuarios y equipos cuyas cuentas se ubican en dicha unidad. De esta forma, es posible aplicar restricciones distintas a subconjuntos de usuarios y equipos del dominio, en función exclusivamente de la unidad organizativa donde se ubican. Por ejemplo, se puede limitar a los usuarios del departamento de contabilidad para que puedan utilizar ciertas aplicaciones, pero que esto no se aplique a los usuarios del departamento de informática.

En muchas organizaciones de pequeño o medio tamaño resulta más adecuado implementar un modelo de dominio único con múltiples unidades organizativas que un modelo de múltiples dominios. Si es necesario, cada unidad

puede administrarse independientemente, con uno o varios administradores delegados y con comportamientos (políticas) diferentes.

2.3 Esquema del Directorio Activo.

El directorio activo es una base de datos compleja, que almacena y manipula los objetos dinámicamente como sea necesario. Este comportamiento complejo se basa en una lista de definiciones, que define valga la redundancia que tipo de objetos pueden ser puestos en el directorio activo, como los objetos son definidos y como los objetos son integrados con otros. Esta lista de definiciones es llamada el esquema de Directorio Activo; el cual es un esquema simple que define como los datos serán almacenados y vistos en el Directorio Activo.

Es posible imaginar que el esquema del directorio activo es como un árbitro en un juego de futbol. El árbitro conoce las reglas dentro y fuera, y su trabajo consiste en que los jugadores jueguen el juego de acuerdo a las reglas.

En el mundo del Directorio Activo, el esquema es el árbitro, y los administradores del servicio, son los jugadores. El trabajo del esquema es conocer las reglas de la base de datos del Directorio Activo conocer las definiciones que determinan que objetos pueden ser almacenados en la base de datos, como son almacenados y como son definidos.

El esquema mantiene la base de datos en igualdad de condiciones y se asegura que ciertos usuarios no cometan equivocaciones. En términos más técnicos, el esquema se compone de una lista de definiciones sobre objetos e incluso las definiciones acerca de las definiciones de los objetos. Todas estas definiciones son llamadas metadatos, lo cual significa datos sobre datos.

Los metadatos saben que las cuentas de usuarios deben de tener cualidades de nombre de usuario, contraseña, dirección física, número telefónico, y así sucesivamente y no cualidades como de un solo lado, color, tamaño de hoja y especie.

2.3.1 Árbol de directorio LDAP.

Un árbol de directorio no es nada más que una manera organizada de proveer contenedores para almacenar diferentes tipos de información.

Los servidores de directorio LDAP almacenan su información de manera jerárquica, no distinto a un sistema de ficheros UNIX. La jerarquía provee de un método para agrupamiento (y sub agrupamiento) lógico de ciertas características juntos. Estos agrupamientos pueden ser útiles en un número de situaciones:

- Delegación de "autoridad" para uno o más grupos de datos a otro servidor o a otro sitio.
- Replicación de datos.
- Seguridad y control de acceso.
- Escalabilidad.

En la Figura 2.3 se muestra un ejemplo de un árbol de directorio simple.

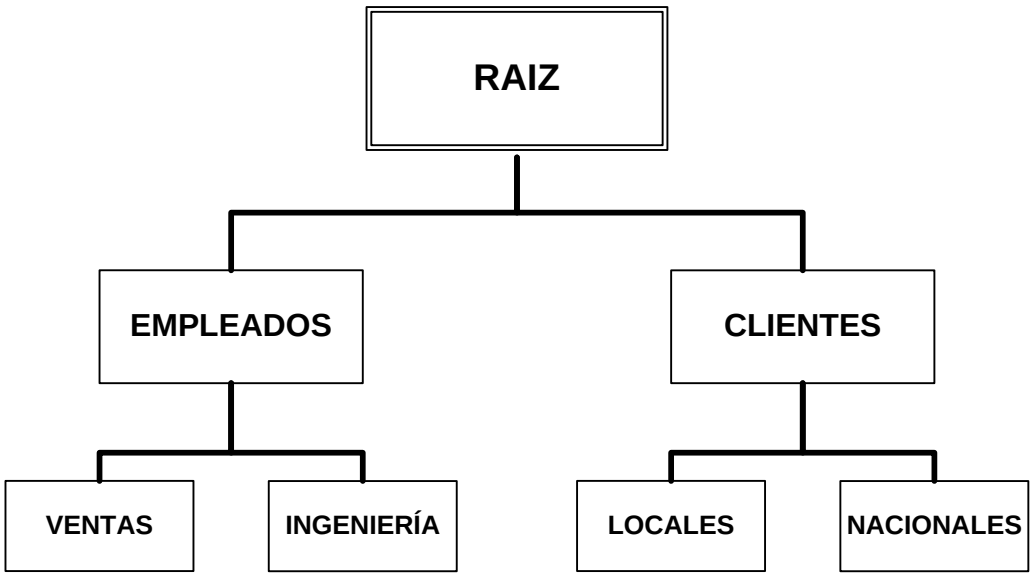

Figura 2.3 .Árbol de Directorio.

Arriba se tiene la raíz, el punto de inicio para el árbol de directorio. Bajo la raíz hay dos categorías, cada una con dos sub categorías. Como se puede observar en el caso de Empleados se están agrupando por departamento, mientras que para Clientes se agrupan geográficamente.

No hay restricciones de como un grupo dado es sub agrupado. En el ejemplo se ha mostrado solo dos grupos por nivel, un directorio puede tener tantos grupos de datos como sean necesarios en cualquier nivel del directorio.

Es importante mencionar que no hay una manera correcta de configurar una estructura de directorio. El diseño de uno puede parecer similar a cualquier otro, y otra vez después puede no parecerlo.

La parte más alta del directorio, referida anteriormente como raíz del árbol del directorio, también es conocida como la base. El nombre de esa base es el Nombre Distinguido de la Base, o base DN.

El modelo de nombres de LDAP define como las entradas son identificadas y organizadas. Las entradas son organizadas en una estructura tipo árbol llamada el Árbol de Información de Directorio, sus siglas en inglés son (DIT-Directory Information Tree). Las entradas son acomodadas dentro del DIT basadas en su nombre distinguido (DN-Distinguished Name), un DN es un nombre único que identifica una entrada sencilla. Los DNs son hechos de una secuencia de nombres distinguidos relativos (RDNs- Relative Distinguished Names).

Cada RDN en un DN corresponde a una rama que va desde la raíz del DIT a la entrada del directorio. Un DN es compuesto de una secuencia de RDNs separados por comas, por ejemplo nc=jlopez, uo=Seminario, o=ESIME.

El administrador puede definir un Árbol de Directorio basado en las necesidades organizacionales, es decir, es posible elaborar el directorio a conveniencia, en esto radica la gran aceptación del protocolo, ya que los campos que aparecen son los que se eligen; a continuación se muestra un ejemplo en la figura 2.4.

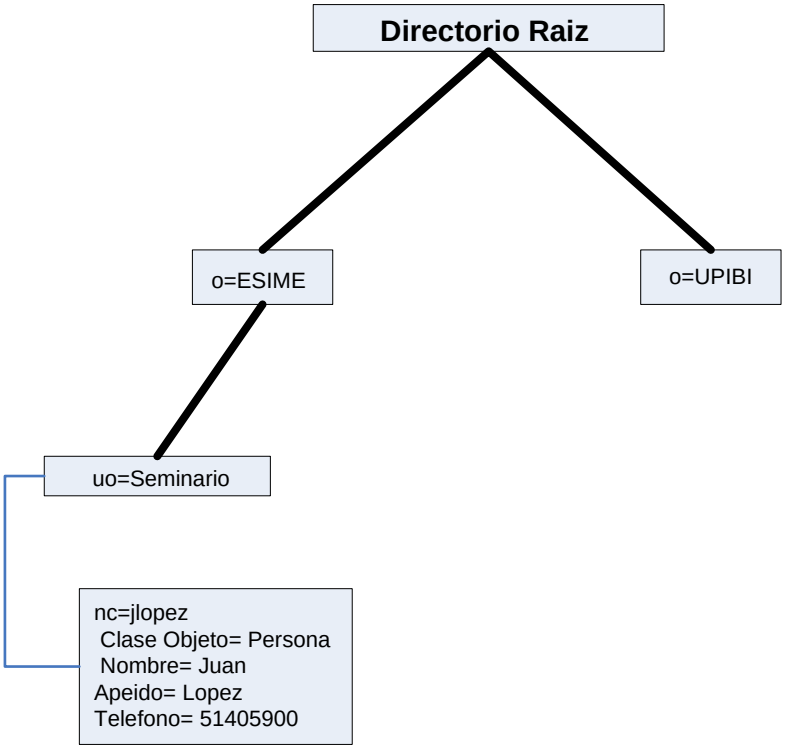

Figura 2.4 Ejemplo de un Árbol de Directorio de Información.

CAPÍTULO III. IMPLEMENTACION DE UN SERVIDOR LDAP CON OpenLDAP

Este capítulo menciona la manera de cómo se configura un Servidor OpenLDAP para controlar el acceso a la información de una Organización y muestra de manera sencilla la forma de cómo obtener el Software de instalación.

Se requiere configurar, debido a que es necesario personalizar la aplicación, aunque se sabe que el Software es libre y se puede obtener desde una dirección electrónica, se tiene la necesidad de adaptarlo a las necesidades de la empresa.

En este caso se usa para autenticar a usuarios dentro de una red y por otro lado autorizar a los usuarios para usar los servicios previamente acordados, en esta ocasión los servicios son archivos.

En la figura 3.1 se muestra el diagrama de conexión para la implementación de un servidor LDAP

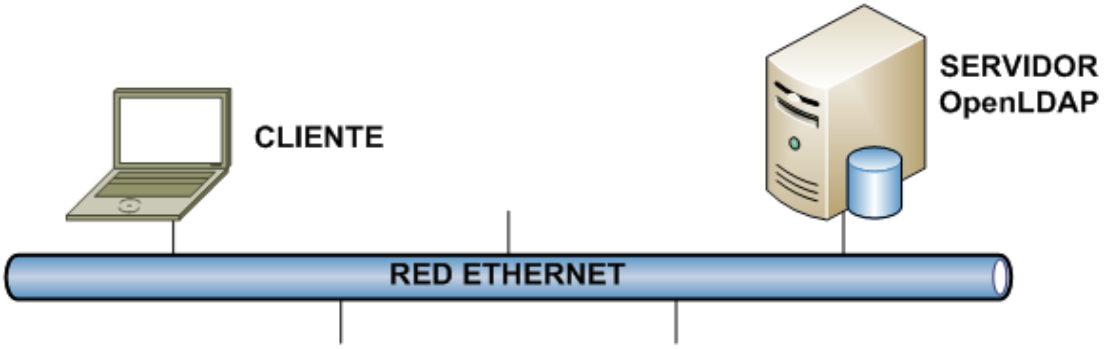

Figura 3.1.Diagrama de conexión.

La configuración de OpenLDAP requiere de:

- Obtención de Software.
- Configuración de la aplicación.

Para fines prácticos de este trabajo, están involucradas tres áreas o perfiles principales de una organización en donde se tiene acceso a la información, dichas áreas són:

- Ingeniería
- Proyectos
- Optimización

Dicho organigrama se muestra en la figura 3.2.

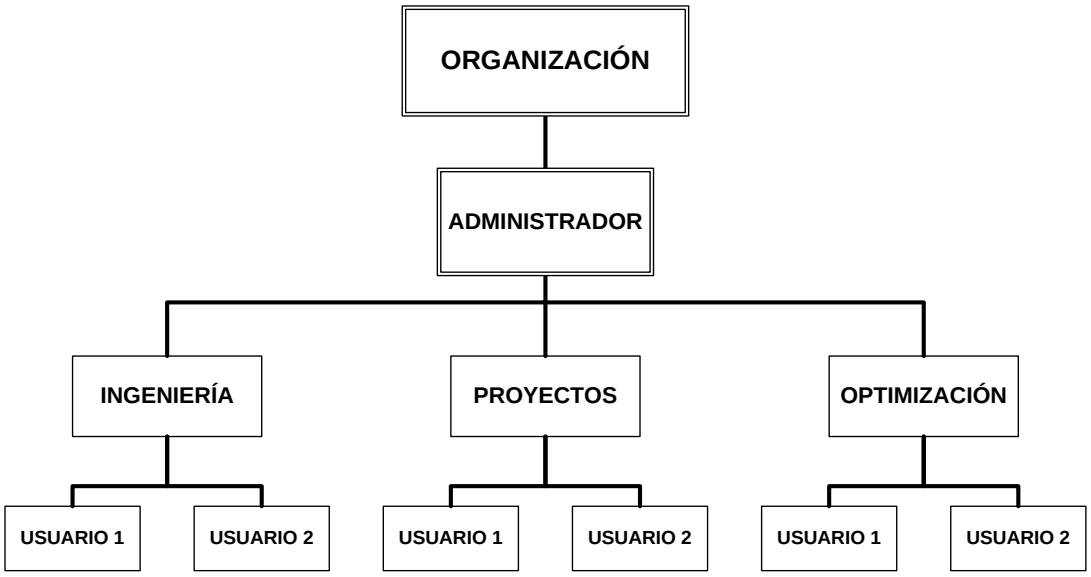

Figura 3.2.Organigrama de la organización.

3.1. Obtención de Software.

Open LDAP es un software libre, por lo tanto es posible poder descargar la aplicación OpenLDAP de la siguiente dirección electrónica: ***http://www.openldap.org/software/download/*** , este software es de uso libre, por lo cual no es necesario pagar por el costo de una licencia, una ventaja más del uso de este software.

OpenLDAP es un servidor que se distribuye bajo licencia GNU (Open Source), que permite que el software se pueda usar de forma gratuita, tanto de forma educativa como profesional. Además se dispone del código fuente para poder realizar modificaciones.

Existen varias versiones de OpenLDAP:

- OpenLDAP Release. (se recomienda verificar la última versión a instalar)
- OpenLDAP Stable Release. (la versión más fiable)
- OpenLDAP Test Releases. (normalmente es una versión de prueba)

Los paquetes que incluyen las distribuciones de OpenLDAP son:

- Servidor LDAP (ldapd)
- Servidor de replicación LDAP (slurpd)
- Software Development Kit (ldap)

El software OpenLDAP funciona en los siguientes sistemas operativos:

- Apple Mac OS X
- Linux: Debian, Red Hat,Suse, Fedora, Mandrake
- Free BSD
- IBM AIX
- Microsoft Windows 2000/NT
- Net BSD
- Solaris

El paquete que incorpora el servidor de OpenLDAP se denomina *openldap-servers* una vez descargado e instalado, este paquete instala los ficheros de

configuración por defecto bajo el directorio /etc/openldap, en este directorio se encuentran los archivos necesarios para poder realizar la configuración de la aplicación, en las líneas siguientes se puede ver el contenido del directorio /etc/openldap.

```
[root@localhost openldap]# ls -l
total 1088
-rw-r--r-- 1 root root      0 Apr 18 19:00 1
-rw-r--r-- 1 root root    368 Apr 19 19:03 adduser1.1.ldif
-rw-r--r-- 1 root root     86 Apr 19 18:42 adduser1.ldif
-rw-r--r-- 1 root root    180 Apr 19 18:40 adduser.ldif
-rw-r--r-- 1 root root    180 Apr 19 18:41 adduser.ldif.bak
-rw-r--r-- 1 root root    118 Apr 26 11:21 agrega1.ldif
-rw-r--r-- 1 root root    361 Apr 26 13:03 agrega2.ldif
-rw-r--r-- 1 root root    135 May  2 12:15 agrega_areas.ldif
-rw-r--r-- 1 root root    311 May  2 12:22 agrega_user1_ing.ldif
drwxr-xr-x 5 ldap ldap   4096 Apr 28 19:15 backup
-rw-r--r-- 1 root root    368 Apr 19 19:01 base_structure1.ldif
-rw-r--r-- 1 root root    180 Apr 19 18:53 base_structure_areas.ldif
-rw-r--r-- 1 root root    180 Apr 19 18:53 base_structure.ldif
drwxr-xr-x 2 ldap ldap   4096 Feb 28 11:29 cacerts
-rw-r--r-- 1 root root     31 Apr 19 18:08 creabd.ldif
-rw-r--r-- 1 root root    438 Apr 19 18:07 creabd.ldif.bak
-rw-r--r-- 1 root root  19762 Apr 19 19:36 create_schema1.ldif
-rwxr-x--x 1 ldap ldap    886 Apr 11 20:04 DB_CONFIG
-rwxr-x--x 1 ldap ldap    886 Apr 11 20:04 DB_CONFIG.bak
-rwxr-x--x 1 ldap ldap    921 Feb 28 11:18 DB_CONFIG.example
-rw-r--r-- 1 root root    163 Apr 13 18:56 init.ldif
-rw-r--r-- 1 root root    165 Apr 18 18:56 init.ldif1
-rwxr-xr-x 1 ldap ldap    327 Apr  4 16:12 ldap.conf
-rw-r--r-- 1 root root 839680 Apr 28 19:15 openldapcfg.tar
-rw-r--r-- 1 root root    442 Apr 28 13:12 perm1.ldif
```

```
-rw-r--r-- 1 root root    193 Apr 18 18:47 prueba.ldif
drwxr-xr-x 3 ldap ldap   4096 Apr 26 12:58 schema
drwxr-xr-x 2 root root   4096 Apr 25 17:26 schemaU
-rwxr-xr-x 1 ldap ldap   2837 Apr 29 18:19 slapd.conf
-rwxr-xr-x 1 root root   2101 Apr 19 17:33 slapd.conf.19abr2011
-rwxr-xr-x 1 root root   2101 Apr 25 17:34 slapd.conf.25abr2011
-rwxr-xr-x 1 root root   2230 Apr 26 13:39 slapd.conf.26abr2011
-rwxr-xr-x 1 root root   2342 Apr 27 13:13 slapd.conf.27abr2011
-rwxr-xr-x 1 root root   2449 Apr 28 12:36 slapd.conf.28abr2011
-rwxr-xr-x 1 root root   2101 Apr 18 18:08 slapd.conf.bak
-rwxr-xr-x 1 ldap ldap   2102 Apr 11 20:09 slapd.conf.old
-rwxr-x--x 1 ldap ldap   3801 Feb 28 11:18 slapd.conf.rpmnew
```

También se incorpora el servicio o "demonio" de LDAP, denominado *slapd.* Curiosamente, el nombre del servicio presente en */etc/rc.d/init.d,* y que se utiliza para iniciar y parar el demonio, se denomina *ldap* y no *slapd*.

3.2. Base de datos.

Para poder construir la base de datos del directorio OpenLDAP, es necesario emplear una base de datos primaria, una de las herramientas que permite hacer esto es *Berkeley DB Sleepycat*. Esta Base de Datos es simple rápida y segura, con poca administración, debido a que funciona como una biblioteca que se enlaza directamente en la aplicación, eliminando la penalización en el rendimiento de los sistemas cliente-servidor y el procesamiento SQL, ideal para consultas estáticas sobre datos dinámicos.

Las principales características de Berkeley son:

- Recuperación de datos de forma secuencial e indexada.
- Procesos múltiples por aplicación e hilos múltiples por proceso.
- Datos de memoria, en disco o ambos.
- Cifrado de datos por el algoritmo AES.
- Registros de hasta 4 GB y tablas de hasta 256 TB.
- Soporte para transacciones distribuidas.
- Respaldos en frio y en caliente.
- Replicación.

3.3. Instalación de un Servidor OpenLDAP.

Si el software ha sido probado con éxito, está listo para instalarse. Para lo cual es necesario tener permiso de escritura en los directorios de instalación. Por defecto OpenLDAP se instala en ***/ usr / local.***

```
[root@localhost local]# pwd
/usr/local
[root@localhost local]# ls -l
total 104
drwxr-xr-x 6 root root 4096 Apr  5 18:54 BerkeleyDB.4.0
drwxr-xr-x 2 root root 4096 Apr 18 19:41 bin
drwxr-xr-x 3 root root 4096 Apr  5 18:51 db
drwxr-xr-x 2 root root 4096 Oct 10  2006 etc
drwxr-xr-x 2 root root 4096 Oct 10  2006 games
```

```
drwxr-xr-x 2 root root 4096 Oct 10  2006 include
drwxr-xr-x 3 root root 4096 Apr 18 19:41 lib
drwxr-xr-x 2 root root 4096 Oct 10  2006 lib64
drwxr-xr-x 2 root root 4096 Oct 10  2006 libexec
drwxr-xr-x 3 root root 4096 Apr  5 19:04 openldap
drwxr-xr-x 2 root root 4096 Oct 10  2006 sbin
drwxr-xr-x 5 root root 4096 Apr 18 19:41 share
drwxr-xr-x 2 root root 4096 Oct 10  2006 src
```

Normalmente, la instalación requiere privilegios de superusuario.

3.4. Configuración del Servidor OpenLDAP.

Cuando el software se haya compilado e instalado, es posible proceder con la configuración directamente en el servidor.

Toda la configuración se realiza mediante el fichero slapd.conf, que se instala en el directorio que haya especificado; en caso de que no se especifique ninguno, de manera predeterminada se instala en: ***/usr/local/etc/openldap***, en este archivo por ejemplo se dan de alta los esquemas, tal como se muestra a continuación:

```
Include         /etc/openldap/schema/core.schema
Include         /etc/openldap/schema/cosine.schema
Include         /etc/openldap/schema/nis.schema
```

Estos esquemas se definen de acuerdo a la siguiente tabla:

Tabla 3.1 Relación de Esquemas

Archivo	Descripción
`core.schema`	Define la estructura primaria de XML para almacenar objetos
`cosine.schema`	Especifica un protocolo de normas para la comunidad de internet.
`nis.schema`	*Network Information Services.* Define los atributos de red para OpenLDAP (IP del Servidor, puerto TCP, etc.)

El archivo *slapd.conf* está compuesto por una serie de opciones globales de configuración que afectan a *slapd* en su conjunto.

El formato general del fichero *slapd.conf* es el siguiente:

```
[root@localhost openldap]# cat slapd.conf
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/nis.schema

# Define global ACLs to disable default read access.

# Load dynamic backend modules:
# modulepath    %MODULEDIR%
# moduleload    back_bdb.la
# moduleload    back_hdb.la
# moduleload    back_ldap.la

# Sample security restrictions
#       Require integrity protection (prevent hijacking)
#       Require 112-bit (3DES or better) encryption for updates
```

```
#       Require 63-bit encryption for simple bind
# security ssf=1 update_ssf=112 simple_bind=64

# Sample access control policy:
#       Root DSE: allow anyone to read it
#       Subschema (sub)entry DSE: allow anyone to read it
#       Other DSEs:
#               Allow self write access
#               Allow authenticated users read access
#               Allow anonymous users to authenticate
#       Directives needed to implement policy:
access to dn.base="dc=astelecom,dc=com" by * read
access to dn.base="ou=People,dc=astelecom,dc=com" by * write
access to dn="ou=People,dc=astelecom,dc=com" attrs=userPassword
   by self write
   by * auth
#
# if no access controls are present, the default policy
# allows anyone and everyone to read anything but restricts
# updates to rootdn.  (e.g., "access to * by * read")
#
# rootdn can always read and write EVERYTHING!

#######################################################################
#
# BDB database definitions
#######################################################################
#

database        bdb
suffix          "dc=astelecom,dc=com"
rootdn          "cn=Manager,dc=astelecom,dc=com"
# Cleartext passwords, especially for the rootdn, should
# be avoid.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
```

```
rootpw          {SSHA}+B7CjCDT+HlBlx/Dg71YfdykTHDbdb7V
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
directory       /var/lib/ldap/openldap-data
# Indices to maintain
index   objectClass     eq
index   uid     eq
loglevel 128
olcAccess: to * by * write
olcAccess: to * by * search
olcSuffix: "dc=astelecom,dc=com"
olcSuffix: "ou=People,dc=astelecom,dc=com"
[root@localhost openldap]#
```

A continuación se describe el fichero de configuración. Primeramente, la parte de configuración global, donde se indican los esquemas que debe de seguir OpenLDAP por medio de la orden ***include***:

```
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/nis.schema
```

La primera línea define la versión de LDAP, los RFCs como por ejemplo el 1274 (uid/dc), el 2079 (URI), el 2247 (dc/dcObject), etc. ,la segunda línea contiene el esquema derivado de X.500 o también llamado arquitectura de nombres y por último la tercera línea contiene la información referente al esquema de comunicaciones.

En la sección global, se definen las listas de acceso:

```
Sample access control policy:
#       Root DSE: allow anyone to read it
#       Subschema (sub)entry DSE: allow anyone to read it
#       Other DSEs:
#               Allow self write access
#               Allow authenticated users read access
#               Allow anonymous users to authenticate
#       Directives needed to implement policy:
access to dn.base="dc=astelecom,dc=com" by * read
access to dn.base="ou=People,dc=astelecom,dc=com" by users write
access to dn="ou=People,dc=astelecom,dc=com" attrs=userPassword
   by self write
   by * auth
#
# if no access controls are present, the default policy
# allows anyone and everyone to read anything but restricts
# updates to rootdn.  (e.g., "access to * by * read")
#
# rootdn can always read and write EVERYTHING!
```

A continuación se tiene la definición de la base de datos que se va a usar, y la definición de la jerarquía que se usa, o sea, compañía.com.

```
#######################################################################
#####
# BDB database definitions
#######################################################################
#####

database        bdb
suffix          "dc=astelecom,dc=com"
rootdn          "cn=Manager,dc=astelecom,dc=com"
```

Para agregar la unidad organizativa (Ingenieria)

```
dn: ou=Ingenieria, dc=astelecom, dc=com
ou: Ingenieria
objectClass: top
objectClass: organizationalUnit
```

Para agregar la unidad organizativa (Proyectos)

```
dn: ou=Proyectos, dc=astelecom, dc=com
ou: Proyectos
objectClass: top
objectClass: organizationalUnit
```

Para agregar la unidad organizativa (Optimización)

```
dn: ou=Optimización, dc=astelecom, dc=com
ou: Optimización
objectClass: top
objectClass: organizationalUnit
```

Para agregar los usuarios de los departamentos (Solo se pondrá un usuario de cada departamento, para que sirva como muestra representativa)

```
dn: uid=Adrian, ou=Ingenieria, ou=People, dc=astelecom, dc=com
uid: Adrian
gidNumber: 517
uidNumber: 517
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
sn: Robles
cn: Adrian Robles
mail: adrian.robles@astelecom.com
```

```
homeDirectory: /home/adrian
loginShell: /bin/bash

dn: uid=Juan, ou=Proyectos, ou=People, dc=astelecom, dc=com
uid: Juan
gidNumber: 520
uidNumber: 520
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
sn: Perez
cn: Juan Perez
mail: juan.perez@astelecom.com
homeDirectory: /home/juan
loginShell: /bin/bash

dn: uid=Israel, ou=Optimización, ou=People, dc=astelecom, dc=com
uid: Israel
gidNumber: 525
uidNumber: 525
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
sn: Castelan
cn: Israel Castelan
mail: Israel.castelan@astelecom.com
homeDirectory: /home/israel
loginShell: /bin/bash
```

En los siguientes renglones se especifica que la contraseña usada para el administrador en este caso está cifrada, es muy aconsejable mantenerla de esta forma debido a que si alguien logra acceder al directorio de contraseñas, no conozca el contenido de las mismas.

```
# Cleartext passwords, especially for the rootdn, should
# be avoid.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw          {SSHA}+B7CjCDT+HlBlx/Dg71YfdykTHDbdb7V
```

Para garantizar la mayor seguridad, es necesario tomar en consideración los siguientes puntos:

- Evitar el uso de LDAPv2, usar LDAPv3 de preferencia.
- Evitar accesos anónimos.
    - disallow anonymous
    - requiere authc
- Establecer límites en el número de entradas que será devuelto al realizar una consulta, por defecto son 500, al bajar este valor se evita cargar al servidor y se incrementa el trabajo a los atacantes que usan ldap injection.
    - Sizelimit

- Establecer tiempos de espera para forzar la desconexión después de un tiempo “idle”, con esto se evita posibles ataques de negación de servicio.
    - Idletimeout
- Reducir los límites de los tiempos de búsqueda que por defecto son 3600 segundos, con esto reducimos el ataque por ldap injection.
    - Timelimit

Por último se tiene el directorio de trabajo:

```
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
directory       /var/lib/ldap/openldap-data
```

CAPÍTULO IV. PRUEBAS DE UN SERVIDOR OpenLDAP

Un servidor con la aplicación OpenLDAP es capaz de poder realizar dos funciones básicas de un sistema de seguridad.

En este capítulo se muestran las diferentes pruebas realizadas al sistema completo, es decir al modelo cliente/servidor.

Se puede ver de una manera muy grafica los resultados derivados de cada una de las acciones hechas para poder demostrar la funcionalidad del modelo.

La aplicación que se planea dar a este sistema en su conjunto es la de poder autorizar a los usuarios dentro de una red local, para lo cual se usa la base de datos que se genera en LDAP (Previamente se deben de dar de alta los usuarios en esta base), lo que debe de hacer el usuario es poner en la pantalla de inicio su usuario y su contraseña, si esta información se encuentra en la base, el

usuario es permitido y ya estando validado en el sistema, puede hacer uso de los recursos.

4.1. Hardware y Software usado.

Para el desarrollo de las pruebas fue necesario instalar en un servidor con sistema operativo Linux la herramienta conocida como OpenLDAP que será con la cual se valida a los usuarios en la base de datos contenida en el mismo.

En la Tabla 4.1 se detallan las características del servidor:

Tabla 4.1 Características del servidor

	HARDWARE	**SOFTWARE**
SERVIDOR	INTEL PENTIUM 4 CELERON DE 2.66GHZ CON PLACA INTEL945 MEMORIA RAM DE 512 DDR-2 DISCO DURO DE 80 GB MULTIGRABADOR DE DVDs/CDs 8 PUERTOS USB 2 FRONTALES 6 POSTERIORES PUERTO DE RED ETHERNET 10/100 VIDEO INTEGRADO DE 224MB	LINUX RED HAT VERSION 2.6.18-53.el5xen
		SAMBA SERVER CONFIGURATION TOOL VERSION 1.2.39
		LDAP MANAGEMENT MADE EASY LUMA 2.4

A este equipo fue necesario el instalarle una serie de paquetería para poder probar la funcionalidad completa, tales como:

- Linux Red Hat.- Es un núcleo de sistema operativo libre, está licenciado bajo la GPL v2 y está desarrollado por colaboradores de todo el mundo.
- Samba.- Es un software que permite al servidor con Linux poder compartir archivos e impresoras con otras computadoras en una misma red local. Utiliza para ello un protocolo conocido como SMB/CIFS compatible con sistemas operativos Windows (XP, NT, 98...), OS/2 o incluso DOS.
- Luma.- Es una utilidad gráfica para acceder y gestionar los datos almacenados en los servidores LDAP.

Para el caso del cliente, se utilizó una computadora portátil tipo laptop marca DELL con una serie de características mencionadas en la Tabla 4.2:

Tabla 4.2 Características del cliente

	HARDWARE	**SOFTWARE**
CLIENTE	DELL MODELO PRECISION M4300 INTEL CORE 2 DUO 4 2 GHZ MEMORIA RAM DE 2GB	MICROSOFT WINDOWS XP PROFESIONAL VERSION 2002 SERVICE PACK 2

	DDR-2 DISCO DURO DE 110 GB MULTIGRABADOR DE DVDs/CDs 3PUERTOS USB PUERTO DE RED ETHERNET 100/1000	pGINA VERSION 1.8.8

Para el caso del cliente, se le instaló el siguiente software:

- Windows XP.- Es un sistema operativo desarrollado por Microsoft, que controla la actividad general de la computadora.
- pGina.- Es un sistema de autenticación abierta que sustituye a la construida en el sistema operativo Microsoft Windows.

4.2. Autenticación.

Es posible poder comprobar la autenticación a través del cliente y el servidor por medio de las siguientes pruebas, dado que en la base de datos del servidor OpenLDAP existe un usuario, y este usuario tiene asignado una contraseña, de tal manera que si un cliente se desea validar dentro de LDAP, será necesario que las credenciales de inicio correspondan a las almacenadas en la base de datos, por ejemplo en esta prueba se valido al usuario *jcperez* quien es un usuario que se encuentra dado de alta en el servidor, y debido a que la contraseña

que introdujo también es correcta se obtiene el siguiente resultado, mostrado en la figura 4.1,dentro de la maquina se crea un perfil personal del usuario autenticado, esta acción es mostrada en la figura 4.2.

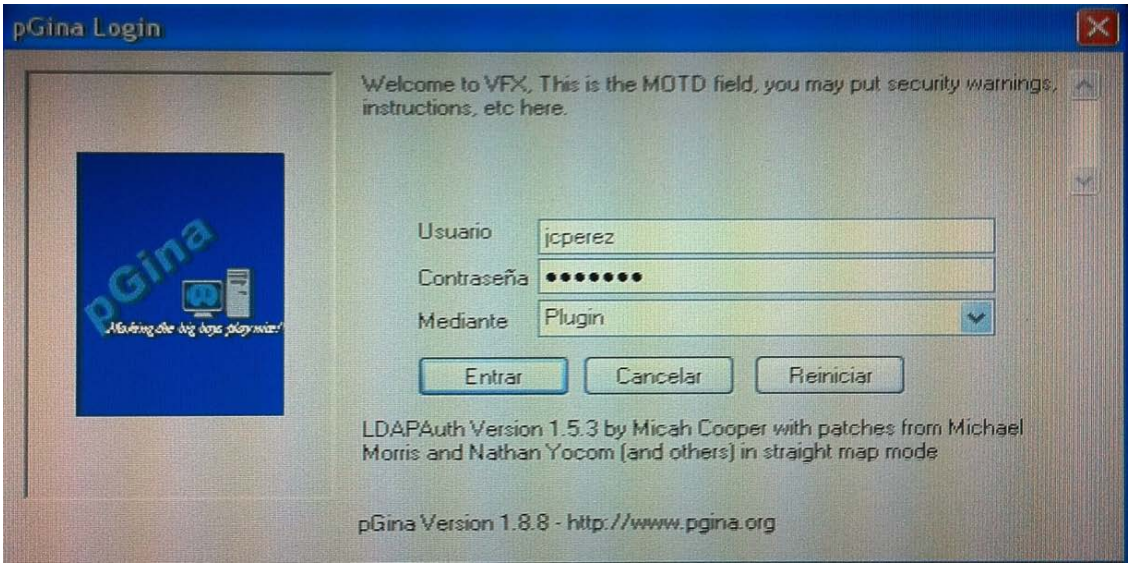

Figura 4.1 Usuario y contraseña correctos.

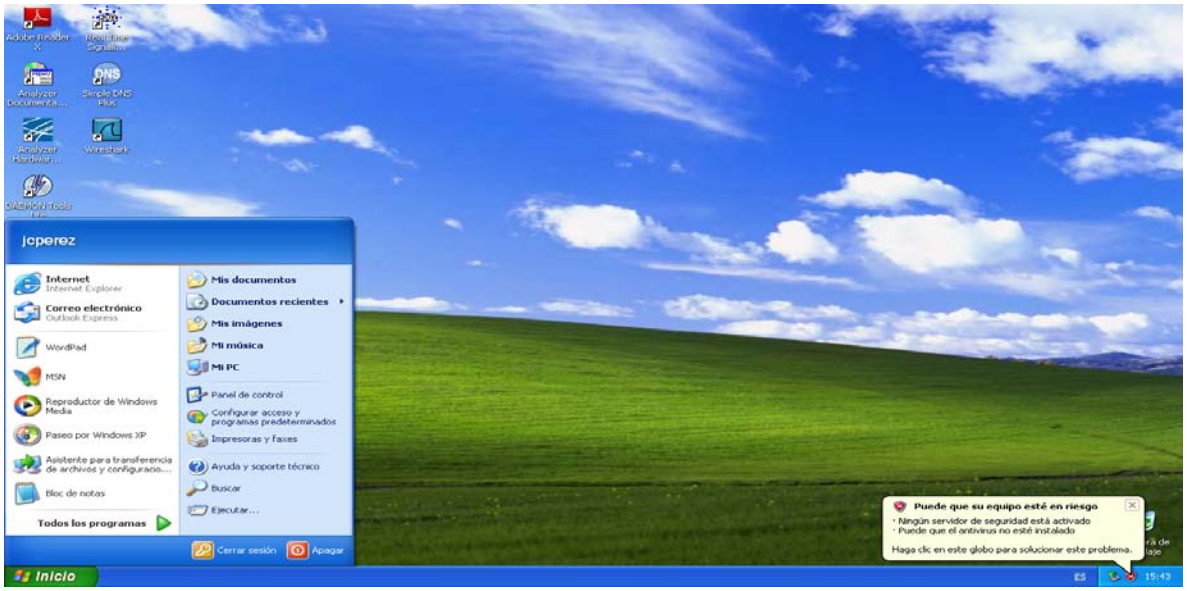

Figura 4.2 Perfil personal del usuario.

En caso de que las credenciales de inicio no correspondan, es decir si el usuario y/o contraseña no están en la base de datos, se tendrá el siguiente resultado:

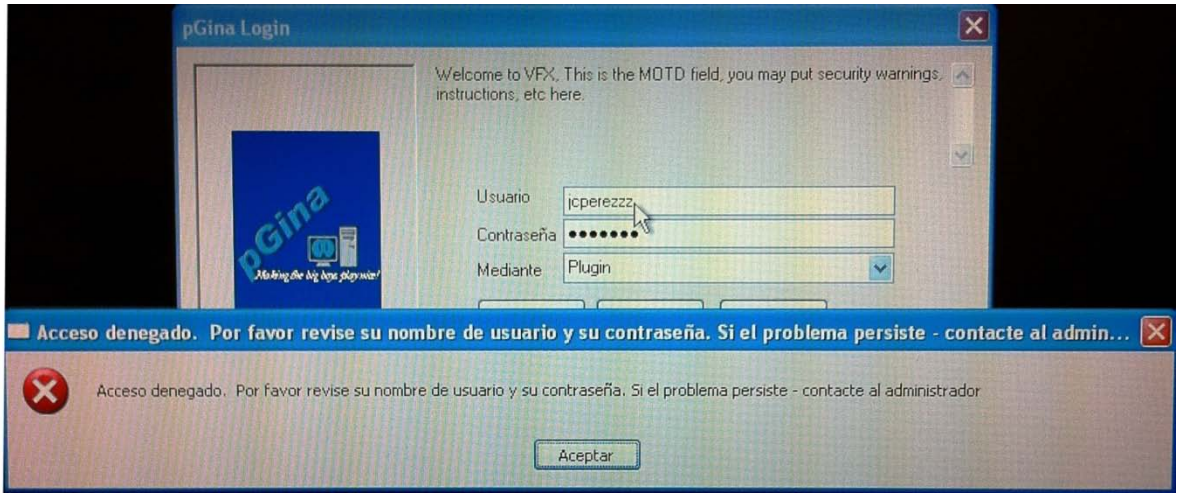

Figura 4.3 Usuario y/o contraseña no validos.

4.3. Autorización.

Es posible también poder comprobar la autorización a través del cliente y el servidor por medio de las siguientes pruebas, dado que en la base de datos del servidor existe un usuario con una contraseña y este a su vez tiene asignado uno o varios archivos en donde cuenta con los permisos de lectura y escritura, solo lectura o solo escritura, es posible por lo tanto verificar primero cuando el usuario cuenta con los privilegios de lectura y escritura de algún o algunos archivos, esto se muestra en las figuras 4.4 y 4.5.

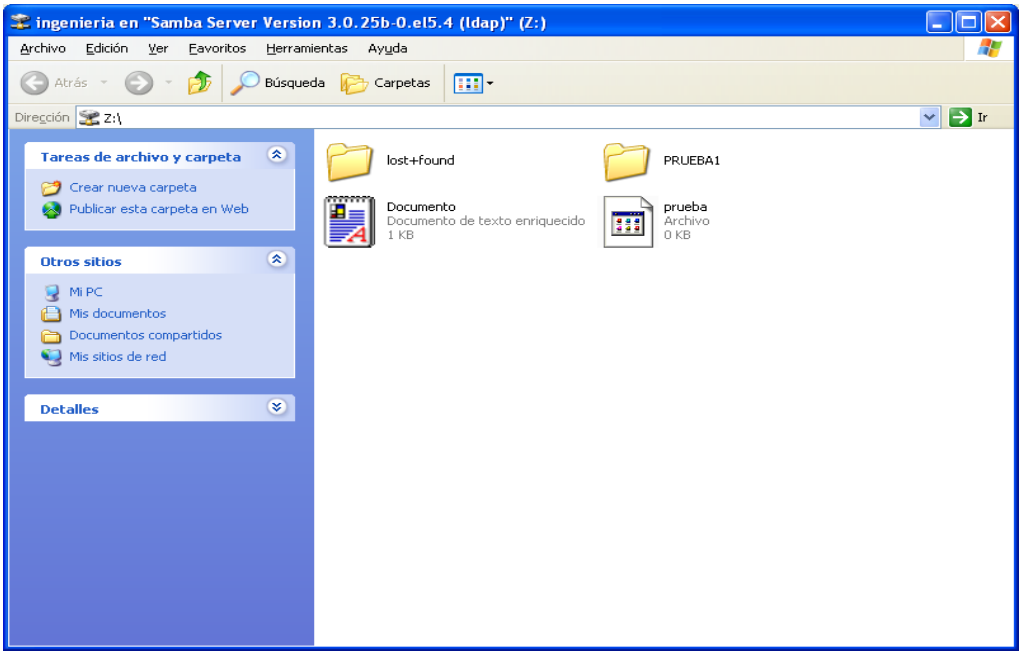

Figura 4.4 Usuario con permiso de lectura.

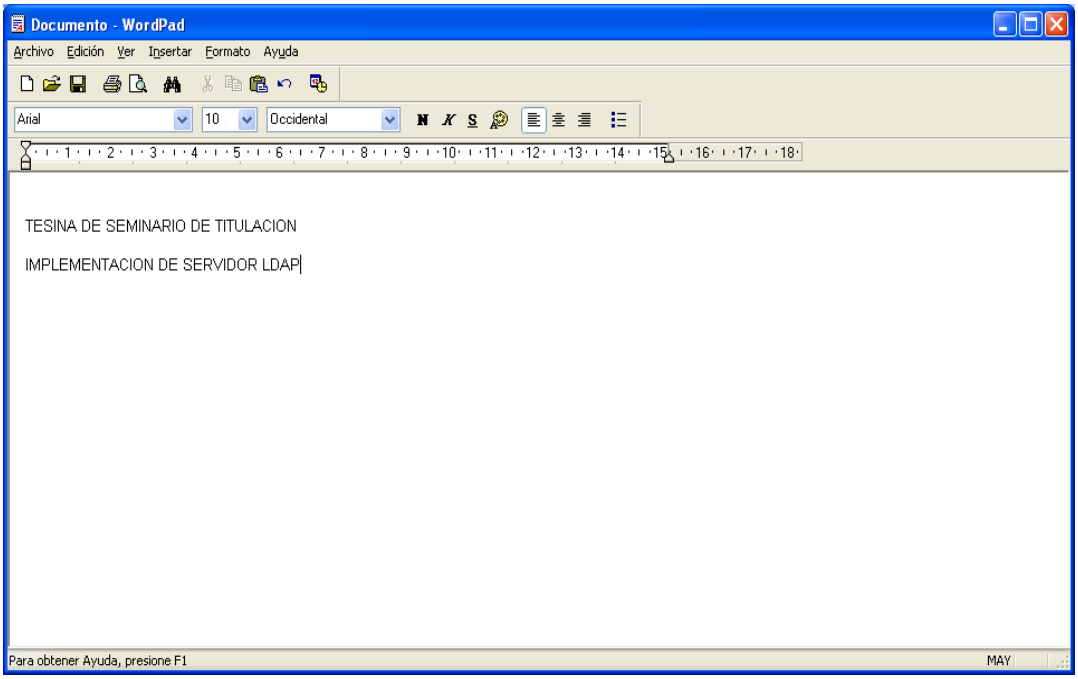

Figura 4.5 Usuario con permiso de escritura

Por último se muestra un usuario que si es válido dentro de la base de datos de OpenLDAP y que no cuenta con el permiso de lectura, por lo que no es capaz ni siquiera de mapear la unidad, esto es mostrado en la figura 4.6, y el resultado obtenido por problemas de permisos se muestra en el mensaje de error obtenido en la figura 4.7

Figura 4.6 Usuario intentando mapear la unidad “z”.

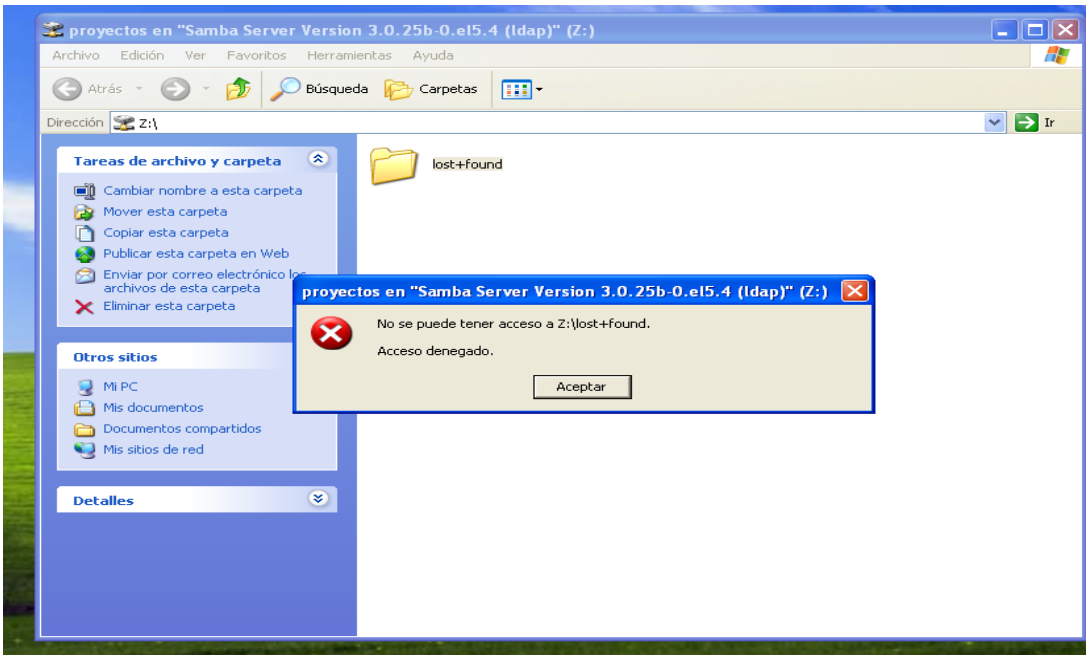

Figura 4.7 Mensaje de error por permisos.

Es de hacer notar que con las pruebas realizadas, es posible probar dos elementos básicos usados en un sistema de seguridad, como es el caso de la autenticación, proceso de permitir o no el acceso a un usuario al sistema, por, medio de un usuario y contraseña.

El siguiente elemento es el de poder autorizar a los usuarios a hacer uso de los recursos, con estas dos acciones simples se logra el objetivo de mitigar el riesgo de alguna alteración no deseada a un sistema con archivos.

CONCLUSIONES.

El por qué se decidió usar Open LDAP, es que además de poder contar con la capacidad de crear directorios con permisos, permite manejar una base de datos en donde se encuentran contenidos los usuarios y contraseñas. De tal manera se puede cumplir con dos requisitos indispensables en un sistema de seguridad, los cuales son la autenticación y la autorización.

En esta ocasión los objetivos fijados fueron cubiertos en su totalidad, ya que se requería contar con un sistema capaz de validar a los usuarios y asignar permisos a estos usuarios ya validados.

Con este sistema pudimos lograr que los usuarios que existen en la base de datos de LDAP cuenten con sus respectivas contraseñas y a su vez asignarles los permisos respectivos.

Otra ventaja más de LDAP es la facultad de usar maquinas en forma de grupo, esto se obtiene debido a que la validación del usuario será realizada por el servidor, ya que el servidor cuenta con una base de datos local en donde es necesario que el usuario que intente acceder a los servicios de archivos, sea un usuario valido, es decir que el nombre y la contraseña existan en esta base de información.

Con el intercambio de credenciales y la validación de las mismas, el sistema crea un perfil de usuario con los permisos que tiene asignado, de esta forma se consigue que cualquier persona puede usar cualquier computadora que este libre para trabajar, esto es que las maquinas ya no deben de ser 100% dedicadas; con lo cual se puede obtener un gran ahorro en cuanto al costo de operación de alguna organización. Aunado a todo esto las licencias de uso de este sistema son libres, es decir, no pagamos por ellas.

Es posible poder usar ambientes gráficos para la operación del sistema, con lo cual la parte administrativa se vuelve más fácil de llevar a cabo, en el sistema se empleo una herramienta llamada luma, la cual también es de uso libre.

REFERENCIAS.

[1] http://searchwinit.techtarget.com/definition/directory

[2] Atif Ghaffar. **"Introducción a LDAP sobre Linux"** Linux Focus. Primera Impresión 2001.

[3] Simmons Curt, "**Active Directory Bible**" IDG Books, First Edition USA 2001.

[4] Ehlenberger Ami, Gorthi Ramakrishna, Leiserson Jay, Macbeth Richard, Owen Nathan, Ranahandola Sunil, Storrs Michael, Tuttle Steven, Yang Chunhui."**Understanding LDAP Design and Implementation"** IBM Corp. Second Edition USA 2004.

[5] Jose Manuel Suárez, "**Curso OpenLDAP**" GOA, Versión 3 1/2010.

[6]Good Gordon, Howes Tim, Smith Mark "**Understanding and Deploying LDAP Directory Services"** Pearson Education, Inc. Second Edition USA 2003.

[7] http://www.openldap.org/doc/admin24/intro.html

[8] http://www.ldapman.org/articles/sp_intro.html

[9]Facundo Hector, **"La Biblia de Linux"** MP Ediciones S.A. Primera Impresión. Buenos Aires Argentina 2003.

[10] Esp. Lidia Prudente **"Seminario Seguridad de la Información"** Mexico D.F

GLOSARIO.

LDAP Lightweight Directory Access Protocol Protocolo ligero de acceso a Directorios. Hace referencia a un protocolo a nivel de aplicación el cual permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

Pool Agrupamiento, en este caso de computadoras que pueden ser compartidas.

Sistema Operativo Windows Es el sistema operativo de mayor difusión entre computadoras personales, servidores pequeños y medianos. Fue desarrollado por Microsoft, por lo cual es una marca registrada de este proveedor.

Sistema Operativo Linux Linux es un sistema operativo diseñado por cientos de programadores de todo el planeta, aunque el principal responsable del proyecto es LinusTovalds. Su objetivo inicial es propulsar el software de libre distribución junto con su código fuente para que pueda ser modificado por cualquier persona,

SQL El lenguaje de consulta estructurado o SQL (por sus siglas en inglés Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales.

ANEXO.

"INFORMACION DEL SW INSTALADO"

```
[root@localhost shell]# rpm -q --info openldap
Name        : openldap                     Relocations: (not relocatable)
Version     : 2.3.43                            Vendor: Red Hat, Inc.
Release     : 12.el5_6.7                    Build Date: Mon 28 Feb 2011
11:18:57 AM CST
Install Date: Tue 12 Apr 2011 05:50:24 PM CDT      Build Host: x86-
002.build.bos.redhat.com
Group       : System Environment/Daemons    Source RPM: openldap-2.3.43-
12.el5_6.7.src.rpm
Size        : 614903                           License: OpenLDAP
Signature   : DSA/SHA1, Mon 07 Mar 2011 05:18:15 AM CST, Key ID
5326810137017186
Packager    : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
URL         : http://www.openldap.org/
Summary     : The configuration files, libraries, and documentation for
OpenLDAP.
Description :
OpenLDAP is an open source suite of LDAP (Lightweight Directory Access
Protocol) applications and development tools. LDAP is a set of
protocols for accessing directory services (usually phone book style
information, but other information is possible) over the Internet,
similar to the way DNS (Domain Name System) information is propagated
over the Internet. The openldap package contains configuration files,
libraries, and documentation for OpenLDAP.
```

"ARCHIVOS QUE REQUIERE OPENLDAP"

```
[root@localhost /]# rpm -q --requires openldap
/sbin/ldconfig
/sbin/ldconfig
config(openldap) = 2.3.43-12.el5_6.7
glibc >= 2.2.3-48
libc.so.6()(64bit)
libc.so.6(GLIBC_2.2.5)(64bit)
libc.so.6(GLIBC_2.3)(64bit)
libc.so.6(GLIBC_2.3.2)(64bit)
libc.so.6(GLIBC_2.3.4)(64bit)
libc.so.6(GLIBC_2.4)(64bit)
libcrypto.so.6()(64bit)
liblber-2.3.so.0()(64bit)
libldap-2.3.so.0()(64bit)
libldap_r-2.3.so.0()(64bit)
libresolv.so.2()(64bit)
libresolv.so.2(GLIBC_2.2.5)(64bit)
libsasl2.so.2()(64bit)
libssl.so.6()(64bit)
```

```
mktemp
rpmlib(CompressedFileNames) <= 3.0.4-1
rpmlib(PayloadFilesHavePrefix) <= 4.0-1
rtld(GNU_HASH)
/sbin/ldconfig
/sbin/ldconfig
config(openldap) = 2.3.43-12.el5_6.7
glibc >= 2.2.3-48
libc.so.6
libc.so.6(GLIBC_2.0)
libc.so.6(GLIBC_2.1)
libc.so.6(GLIBC_2.1.2)
libc.so.6(GLIBC_2.1.3)
libc.so.6(GLIBC_2.3)
libc.so.6(GLIBC_2.3.2)
libc.so.6(GLIBC_2.3.4)
libc.so.6(GLIBC_2.4)
libcrypto.so.6
liblber-2.3.so.0
libldap-2.3.so.0
libldap_r-2.3.so.0
libresolv.so.2
libresolv.so.2(GLIBC_2.2)
libsasl2.so.2
libssl.so.6
mktemp
rpmlib(CompressedFileNames) <= 3.0.4-1
rpmlib(PayloadFilesHavePrefix) <= 4.0-1
rtld(GNU_HASH)
```

"CONFIGURACION DE SLAPD.CONF"

```
[root@localhost openldap]# cat slapd.conf
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/nis.schema
include         /etc/openldap/schema/core.ldif

# Define global ACLs to disable default read access.

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral       ldap://root.openldap.org

pidfile         /var/lib/ldap/run/slapd.pid
argsfile        /var/lib/ldap/run/slapd.args
```

```
# Load dynamic backend modules:
# modulepath    %MODULEDIR%
# moduleload    back_bdb.la
# moduleload    back_hdb.la
# moduleload    back_ldap.la

# Sample security restrictions
#       Require integrity protection (prevent hijacking)
#       Require 112-bit (3DES or better) encryption for updates
#       Require 63-bit encryption for simple bind
# security ssf=1 update_ssf=112 simple_bind=64

# Sample access control policy:
#       Root DSE: allow anyone to read it
#       Subschema (sub)entry DSE: allow anyone to read it
#       Other DSEs:
#               Allow self write access
#               Allow authenticated users read access
#               Allow anonymous users to authenticate
#       Directives needed to implement policy:
access to dn.base="dc=astelecom,dc=com" by * read
access to dn.base="ou=People,dc=astelecom,dc=com" by * write
access to *
  by self write
  by users write
  by anonymous write
  by dn.exact="ou=People,dc=astelecom,dc=com" write
access to dn="ou=People,dc=astelecom,dc=com" attrs=userPassword
   by self write
   by dn.exact="ou=People,dc=astelecom,dc=com" write
   by * auth
#
# rootdn can always read and write EVERYTHING!

#######################################################################
# BDB database definitions
#######################################################################

database        bdb
suffix          "dc=astelecom,dc=com"
rootdn          "cn=Manager,dc=astelecom,dc=com"
# Cleartext passwords, especially for the rootdn, should
# be avoid.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw          {SSHA}+B7CjCDT+HlBlx/Dg71YfdykTHDbdb7V
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
directory       /var/lib/ldap/openldap-data
# Indices to maintain
index   objectClass     eq
index   uid     eq
loglevel 128
```

```
olcAccess: to * by * write
olcAccess: to * by * search
olcSuffix: "dc=astelecom,dc=com"
olcSuffix: "ou=People,dc=astelecom,dc=com"
[root@localhost openldap]#
```

"DIRECTORIOS DE OPENLDAP"

```
[root@localhost ~]# cd oldap
[root@localhost oldap]# ls -ltr
total 1728
-rwxrwxrwx 1 2000 2000   2214 Nov 24  2003 LICENSE
-rwxrwxrwx 1 2000 2000 249747 Oct 29  2005 aclocal.m4
-rwxrwxrwx 1 2000 2000   2345 Jan  4 17:49 COPYRIGHT
-rwxrwxrwx 1 2000 2000   3836 Jan  4 17:49 ANNOUNCEMENT
-rwxrwxrwx 1 2000 2000   3553 Jan  4 17:49 README
-rwxrwxrwx 1 2000 2000   1035 Jan  4 17:49 Makefile.in
-rwxrwxrwx 1 2000 2000   4414 Jan  4 17:49 INSTALL
-rwxrwxrwx 1 2000 2000  95115 Mar 23 19:18 configure.in
-rwxrwxrwx 1 2000 2000 696421 Mar 23 19:52 configure
-rwxrwxrwx 1 2000 2000  51836 Mar 25 15:08 CHANGES
-rwxrwxrwx 1 root root   5032 Apr  4 16:09 install.log.syslog
-rwxrwxrwx 1 root root  59106 Apr  4 16:12 install.log
-rwxrwxrwx 1 root root   1619 Apr  4 16:13 anaconda-ks.cfg
-rwxrwxrwx 1 root root    195 Apr  4 17:25 scsrun.log
drwxrwxrwx 2 root root   4096 Apr  4 17:26 Desktop
drwxrwxrwx 3 root root   4096 Apr  4 17:54 servers
drwxrwxrwx 8 root root   4096 Apr  4 17:54 libraries
drwxrwxrwx 3 root root   4096 Apr  4 17:54 include
drwxrwxrwx 7 root root   4096 Apr  4 17:54 contrib
drwxrwxrwx 3 root root   4096 Apr  4 17:54 clients
drwxrwxrwx 2 root root   4096 Apr  4 17:54 build
drwxrwxrwx 8 root root   4096 Apr  4 17:54 doc
-rwxr-xr-x 1 root root 209584 Apr  5 19:01 libtool
-rw-r--r-- 1 root root 179371 Apr  5 19:02 config.log
drwxrwxrwx 5 root root   4096 Apr 12 19:02 tests
[root@localhost oldap]#
```

"INSTRUCCIONES DE OPENLDAP"

```
[root@localhost slapd]# ls -ltr
total 2928
-rw-rw-rw- 1 2000 2000   2105 Feb 13  2007 slapd.conf~
-rwxrwxrwx 1 2000 2000   2596 Nov 20  2007 slapd.ldif
-rwxrwxrwx 1 2000 2000    921 Dec 18  2007 DB_CONFIG
-rwxrwxrwx 1 2000 2000  15895 Jan  4 17:50 Makefile.in
-rwxrwxrwx 1 2000 2000  23033 Jan  4 17:50 at.c
-rwxrwxrwx 1 2000 2000   2030 Jan  4 17:50 alock.h
-rwxrwxrwx 1 2000 2000  15501 Jan  4 17:50 alock.c
```

```
-rwxrwxrwx 1 2000 2000  30209 Jan  4 17:50 ad.c
-rwxrwxrwx 1 2000 2000  72375 Jan  4 17:50 aclparse.c
-rwxrwxrwx 1 2000 2000  44738 Jan  4 17:50 aci.c
-rwxrwxrwx 1 2000 2000   4067 Jan  4 17:50 abandon.c
-rwxrwxrwx 1 2000 2000  30454 Jan  4 17:50 backover.c
-rwxrwxrwx 1 2000 2000   3858 Jan  4 17:50 cancel.c
-rwxrwxrwx 1 2000 2000  12541 Jan  4 17:50 bind.c
-rwxrwxrwx 1 2000 2000   6283 Jan  4 17:50 config.h
-rwxrwxrwx 1 2000 2000   2454 Jan  4 17:50 component.h
-rwxrwxrwx 1 2000 2000  32132 Jan  4 17:50 component.c
-rwxrwxrwx 1 2000 2000  10575 Jan  4 17:50 compare.c
-rwxrwxrwx 1 2000 2000   2978 Jan  4 17:50 ch_malloc.c
-rwxrwxrwx 1 2000 2000  25089 Jan  4 17:50 entry.c
-rwxrwxrwx 1 2000 2000   6703 Jan  4 17:50 delete.c
-rwxrwxrwx 1 2000 2000   5422 Jan  4 17:50 ctxcsn.c
-rwxrwxrwx 1 2000 2000  10519 Jan  4 17:50 cr.c
-rwxrwxrwx 1 2000 2000  50531 Jan  4 17:50 controls.c
-rwxrwxrwx 1 2000 2000  32112 Jan  4 17:50 limits.c
-rwxrwxrwx 1 2000 2000   7727 Jan  4 17:50 ldapsync.c
-rwxrwxrwx 1 2000 2000   6882 Jan  4 17:50 init.c
-rwxrwxrwx 1 2000 2000   2496 Jan  4 17:50 index.c
-rwxrwxrwx 1 2000 2000   1175 Jan  4 17:50 globals.c
-rwxrwxrwx 1 2000 2000   4817 Jan  4 17:50 frontend.c
-rwxrwxrwx 1 2000 2000  24254 Jan  4 17:50 filterentry.c
-rwxrwxrwx 1 2000 2000  12645 Jan  4 17:50 mr.c
-rwxrwxrwx 1 2000 2000   6197 Jan  4 17:50 mra.c
-rwxrwxrwx 1 2000 2000   8159 Jan  4 17:50 module.c
-rwxrwxrwx 1 2000 2000  12012 Jan  4 17:50 mods.c
-rwxrwxrwx 1 2000 2000  27523 Jan  4 17:50 modify.c
-rwxrwxrwx 1 2000 2000   7237 Jan  4 17:50 matchedValues.c
-rwxrwxrwx 1 2000 2000  25206 Jan  4 17:50 main.c
-rwxrwxrwx 1 2000 2000   2135 Jan  4 17:50 lock.c
-rwxrwxrwx 1 2000 2000  10600 Jan  4 17:50 phonetic.c
-rwxrwxrwx 1 2000 2000  14493 Jan  4 17:50 passwd.c
-rwxrwxrwx 1 2000 2000   5148 Jan  4 17:50 operation.c
-rwxrwxrwx 1 2000 2000   2169 Jan  4 17:50 operational.c
-rwxrwxrwx 1 2000 2000   5209 Jan  4 17:50 oidm.c
-rwxrwxrwx 1 2000 2000  19037 Jan  4 17:50 oc.c
-rwxrwxrwx 1 2000 2000   2776 Jan  4 17:50 nt_svc.c
-rwxrwxrwx 1 2000 2000  21097 Jan  4 17:50 schema_check.c
-rwxrwxrwx 1 2000 2000   4343 Jan  4 17:50 schema.c
-rwxrwxrwx 1 2000 2000  47679 Jan  4 17:50 sasl.c
-rwxrwxrwx 1 2000 2000  11651 Jan  4 17:50 root_dse.c
-rwxrwxrwx 1 2000 2000   7328 Jan  4 17:50 referral.c
-rwxrwxrwx 1 2000 2000  17435 Jan  4 17:50 sl_malloc.c
-rwxrwxrwx 1 2000 2000   2393 Jan  4 17:50 sets.h
-rwxrwxrwx 1 2000 2000  18997 Jan  4 17:50 sets.c
-rwxrwxrwx 1 2000 2000  11052 Jan  4 17:50 search.c
-rwxrwxrwx 1 2000 2000   9915 Jan  4 17:50 schemaparse.c
-rwxrwxrwx 1 2000 2000   2520 Jan  4 17:50 slapindex.c
-rwxrwxrwx 1 2000 2000   2270 Jan  4 17:50 slapdn.c
-rwxrwxrwx 1 2000 2000   3452 Jan  4 17:50 slapcommon.h
-rwxrwxrwx 1 2000 2000  20119 Jan  4 17:50 slapcommon.c
```

```
-rwxrwxrwx 1 2000 2000   3882 Jan  4 17:50 slapcat.c
-rwxrwxrwx 1 2000 2000   3820 Jan  4 17:50 slapauth.c
-rwxrwxrwx 1 2000 2000  15097 Jan  4 17:50 slapadd.c
-rwxrwxrwx 1 2000 2000   9273 Jan  4 17:50 slapacl.c
-rwxrwxrwx 1 2000 2000   2086 Jan  4 17:50 str2filter.c
-rwxrwxrwx 1 2000 2000   2911 Jan  4 17:50 starttls.c
-rwxrwxrwx 1 2000 2000   2419 Jan  4 17:50 slaptest.c
-rwxrwxrwx 1 2000 2000   3761 Jan  4 17:50 slapschema.c
-rwxrwxrwx 1 2000 2000   4668 Jan  4 17:50 slappasswd.c
-rwxrwxrwx 1 2000 2000  24509 Jan  4 17:50 zn_malloc.c
-rwxrwxrwx 1 2000 2000  18703 Jan  4 17:50 value.c
-rwxrwxrwx 1 2000 2000   3723 Jan  4 17:50 user.c
-rwxrwxrwx 1 2000 2000   1754 Jan  4 17:50 unbind.c
-rwxrwxrwx 1 2000 2000   5194 Jan  4 17:50 txn.c
-rwxrwxrwx 1 2000 2000   9543 Jan  4 17:50 syntax.c
-rwxrwxrwx 1 2000 2000  37195 Jan 26 17:23 backglue.c
-rwxrwxrwx 1 2000 2000  43453 Jan 26 17:23 backend.c
-rwxrwxrwx 1 2000 2000   9465 Jan 26 17:23 extended.c
-rwxrwxrwx 1 2000 2000  51946 Jan 26 17:23 connection.c
-rwxrwxrwx 1 2000 2000 103074 Jan 26 17:23 slap.h
-rwxrwxrwx 1 2000 2000  62629 Jan 26 17:59 acl.c
-rwxrwxrwx 1 2000 2000  46614 Jan 28 12:53 result.c
-rwxrwxrwx 1 2000 2000 141496 Jan 28 15:04 syncrepl.c
-rwxrwxrwx 1 2000 2000   3708 Jan 31 14:47 ava.c
-rwxrwxrwx 1 2000 2000  36317 Feb  2 15:26 filter.c
-rwxrwxrwx 1 2000 2000 171777 Feb  2 15:35 schema_init.c
-rwxrwxrwx 1 2000 2000  51280 Feb  4 14:37 saslauthz.c
-rwxrwxrwx 1 2000 2000  68999 Feb  4 15:03 proto-slap.h
-rwxrwxrwx 1 2000 2000  16019 Mar 23 19:44 modrdn.c
-rwxrwxrwx 1 2000 2000  27249 Mar 23 19:51 dn.c
-rwxrwxrwx 1 2000 2000  17169 Mar 23 19:56 add.c
-rwxrwxrwx 1 2000 2000  14911 Mar 23 19:56 attr.c
-rwxrwxrwx 1 2000 2000  48547 Mar 23 20:02 schema_prep.c
-rwxrwxrwx 1 2000 2000 191354 Mar 23 20:11 bconfig.c
-rwxrwxrwx 1 2000 2000  58059 Mar 23 20:22 config.c
-rwxrwxrwx 1 2000 2000  80488 Mar 24 12:22 daemon.c
drwxrwxrwx 2 root root   4096 Apr  4 17:54 slapi
drwxrwxrwx 2 root root   4096 Apr  4 17:54 shell-backends
drwxrwxrwx 2 root root   4096 Apr  4 17:54 schema
drwxrwxrwx 2 root root   4096 Apr  4 17:54 overlays
drwxrwxrwx 4 root root   4096 Apr  4 17:54 back-sql
drwxrwxrwx 2 root root   4096 Apr  4 17:54 back-sock
drwxrwxrwx 2 root root   4096 Apr  4 17:54 back-shell
drwxrwxrwx 2 root root   4096 Apr  4 17:54 back-relay
drwxrwxrwx 2 root root   4096 Apr  4 17:54 back-perl
drwxrwxrwx 2 root root   4096 Apr  4 17:54 back-passwd
drwxrwxrwx 2 root root   4096 Apr  4 17:54 back-null
drwxrwxrwx 2 root root   4096 Apr  4 17:54 back-ndb
drwxrwxrwx 2 root root   4096 Apr  4 17:54 back-monitor
drwxrwxrwx 2 root root   4096 Apr  4 17:54 back-meta
drwxrwxrwx 2 root root   4096 Apr  4 17:54 back-ldif
drwxrwxrwx 2 root root   4096 Apr  4 17:54 back-ldap
drwxrwxrwx 2 root root   4096 Apr  4 17:54 back-hdb
```

```
drwxrwxrwx 2 root root   4096 Apr  4 17:54 back-dnssrv
drwxrwxrwx 2 root root   4096 Apr  4 17:54 back-bdb
-rwxrwxrwx 1 2000 2000   2105 Apr  5 13:41 slapd.conf
[root@localhost slapd]#
```