Implementación de un Sistema de Directorios LDAP para la Universidad de Concepción

Salvador Ramírez Flandes

28 de marzo de 2001

Índice General

Índice de Tablas

Índice de Figuras

Resumen

Internet ya no es sólo la red académica y de investigación que fue en sus comienzos. Actualmente muchas operaciones cotidianas (por ejemplo comerciales) son llevadas a cabo a través de esta gran red, evitando problemas tan comunes como por ejemplo las largas filas de muchos servicios. Sin embargo, en las operaciones a través de la red es más difícil estar seguro que quien está detrás de alguna operación es realmente quien clama ser.

El presente proyecto tiene como objetivo el estudio e implementación de una plataforma software que permita la integración de herramientas informáticas actuales para solucionar el problema anterior, más otros problemas presentes en los servicios Internet que ofrece nuestra Universidad. Dentro de estas herramientas se encuentran la criptografía de clave pública, necesaria para llevar a cabo operaciones seguras a través de redes inseguras. Entre los problemas presentes en los servicios de red de nuestra Universidad, se encuentra la redundancia en el almacenamiento de los datos de los usuarios para diferentes servicios, como son los servicios Internet y el servicio de reserva y uso de los computadores personales en los laboratorios de computación de alumnos.

Específicamente el proyecto presenta un nuevo modelo de cómo almacenar y publicar la información que los usuarios necesitan para el uso de los servicios de red que ofrece nuestra Universidad. Este nuevo modelo se basa en el uso de los *Directorios LDAP*. Las principales ventajas de este nuevo modelo son la unificación de la base de datos de información de usuarios, que actualmente se encuentra distribuída (y muchas veces repetida) en muchas bases de datos propias de cada servicio; y la posibilidad de publicar, en estos directorios, información que permita autentificar a cada usuario y así permitir llevar a cabo operaciones seguras a través de la red de nuestra Universidad. Estas operaciones seguras incluyen, por ejemplo, la inscripción de asignaturas por parte

de los alumnos, consultas y pagos de deudas y la modificación de la carga académica de docentes, entre otras.

Una condición importante impuesta al proyecto, es el uso de herramientas software abierto y de uso gratuito (al menos en organismos educacionales) para llevar a cabo los objetivos. Actualmente este tipo de software tiene mucho apoyo y desarrollo a lo largo del mundo. Sin embargo la experiencia, en este proyecto, con estas herramientas tuvo ciertos problemas, aunque no mayores como para impedir el cumplimiento de las metas propuestas.

Por problemas de tiempo y disponibilidad de los recursos para la implementación, ésta no ha podido llevarse a cabo aún en el ambiente final, sino en un ambiente de menor escala que ha servido de prueba del sistema de directorios y su conectividad con las otras herramientas relacionadas, como son los programas de servicios de Internet. En este ambiente, la integración de LDAP con los demás servicios resultó exitosa, por lo que se espera igual resultado en la instalación final de uso para la Universidad.

Capítulo 1

Introducción

Actualmente Internet es uno de los medios de comunicación más masivos del planeta. La cantidad de usuarios en línea en Internet actualmente se estima en 407 millones[1]. Esto en gran parte es debido a que Internet ofrece variadas posibilidades de comunicación entre usuarios. Estas posibilidades se distribuyen entre los distintos servicios que Internet permite. Entre estos servicios, los de más uso son el correo electrónico y el servicio de páginas Web, WWW. Este último servicio, el WWW, almacena tanta información alrededor del mundo que ningún computador actual sería capaz de indexar toda esa información.

Aparte de estos dos servicios, en Internet existen otros tales como el sistema de transferencia de archivos *ftp*, el sistema de noticias *news* y el sistema de conversación *irc*. Muchos de estos servicios son incluídos en *WWW*, pero típicamente este último se relaciona con el servicio de transferencia de páginas Web, *http*, más que con los otros servicios mencionados.

Dada la cantidad de información a la que se puede acceder a través de estos servicios y la rapidez a la cual este acceso puede ser posible, estos servicios son una necesidad para cualquier organización que desee mantenerse vigente hoy en día.

En nuestra Universidad, actualmente, tanto alumnos como funcionarios tienen derecho a una cuenta para acceder a los servicios Internet mencionados anteriormente. Hace algún tiempo atrás esto no era así y sólo algunos alumnos poseían una cuenta

[1] Según información de *Nua Internet Surveys* (http://www.nua.ie/surveys).

(en los casos en que un académico justificara su uso) y pocos funcionarios (los más relacionados con el tema informático) hacían uso de ellas. El año 1997 se inició un proyecto en la Dirección de Tecnologías de Información (DTI)[2], destinado a hacer posible la creación automática de cuentas para acceder a los servicios Internet que ofrece nuestra Universidad. La explosión de creación de cuentas superó las espectativas, sobretodo por parte de los alumnos. Actualmente muchos alumnos y funcionarios de nuestra Universidad tienen una cuenta y pueden acceder a los servicios Internet de la Universidad. Sin embargo, actualmente existen otros problemas, tal como se verá en este mismo capítulo, en la sección de motivación. También se verán los objetivos del presente proyecto, que apuntan a solucionar estos problemas.

En el capítulo 2 (Antecedentes Generales) se presenta una breve descripción de algunos conceptos usados más adelante. Se describen primero los principales servicios Internet; las cuentas de usuario en sistemas Unix, que permiten el uso de los servicios Internet por parte de los usuarios; la seguridad en general para los servicios Internet mencionados y el funcionamiento general de los sistemas criptográficos de uso actual; y por último una descripción general de herramientas de uso común en WWW, como son el HTML, los programas CGI y lenguajes para la construcción de páginas Web dinámicas, tales como PHP.

En el capítulo 3 (Directorios y LDAP) se explica el concepto de *directorio* desde sus orígenes, en el estándar OSI X.500, hasta la última versión (versión 3) del estándar LDAP. Se explican también las ventajas de mantener las cuentas de usuario en directorios LDAP, así como los software actuales relacionados con este estándar.

En el capítulo 4 (Servicios DTI-UDEC) se describen los servicios Internet ofrecidos por la DTI de nuestra Universidad, además del actual sistema automático de creación de cuentas de usuario.

Por último, en el capítulo 5 (Proyecto de Directorios LDAP en DTI-UDEC), se describe el proyecto de instalación de directorios LDAP en los computadores centrales de la DTI. Se describe además el nuevo modelo propuesto en el proyecto para las

[2]En aquel tiempo ésta se llamaba División de Planificación e Informática (DPI).

cuentas de usuario y el nuevo sistema automático de creación de cuentas LDAP para este nuevo modelo de servicios.

1.1 Motivación

Actualmente muchos alumnos y funcionarios de nuestra Universidad tienen acceso a Internet y a los servicios ofrecidos en esta misma organización a través de la DTI. Para acceder a estos servicios muchos funcionarios usan computadores que ellos tienen asignados en sus respectivas oficinas, mientras que la mayoría de los alumnos, como no poseen un computador personal dentro de la Universidad, deben compartirse el uso de los computadores de los laboratorios[3].

El problema del uso de los computadores de un laboratorio es que muchos servicios Internet requieren de cierta configuración y que por el hecho de que los computadores son compartidos no es practicable el dejar grabados allí los datos de la configuración. Un *directorio*[4] es capaz de mantener toda esta información en forma centralizada y además poder ser accesible desde muchas aplicaciones que actualmente soportan el protocolo LDAP para el acceso a directorios.

Otro problema que motivó la implementación de directorios es que el almacenamiento de la información de autentificación de los usuarios, para muchos servicios, está distribuída (y replicada en varios casos) en muchos servidores. Esto provoca problemas al usuario al tener la necesidad de recordar contraseñas para cada servicio. A su vez es un problema para la administración de los servicios el necesitar administrar información en muchos lugares. Mantener los datos requeridos por los diferentes servicios en una base de datos centralizada solucionaría el problema. Esta base de datos puede ser implementada en un directorio LDAP. De esta manera, es posible además, especificar diferentes tipos de autentificación de usuarios para diferentes aplicaciones y no lo que es común, que las aplicaciones definan sus usuarios de acuerdo a los usuarios del sistema

[3] Ver sección 4.2 en la página 41 (Servicio Laboratorios de Computación de la DTI).

[4] Concepto que se revisará en detalle en el capítulo 3 (Directorios y LDAP). Para una definición rápida veáse el Glosario de Términos.

sobre el cual está ejecutándose la aplicación.

Además de los problemas descritos anteriormente existe aún una necesidad, en nuestra Universidad, de una plataforma para soportar la comunicación encriptada usando criptografía de clave pública[5]. Actualmente existe un servidor de certificados digitales en nuestra Universidad pero es necesario asignar a cada usuario un certificado y publicarlo en un servidor LDAP para que estos puedan ser accedidos desde cualquier aplicación que los necesite. Para enviar correos electrónicos firmados digitalmente, Netscape Messenger necesita acceder a un servidor LDAP para obtener el certificado digital del destinario de un determinado mensaje. Un correo electrónico firmado digitalmente tiene la característica de que se puede tener gran seguridad de que el emisor efectivamente envió tal mensaje y no ha sido falsificado ni su contenido ni su emisor. Esto es una necesidad en ambientes donde se pretenda reemplazar la mensajería tradicional por mensajería electrónica.

La identificación inequívoca de los usuarios es también necesaria para poder llevar a cabo operaciones a través de la red. Actualmente, en nuestra Universidad se están desarrollando (y otras se empezarán a desarrollar pronto) muchas aplicaciones a través de la red. Entre estas aplicaciones se encuentran la inscripción de asignaturas, las consultas y pagos de deudas o crédito universitario y la modificación de la carga académica de docentes. A través de criptografía de clave pública (con certificados publicados en directorios LDAP) es posible autentificar a los usuarios en el momento en que éstos solicitan llevar a cabo una determinada operación. Evitando así que impostores puedan realizar operaciones que no les está permitido realizar.

Por último, el uso de directorios LDAP a lo largo del mundo es una tendencia dentro de las tecnologías de información. Las organizaciones están invirtiendo en mantener su información en directorios pues es cómodo su acceso desde una gran variedad de aplicaciones en diversas plataformas. El cambio temprano a esta tendencia es una ventaja signficativa en el uso de las tecnologías actuales de información.

[5] Ver sección 2.4 en la página 16 (Criptografía).

1.2 Definición del Proyecto de Directorios LDAP en DTI-UDEC

El objetivo general del proyecto es estudiar e implementar un sistema de directorios en nuestra Universidad, con el fin de que cada usuario (funcionarios y alumnos de la Universidad) tenga un directorio asociado con información que le permita hacer uso de todos los actuales servicios Internet más el uso de certificados digitales en operaciones seguras.

Los objetivos específicos del proyecto se listan a continuación:

1. Instalar, configurar y poner en marcha un servidor de directorios LDAP.
2. Poblar los directorios de tal servidor con la información de cada usuario de la Universidad. Esto incluye tanto a funcionarios como a alumnos. Esto es equivalente a decir el traspaso de las cuentas Unix (en NIS) al sistema de directorios.
3. Integrar el Correo Electrónico, FTP y WWW de la Universidad (dominio udec.cl) con la información de usuarios del servidor de directorios.
4. Desarrollar herramientas para la creación de directorios de usuario via Web. Estas herramientas serán desarrolladas usando PHP.
5. Integrar los directorios de usuarios con el servidor de certificados digitales para que en la creación de cada directorio de usuario se asigne un certificado digital automáticamente y luego éste pueda ser usado para llevar a cabo operaciones seguras a través de la red de nuestra Universidad.

Se divide el proyecto en dos grandes etapas, la etapa de implementación y pruebas, y la etapa de implementación final.

En la primera etapa se trabajará en un ambiente que incluya todo lo que implica la implementación final pero que no afecte los servicios que diariamente usan tanto alumnos como funcionarios de nuestra Universidad. Este ambiente incluye un servidor LDAP, un servidor de correo electrónico, un servidor Web y un servidor FTP. Esta

etapa incluye también el aprendizaje y prueba de las distintas herramientas que se utilizarán para llevar a cabo los objetivos del proyecto, además de la construcción de las herramientas para la creación de directorios de usuario vía Web.

Una vez que todo este ambiente de pruebas funcione correctamente, se procederá a la implementación que finalmente será usada en la Universidad.

Capítulo 2

Antecedentes Generales

En este capítulo se introducen conceptos usados a lo largo de todo el documento. Estos conceptos involucran los servicios tradicionales de Internet (como son el correo electrónico, el FTP y el WWW), el cómo estos servicios se implementan para cada usuario en un sistema Unix (cuentas de usuario) y la seguridad en general, de estos servicios.

2.1 Servicios Internet Tradicionales

A pesar que existen muchos servicios Internet, los dos más conocidos por todos son el WWW y el correo electrónico. Gran parte de los otros servicios Internet han sido absorbidos en el concepto que involucra el WWW. En este sistema es posible acceder a múltiples servicios a través un mismo programa. Algunos de los servicios a los cuales se puede acceder a través de WWW son: HTTP (HyperText Transfer Protocol), FTP (File Transfer Protocol) y NNTP (Network News Transfer Protocol). A continuación se ofrece una breve descripción de los servicios de correo electrónico, HTTP y FTP.

2.1.1 Correo Electrónico

El correo electrónico es un sistema en el cual, normalmente, se encuentran involucrados más de un servicio Internet. Uno de estos servicios es usado para el envío y recepción de los correos (SMTP) y otros servicios para la lectura de éstos desde los computadores

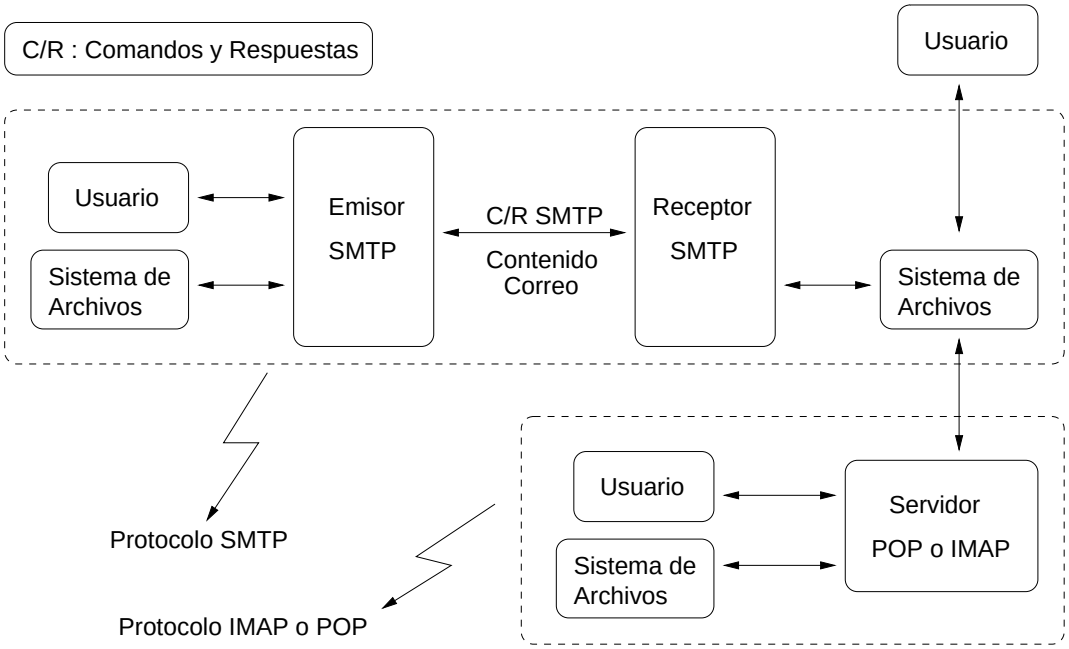

Figura 2.1: Esquema de un sistema de correo electrónico

de escritorio de los usuarios finales (POP e IMAP).

El protocolo Simple Mail Transfer Protocol (SMTP) es usado para la transferencia de correos electrónicos entre dos servidores SMTP, uno emisor y otro receptor de un correo, en un momento dado. El proceso de envío de un correo electrónico se compone, comúnmente, de los siguientes tres pasos:

1. Un usuario, a través de un programa de correo electrónico[1], se conecta a un servidor SMTP (emisor SMTP) para enviar un correo a una dirección electrónica dada. En este proceso se transmite el contenido del correo (texto escrito y/o un archivo del sistema), la dirección destino y otros posibles parámetros al servidor SMTP.

2. El servidor SMTP procesa el requerimiento buscando información acerca de la dirección destino, a la cual el correo electrónico va dirigido. El dato más importante a conseguir es la dirección IP del servidor SMTP (receptor SMTP) encargado de recibir el correo electrónico para la dirección dada. Este dato es

[1]Ejemplo de programas de este tipo son: pine, elm, Netscape Mail y Outlook Express.

conseguido a través de una consulta al sistema DNS (Domain Name System). Una vez conseguida la dirección IP del receptor SMTP, se inicia una conexión a este servidor, en donde finalmente (si es aceptado el correo hacia el destinatario) se termina por transferir el contenido del correo electrónico para éste. Finalizado este proceso la casilla electrónica del usuario (un archivo o directorio del sistema de archivos del servidor) ya posee un nuevo correo.

3. Una vez que el correo electrónico se encuentra en la casilla del usuario en el servidor, existen al menos tres formas a través de las cuales el usuario final puede leer su correo. La primera es que el usuario conectado al sistema a través de un login, pueda leer su casilla directamente (en muchos sistemas Unix ésta se encuentra en /var/mail o /var/spool/mail) con un programa tal como pine, elm o netscape (con la opción movemail)[2]. Las otras dos opciones son el uso de un servidor ya sea de POP (Post Office Protocol) o IMAP (Internet Message Access Protocol). Estos servicios Internet permiten a un cliente acceder a las casillas de correos electrónico en forma remota y revisar cada cierto tiempo si ha llegado un nuevo correo a la casilla. Ejemplos de programas de correo que soportan POP son Netscape Mail y Outlook Express, y un ejemplo de programa que soporte IMAP es Netscape Mail.

Notar que sólo los dos primeros puntos anteriores corresponden al protocolo SMTP, mientras que el último punto tiene que ver con la lectura de los correos, lo que no necesariamente necesita de un protocolo, dado que éstos pueden ser leídos directamente en el servidor, tal como se mencionó anteriormente. Es decir, un servidor POP o IMAP es opcional (aunque cómodo) en muchos casos.

La diferencia fundamental entre POP e IMAP se relaciona con la habilidad de éste último para manipular carpetas de correos en el servidor. Es decir, IMAP no sólo es capaz de transferir los correos de la casilla electrónica del usuario, sino que además

[2]También es posible que el servidor comparta el sistema de archivos donde se encuentran las casilla a otros computadores (típicamente estaciones de trabajo) a través de NFS por ejemplo. En este caso los usuarios también podrían leer sus correos directamente conectándose a estos otros sistemas.

puede mantener otras carpetas en el servidor, a diferencia de POP que sólo mantiene las carpetas en el computador cliente. Esto es útil cuando el usuario no tiene un computador de uso fijo.

2.1.2 Servicio de Páginas Web

El servicio de páginas Web consiste de un servidor que alberga páginas Web, comúnmente en formato HTML (HyperText Markup Language), y que son accedidas desde clientes llamados browsers o navegadores. El lenguaje de comunicación entre clientes y servidores en este caso es HyperText Tranfer Protocol (HTTP) que es un protocolo de nivel de aplicación para la transferencia de datos multimediales. Se habla de datos multimediales pues las páginas Web pueden incluir referencias a archivos de imágenes, audio o video, presentes en el servidor Web.

La forma en que se especifican los tipos de archivos (con datos multimediales) en HTTP es a través del Multipurpose Internet Mail Extensions (MIME). Cada vez que el cliente o servidor tiene que enviar datos al otro, se envía primero un encabezado con información, en formato MIME, del tipo de archivo a enviar[3]. De esta manera, tanto el cliente como el servidor saben como manejar los datos a recibir.

Por último cabe mencionar que al conjunto de servidores Web, compartiendo sus páginas Web y datos en general, a lo largo del mundo se le llama el sistema Worl-Wide Web o simplemente WWW.

2.1.3 FTP

FTP es la abreviatura de File Transfer Protocol, y es un protocolo para transferir[4] archivos entre computadores conectados a una red TCP/IP, como lo es Internet.

Este servicio de transferencia de archivos es uno de los servicios clásico de Internet y como todos estos, se implementa a través de un protocolo cliente/servidor. Los servidores de FTP siempre están esperando por conexiones desde clientes para solicitar

[3]Este encabezado normalmente posee otros datos también, más no importantes de mencionar aquí.
[4]Para ser exactos, los bytes de un archivo no son transferidos sino copiados.

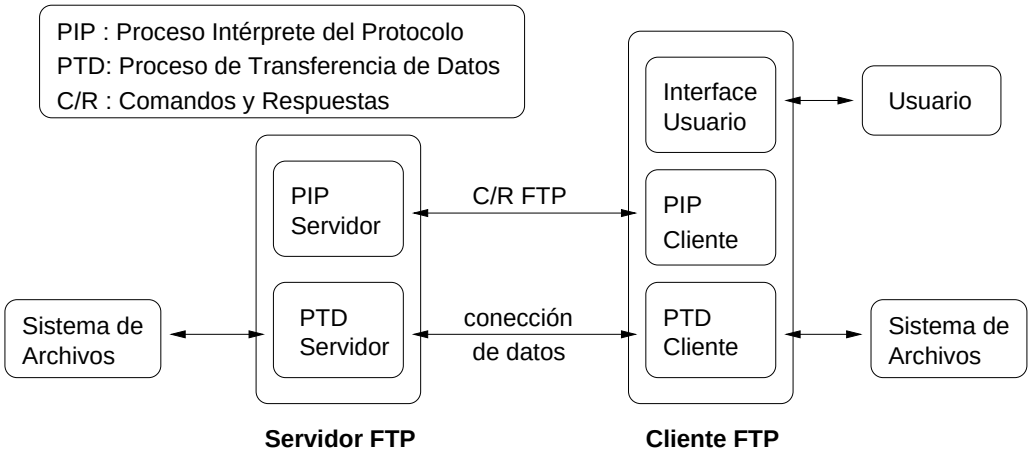

Figura 2.2: Funcionamiento del Protocolo de Transferencia de Archivos (FTP)

la transferencia de archivos ya sea desde o hacia el servidor. Una vez que cliente y servidor se han puesto de acuerdo en el archivo a tranferir, se abre una nueva conexión para la transferencia de este archivo, quedando la conexión original para el envío de comandos entre cliente y servidor (figura 2.2).

Un cliente FTP, para conectarse a un servidor FTP, necesita de una cuenta en este último sistema, para poder acceder a los archivos. Esta cuenta es una identificación de un usuario, con una determinada contraseña o clave y un espacio en el sistema de archivos, en donde se encuentran los archivos, a los cuales el cliente accede.

El uso más frecuente de FTP se conoce como FTP anónimo, esto es, que la cuenta a la cual se accede en el servidor es de uso público y por tanto no necesita de una contraseña. De esta manera cualquier persona a través de un cliente puede acceder. Muchas utilidades se han dado a este servicio entre los cuales están la distribución de software o bases de datos de interés científico.

2.2 Cuentas de Usuario

Dada la estrecha relación entre el desarrollo del sistema operativo Unix e Internet en sus comienzos, muchos de los servicios Internet son nativos de este sistema operativo. Toda distribución de Unix consta, por ejemplo, con un servidor de correo electrónico y

FTP. El servidor HTTP (para servir páginas Web en WWW) de mayor uso actualmente (Apache Web Server) se desarrolla en Unix así como muchos otros programas de servicios en Internet. De esta manera, es común el uso del término cuenta de usuario como un requisito para obtener los servicios Internet ya mencionados.

Una cuenta es un acceso a un sistema Unix, que se compone por un nombre (username), una clave (password) y un espacio en disco (home directory). El username y password constituyen los datos para la autentificación del usuario en el sistema. Es decir, si un usuario ingresa un username y password válido entonces el sistema lo reconoce como el usuario dueño de la cuenta respectiva. De esta manera, para poder ingresar a un servicio FTP remoto es necesario poseer una cuenta en tal sistema. Lo mismo sucede para otros servicios Internet tales como POP e IMAP para leer el correo electrónico.

En muchos casos es útil que una serie de computadores posean las mismas cuentas. Se puede pensar en un conjunto de computadores, cada uno asignado a un servicio Internet distinto, donde se desee que todos los usuarios posean todos los servicios. En este caso habría que mantener las mismas cuentas en distintos computadores, lo que provocaría dificultades en la administración de cuentas. Crear una cuenta significaría crearla en todas las máquinas, así también el borrado de estas sería un trabajo engorroso.

La empresa Sun Microsystems, creó un sistema para solucionar este problema. El sistema fue llamado Yellow Pages y luego pasó a llamarse NIS (Network Information Service) por un problema de patente de nombre. NIS es un sistema en donde un conjunto de computadores (con sistema operativo Unix) comparten las mismas cuentas, presentes en un sólo servidor NIS, donde se efectúa la administración centralizada de las cuentas. Cuando se desea que un computador posea las mismas cuentas que el servidor NIS entonces se hace este computador cliente NIS del servidor.

Este sistema NIS se complementa muy bien con otro sistema creado también por Sun Microsystem, llamado NFS (Network File System). Con NFS es posible compartir sistemas de archivos entre distintos computadores. De esta manera, los usuarios con las mismas cuentas en muchos computadores, con este sistema, ahora también

disponen del mismo sistema de archivos en todos esos computadores, logrando cierta transparencia, cuando los sistemas operativos de los computadores son los mismos. Esta transparencia se traduce en que para el usuario simpre será lo mismo conectarse a cualquier computador de la red, pues en todos ellos están sus mismos archivos y además sus mismo datos de autentificación.

En la mayoría de estos servicios distribuídos, cuando un usuario se conecta a un servicio remoto, los datos de la autentificación (username y password), se transmiten en *texto plano* por la red, sin ningún tipo de encriptación. Esto quiere decir que si un tercero pudiera leer lo que se transmite por la red en el momento de la autentificación, entonces podría tener acceso a los datos de la autentificación de otro usuario y así poder suplantarlo en el futuro. Esto constituye un problema de seguridad que será abordado en mayor detalle en la siguiente sección.

2.3 Seguridad

La seguridad es un tema importante para las aplicaciones de red que proveen los servicios mencionados y en general todos los servicios de Internet. Esto se debe al hecho de que la mayoría de las redes en Internet y los protocolos que gobiernan su comunicación, son inseguros. Un mensaje enviado desde un emisor a un destinatario lejano es típicamente dividido en muchos paquetes (que juntos componen el mensaje original) y cada uno de éstos son enviados a través de rutas muchas veces distintas, pasando por un conjunto de computadores que van dirigiendo estos paquetes. Dada la cantidad de redes involucradas en el viaje de un paquete, es imposible asegurar que éste no haya sido leído por terceros en algún lugar.

Por otro lado, la mayoría de las redes de area local (LAN) son redes de difusión, en donde un mensaje dirigido a un destino particular puede ser interceptado (dependiendo de la topología de la red[5]) por cualquier otro computador. La configuración normal de

[5]Si la topología de la red es de bus o estrella a un hub o repetidor, entonces los mensajes pueden ser interceptados por otros. Si la topología es estrella con todos conectados a un switch entonces no sucede esto pues el switch es un dispositivo que no repite los paquetes a todas las demás puertas.

los dispositivos de conexión a estas redes, en los computadores conectados a la red, solo leen de la red los paquetes dirigidos al computador local. Sin embargo, con los privilegios apropiados, un programa puede instruir al dispositivo de red para leer todos los paquetes y así leer todos los mensajes pasando por la red, aún cuando éstos no hayan sido dirigidos a ese computador. A este tipo de redes, en el contexto de seguridad, se les llama redes inseguras.

Tal como se mencionó en la sección anterior, la mayoría de los servicios Internet, por sí solos son inseguros pues no incluyen una encriptación de datos sensibles ni el momento de la autentificación ni en la transferencia misma de la información que se transmite a través de tal servicio. El uso de estos servicios entonces, en redes inseguras, es un peligro, pues otras personas podrían capturar los datos de autentificación y así falsear su identidad.

En general, el concepto de seguridad en Internet involucra cuatro aspectos:

1. **Autentificación**. El hecho de probar que una persona o programa es quien realmente clama ser.

2. **Integridad**. Seguridad de que la información recibida es idéntica a la información enviada[6].

3. **Confidencialidad**. Evitar que los datos sean leídos por receptores no autorizados para ello.

4. **Autorización**. Seguridad de que una persona o programa está autorizado para llevar a cabo la acción que solicita[7].

Se han desarrollado tecnologías para abordar el problema de los tres primeros aspectos mencionados. El tema de la autorización lo maneja, por lo general, la aplicación.

Entre las tecnologías más usadas actualmente para la seguridad de las transacciones en Internet, están SSL y TLS. Estos protocolos de seguridad encriptan todos los

[6] Y por tanto no fue alterada durante la transmisión.
[7] Esto normalmente es revisado después del proceso de autentificación.

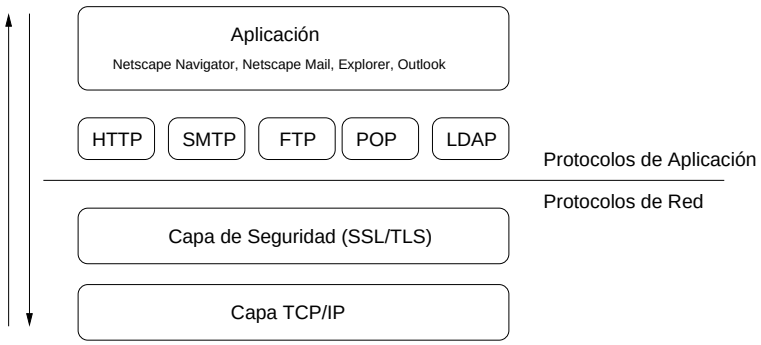

Figura 2.3: Relación de SSL/TLS con los otros protocolos.

datos que fluyen entre un cliente y un servidor a través de TCP/IP. SSL, que es la tecnología con mas tiempo de uso de estas dos, ha sido ampliamente usada en el WWW, proveyendo una plataforma segura para operaciones tales como el comercio electrónico, donde es necesario asegurar cierta información del usuario, como son, por ejemplo, los números de tarjetas de crédito.

SSL fue desarrollado por la empresa Netscape Communications, mientras que TLS es un protocolo que está siendo desarrollado en la IETF[8] como un estándar para Internet, basado en SSL. Se supone que con el tiempo TLS reemplazará a SSL, aunque actualmente estos protocolos ofrecen los mismos servicios.

Aparte de proveer la encriptación de una sesión de transmisión de datos mediante TCP/IP, SSL/TLS permite la autentificación mutua, entre clientes y servidores, a través de certificados criptográficos basados en el estándar X.509. Esta autentificación es necesaria para que tanto clientes como servidores, estén seguros de la identidad del cliente o servidor con el cual se están comunicando. Más detalles de esta autentificación se dan en la siguiente sección.

SSL/TLS es uno de los mecanismos usados para llevar a cabo la autentificación entre clientes y servidores en Internet. Aparte de este, existen muchos otros, tales como Kerberos y S/Key.

Existe un protocolo llamado SASL (Simple Authentication and Security Layer), que sirve para que un cliente y un servidor puedan negociar el mecanismo de autentificación

[8]The Internet Engineering Task Force. http://www.ietf.org

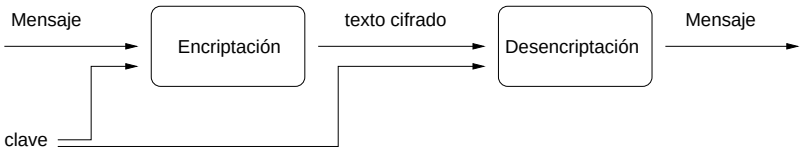

Figura 2.4: Criptografía de clave secreta. La misma clave se usa tanto para encriptación como para desencriptación.

a usar, en el caso de que exista más de un mecanismo disponible. Este protocolo es importante pues libera a la aplicación de la implementación de un mecanismo de autentificación en particular y se deja esta labor a la capa SASL. En caso que un nuevo mecanismo de autentificación se desarrollara, la capa SASL lo implementa, sin necesidad de modificar las aplicaciones que harán uso de este nuevo mecanismo.

2.4 Criptografía: Certificados y Firmas digitales

La Criptografía es el arte de mantener mensajes en secreto, protegiéndolos contra la lectura de terceros no autorizados. Para llevar a cabo esto es necesario *transformar* el mensaje original en un mensaje que no pueda ser leído por otros sino sólo por quienes participan de la comunicación. En la jerga, al mensaje original se llama *texto plano*, al mensaje transformado se llama *texto cifrado*, a la acción de transformar el texto plano en texto cifrado se llama *encriptación*, mientras que la acción de transformar el texto cifrado en texto plano (para su lectura) se llama *desencriptación*.

Actualmente los algoritmos usados para la encriptación y desencriptación usan *claves*. A estos algoritmos se les llama (con el nombre poco sorprendente de) *algoritmos criptográficos basados en clave.* Existen dos tipos de algoritmos criptográficos basados en claves: **simétricos** (o de clave secreta, secret-key) y **asimétricos** (o de clave pública, public-key). Habitualmente se llama criptografía de clave secreta (secret-key cryptography) al conjunto de sistemas de encriptación que utilizan algoritmos simétricos y criptografía de clave pública (public-key cryptography) al conjunto de sistemas de encriptación que utilizan algoritmos asimétricos.

En la criptografía de clave secreta una clave es compartida por dos o más partícipes

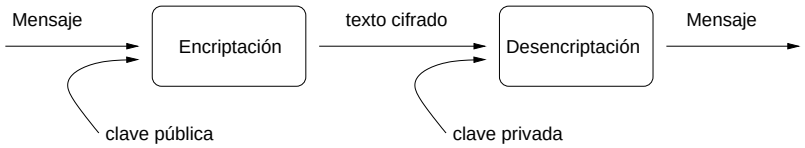

Figura 2.5: Criptografía de clave pública. Cada usuario posee dos claves y un mensaje encriptado con una de ellas sólo puede ser desencriptado con el par correspondiente.

de alguna comunicación. Para encriptar un mensaje este tipo de algoritmos toma la clave y el mensaje original como entradas para una función matemática que entregará el mensaje original encriptado (texto cifrado). Para la desencriptación del texto cifrado es necesario usar la misma clave.

En la criptografía de clave pública en cambio, cada usuario posee una clave privada y una clave pública. Un mensaje que es encriptado con la clave privada sólo puede ser desencriptado con la clave pública correspondiente y viceversa. El usuario debe mantener en secreto su clave privada, mientras que la clave pública debe dejarla a libre disposición para que otros usuarios, que deseen comunicarse con él, puedan encriptar sus mensajes con esta clave pública. De esta manera sólo el usuario poseedor de la clave privada podrá desencriptar el texto cifrado.

Tanto la criptografía de clave secreta como la de clave pública tienen sus ventajas y desventajas. La principal ventaja de la criptografía de clave pública es que ofrece mayor seguridad, al evitar tener que transmitir las claves secretas cada vez que se necesite comunicar (en forma segura) con un nuevo usuario. En la criptografía de clave pública, un usuario que desee enviar un mensaje a otro sólo necesita tener acceso a su clave pública (típicamente publicada como certificado digital en directorios LDAP) y encriptar el mensaje usando aquella clave. Dado que el mensaje encriptado de esta manera sólo puede ser desencriptado usando la clave privada correspondiente, sólo el usuario destinatario será capaz de leer el mensaje.

Otra ventaja importante de la criptografía de clave pública es que permite autentificación y firmado digital. La autentificación es posible dado que un mensaje encriptado que pueda ser desencriptado con la clave pública exitosamente, necesariamente debió ser encriptado con la clave privada correspondiente, que sólo debe poseer el usuario. Una

firma digital actúa de una manera similar y a continuación se presenta el procedimiento que sigue un programa para enviar un mensaje encriptado y firmado digitalmente:

1. El emisor usa un algoritmo especial para generar un *message-digest* del mensaje. El message-digest es una representación del mensaje, que es más corta y por tanto más rápida de encriptar/desencriptar.
2. El emisor encripta el message-digest usando su clave privada.
3. El emisor genera una clave secreta randómica.
4. El emisor adjunta el message-digest al mensaje original y los encripta usando la clave secreta randómica.
5. El emisor encripta la clave randómica con la clave pública del receptor, así sólo el receptor podrá conocer la clave y luego desencriptar el mensaje.
6. El emisor envía el mensaje firmado encriptado (con la clave secreta) más la clave secreta encriptada (con la clave pública del receptor) al receptor.

Como se puede apreciar en el procedimiento anterior, en una misma aplicación se usa tanto criptografía de clave secreta como criptografía de clave pública para aprovechar las ventajas de cada una. La principal ventaja de la criptografía de clave secreta es que por lo general sus algoritmos son poco consumidores de recursos computacionales, así la encriptación y desencriptación de un texto se realiza muy rápidamente o al menos más rápido que como sucede con la criptografía de clave pública. Por esta razón, el mensaje mismo (que puede ser largo) se encripta usando criptografía de clave secreta.

Para tener la seguridad de que una clave pública realmente pertenece al usuario que la ha publicado, existen los certificados digitales. Un certificado es un documento digital que acredita que una determinada entidad (ya sea usuario u organización) posee la clave pública que en el mismo certificado se especifica. Los certificados son emitidos por entidades llamadas *certificate authorities* o CA y son las responsables de verificar la identidad de los usuarios u organizaciones según sea el caso.

El protocolo SSL/TLS, mencionado en la sección anterior, funciona básicamente igual que el procedimiento anterior para los mensajes firmados digitalmente. Un programa servidor envía a un programa cliente su certificado digital (con su clave pública) y puede o no requerir la misma acción por parte del cliente. En seguida el cliente genera una clave secreta randómica y la envía al servidor (encriptada con la clave pública del certificado) para luego encriptar toda la sesión con aquella clave.

2.5 HTML, CGI y PHP

Inicialmente el sistema WWW se componía, prácticamente, sólo de páginas HTML (HyperText Markup Language) planas, componiendo hipertexto con enlaces a imágenes, audio y video. HTML es un lenguaje que define la estructura de un documento en una página Web, y por tanto es un lenguaje que "entiende" el programa navegador. Con el tiempo se han dado muchas aplicaciones al WWW y ya las páginas HTML, por sí solas, no fueron suficientes. Aplicaciones para procesar formularios por la red, por ejemplo, requieren de la ejecución de un programa en el servidor para procesar la información del formulario. Esto dió lugar al sistema CGI (Common Gateway Interface), que es una aplicación para el intercambio de datos entre un servidor WWW y un programa ejecutable local[9]. A este programa local típicamente se le nombra programa CGI o CGI simplemente.

Hay que destacar que un CGI puede procesar datos desde una página Web y además escribir datos que el servidor finalmente dirigirá hacia el navegador. El lenguaje o formato en el cual deben ser escritos esos datos, entonces, debe ser necesariamente uno que entienda el navegador (lenguaje HTML por ejemplo).

Uno de los problemas de los programas CGI es que, dependiendo del lenguaje de programación a usar, son difíciles de crear y consumen muchos recursos dependiendo de la cantidad de clientes efectuando requerimientos. Por estas y otras razones existen sistemas de programación de *script* incluídos como módulos en el mismo

[9]Este programa puede ser tanto ejecutable binario como interpretado, es decir puede ser escrito en C, Fortran o Perl (entre otros muchos lenguages de programación).

programa servidor WWW y que interpreta los comandos de los programas escritos. La ventaja de estos sistemas de programación es que son especializados en tareas para el procesamiento de datos del Web, a diferencia de los lenguajes de programación de propósito general, en el cual son escritos los programas CGI. Ejemplos de sistemas de programación de *script* incluídos en servidores Web son el ASP (Active Server Pages) y PHP (PHP Hypertext Preprocessor). El sistema ASP es propietario de Microsoft y como tecnología cerrada es sólo incluído en el servidor WWW de Microsoft. PHP por su parte es un software abierto y funciona sobre muchas plataformas, aunque su plataforma nativa es Unix/Linux.

Actualmente, en muchos casos las páginas HTML ni siquiera existen en los servidores WWW, sino que se crean dinámicamente *on-the-fly*, escritas ya sea por un programa CGI, ASP o PHP.

Capítulo 3

Directorios y LDAP

3.1 Breve Historia

En la década de los 70 dos estándares de comunicación fueron desarrollados separadamente, OSI y TCP/IP. El modelo OSI estaba constituído por siete capas bien definidas y su desarrollo (que no implicaba la implementación de las aplicaciones) estaba en manos de un comité formal, a saber el CCITT (Consultative Commitee on International Telephony and Telegraphy) en conjunto con la ISO (International Standards Organization). TCP/IP en cambio se desarrolló de una manera mucho menos formal en donde cualquier experto proponía estándares a través de los llamados RFC (Request For Comments) y las implementaciones corrían por cuenta de cualquier persona que quisiera implementar algún RFC dado. Además, las siete estrictas capas definidas en el modelo OSI hacían de éste una implementación software más consumidora de recursos computacionales que TCP/IP, que no era tan rígido en esta definición de las capas. Este hecho llevó a que el modelo OSI no pudiese implementarse en algunos computadores (de escritorio o personales por ejemplo) de aquella época.

Así fue como OSI se quedó con sus siete capas definiendo protocolos y aplicaciones y que hoy día prácticamente sólo sirven como referencia dado que su desarrollo fue muy lento comparado con el de TCP/IP. Este último, finalmente, se convirtió en el estándar para las comunicaciones en Internet.

A pesar de esto, OSI definió algunos estándares de gran importancia y que sirvieron

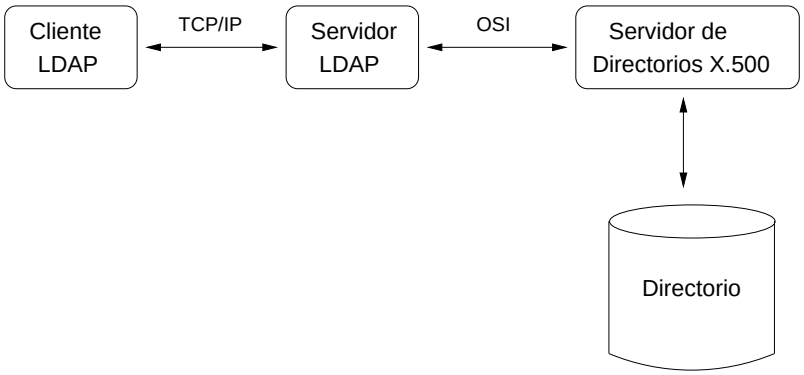

Figura 3.1: LDAP como intermediario o gateway TCP/IP para servidores X.500

para inspirar símiles en TCP/IP. Uno de estos protocolos fue el servicio de directorios X.500, desarrollado en 1988. X.500 propone una organización de las entradas en los directorios en un espacio de nombres ordenados jerárquicamente. X.500 también define capacidades de búsqueda poderosas para hacer más fácil y óptimo el rescate de la información desde los directorios. Para la interacción entre un cliente y el servidor de directorios, el estándar X.500 definió el Directory Access Protocol (DAP). Sin embargo para que DAP pudiese operar se requería de las capas OSI. Este hecho llevo al desarrollo de un protocolo de acceso a los directorios menos consumidor de recursos y que por tanto usara TCP/IP en lugar del modelo OSI.

Este nuevo protocolo fue inicialmente definido en el RFC 1487, “X.500 Lightweight Access Protocol”, el que luego fue reemplazado por el RFC 1777, “Lightweight Directory Access Protocol”, de aquí LDAP. La última versión de LDAP se propuso en el RFC 2251, “Lightweight Directory Access Protocol (v3)”[1].

Inicialmente se definió LDAP como un protocolo para acceder a un servidor de directorios X.500 a través de TCP/IP dado que los servidores X.500 soportan el protocolo DAP para acceder a sus directorios. Así el servidor LDAP servía como intermediario entre un cliente que no podía implementar DAP dado que no tenía OSI sino TCP/IP en su torre de protocolos, tal como se muestra en la figura 3.1.

Sin embargo, como ya se ha mencionado, TCP/IP creció y cada vez más clientes

[1]Las principales diferencias entre la versión 2 y 3 del protocolo LDAP son que este último es capaz de negociar el mecanismo de autentificación, a través de un protocolo llamado SASL, el que se describirá mas adelante; y además a LDAPv3 se le pueden extender sus operaciones.

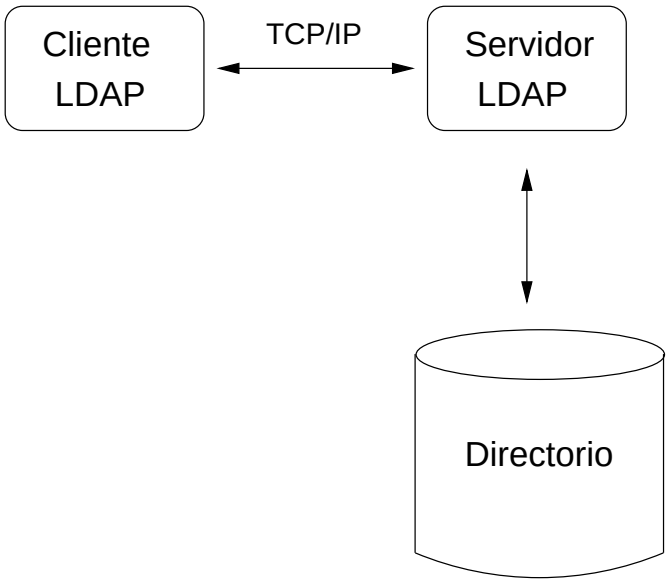

Figura 3.2: Servidor LDAP independiente o stand-alone.

LDAP (por ejemplo aplicaciones tales como servidores de correo electrónico que mantengan sus usuarios en directorios, servidores de FTP, navegadores, etc.) fueron surgiendo. Más aplicaciones clientes LDAP se fueron desarrollando que aplicaciones clientes para directorios X.500.

Dado este desarrollo, la necesidad del servidor de directorios X.500 fue desapareciendo, y se pensó en que mejor el servidor LDAP accesará los directorios directamente. El RFC 1777 propone el acceso directo a los directorios X.500.

A un servidor LDAP que pueda acceder directamente a los directorios se llama Servidor LDAP *stand-alone* o independiente (figura 3.2).

Servidores LDAP independientes son los servidores de uso común actualmente, por lo que de aquí en adelante cuando se refiera a un servidor LDAP se estará refieriendo a un servidor LDAP independiente. Finalmente cabe desctacar que todo el desarrollo inicial de LDAP fue realizado en la Universidad de Michigan, USA.

3.2 Estructura y Organización de los Directorios en LDAP

El estándar LDAP, definido en los RFC mencionados en la sección anterior, define

básicamente dos cosas:

1. Cuatro modelos que definen la estructura y organización de los directorios
2. Un protocolo de red, del cual se hablará en la siguiente sección

Los modelos que describen los directorios LDAP toman muchos conceptos de X.500. Los cuatro modelos LDAP son los siguientes:

- El modelo de información de LDAP, que define los datos que se pueden almacenar en el directorio
- El modelo de nombres de LDAP, que define como organizar y como referirse a los datos en el directorio
- El modelo funcional de LDAP, que define como acceder y actualizar la información del directorio
- El modelo de seguridad de LDAP, que define como proteger la información del directorio contra accesos no autorizados

A continuación se describirán estos modelos en mayor detalle.

3.2.1 El modelo de información de LDAP

El modelo de información de LDAP define los tipos de datos y unidades básicas de información que se pueden almacenar en un directorio.

La unidad básica de información en un directorio es la *entrada*, que es un conjunto de datos acerca de un objeto, como por ejemplo una persona o una organización. Las entradas en un directorio se distribuyen en forma jerárquica, tal como se muestra en la figura 3.3 en la página siguiente.

Una entrada se compone de un conjunto de atributos, cada uno de los cuales describe una característica particular del objeto. Cada uno de estos atributos tiene un tipo de dato y uno o más valores. Por ejemplo el objeto persona puede tener el atributo apellido,

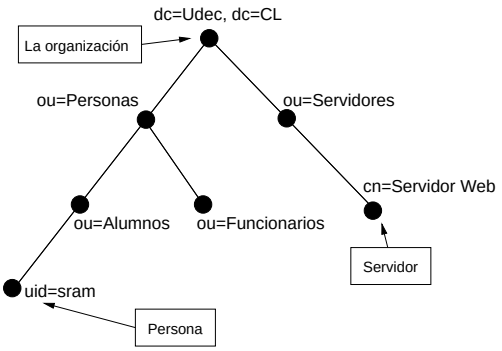

Figura 3.3: Ejemplo de organización de un directorio. Cada nodo corresponde a una entrada en el directorio.

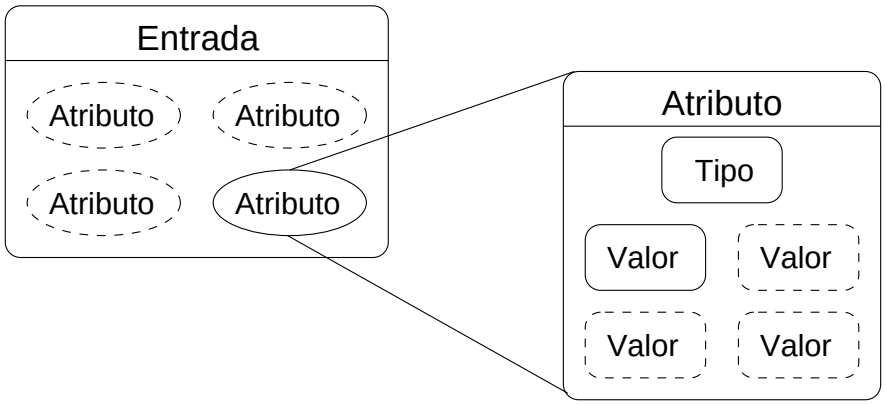

Figura 3.4: Entradas, Atributos y Valores

cuyo tipo de dato es *string* y su valor *Ramírez*. Un ejemplo de un atributo con más de un valor puede ser el atributo email de una persona, que por supuesto puede tener más de un valor. Esta característica de los atributos, de ser multivaluados, es configurable en el directorio para cada atributo, así una entrada puede tener atributos que si aceptan múltiples valores mientras que otros no (ver figura 3.4).

Una entrada en un directorio posee un conjunto de atributos que son requeridos mientras que otros son sólo permitidos u opcionales. Sin embargo cualquier otro atributo que no sea ni requerido ni opcional no es permitido. Al conjunto de atributos requeridos y permitidos se llama *esquema*. Un esquema típicamente define una *clases de objeto* donde se define los tipos de datos que los atributos de las entradas pueden mantener. Una entrada puede tener asociada una o más de una clase de objeto o

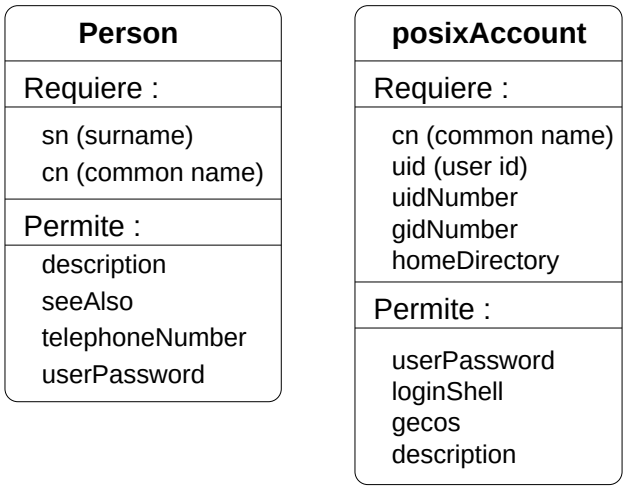

Figura 3.5: Ejemplos de esquemas definiendo las clases de objecto *Person* y *posixAccount*

esquema[2]. Ejemplos de esquemas se pueden ver en la figura 3.5.

3.2.2 El modelo de nombres de LDAP

El modelo de nombres de LDAP define como se organizan y referencian las entradas en el directorio. Este modelo de nombres especifica que las entradas en un directorio son organizadas como un árbol invertido, en una estructura típica de organización de directorios en un sistema de archivos.

La manera en que se referencian las entradas en un directorio hace explícita la estructura jerárquica del directorio. Por ejemplo, en la figura 3.6 en la página siguiente, la raíz es “dc=Udec, dc=CL”, sobre la cual existe la clase de objetos, ou=Personas. Sobre ou=Personas existen dos clases de objetos llamados ou=Funcionarios y ou=Alumnos . Sobre este último se encuentra el objeto alumno con uid=sram. La referencia completa de este último objeto en el directorio es: “uid=sram, ou=Alumnos, ou=Personas, dc=Udec, dc=CL”.

El modelo de nombres de LDAP es importante pues permite dar un nombre único a cada entrada de un directorio. Este nombre único en LDAP se llama *distinguished name* o simplemente DN, que se construye a partir de una serie de *relative distinguished*

[2]Toda entrada en un directorio debe tener al menos un esquema asociado. Una entrada define un esquema a implementar mediante el atributo *objectClass*.

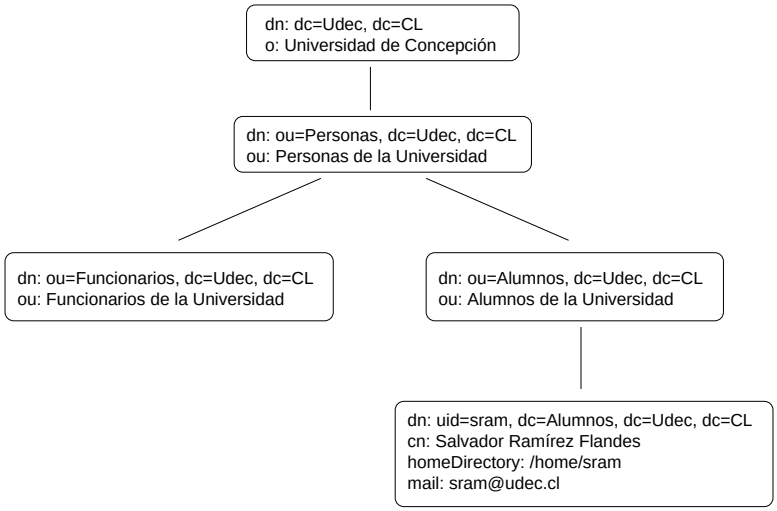

Figura 3.6: Estructuras y DNs de los objetos de un directorio

names (RDN) del objeto particular, hacia arriba del árbol. En el ejemplo anterior uid=sram es el RDN del objeto particular a referirse, y su DN es : "uid=sram, ou=Alumnos, ou=Personas, dc=Udec, dc=CL".

3.2.3 El modelo funcional de LDAP

El modelo funcional de LDAP describe como acceder a los datos en el directorio a través de operaciones que se pueden llevar a cabo usando el protocolo LDAP.

Las operaciones del modelo funcional de LDAP están divididas en tres grupos:

- Las operaciones de consulta, que permiten realizar búsquedas en el directorio y la recuperación de datos del mismo
- Las operaciones de actualización, que permiten agregar, borrar, renombrar o cambiar entradas en un directorio, y
- Las operaciones de autentificación y control, que permiten a los clientes ser identificados ante el servidor de directorios y por tanto controlar ciertos aspectos de la sesión, como por ejemplo restringir o permitir el acceso a ciertas entradas.

Además de estas operaciones, la versión 3 del protocolo LDAP define una manera para agregar nuevas operaciones que permitan hacer búsquedas o recuperaciones de

datos especializados. Estas operaciones se llaman las operaciones extendidas de LDAP. A continuación se presenta una descripción mas detalladas de los tres grupos de operaciones más importantes mencionadas primero.

3.2.3.1 Las operaciones de consulta

Las operaciones de consulta permiten a los clientes realizar búsquedas y recuperación de los datos de esas búsquedas de una sola vez, es decir, no existen operaciones de lectura de una entradas en particular. En el caso de que se requiera esto, se debe realizar una búsqueda que se restrinja a la entrada que se quiera obtener.

Para describir una búsqueda sobre un directorio se pueden especificar algunos parámetros que caractericen las entradas a obtener. A continuación se decriben algunos de estos parámetros:

- **Objeto base.** Corresponde al subárbol del directorio (expresado como un DN) sobre el cual la búsqueda estará restringida. (ver figura 3.7 en la página siguiente)
- **Alcance**. Indica la profundidad de la búsqueda sobre el objeto base. Existen tres tipos de alcances: *subtree*, que indica una búsqueda sobre todo el subárbol bajo el objeto base indicado; *onelevel*, que indica una búsqueda sobre el nivel siguiente al objeto base solamente; y *base* que es una búsqueda sólo en el objeto base, y que es usado para obtener una entrada particular del directorio. (ver figura 3.7 en la página siguiente)
- **Cantidad máxima de entradas a recibir**. Si el servidor encuentra 30 entradas pero se especifican en la búsqueda un máximo de 10 entonces sólo las 10 primeras entradas serán enviadas al usuario. Un valor de 0 (cero) en este parámetro indica que se desean recibir todas las entradas encontradas[3].

[3]Mediante configuraciones especiales de un servidor LDAP se puede restringir la cantidad máxima de entradas enviadas al usuario. Si el valor especificado por el cliente LDAP es mayor al impuesto por el servidor entonces se ignora el valor máximo solicitado por un cliente no privilegiado.

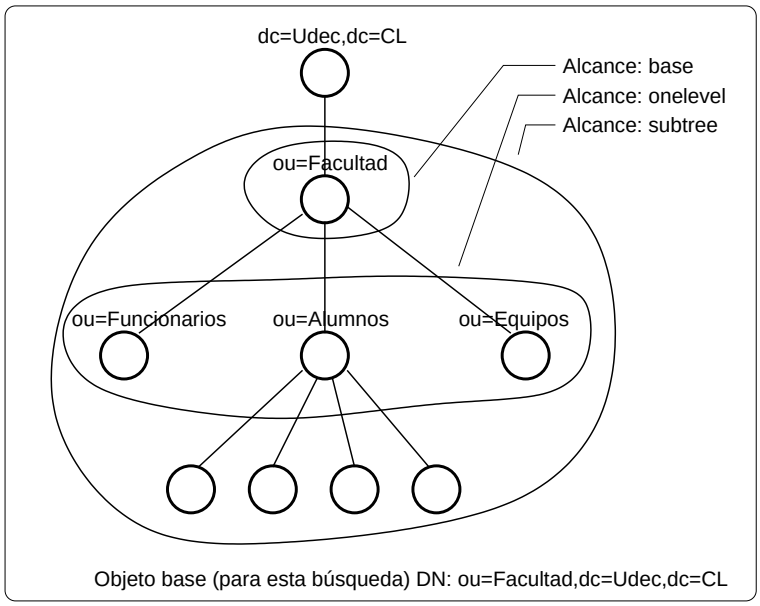

Figura 3.7: Entradas obtenidas de distintas búsquedas con el mismo objeto base de búsqueda pero distintos parámetros de alcance.

- **Cantidad máxima de tiempo**. Indica la cantidad de tiempo (en segundos) a usar como máximo para realizar la búsqueda[4] .

- **Filtro**. Describe los tipos de entradas en los cuales se está interesado en encontrar. Un filtro LDAP corresponde a una expresión que se describirá mas adelante. Para adelantar, (uid=sram) sería un filtro que caracteriza a las entradas cuyo atributo *uid* sea igual a *sram*.

- **Lista de atributos**. Para cada entrada encontrada, de acuerdo a los parámetros de búsqueda anteriormente mencionados, se especifican los atributos en particular que se desean recibir. Esto es útil para evitar tráfico innecesario desde el servidor al cliente LDAP. Si no se especifica una lista de atributos entonces se obtienen todos los atributos de las entradas.

Filtros LDAP

Los filtros LDAP son combinaciones de afirmaciones booleanas compuestas por

[4]Si el servidor de LDAP posee un tiempo máximo menor entonces se usa este último.

duplas atributo-valor. Cada afirmación se encierra entre paréntesis para separar una afirmación de otra. En estas afirmaciones se pueden usar comodines como el * (asterisco). Por ejemplo (uid=sram), indica todas las entradas cuyo atributo *uid* tenga el valor *sram*. El filtro (uid=sr*) indica todas las entradas cuyo atributo *uid* comience con *sr*, así las entradas con *uid* igual a *srojas* y *sruiz* también serían obtenidas por este filtro. Otra forma de uso del comodín asterisco son los filtros de presencia. Por ejemplo, (mail=*) es un filtro que en una búsqueda entregaría todas las entradas que tengan el atributo mail.

Los filtros LDAP también soportan los operadores *mayor que (>)* y *menor que* (<). Así por ejemplo el filtro (edad>28) indica todas las entradas cuyas entradas tengan su atributo *edad* mayor que 28.

La negación en los filtros es realizada con el signo de exclamación (!). Así, (!(edad>28)) indica las entradas cuyo atributo edad no es mayor que 28 (es decir, las edades menores o iguales a 28).

Para componer filtros mas complejos se usan los operadores AND (&) y OR (|). Se usa notación prefija para la aplicación de estos operadores. Por ejemplo el filtro: (&(sn=Ramírez)(cn=Salvador)), indica todas las entradas cuyo atributo *sn* sea Ramírez y cuyo atributo *cn* sea Salvador.

3.2.3.2 Las operaciones de actualización

Las operaciones de actualización son cuatro:

1. *add*, permite agregar una entrada a un directorio. Para llevar a cabo esta operación es necesario especificar los parámetros DN y una lista de duplas atributo-valor, que caracterizan la nueva entrada a agregar. Para que esta operación se puede realizar es necesario que:
   - El DN padre del DN especificado como parámetro ya debe existir en el directorio
   - No debe haber una entrada con el mismo DN
   - Los atributos de la entrada deben corresponder a los esquemas en cuestión
   - Los controles de acceso deben permitir la operación de agregado

2. *delete*, permite borrar una entrada de un directorio. Esta operación consta de un sólo parámetro, que es el DN de la entrada a borrar. Para que esta operación pueda llevarse a cabo es necesario que:
   - La entrada (especificada por su DN) a ser borrada exista
   - La entrada no tenga entradas hijas
   - Los controles de acceso permitan la operación de borrado

3. *rename*, permite modificar el nombre (DN) de una entrada. Entre los parámetros de esta operación se encuentran el DN de la entrada a borrar, su nuevo RDN y un parámetro opcional de nuevo DN padre. Para que esta operación se pueda realizar es necesario que:
   - La entrada (especificada por su DN) exista
   - El nuevo nombre de la entrada (especificada por su nuevo RDN) no esté en uso
   - Los controles de acceso permitan llevar a cabo esta operación

4. *modify*, permite modificar el valor de una entrada en un directorio. Los parámetros a especificar para esta operación son el DN de la entrada a modificar y una lista de duplas atributo-valor que reemplazarán a las ya existentes. Para poder realizar esta operación, es necesario que:
   - La entrada a ser modificada exista
   - Las modificaciones de cada uno de los atributos sea aceptada
   - La entrada resultante aún se rija por el esquema en cuestión
   - Los controles de acceso permitan esta operación

3.2.3.3 Las operaciones de identificación y control

Las operaciones de autentificación de LDAP son *bind* y *unbind*, y la operación de control es *abandon*.

La operación bind permite a un cliente LDAP autentificarse ante un servidor de directorios LDAP. Para esta autentificación, es necesario que el cliente presente al servidor un DN y un conjunto de credenciales para que el servidor pueda decidir si

dar o no los permisos correspondientes al cliente. Estos permisos dependerán de la identidad del cliente, determinada por el DN y las credenciales.

En el proceso de bind, existe un traspaso de datos de carácter sensible, desde el cliente al servidor. Estos datos sensibles corresponden a las credenciales y DN, que si son interceptadas por terceros, la seguridad de los datos del cliente se vería afectada. Existen varias formas de realizar este traspaso de datos en el proceso de bind. El método mas común y sencillo es el bind simple. En este proceso, el cliente envía el DN y una clave, en texto plano y a través de la red, al servidor LDAP. El servidor entonces verifica que la clave provista corresponda al valor del atributo *userpassword* del DN en cuestión. Si las claves corresponden entonces el servidor envía un código de éxito al cliente, en caso contrario se envía un código de fracaso de la operación.

Para superar el problema de la seguridad del bind simple, se puede usar Secure Socket Layer (SSL) o Transport Layer Security (TLS), que son protocolos para encriptar una sesión de transporte, como lo es el protocolo de red de LDAP. De esta manera un cliente LDAP continúa enviando los datos del bind en texto plano pero la plataforma SSL o TLS se encarga de encriptar toda la sesión y por tanto los datos ya no son transferidos en texto plano por la red.

Otros tipos de bind son soportados en la versión 3 del protocolo LDAP. Un ejemplo de estos tipos de bind es el bind SASL. SASL significa *Simple Authentication and Security Layer* y es un mecanismo para proveer una plataforma de autentificación a los protocolos de red[5]. Una de las ventajas más importantes del uso de SASL es que la incorporación de nuevos métodos de autentificación entre clientes y servidores LDAP no requieren de algún cambio en el protocolo LDAP sino sólo en el standard SASL.

La otra operación de autentificación es *unbind*. Esta operación realiza el proceso contrario a bind, es decir retira los privilegios asignados previamente al cliente, por parte del servidor LDAP sobre el directorio.

La operación de control, *abandon*, sirve para cancelar una operación previamente solicitada a un servidor por un cliente LDAP. Esto puede ser útil en aplicaciones

[5]Recurrir a la sección 2.3 en la página 13 (Seguridad) para más información acerca de SASL.

interactivas donde el tiempo de espera por la finalización de alguna operación ya sea demasiado y el usuario, impaciente, desee cancelar aquella operación.

3.2.4 El modelo de seguridad de LDAP

El propósito del modelo de seguridad de LDAP es proteger la información de un directorio contra accesos no autorizados. La autentificación de un cliente LDAP en un servidor es parte de la seguridad de un directorio, por lo que el modelo anteriormente visto, el modelo de autentificación, forma parte de este.

Otra parte del modelo de seguridad de LDAP la conforman los controles de acceso, que son la forma en la cual se especifican los privilegios que poseen ciertos usuarios, después de ser autentificados exitosamente. Estos controles de acceso a un directorio aún no han sido estandarizados aunque se está trabajando en ello[6]. Si bien la forma en la cual se especifican estos controles de acceso aún no está estandarizada, si es claro que los tipos de privilegios básicos a asignar a clientes sobre determinados DN de un directorio son escritura y/o lectura. Típicamente el dueño de una determinada entrada en un directorio podrá tener privilegios de escritura sobre esa entrada, mientras que otros sólo lectura a sus datos públicos y ningún privilegio sobre los datos sensible del usuario en cuestión. Cuando un cliente hace bind sobre un directorio para leer datos públicos de una entrada, no necesita proveer una clave (dado que los datos son públicos). En este caso se habla de un *bind anónimo*.

Finalmente, como ya se mencionó en la sección anterior, es posible encriptar toda una sesión entre un cliente y un servidor LDAP mediante el uso de SSL/TLS. Esto es útil, primero en el momento de la autentificación y luego para la transferencia de datos sensibles entre el cliente y el servidor LDAP.

3.3 El protocolo LDAP

El protocolo LDAP es un protocolo cliente/servidor orientado al mensaje. Esto significa

[6] Actualmente cada servidor LDAP (OpenLDAP por ejemplo) posee su propio control de acceso.

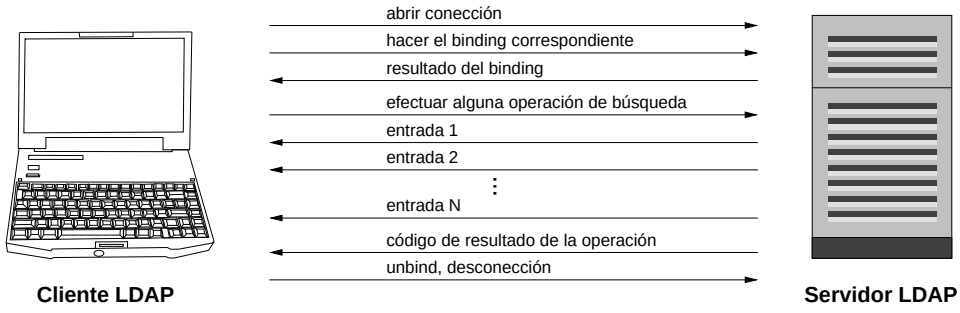

Figura 3.8: Un cliente efectuando una búsqueda sobre un directorio en un servidor LDAP

que un programa cliente, ejecutándose en algún computador, construye un mensaje de requerimiento de cierta acción y lo envía a través de la red a otro computador que está ejecutando un programa servidor[7]. Este programa servidor recibe el requerimiento, lleva a cabo alguna acción y retorna al cliente algún resultado. Ejemplos de protocolos cliente/servidor son el HyperText Transfer Protocol (HTTP), usado para servir páginas Web en la red; y el Simple Mail Transfer Protocol (SMTP), que es un protocolo usado para la transferencia de los correos electrónicos a través de Internet.

La interacción típica entre un cliente y un servidor LDAP es la siguiente:

– El cliente establece un sesión TCP/IP con el servidor LDAP. Esta operación es siempre acompañada por el *binding* del cliente en el directorio del servidor LDAP, por lo que a estas dos operaciones se acostumbra a llamar *binding* del cliente al servidor LDAP. El establecimiento de una sesión implica que el cliente LDAP debe especificar la dirección IP o nombre del servidor, además del puerto TCP sobre el cual el servicio LDAP está disponible[8]. Luego en el binding al directorio, el cliente debe proveer un username y una clave[9] para la autentificación y así acceder a los permisos y accesos apropiados para la ejecución de las tareas del siguiente punto.

– El cliente efectúa una o más operaciones de búsqueda (lectura) o de actualización en el directorio. Estas operaciones son las vistas en la sección del Modelo

[7]Es también posible que tanto el programa servidor como el programa cliente se encuentren ejecutándose en un mismo computador.

[8]El puerto TCP comúnmente usado para el servicio LDAP es el 389.

[9]Aunque cuando sea un binding anónimo para acceder a datos públicos, esta clave existe, aunque nula.

Funcional de LDAP. El cliente recibe los resultados de su operación por parte del servidor. Cada uno de estos resultados se recibe en mensajes separados, de ahí que LDAP sea un protocolo orientado al mensaje. Además de esto el cliente recibe por parte del servidor un código de resultado de la operación.

- El cliente termina y se desconecta del servidor.

El protocolo LDAP también permite que el cliente efectúe más de una operación a la vez, en cuyo caso cada mensaje por parte del servidor tendrá un código que especifique la operación a la cual pertenece tal mensaje.

Los mensajes intercambiados por clientes y servidores LDAP deben estar codificados de una manera especial, para asegurar la compatiblidad entre las distintas plataformas sobre las cuales las aplicaciones y servidores LDAP estén disponibles. LDAP usa una versión simplificada de Basic Encoding Rules (BER), llamada Lightweight BER (LBER). BER es un conjunto de reglas para la codificación de varios tipos de datos, tales como enteros y strings, en una forma independiente del sistema sobre el cual se esté usando. BER también define formas para combinar esos tipos de datos primitivos en estructuras de datos más complejas. La versión simplificada que usa LDAP, LBER, sólo posee los tipos de datos más usados actualmente y representa la mayoría de estos como strings simples.

3.4 Cuentas de Usuario en LDAP

Tal como se explicó en secciones anteriores, una cuenta es un acceso a un sistema para poder acceder a los servicios que esta provea. Una cuenta se compone, típicamente, de un nombre (username), una clave (password) y un espacio en disco (home directory). El username y password constituyen los datos para la autentificación del usuario en el sistema.

En LDAP, es posible implementar una cuenta de usuario como una entrada, con un esquema apropiado. Este esquema debe poseer atributos tales como un nombre de usuario, username o identificación (uid), una clave de acceso (userPassword) y un

directorio asociado en el sistema de archivos (homeDirectory), entre otros atributos opcionales.

Las ventajas de mantener las cuentas de usuario en un servidor LDAP son las siguientes:

1. **Flexibilidad**. A una cuenta LDAP es posible agregarle atributos mediante la agregación de esquemas a las entradas que constituyen las cuentas. Las cuentas en Unix por ejemplo tienen una estructura fija y no es posible agregar otros campos. Lo mismo sucede con cuentas en otros sistemas como WindowsNT. Sin embargo en LDAP, se pueden agregar a los datos del usuario atributos tales como la fotografía digital del usuario, su certificado criptográfico digital, etc. Si en el futuro se crea una aplicación que necesite de otro atributo para cada usuario entonces se puede agregar ese atributo sin problemas a la cuenta (entrada) del usuario en el directorio LDAP.

2. **Accesibilidad**. Actualmente existen muchas aplicaciones, tanto para Unix como para Windows, que soportan autentificación de usuario desde un servidor LDAP.

3. **Centralización**. Esto es una consecuencia del punto anterior. Dado que es posible mantener las cuentas de usuario en un servidor LDAP y éstas puedan accederse desde diferentes sistemas operativos, no es necesario que cada sistema maneje sus cuentas propias, sino que en lugar de ello se use un servidor de cuentas LDAP y desde los demás servidores (ya sea corriendo bajo Unix o Windows) sean accedidas.

Actualmente, el esquema estándar para las cuentas en LDAP se llama posixAccount y posee los atributos obligatorios de: cn (common name), uid (user identification), uidNumber, gidNumber y homeDirectory. Además posee los atributos opcionales de userPassword, loginShell, gecos y description. Otros esquemas permiten el atributo userCertificate (certificado de usuario) que es el usado para autentificar a un usuario y para habilitarlo a recibir y enviar datos encriptados ya sea en transacciones a través

del Web o correos electrónicos, tal como se explica en la sección 2.4, Criptografía: Certificados y Firmas digitales.

3.5 Software Actuales para el Servicio LDAP

Como se ha mencionado, LDAP es un protocolo cliente/servidor, lo que implica la existencia de al menos dos programas involucrados, un cliente y un servidor. En esta sección se revisan algunos programas servidor de LDAP, mientras que en la sección siguiente programas cliente de LDAP.

Actualmente existen muchos productos LDAP, dada la importancia que ha adquirido esta tecnología últimamente. A continuación se presentan los productos mas conocidos y que por tanto tienen un mayor uso en la actualidad:

- OpenLDAP. Esta es un implementación de desarrollo abierto de LDAP. La última versión estable (1.2.11) a la fecha de este producto soporta la versión 2 del protocolo LDAP. La versión 2.0.7, aunque no estables aún, soporta LDAP versión 3. Este producto es gratuito y de desarrollo libre. Su página Web es http://www.openldap.org.

- Netscape Directory Server. Este servicio es parte de la suite de protocolos de Netscape. Este servicio es una implementación nativa LDAP que soporta la versión 2 y 3 de éste. Este producto es comercial. La página Web de este producto se puede encontrar en http://www.netscape.com.

- IBM eNetwork LDAP Directory. Esta es una implementación nativa de LDAP soportando la versión 2 y 3 de este protocolo. Este producto también es comercial. La página Web de este producto se puede encontrar en http://www.ibm.com.

- Lotus Domino. Este es un conjunto de servicios software que ofrece la empresa Lotus Development Corp. Es un producto comercial cuya última versión soporta LDAP 2 y 3.

Software	Comercial	Versión LDAP	Estable	Página Web
OpenLDAP 1.2.11	No	2	Si	openldap.org
OpenLDAP 2.0.7	No	2 y 3	No	openldap.org
Netscape Directory	Si	2 y 3	Si	netscape.com
IBM LDAP Dir.	Si	2 y 3	Si	ibm.com
Lotus Domino	Si	2 y 3	Si	lotus.com
NDS	Si	2 y 3	Si	novell.com
Microsoft Active Dir.	Si	-	Si	microsoft.com

Tabla 3.1: Comparación de las características de los distintos software servidores LDAP

- Novell Directory Services (NDS). Es el servicio de directorios que viene con el sistema operativo NetWare de Novell. Este es un producto comercial. Página Web: http://www.novell.com.

- Microsoft Active Directory. Esta es la implementación de Microsoft del servicio de directorios y por supuesto es un producto comercial. Este servicio corre bajo Windows NT. Este servicio de directorios no es compatible con LDAP por sí sólo y dado el auge que ha tenido LDAP en este último tiempo se le ha agregado una interface LDAP que permite a los clientes LDAP acceder a la información almacenada en estos directorios. Página Web http://www.microsoft.com.

3.6 Servidores y Aplicaciones Compatibles con LDAP

Tal como se ha dicho, actualmente existen muchas aplicaciones que soportan LDAP como mecanismo para autentificación de usuarios. Estas aplicaciones varían desde lectores de correos hasta programas para servicios en Internet, tales como servidores de ftp, correo electrónico, POP e IMAP. A continuación se presenta una lista de las aplicaciones que soportan LDAP y que son de interés en el presente proyecto, junto con una descripción breve del uso de LDAP en sus servicios:

- **Netscape Communicator**. Este software comprende tanto un navegador para WWW como un lector de correo electrónico (entre otros utilitarios para Internet). Este software además es capaz de mantener la configuración y otros datos de un

usuario en un directorio de un servidor LDAP, y obtenerlos desde éste último cada vez que se inicia una sesión.

Entre los datos de usuario que este software puede manteneren en un directorio remoto LDAP, se encuentran los *bookmarks*, el *history*, los *cookies*[10] y las claves privadas para uso con SSL/TLS y el envío de email firmados digitalmente. La opción de Netscape Communicator para el uso de directorios LDAP se llama *Roaming User* y se puede encontrar en el menú Edit->Preferences->Roaming User.

- **Qmail**. Este software es un servidor de correo electrónico o servidor SMTP[11].

  En el proceso de recepción de un correo electrónico, Qmail puede reconocer usuarios en directorios LDAP. Las entradas que representan las cuentas deben implementar un esquema LDAP propio de Qmail, lo que implica que las entradas deben tener ciertos atributos específicos, requeridos por Qmail. Entre estos atributos se encuentran el username del usuario, el directorio en el cual se dejará el correo y la cantidad de espacio de disco máximo que puede usar el usuario en correos electrónicos.

- **Cyrus-IMAP-POP**. Este software incluye un servidor POP e IMAP, para la lectura de los correos electrónicos desde los computadores de escritorio de los usuarios[12].

  Cuando un usuario desea su correo electrónico, este debe proveer su username y clave de acceso. Con estos datos este software Cyrus-IMAP-POP se conecta al servidor LDAP y autentifica al usuario, ademas de darle acceso a la lectura de sus correos electrónicos, almacenados en el servidor.

- **ProFTPD**. Este software es un servidor de FTP.

  Tal como se mencionó en la sección 2.1 en la página 7 (Servicios Internet Tradicionales) , cuando un usuario se conecta a un servidor FTP, necesita de un

[10] Los *bookmarks* son el listado de páginas Web que son de interés para el usuario. El *history* es el historial de páginas Web visitadas por el usuario y los *cookies* son información que un servidor Web desea mantener del usuario pero que es guardada por el navegador en los archivos locales del usuario.

[11] Referirse a la sección 2.1 en la página 7 (Servicios Internet Tradicionales).

[12] Ídem.

username y una clave de acceso (excepto cuando se conecta anónimamente). El servidor ProFTPD puede autentificar a los usuarios en un servidor LDAP, en lugar de los servidores FTP tradicionales que pueden autentificar sólo a usuarios como cuentas locales Unix (o en servidores para Windows, en bases de datos de cuentas FTP locales).

Capítulo 4

Servicios DTI-UDEC

4.1 Servicios Internet en la DTI

La Dirección de Tecnologías de Información (DTI) es el organismo encargado de los sistemas centrales de conexión a Internet en nuestra Universidad. Además este organismo ofrece servicios Internet para cualquier alumno o funcionario interno. Estos servicios incluyen, principalmente, una casilla de correo electrónico y una página Web. Los servidores que ofrecen estos servicios se encuentran instalados en computadores con sistema operativo Unix (ya sea Linux, Digital Unix o AIX) y las cuentas de usuario son mantenidas en un servidor NIS para facilitar la administración de cuentas entre los distintos computadores con Unix. Sin embargo, tal como se puede apreciar en la figura 4.1 en la página siguiente, las cuentas NIS no pueden ser usadas por las aplicaciones de Windows.

La forma en que los alumnos y funcionarios solicitan una cuenta de acceso a estos servicios es a través del sistema automático de creación de cuentas, del cual se habla en una sección posterior.

4.2 Servicio de Laboratorios de Computación

La Universidad dispone de dos salas de computación con computadores personales. Dichas salas se encuentran habilitadas con equipamiento, software y administradores

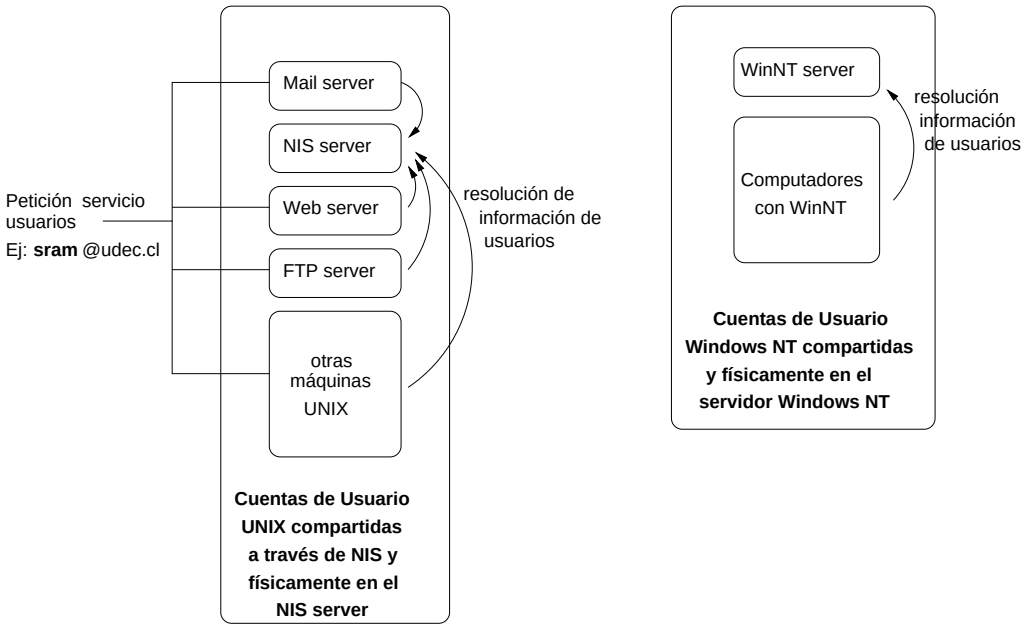

Figura 4.1: Modelo de autentificación de usuarios de los servicios Internet ofrecidos por la DTI

de sala, para asesorar en el uso de los recursos.

Estas salas están disponibles para todos los alumnos y funcionarios de la Universidad de Concepción, principalmente para estudiantes de aquellas unidades académicas que no cuentan con medios computacionales.

Modo de uso:

- Reserva masiva: solicitar formulario en Oficina Atención Usuarios DTI.
- Acceso libre alumnos y funcionarios : reservar en la misma sala o vía teléfono y al ingreso a la sala, acreditar que es alumno o funcionario de la Universidad.

Los Laboratorios de Computación, actualmente, son 2: Sala Udec1 y Sala Udec2.

A continuación se describen algunos detalles de cada uno de estos laboratorios:

Sala Udec1

Descripción : Sala con equipamiento PC's compatibles y terminales conectados a la red del campus.

Ubicación : Edificio D.T.I. (al costado del foro)

Horario : Lunes a Viernes de 10:00 a 19:00 hrs.

Anexo : 3797

Configuración :

- 16 computadores Pentium 166 Mhz, 32 MB RAM, con Windows 95, Office 97, Software conexión a la red.
- 4 computadores multimedia (con tarjeta de sonido) Pentium 166 Mhz, 32 MB RAM, con Windows 95, Office 97, Software conexión a la red
- 1 impresora láser
- 1 impresora de inyección de tinta color
- 1 scanner
- 3 terminales

Sala Udec2

Descripción : Sala con equipamiento PC's compatibles y terminales conectados a la red.

Ubicación : Acceso Edificio Facultad de Cs. Económicas y Administrativas

Horario : Lunes a Viernes de 9:00 a 20:00 hrs.

Anexo : 4760

Configuración :

- 14 computadores Pentium 100 Mhz, 24 MB RAM, Windows NT Workstation, Office 97, software conexión a la red
- 1 impresora láser
- 2 terminales

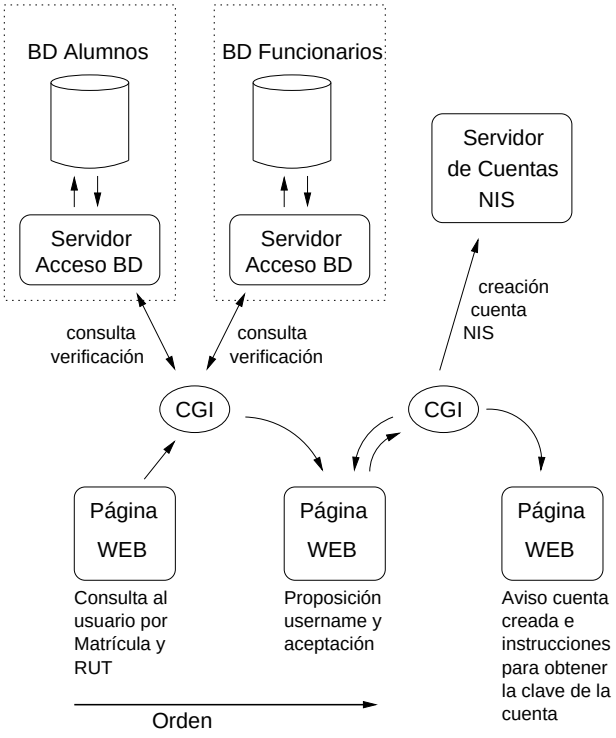

Figura 4.2: Esquema del sistema actual de creación automática de cuentas.

4.3 Creación Automática de Cuentas de Usuario

En el año 1997 se implementó en los computadores centrales de nuestra Universidad un sistema de creación automática de cuentas de usuario que permitió que todo usuario posea una cuenta de acceso a los servicios Internet ofrecidos por la DTI.

Este sistema automático de creación de cuentas es accesible a través de WWW por una página Web, donde el usuario debe introducir su número de matrícula (de funcionario o alumno) y RUT para que el sistema verifique si los datos corresponden a un alumno o funcionario de la Universidad. Esta verificación se hace conectándose a las bases de datos corporativas de la Universidad, que mantiene la información de cada alumno y funcionario. Si el usuario no posee una cuenta entonces el sistema le crea una automáticamente y le muestra información acerca de cómo obtener la clave de acceso de la cuenta recién creada (ver figura 4.2).

Uno de los problemas que surgieron para la creación de este sistema fue que en ese tiempo las librerias para la conexión remota a las bases de datos Oracle eran sólo

accesibles para el sistema sobre el cual el software Oracle había sido adquirido (en este caso Digital-Unix), y el acceso remoto tenía que realizarse desde el servidor WWW (Linux), que es donde se ejecuta el programa CGI, después de invocado desde la página Web, correspondiente a la creación de cuentas. Este problema se solucionó creando un servicio TCP/IP, en los computadores en los cuales residen las bases de datos, que acepta conexiones remota a través de la red y se conectaba localmente a las bases de datos. En el esquema, este servidor se nombra como *Servidor de Acceso BD.*

Capítulo 5

Proyecto de Directorios LDAP en DTI-UDEC

5.1 Modelo de Servicios y Aplicaciones

En el nuevo esquema de cuentas, en directorios LDAP, la autentificación de usuarios en los servicios Internet es a través de un servidor LDAP, en lugar de un servidor NIS. Esta misma cuenta LDAP sirve también para el almacenamiento de los datos de aplicaciones tales como Netscape Communicator, para que cada vez que un usuario use un computador y ejecute Netscape, este configure automáticamente lo necesario con datos obtenidos de la cuenta LDAP (ver figura 5.1 en la página siguiente). Tal como se ha mencionado, esta configuración incluye datos como el *bookmark*, el *history*, *claves privadas,* servidor de correo electrónico y *username,* entre otros.

Por otro lado, muchas aplicaciones para Windows son capaces de autentificar usuarios en un servidor LDAP. Esto simplifica la labor de administración, al evitar la necesidad de mantener una base de datos de usuario aparte para Unix y Windows. A su vez, con el proceso de creación automática de cuentas de usuario LDAP, descrito mas adelante, se creará de una sola vez un punto de acceso tanto para los servicios Internet como para el uso de los computadores de los laboratorios de computación, que actualmente usan Windows.

Cabe notar que este modelo de servicios propuesto no limita que cada servicio esté instalado en un computador diferente, como se podría pensar después de ver la figura 5.1 en la página siguiente. En esta figura las cajas representan servidores pero como

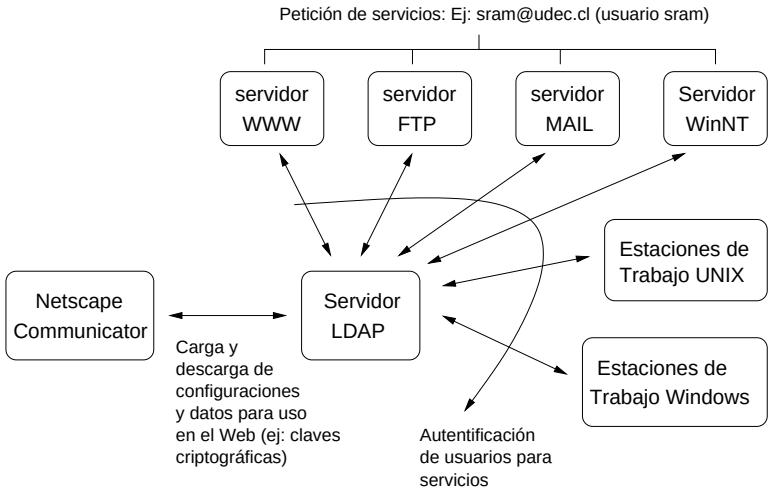

Figura 5.1: Autentificación de usuarios por parte de aplicaciones y servicios Internet en el modelo propuesto

servicios software y no computadores servidores. De esta manera bien podrían estar todos los servicios en un sólo computador, incluso el mismo servidor LDAP. En este caso, desde el punto de vista de las cuentas, sería lo mismo tener cuentas locales Unix o en un servidor LDAP, tal como se ha propuesto. Sin embargo esto, un servidor LDAP tiene otras funciones importante también, que no podrían ser reemplazadas. Estas otras funciones son que un servidor LDAP es donde se publican los certificados digitales de los usuarios[1], además de mantener la información de los usuarios de Netscape Communicator[2]. La utilidad de un servidor LDAP en una organización entonces, no depende de la configuración de computadores y servicios en ellos.

5.2 Creación Automática de cuentas LDAP

En el año 1997, año en que se desarrolló el primer sistema automático de creación de cuentas, las librerias de desarrollo para la conexión a bases de datos Oracle no eran accesibles para el sistema operativo Linux, lo que llevó al desarrollo de otro servicio, que sirviera de intermediario entre un cliente remoto y la base de datos local. Este problema no existe actualmente, pues Oracle ha liberado una versión gratis de las

[1]Ver sección 2.4 en la página 16 (Criptografía: Certificados y Firmas Digitales).
[2]Ver sección 3.6 en la página 38 (Servidores y Aplicaciones Compatibles con LDAP).

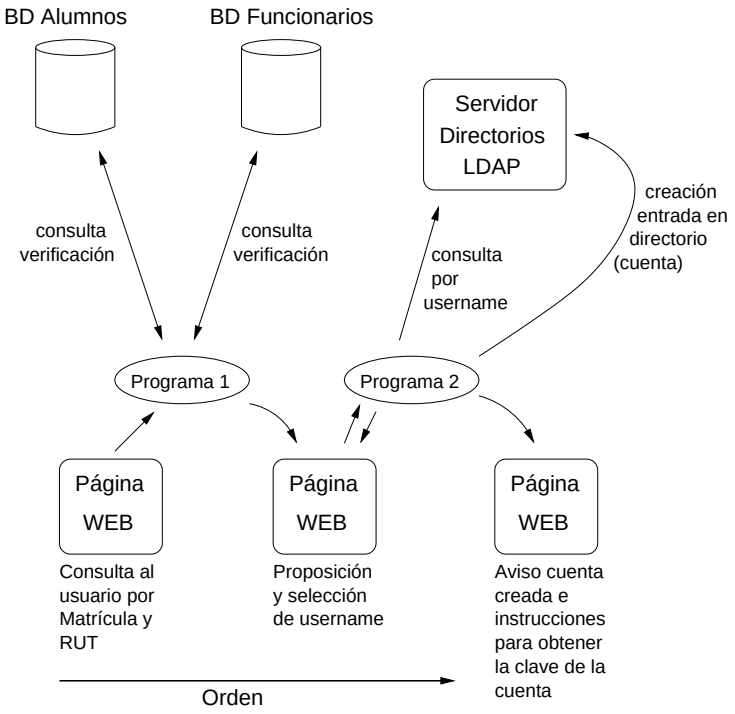

Figura 5.2: Esquema del sistema automático de creación de cuentas LDAP

librerias de desarrollo para la conexión a sus bases de datos. De esta manera, ya no es necesario el *Servicio Acceso a BD* mostrado en el esquema del anterior sistema automático de creación de cuentas.

Por otro lado, el sistema automático de creación de cuentas fue programado en lenguaje C, como un programa CGI. Esto ha hecho un tanto difícil el mantenimiento de este software dada la dificultad que para muchos presenta la programación en este lenguaje. Mas aún, cualquier programa hecho en C es un programa largo, de varias líneas de código, que requiere de buena documentación para poderlo mantener en el tiempo.

Actualmente, tal como se mencionó en la 2.5 en la página 19, existen lenguajes *script* especializados en labores del Web, lo que facilita y acorta significativamente los programas. Un ejemplo de estos lenguajes es PHP, que hoy en día es un software abierto y con mucho desarrollo .

En el modelo del sistema automático de creación de cuentas propuesto para el nuevo esquema de cuentas LDAP, se propone PHP sobre un servidor Unix como

plataforma software de desarrollo para los dos programas que componen el nuevo sistema automático de creación de cuentas.

El sistema automático de creación de cuentas LDAP tiene, básicamente, la misma forma que el sistema anterior. El sistema se compone de dos programas (programa 1 y programa 2) y una página Web inicial, aunque cada uno de los dos programas mencionados generan como salida otras páginas Web, dado que ésta es la interfaz de usuario (ver figura 5.2 en la página anterior). En la página Web inicial se consulta al usuario por su número de matrícula (ya sea de alumno o funcionario) y su RUT. Estos datos son tomados por el programa 1 y verificados directamente en las bases de datos corporativas, para probar la validez de ellos. Después de ésto, se presenta al usuario una página Web con una serie de *usernames* que este puede elegir o proponer alguno. El programa 2 toma el dato del *username* y verifica si ya existe tal en el directorio LDAP. Si ya existe entonces se vuelve la página de proposición y selección de *username*, en caso contrario se crea la entrada en el directorio del servidor LDAP con toda la información necesaria, incluyendo el certificado digital que contiene la clave pública del usuario. Se termina por mostrar al usuario información acerca de cómo obtener su clave de acceso y su certificado digital conteniendo su clave privada.

Como se puede apreciar, el diseño de estos programas es muy simple y por tanto no merecen un mayor énfasis en este aspecto. En efecto, estos programas en total hacen alrededor de 100 líneas de código útil y fueron programados por el autor en cuestión de horas, luego de aprender el lenguaje PHP. Este nuevo sistema de creación de cuentas LDAP ha estado en una fase de prueba desde entonces, sin presentar fallas.

5.3 Resultados de la etapa de Pruebas

Tal como se menciona en el capítulo de definición del proyecto, éste se divide en dos etapas: de pruebas y de implementación final. En la etapa de pruebas se ha instalado un servidor LDAP OpenLDAP versión 2.0.7, dado que esta soporta la última versión del protocolo LDAP. Sin embargo, como esta versión aún no es una versión estable se encontraron algunos problemas de compatibilidad con el software Netscape

Communicator para almacenar la información de *Roaming User*[3] en el servidor LDAP. Debido a este problema se debió usar OpenLDAP 1.2.11[4].

Un problema parecido ocurrió con el software PHP. Inicialmente se usó PHP en su último versión, la versión 4. El problema surgió cuando se intentó usar este lenguaje para hacer una conexión a un servidor LDAP, fue imposible realizarla. Sin embargo con la versión 3 no hubieron problemas.

Después de tener instalado el servidor LDAP se crearon los directorios de algunos usuarios para probar la conectividad con Netscape Communicator y todos los demás servidores que se especifican en la figura 5.1 en la página 47. Como servidor WWW se usó Apache, como servidor FTP se usó ProFTPD y como servidor de correo electrónico se usó Qmail. No se realizaron pruebas de conectividad con computadores con WindowsNT por problemas de tiempo y disponibilidad de equipos con este sistema operativo. Se efectuaron además pruebas recibiendo correo electrónico para los usuarios del directorio LDAP. Todo esto funcionó perfectamente.

Finalmente se crearon las herramientas para la creación automática de cuentas LDAP a través del Web[5]. Estas herramientas fueron creadas en PHP versión 3, dado el problema mencionado anteriormente. Estos programas son muy simples y funcionan sin problemas.

5.4 Detalles de la etapa de Instalación final

En esta etapa se implementan todas las herramientas y servicios de la etapa de pruebas en el escenario final, donde funcionan actualmente los servicios que ofrece la DTI, esto es, en los computadores centrales de este organismo.

Para llevar a cabo la transición desde el actual sistema de cuentas al nuevo, es necesario considerar algunos aspectos para lograr la transparencia para el usuario. Estos aspectos incluyen:

[3] Ver sección 3.6 en la página 38 (Servidores y Aplicaciones compatibles con LDAP).
[4] Ver sección 3.5 en la página 37 (Software Actuales para el Servicio LDAP).
[5] En la sección 5.2 en la página 47 (Creación Automática de Cuentas LDAP) se describe este sistema en detalle.

1. Migrar todas las cuentas actuales Unix (NIS) a LDAP. Dada la cantidad de cuentas Unix actuales (alrededor de 10.000), es necesario realizar esta migración con la ayuda de un programa que lea la información de las cuentas y construya entradas en el directorio del servidor LDAP.

2. Reemplazar todos los programas servidores actuales (correo electrónico, ftp, etc.) que trabajan con cuentas Unix (NIS) a programas que soporten LDAP. Estos programas corresponden a los servicios cuyo modelo se presentó en la sección anterior.

Si los dos puntos anteriores son logrados exitosamente, el usuario no debería percatarse del cambio y continuar trabajando sin problemas tal como antes. Sin embargo hay que considerar además que para la realización de los dos puntos anteriores, es necesario un tiempo que dependerá de la capacidad de proceso y entrada/salida del servidor.

El programa mencionado en el punto 1 anterior, debe leer una serie de cuentas Unix almacenadas en un archivo ASCII (una cuenta por línea) y escribir la información de aquellas cuentas (sumado a otros datos tales como el certificado del usuario) en un archivo con un formato adecuado para el posterior poblado de la base de datos del directorio LDAP. Este formato se llama LDIF (LDAP Data Interchange Format) y se define dentro del estádar LDAP. Una vez creado este archivo de salida, en formato LDIF, se usan las herramientas para poblar directorios. Estas herramientas se distribuyen junto a los software servidor LDAP, revisados en la sección 3.5.

La plataforma más apropiada sobre la cual se debe crear este programa es Unix, debido a que el servidor LDAP se instalará en un computador con este sistema operativo. El lenguaje a usar para la construcción del programa puede ser cualquier lenguaje de programación de propósito general, aunque se recomienda Perl, o mejor aún C, por su eficiencia, la que se necesitará por el volumén de cuentas actuales.

Dada la simplicidad de este programa, no es necesario un mayor análisis ni diseño.

Sin embargo lo anterior, esta etapa aún no ha sido llevada a cabo por diferentes razones, entre ellas la disponibilidad de recursos para la implementación del servicio

LDAP (se necesita de un computador ojalá exclusivo para esta labor, dado el alto tráfico que tendrá) así como la organización y sincronización de las tareas de las distintas personas involucradas en la administración de los servicios ofrecidos en la DTI.

Capítulo 6

Conclusiones

Las ventajas de ofrecer servicios a través de Internet son muchas, entre estas se pueden citar el servicio de atención permamente y la asistencia inmediata. Por estas razones muchos bancos y otras organizaciones han hecho grandes inversiones para llevar sus servicios a Internet. Actualmente en nuestra Universidad pocos servicios se ofrecen a través de la red, principalmente debido al hecho de la carencia de una plataforma software que provea seguridad en las transacciones llevadas a cabo a través de este medio.

Una memoria de título reciente aborda este tema y como resultado de aquel proyecto actualmente existe un servidor de certificados digitales en nuestra Universidad. Sin embargo la integración de tal sistema con los demás servicios, el uso de una base de datos común para el almacenamiento de la información de los usuarios y la creación de herramientas que permitan la creación automática de tal información aún siguen siendo una necesidad insatisfecha.

A lo largo de este documento se presentó un estudio de las tecnologías necesarias para solucionar los problemas anteriormente mencionados. También se describen los detalles para su implementación como solución informática a esos problemas.

A continuación se presentan las conclusiones de este proyecto.

6.1 de la Experiencia con Software Abierto

A pesar del gran desarrollo del software libre en el último tiempo, muchos de éstos aún requieren de la inversión de bastante tiempo para lograr ponerlos en marcha con las opciones que se necesitan. Falta de documentación y problemas en algunas versiones son la causa de esto, en cuyos casos es necesario ponerse en contacto con los desarrolladores (a través de Internet) para satisfacer las dudas y resolver los problemas. En muchos casos no es posible tener la seguridad de que la causa del mal funcionamiento de una opción de algún software es un problema del software mismo o es el mal uso por parte del instalador del software. En todo este proceso es imprescindible el dominio del idioma inglés para llevar a cabo toda esta interacción, dado que la mayor parte de estos software usan este lenguaje como lenguaje de comunicación entre los distintos desarrolladores a lo largo del mundo.

Sin embargo este no es el caso de todos los software de este tipo. Existen muchos software, como por ejemplo Apache (servidor WWW), cuya instalación y puesta en marcha es trivial y no consume más que unos minutos. Esto depende mucho del tiempo que tenga el proyecto de desarrollo del software libre. En el ejemplo, Apache es un proyecto de más 5 años, mientras que el desarrollo de OpenLDAP o PHP no tienen más de 3 años.

6.2 del Desarrollo de Programas para el Web en PHP

Como lenguaje de desarrollo de programas para el Web, PHP resulta de gran ayuda. Este lenguaje posee conectividad con muchas bases de datos actuales, entre ellas Oracle (que es el motor de bases de datos usado en nuestra Universidad) y LDAP. La característica de lenguaje script además de la gran cantidad de funciones incorporadas en PHP hacen que los programas desarrollados en este lenguaje sean muy cortos y por tanto fáciles de mantener. Este es el caso de los programas para la creación automática de cuentas LDAP, desarrollado en este proyecto.

6.3 de los Costos y Beneficios del uso de LDAP

En la sección 3.4 (Cuentas de Usuario en LDAP) se revisaron en detalle las ventajas de mantener las cuentas de usuario en un servidor LDAP. A través de este esquema es posible publicar los certificados de usuario y por tanto permitir llevar a cabo operaciones seguras a través de la red. Esta importante ventaja sumada a la disminución en el costo de admistración de los sistemas de cuentas (debido a que muchas aplicaciones de diferentes plataformas pueden reconocer usuarios en directorios LDAP), muestra que el uso de LDAP para el almacenamiento de los datos de los usuarios para los servicios Internet, constituye una ventaja importante sobre otros esquemas.

Una desventaja del esquema LDAP es la relativamente poca documentación disponible. Esto es debido a que sólo recientemente este estándar ha sido incorporado a los sistemas operativos actuales. Sin embargo esto está cambiando rápidamente dado que muchas empresas están portando sus sistemas de cuentas a LDAP por las ventajas ya mencionadas.

6.4 del Proyecto LDAP en DTI-UDEC

Tal como se ha mencionado, el proyecto de instalación del sistema de Directorios LDAP en el sistema de computadores centrales de la DTI aún no ha sido llevado a cabo. Sin embargo en la etapa de pruebas se ha realizado exitosamente en un ambiente, que si bien de menor escala, posee las mismas cualidades del sistema final en cual se instalará el sistema. El único hecho que diferencia al sistema de prueba con el sistema final es la cantidad de usuarios (entradas en el directorio de usuarios) a manejar. El sistema de pruebas se llevó a cabo con 10 usuarios, mientras que el sistema final tendrá la cantidad de cuentas actuales que maneja el sistema central de la DTI, es decir, alrededor de 12000 cuentas.

Esto último no representa un problema dado que el actual sistema de cuentas trabaja con NIS, un sistema de cuentas de red que a bajo nivel trabaja con el mismo sistema de base de datos con el cual trabaja OpenLDAP. Este sistema de base de datos se llama

DBM.

6.5 del Futuro del Proyecto: Recomendaciones y Sugerencias.

Para la instalación final del sistema de cuentas LDAP en el sistema de computadores centrales de la DTI se recomienda usar versiones estables de los respectivos software abiertos que se usarán, como son OpenLDAP, Qmail, Apache, PHP y ProFTPD. En las conclusiones anteriores acerca de la experiencia con software abierto se explica la razón de esto. Típicamente los proyectos de software abierto tienen disponible dos grandes versiones: una que es la más avanzada y que es la que se encuentra en pleno desarrollo y otra versión que si bien tiene menos cualidades, es más estable y ha sido probada por más tiempo, lo que no implica, por supuesto, que esté cien por ciento libre de errores. En los grandes proyectos de software abierto, la etapa de prueba del software es muy importante, principalmente por la gran cantidad de desarrolladores involucrados, lo que implica muchas formas distintas de programar, aumentando así la probabilidad de la creación de errores en el código. Sin embargo, la etapa de pruebas es muy fuerte en estos proyectos, dado que no sólo los desarrolladores la llevan a cabo sino muchas personas alrededor del mundo que, con tiempo, instalan estas últimas versiones de desarrollo, encontrando errores y posiblemente sugiriendo, a través de lo medios adecuados (listas de correo por ejemplo), un parche a tal error.

Por otra parte, en el actual proyecto se consideró un sistema automático de creación de cuentas, que ayuda en forma considerable a la administración. Sin embargo, para completar esta automatización, se debe pensar además en la eliminación automática de cuentas. Cada vez que una persona, con cuenta de usuario, se retira de nuestra Universidad (por ejemplo funcionarios que se retiran o alumnos egresados), su cuenta de usuario se mantiene por largo período, sin haber un procedimiento automático que cuando se detecte un retiro ya sea de alumno o funcionario (en las bases de datos corporativas) se elimine o al menos se advierta la eliminación de la cuenta de usuario también. El presente proyecto no incluye esto, aunque sería de gran utilidad para automatizar aún más la administración y así poder brindar un mejor servicio dada la

cantidad de usuarios que existen actualmente en nuestra Universidad.

Capítulo 7

Glosario de Términos

ASCII Abreviación de *American Standard Code for Information Interchange (Código estándar americano para el intercambio de información).* Corresponde a un código que asigna a cada caracter alfanumérico (letras, números y símbolos) un número o clave, para hacer posible el intercambio de datos entre computadores.

ASP Abreviación de *Active Server Pages*, especificación para la creación dinámica de páginas Web. En este esquema, el servidor Web crea la página Web en el instante en que el cliente solicita el recurso.

BER Abreviación de *Basic Encoding Rules*. Estándar para la codificación de tipos y estructuras de datos, independientes de la plataforma sobre la cual se implemente. Dado que muchos computadores poseen tipos de datos incompatibles entre ellos (arquitecturas *big-endian* y *little-endian* por ejemplo), estos estándares son necesarios para hacer posible la comunicación en éstos.

CCITT Abreviación de *Consultative Commitee on International Telephony and Telegraphy.* Una organización que actualmente se conoce como ITU (*International Telecommunication Union*) y que se encarga de dictar estándares de comunicaciones.

CGI Abreviación de *Common Gateway Interface*. Una especificación para el intercambio de datos entre un cliente WWW y un programa a ejecutarse en un servidor

WWW, al cual el cliente se conecta. El cliente solicita un recurso a un servidor WWW, este le envía un documento al cliente, quien opcionalmente lo llena con datos y los envía al servidor, quien posteriormente pasará los datos al programa CGI para su ejecución. La salida del programa CGI es un documento Web, que el servidor pasa al cliente finalmente.

Directorio Una base de datos especializada en operaciones de lectura, que organiza los datos en una estructura jerárquica de árbol y que es usada principalmente para almacenar datos de los usuarios de una organización.

DN Abreviación de *Distinguished Name,* y corresponde a la ruta jerárquica completa de un elemento en un directorio. Un DN es la clave de un elemento en un directorio. Ejemplo: “uid=sram, dc=Udec, dc=CL”.

DNS Abreviación de *Domain Name System,* que es el sistema usado en Internet para la obtención de un número IP (ejemplo 152.74.16.83) a partir de un nombre, tal como www.udec.cl. El sistema también permite lo contrario, es decir obtener un nombre a partir de un número IP.

FTP Abreviación de *File Transfer Protocol*. Sistema usado en Internet la transferencia de archivos de un computador a otro.

HTML Abreviación de *Hyper Text Markup Language,* lenguaje usado para crear los documentos Web.

HTTP Abreviación de *Hyper Text Transfer Protocol*, protocolo usado para transferir los documentos Web entre servidores y clientes del WWW.

IETF Abreviación de *Internet Engineering Task Force,* la principal organización de estándares para Internet.

IMAP Abreviación de *Internet Message Access Protocol,* un protocolo usado para que los usuarios, a través de un cliente especializado, puedan obtener sus correos electrónicos desde un servidor de estos. IMAP tiene más funcionalidades que POP

(protocolo usado para los mismos fines), entre las cuales se destacan la posibilidad de trabajar con carpetas en el servidor.

IRC Abreaviación de *Internet Relay Chat,* un sistema de texto-conferencia a través de internet en el cual pueden interactuar muchos usuarios a la vez en múltiples canales de discusión.

Kerberos Sistema de autentificación, en el cual clientes y servidores comparten unos datos secretos, llamados *tickets,* los que ayudan a la identificación de los protagonistas de una determinada conexión.

LAN Abreviación de *Local Area Network*, una red que conecta computadores de un area local reducida, por ejemplo, los computadores de un oficina o edificio. Estas redes se caracterizan por su gran ancho de banda.

LBER Abreviación de *Light Basic Encoding Rules,* versión simplificada de BER, usada en LDAP.

LDAP Abreviación de *Lightweight Directory Access Protocol,* estándar que define la estructura y organización de los directorios y el protocolo de red a través del cual un cliente LDAP accesa los datos del directorio a través de requerimientos al servidor LDAP.

LDIF Abreviación de *LDAP Data Interchange Format,* un formato que representa los datos de un directorio LDAP en ASCII.

MIME Abreviación de *Multipurpose Internet Mail Extensions*, una especificación para dar formato ASCII a datos que no lo son, permitiendo así su transparente transferencia a través de diferentes computadores en Internet.

NFS Abreviación de *Network File System,* un sistema de archivos de red que permite compartir un sistema de archivos local (en un disco duro local por ejemplo) a través de múltiples computadores en una red.

NIS Abreviación de *Network Information System,* un sistema a través del cual es posible mantener centralizado los datos usados por los usuarios de computadores usando Unix, en un solo servidor, haciendo posible la transparencia en un red de computadores en el sentido de que el usuario use datos únicos que serán válidos no importando el computador sobre el cual se encuentre trabajando en un momento determinado.

NNTP Abreviación de *Network News Transfer Protocol,* protocolo usado para publicar y distribuir noticias o mensajes a través de una red de computadores, tal como Internet.

OSI Abreviación de *Open System Interconnection*, un estándar ISO que especifica la comunicación en una red de datos. Este modelo actualmente no se encuentra en uso en Internet, sino TCP/IP. Veáse TCP/IP.

Password Palabra de paso o clave usada para acceder a algún servicio.

PHP Abreviación de *PHP Hypertext Preprocessor,* lenguaje usado para la creación de páginas Web dinámicas en un servidor Web.

POP Abreviación de *Post Office Protocol*, protocolo para obtener los correos electrónicos de un servidor. Veáse IMAP.

RDN Abreviación de *Relative DN*, cada uno de los componentes de un DN. Veáse DN.

RFC Abreviación de *Request For Comments,* un documento que especifica algún cambio o una nueva idea en algún ámbito de Internet. Cualquier persona puede crear un RFC y si este gana interés entonces puede convertirse en un estándar.

SASL Abreviación de *Simple Authentication and Security Layer,* un protocolo para agregar métodos de autentificación a protocolos de aplicaciones.

SMTP Abreviación de *Simple Mail Transfer Protocol,* un protocolo para la transferencia de correos electrónicos en Internet.

SSL Abreviación de *Secure Socket Layer*, un protocolo, desarrollado por la empresa Netscape, para la transferencia de datos encriptados en Internet.

TCP/IP Abreviación de *Transmission Control Protocol/Internet Protocol,* un conjunto de protocolos que especifican la comunicación en Internet. Es el conjunto de protocolos más usado en Internet actualmente.

TLS Abreviación de *Transport Layer Security,* un protocolo, desarrollado por la IETF, para la transferencia de datos encriptados en Internet.

UID Abreviación de *User IDentification,* número identificador de un usuario en un sistema Unix.

Username Nombre que identifica a un usuario de otro.

WWW Abreviación de *World Wide Web*, red que conforman todos los servidores de documentos Web en Internet.

Capítulo 8

Bibliografía y Referencias

1. W. Yeong, T. Howes, y S. Kille, “X.500 Lightweight Directory Access Protocol”, RFC 1487, Julio de 1993.

2. W. Yeong, T. Howes, y S. Kille, “Lightweight Directory Access Protocol”, RFC 1777, Marzo de 1995.

3. M. Wahl, T. Howes, y S. Kille, “Lightweight Directory Access Protocol (v3)”, RFC 2251, Diciembre de 1997.

4. M. Wahl, A. Coulbeck, T. Howes, y S. Kille, “Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions”, RFC 2252, Diciembre de 1997.

5. M. Wahl, S. Kille, y T. Howes, “Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names”, RFC 2253, Diciembre de 1997.

6. T. Howes, “The String Representation of LDAP Search Filters”, RFC 2254, Diciembre de 1997.

7. T. Howes, y M. Smith, “The LDAP URL Format”, RFC 2255, Diciembre de 1997.

8. M. Wahl, “A Summary of the X.500(96) User Schema for use with LDAPv3”, RFC 2256, Diciembre de 1997.

9. T. Howes, y M. Smith, “The LDAP Application Program Interface”, RFC 1823, Agosto de 1995.

10. J. Postel, y J. Reynolds, “File Transfer Protocol (FTP)”, RFC 959, Octubre de 1985.

11. Jonathan B. Postel, “Simple Mail Transfer Protocol (SMTP)”, RFC 821, Agosto de 1982.

12. J. Myers, y M. Rose, “Post Office Protocol (POP) - version 3”, RFC 1939, Mayo de 1996.

13. M. Crispin, “Internet Message Access Protocol (IMAP) - Version 4rev1”, RFC 2060, Diciembre de 1996.

14. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, “Hypertext Transfer Protocol”, Junio de 1999.

15. “X500 Directory Standard Home Page”, http://www.nexor.com/info/directory.htm

16. “Community developed LDAP software”, http://www.openldap.org

17. “DevEdge Online - DevEdge Home Page”, http://developer.netscape.com

18. “PHP: Hypertext Preprocessor”, http://www.php.net

19. “Qmail: a replacement to Sendmail”, http://www.qmail.org

20. “Sendmail Home Page”, http://www.sendmail.org

21. “Exim Internet Mailer”, http://www.exim.org

22. “The Postfix (formerly VMailer) Home Page”, http://www.postfix.org

23. “The Apache Software Foundation”, http://www.apache.org