5. RESULTADOS

El resultado de este proyecto es el desarrollo de la Intranet Corporativa para Auteco S.A. A continuación se explica cómo instalar y configurar la aplicación y las herramientas necesarias para su correcto funcionamiento y, finalmente, se dan unas pautas para la correcta utilización de la Aplicación Web.

5.1 Instalación Completa. Requisitos

Para poder utilizar la Intranet-Auteco se facilita un método completo de instalación en función del software que se tiene instalado en el servidor.

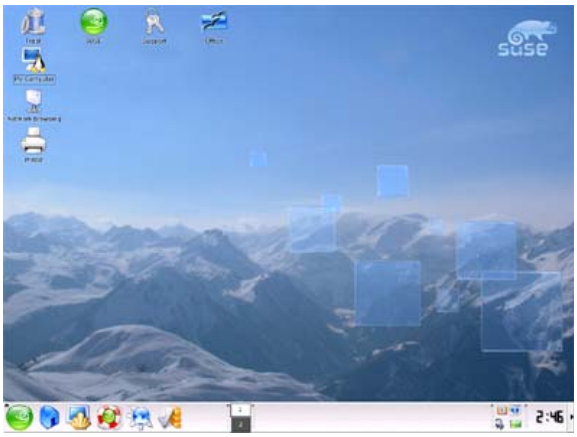
Para la correcta instalación de la aplicación necesitan los siguientes requisitos técnicos mínimos iniciales:

- Procesador de 800Mhz o más, y 256 MB RAM.
- SUSE Linux como Sistema Operativo.
- Internet Explorer 4.0, Netscape Navigator 5 u otro navegador que soporte las funcionalidades de los anteriores.
- Disco duro con 500 MB libres.

Los pasos para la instalación son los siguientes:

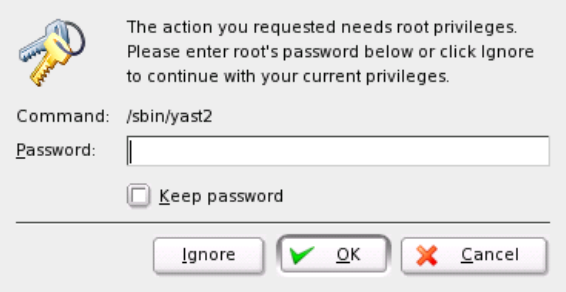
5.1.1 Instalación Servidor Apache, PHP y MySQL

Encender el Servidor. Ingresar como Usuario (no como root).

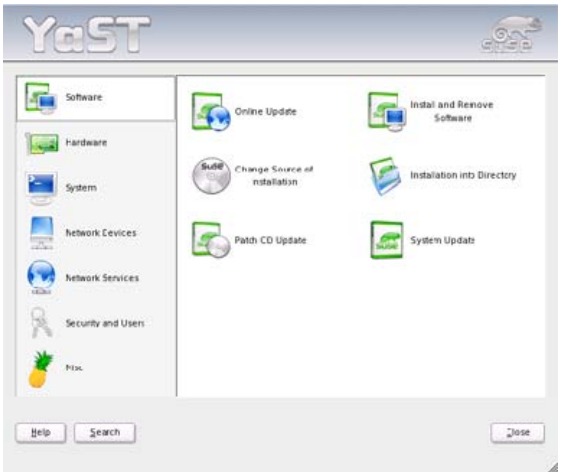


Iniciar el YaST. (Hacer esto dando clic en el botón "Start" de la barra de herramientas del SUSE => menú "System" => "YaST"

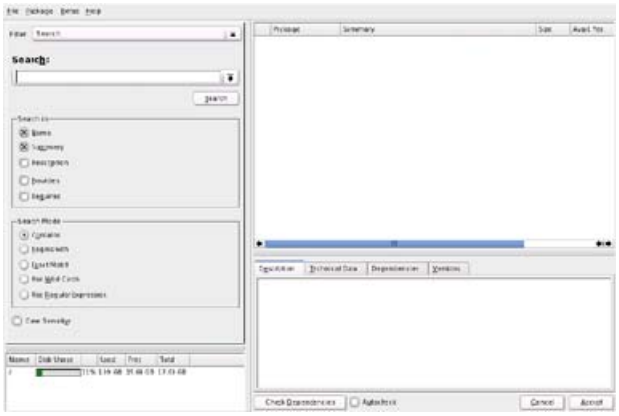
Un cuadro de diálogo del YaST se abre, pidiendo el password de root antes de continuar. Ingresar el password de root.



En la ventana que abre el YaST, dar clic en la casilla "Software" en el menú de la izquierda y luego al ícono "Install and Remove Software" al lado derecho.



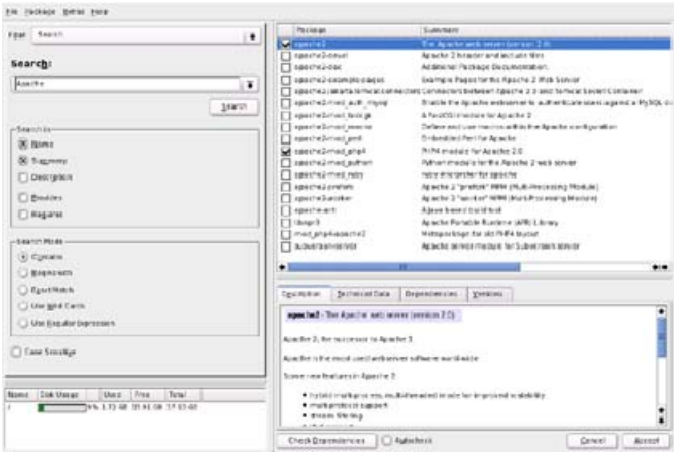
YaST toma un momento inventariando qué hay instalado y qué se puede instalar. Luego abre una ventana con un cuadro de búsqueda.



5.1.1.1 Instalar Apache

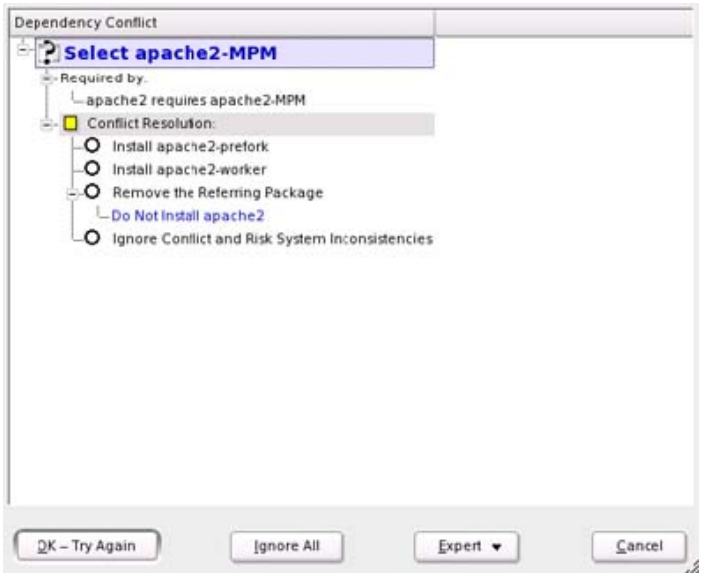
Teclee "apache" en el cuadro de búsqueda y de clic en el botón "Search".

Seleccione el módulo que desee instalar. En este caso instalaremos el Apache Básico así que seleccionaremos "apache2", "apache2-mod_php4".

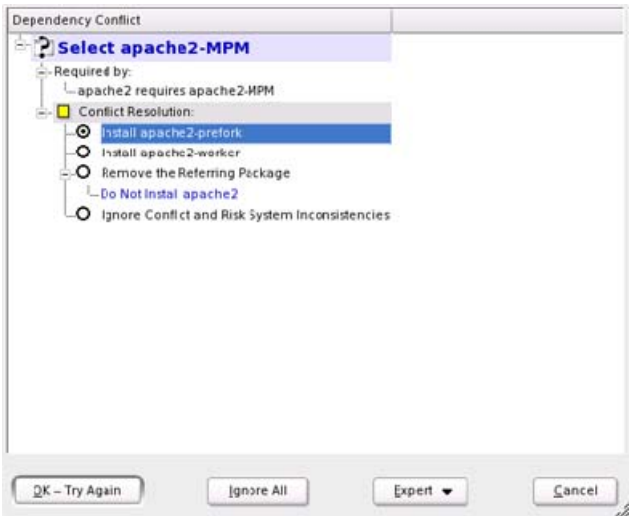


De clic en el botón “Accept".

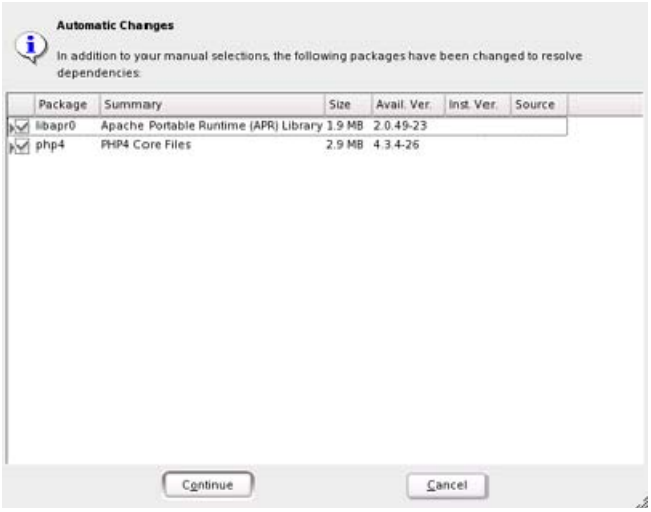
Luego de que el YaST instala lo seleccionado sale un pantallazo informando que ha ocurrido un conflicto y ofrece unos pasos lógicos a seguir para resolverlo.



Seleccionar "install apache2-prefork" y presionar el botón "OK - Try Again".

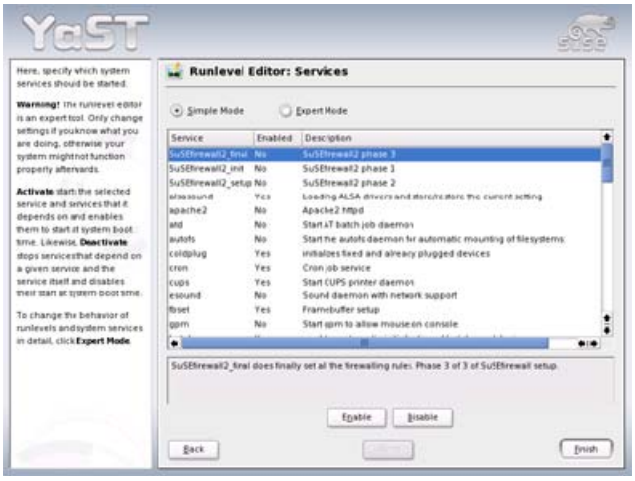


YaST notifica que no ha terminado resolver los conflictos pendientes y que existen otros dos paquetes que deben ser instalados para resolverlos. Así que hay que repetir el paso anterior por cada uno de los paquetes.



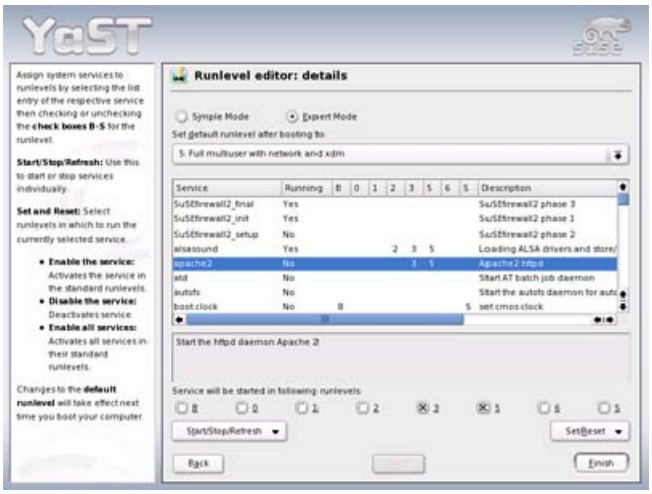
Ahora lo que hay que hacer para que el Servicio de Apache se inicie cada vez que se prenda el servidor es dar clic en el ícono de “System” en el panel izquierdo del YaST.

Dar clic en "Runlevel Editor" en el panel derecho.



Dar clic en el botón de "Expert Mode" en la parte superior de la ventana del YaST.

Seleccionar “apache2” de la lista desplegada.

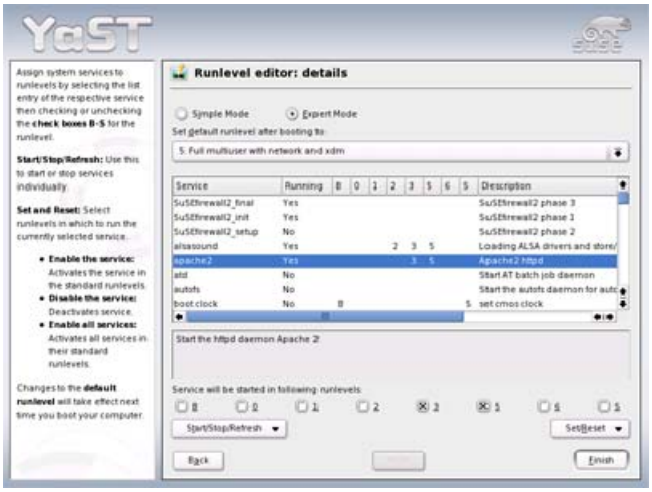


Debajo del botón "Set/Reset" en la parte inferior derecha, elegir "Enable the Service".

Esto debe seleccionar automáticamente los niveles apropiados que deben ser usados cuando arranque el servicio.

Ahora, debajo del botón "Start/Stop/Refresh", seleccionar "Start now ..."

El servicio de Apache debe iniciarse y el editor de los niveles debe lucir como se muestra a continuación:



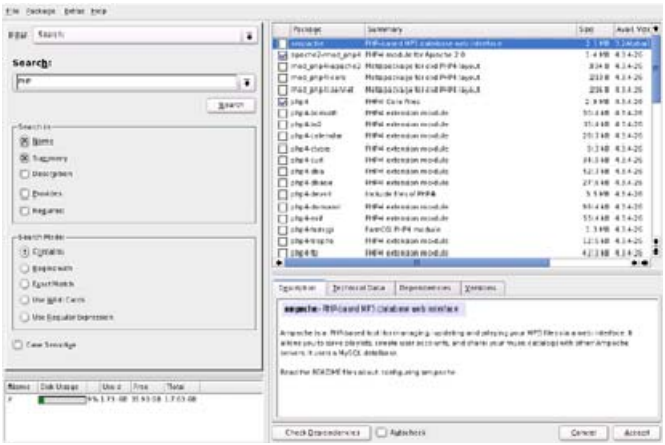
Dar clic en "Finish".

Guardar los cambios hechos.

5.1.1.2 Instalar PHP

Teclee "php" en el cuadro de búsqueda y de clic en el botón "Search".

En este momento el YaST mostrará que ya está instalado el "apache2-mod_php4" y "php4".



Ahora, por medio del Navegador de internet, bajar la última versión estable del phpMyAdmin desde http://www.phpmyadmin.net/home_page/

Salvar el archivo (gzip) en el directorio /srv/www/htdocs.

Dar clic en el Home y buscar el archivo.

Dar clic derecho sobre el archivo y seleccionar “Extract Here”.

Una vez que los archivos hayan sido descomprimidos, cambiar el nombre de la carpeta por “phpMyAdmin” (Probablemente al descomprimirlo el nombre sea algo como phpMyAdmin-2.6.0-pl3).

Siga las instrucciones para modificar el archivo de configuración (config.inc.ini) que se encuentra en la carpeta phpMyAdmin.

Ahora apunte el navegador a http://127.0.0.1/phpMyAdmin/ y ahí tienes.

5.1.1.3 Testear lo hecho hasta ahora

Se puede testear el funcionamiento de Apache y PHP creando un archive “.php” en el directorio del servidor y viendo si el HTTP Apache Server es capaz de subirlo.

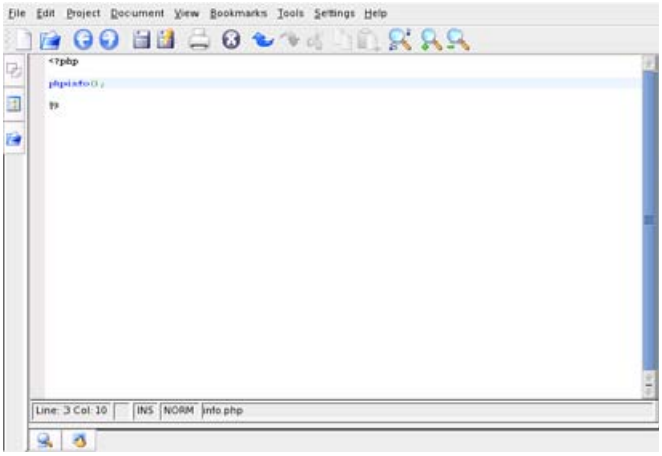
Abrir un editor de texto. "Start" menu => "Utilities" => "Editor".

Copie como contenido lo siguiente:

<?php

phpinfo();

?>

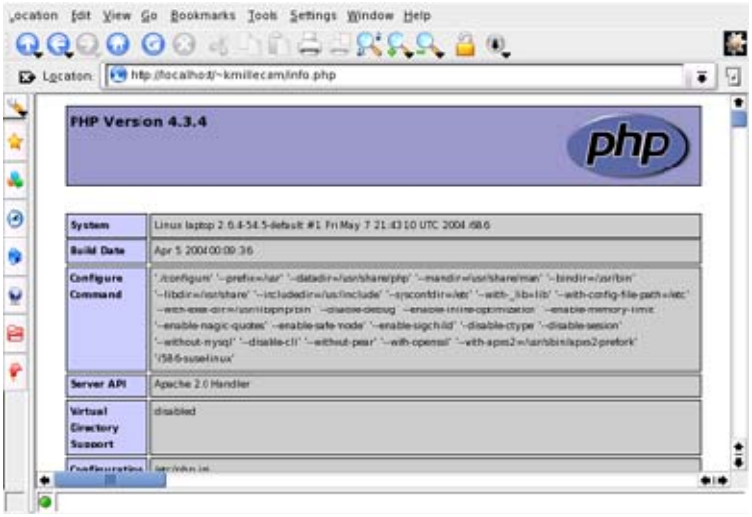


Guarde el archive como “info.php” en el directorio "/home/auteco/public_html/".

Una de las ventajas de Linux y Apache es que cada uno de los usuarios puede tener su propio directorio personal, el cual puede usar para “hostear” lo que desee.

Ahora, por medio del navegador en la dirección URL: http://localhost/~auteco/info.php podremos ver la configuración detallada de la instalación del PHP.

Ya está listo; ambos funcionan correctamente.



5.1.1.4 Instalar MySQL

Teclee " mysql " en el cuadro de búsqueda y de clic en el botón "Search".

Agregue “mysql” y “mysql-client” a la lista de los componentes previamente instalados.

Dar clic en “Accept”.

Inspeccione las adiciones sugeridas

Dar clic en "Continue".

5.1.1.5 Testear MySQL

Como ya se ha instalado el phpMyAdmin, se puede probar la instalación de MySQL abriendo a http://localhost/phpMyAdmin/ por medio del navegador. Hay que tener en cuenta que en Linux, las URLs son sensibles a letras mayúsculas y minúsculas, por tanto hay que tener cuidado en la digitación de phpMyAdmin.

Ahora ya se tiene listo el phpMyAdmin usando Apache para comunicarse con MySQL, y lucirá de la siguiente forma:

5.1.2 Instalación eGroupware

La instalación tiene muchas opciones y, a primera vista, se presenta bastante complicada, pero sólo vamos a elegir las opciones necesarias, y posteriormente se pueden modificar las que queramos desde la administración del programa.

Para poder instalarlo, el servidor debe tener los siguientes requisitos básicos:

- Sistema Operativo: Windows (98/ME, NT/2000/XP/Vista), Linux, Mac OS X (10.3.4 o más).
- El último paquete oficial del eGroupWare. (disponible en http://www.egroupware.org).
- Webserver - Apache, IIS o algún otro que soporte PHP.
- PHP versión 4.3 como mínimo.
- SQL Database Server - MySQL, Postgresql, MaxDB, o Microsoft SQL Server
- Navegador Web - Firefox, Internet Explorer, Safari, Opera, Camino

Para empezar, descargaremos el archivo *.tar.gz de eGroupWare, que pesa aproximadamente 40 MB. A continuación, copiaremos la carpeta descomprimida en la carpeta **srv/www/htdocs** del servidor, y la renombraremos como egroupware.

Una vez realizados estos preparativos, abrimos el programa de instalación en el navegador escribiendo http://localhost/egroupware en la barra de direcciones.

Entonces aparecerá el primer documento, que nos pide que elijamos el idioma. También haremos clic en la opción de Ejecutar pruebas de instalación, y se

comprobará que algunas extensiones de PHP no están instaladas, lo que suele ocurrir con la extensión GD, que tiene que ver con el manejo de imágenes en la programación de la aplicación, que no entorpece la marcha de la instalación.

Existen tres posibles resultados para cada una de la pruebas:

-  si la variable está OK
-  es una advertencia. Alguna información es adicionada. La mayoría de los cambios se refieren al archive 'php.ini'.
-  indica un error, y debe ser corregido.

Lista de Chequeo del eGroupware:

-  Checking required PHP version 4.3+ (recommended 5.0): 5.0.5 ==> True
-  Checking php.ini: safe_mode = Off: ini_get('safe_mode')='' = Off
-  Checking php.ini: magic_quotes_runtime = Off: ini_get('magic_quotes_runtime')='0' = Off
-  Checking php.ini: register_globals = Off: ini_get('register_globals')='0' = Off
-  Checking php.ini: memory_limit >= 16M: ini_get('memory_limit')='8'

  memory_limit (maximum memory available for a script) is set to less than 16MB: some programs require more than the recommended 8MB - be prepared for occasional errors! Please make the following changes in your 'php.ini' file (path/php.ini): memory_limit=16M. And reload your Webserver, in order to implement the changes: e.g. /etc/init.d/httpd reload.

-  Checking php.ini: max_execution_time >= 30: ini_get('max_execution_time')='60'
-  Checking php.ini: file_uploads = On: ini_get('file_uploads')='1' = On
-  Checking php.ini: include_path contain: ini_get('include_path')='. :/Applications/xampp/xamppfiles/lib/php'
-  Checking extension mysql is loaded or loadable: True
-  Checking extension pgsql is loaded or loadable: False

The pgsql extension is needed, if you plan to use a pgSQL database. If you plan to use MySQL as the database, you can ignore this warning

-  Checking extension odbc is loaded or loadable: True
-  Checking extension oci8 is loaded or loadable: False

  The oci extension is needed, if you plan to use an Oracle database.

-  Checking extension mbstring is loaded or loadable: True
-  Checking php.ini: mbstring.func_overload = 7: ini_get('mbstring.func_overload')=''

  You only need to change this parameter in the php.ini file if you will be working with different languages (different character sets!) simultaneously in eGroupWare. Use the 'unicode utf-8' character set to cover all languages, or, e.g. the 'iso-8859-1' for all West-European languages.

-  Checking extension imap is loaded or loadable: False

  The extension imap (php extension) is needed by both email programs (even if you use the POP3-protocol). '"Under Windows, all php extensions must be explicitly loaded in the php.ini file. To do this, remove the semi-colon '";'" from in front of the corresponding line, 'extension=php_imap.dll'. Under Linus, the php extensions are usually individual packages, that is, for 'imap' the package 'php(4|5)_imap' must be installed.'"

-  Checking extension session is loaded or loadable: True
-  Checking PEAR is installed: True
-  Checking PEAR::Log is installed: False

  PEAR::Log is needed by SyncML. '"Under Linux and MacOS?-X, enter the following line, as 'root' in a terminal window: pear install log

-  Checking for GD support...: True
-  Checking file-permissions of . for not world writable: $sysuser/admin drwxr-xr-x

  This might take a while, please wait ...

Please fix the above errors () and warnings () or continue with Header Management

Después, hay que hacer clic en el enlace de “Continuar” para administrar los encabezados. En el siguiente paso llegamos a la instalación propiamente dicha, y aquí escribiremos algunos datos importantes para la configuración, sin olvidarnos de la contraseña para la configuración. Y pulsamos el botón “Continuar” para que se cree el archivo header.inc.php.

El archive header.inc.php es un archivo de texto que contiene fundamentalmente la configuración de las herramientas del eGroupware. Este archivo contiene características y configuraciones que no pueden ser establecidas en la base de datos, por ejemplo, el nombre de la base de datos misma.

Se determinan las siguientes configuraciones:

- Server Root for Linux: */var/www/html/egroupware*. Esta configuración debería ser reconocida automáticamente y no debería cambiarse.
- Include Root (debería ser el mismo Server Root)
- Admin user for header manager por ejemplo *admin*, y, Admin password to header manager . Este usuario admin es necesario si se van a hacer algunos cambios en los encabezados después de la instalación.
- Limit access to setup to the following addresses, networks or hostnames (e.g. 127.0.0.1, 10.1.1, myhost.dnydns.org)
- Persistent connections: Se quieren conexiones persistentes?
- Sessions Type: Qué tipo de sesiones se quieren usar?
    - PHP session management
    - PHP plus restore
    - Database. Guard alas sesiones en BD
- Enable MCrypt.
- MCrypt Version.
- MCrypt initialisation vector.
- Domain select box on login.
- Datebase.

- DB Type.
- DB Host. Hostname/IP del servidor Base de datos.
- DB Port.
- DB Name. Por defecto es "egroupware".
- DB User.
- Configuration User y Configuration Password.
- Abajo de estas opciones hay tres botones:
    - [Create Configuration]: Guarda la configuración en el archivo 'header.inc.php'.
    - [Download]: Aquí se pueden bajar las configuraciones del servidor a n PC local.
    - [View]: Muestra el archive con todas las especificaciones.

A continuación, rellenamos los Datos del Usuario administrador para Instalación/Configuración que hemos escrito en el paso anterior.

A continuación, pulsaremos el botón Instalar todas las aplicaciones, y después ya se habrán completado todos los pasos.

5.1.3 Descarga, compilación e instalación de OpenLDAP

Esta explicación está dada a partir de los fuentes, ya que para utilizar la capa segura deberemos recompilar el OpenLDAP con soporte OpenSSL.

Lo primero que haremos es apuntar a *http://www.openldap.org/software/download/* y descargaremos la última versión que tengamos disponible de uno de sus mirrors. En este caso se trabajará con la versión 1.2.9. A continuación se mostrará el proceso desde las descarga.

5.1.4 Instalación LDAP[1]

La finalidad principal de los siguientes pasos es configurar y utilizar un Servidor de directorio LDAP en una máquina Linux. Se explicará cómo instalar, configurar, ejecutar

[1] LDAP-Linux-Como, Luiz Ernesto Pinheiro Malere, Traducción de J. Iván Juanes Prieto, v1.01, 15 febrero 2000

y mantener el servidor LDAP. Después, se explicará cómo almacenar, recuperar y actualizar información en su Directorio, utilizando las utilidades y clientes LDAP.

5.1.4.1 Instalación del servidor LDAP

Cuatro pasos son necesarios para instalar el servidor: obtener el paquete, descomprimir y desempaquetar el servidor, configurar los ficheros Makefile del programa y compilar el servidor.

5.1.4.1.1 Obtención del paquete

Existen dos servidores LDAP que se distribuyen libremente: el servidor LDAP de la Universidad de Michigan y el servidor *OpenLDAP*. También está el *Netscape Directory Server*, que es libre sólo bajo ciertas condiciones (por ejemplo, las instituciones educativas lo pueden obtener gratis). El servidor *OpenLDAP* está basado en la última versión del servidor de la Universidad de Michigan, y hay disponibles listas de correo y documentación adicional para él. Este documento está basado en el uso del servidor *OpenLDAP*.

Como se había mencionado anteriormente, nos hemos basado en la última versión estable de *OpenLDAP*, utilizando OpenLDAP 1.2.9 sobre un servidor Linux con núcleo.

En la sede Web de *OpenLDAP* encontrará las últimas versiones estables y de desarrollo del servidor OpenLDAP. En el momento de actualizar este documento, la última versión era /openldap-stable-20000129.tgz. La última versión de desarrollo era openldap-1.2.9.tgz (eran las últimas versiones en el momento de traducir, N. del T.)

5.1.4.1.2 Desempaquetado del servidor

Ahora que ya se dispone del archivo .tar.gz en su máquina local, puede desempaquetarlo.

En primer lugar, copie el paquete en un directorio de su conveniencia, por ejemplo `/usr/local`.

Luego utilice la siguiente orden:

```
tar xvzf openldap-stable.tgz
```

También puede usar la siguiente orden:

```
gunzip openldap-stable.tgz | tar xvf –
```

5.1.4.1.3 Configuración del programa

Hay varias opciones que tal vez desee personalizar, de manera que el programa se construya de la forma más adaptada a su sistema.

Para configurar el programa sólo necesita dos pasos:

Edite el fichero `ldapconfig.h.edit`, situado en el subdirectorio `include/` que cuelga del directorio en el que ha desempaquetado el programa.

Ejecute el guión `./configure`.

En el fichero `include/ldapconfig.h.edit` puede configurar opciones como el emplazamiento de los demonios *slapd* y *slurpd*. El fichero propiamente dicho contiene muchos comentarios y sus opciones por defecto también reflejan las opciones que los administradores eligen con más frecuencia, de manera que si tiene prisa puede saltarse este paso y ejecutar directamente:

```
vi include/ldapconfig.h.edit
```

El código fuente de *OpenLDAP* se distribuye con un guión de configuración para ajustar opciones como por ejemplo, el directorio de instalación y las ««banderas»» del compilador y del enlazador. Escriba la orden siguiente en el directorio donde haya desempaquetado el programa:

```
./configure --help
```

Ello mostrará por pantalla todas las opciones que puede personalizar con el guión configure antes de compilar el programa. Algunas opciones útiles para establecer los directorios de instalación son `--prefix=pref`, `--exec-prefix=eprefix` y `--bindir=dir`. Normalmente, si ejecuta `./configure` sin opciones, él mismo autodetectará las opciones adecuadas y se preparará para construir el paquete en la localización común predeterminada. Así pues, teclee:

```
./configure
```

Y observe el resultado por pantalla para verificar que no se produce ningún error.

5.1.4.1.4 Compilación del servidor

Después de configurar el programa puede empezar a compilarlo. Primero construya las dependencias mediante la orden:

```
make depend
```

Después compile el servidor mediante la orden:

```
make
```

Si todo va bien, el servidor se compilará tal y como se haya configurado. En caso contrario, vuelva al paso anterior para revisar las opciones de configuración. Revise las sugerencias específicas para su plataforma, que se hallan en la ruta `doc/install/hints` que cuelga del directorio en el que desempaquetó el software.

Instale luego los ejecutables y las páginas de manual. Es posible que precise de permisos de superusuario para poder llevarlo a cabo (depende del lugar en donde instale los ficheros):

```
su
make install
```

Es todo. Ya dispone del ejecutable del servidor y de los ejecutables de otras varias utilidades.

Si antes de aprender cómo se configura su servidor LDAP desea verificar los ejecutables recién compilados, las últimas versiones del servidor *OpenLDAP* vienen con un guión de verificación. Cuando se actualizó este documento el guión de verificación no era estable al 100% para todos los diagnósticos que llevaba a cabo. De todas maneras pruébelo y ejecútelo, si algún aspecto del guión no funciona bien, siempre puede pararlo pulsando Ctrl+C. En nuestro caso, antes de que el guión o script se detuviera, pudimos observar algunos mensajes que mostraban que los diagnósticos más comunes se habían llevado a cabo con éxito. Para ejecutar el guión de verificación, cámbiese al subdirectorio test/ que cuelga de la ruta en la que desempaquetó el software, y luego teclee:

```
make
```

5.1.4.2 Configuración del servidor LDAP

Cuando el software se haya compilado e instalado, ya puede configurarlo para utilizarlo en su servidor. Toda la configuración en tiempo de ejecución de *slapd* se realiza mediante el fichero slapd.conf, que se instala en el directorio que haya especificado en --prefix en el guión de configuración, o bien, si no especificó ninguno, en `/usr/local/etc/openldap` de forma predeterminada.

En este directorio hallará asimismo los ficheros slapd.oc.conf y slapd.at.conf que se incluyen en el fichero slad.conf y que incluyen, respectivamente, las definiciones de clases de objetos (*objectclasses*) y atributos para la base de datos de segundo plano

de LDAP (*backend*). Lo que sigue es una descripción del formato general del fichero de configuración, y continuaremos con una descripción detallada de cada opción del fichero de configuración.

5.1.4.2.1 Formato del fichero de configuración

El fichero slapd.conf está compuesto por una serie de opciones globales de configuración que afectan a slapd en su conjunto (incluyendo todas las bases de datos de segundo plano o *backends*), seguido por cero o más definiciones de *backends*, las cuales contienen información específica de una instancia de *backend*.

Las opciones globales de configuración pueden anularse en un *backend* determinado (para opciones que aparecen más de una vez, se usa la última aparición en el fichero de configuración slapd.conf). Se ignoran las líneas en blanco y las líneas de comentario que comienzan por el carácter de «#». Si una línea comienza por un espacio en blanco, se considera una continuación de la línea anterior. El formato general del fichero slapd.conf es el siguiente:

```
# comentario - estas opciones se aplican a cualquier base de datos
<opciones de configuración globales>
# definición de la base de datos y opciones de configuración
database <backend tipo 1>
<opciones de configuración específicas del backend tipo 1>
#definición de la segunda base de datos y opciones de configuración
database <backend tipo 2>
<opciones de configuración específicas del backend tipo 2>
# definiciones subsiguientes de bases de datos y opciones de configuración
...
```

Los argumentos de la línea de configuración están separados por espacios en blanco. Si un argumento contiene espacios en blanco, el argumento debe encerrarse entre comillas dobles "de esta manera". Si un argumento contiene unas dobles comillas o una barra invertida `\', el carácter ha de ir precedido de una barra invertida `\', (p. ej. `\\d').

La distribución de *OpenLDAP* contiene un fichero de configuración de ejemplo que se instalará en el directorio de configuración especificado en --prefix. También se

proporcionan un slapd.at.conf, que contiene muchas definiciones de atributos utilizadas a menudo, y slapd.oc.conf, que contiene muchas definiciones de clases usadas con frecuencia.

5.1.4.2.2 Opciones globales

Las opciones que se describen en esta sección se aplican a todos los *backends*, a menos que se sobreescriban o anulen específicamente en la definición de un *backend* concreto. Los argumentos de opción que han de sustituirse por texto de verdad se muestran entre signos de «mayor que» y «menor que» <>.

```
access to <algo> [ by <quién> <niveldeacceso> ]+
```

Esta opción concede acceso (especificado en <niveldeacceso>) a una serie de entradas o atributos (especificados por <algo>) para uno o más peticionarios (especificados en <quién>). Véanse los ejemplos de control de acceso para más detalles.

```
attribute <nombre> [<nombre2>] { bin | ces | cis | tel | dn }
```

Esta opción asocia a una sintaxis con un nombre de atributo. Por defecto se supone que un atributo tiene sintaxis «cis». Se le puede proporcionar a un atributo un nombre alternativo opcional. Las posibles reglas de sintaxis y su significado son éstas:

bin: binario

ces: cadena con mayúsculas y minúsculas exactas (las mayúsculas y minúsculas son significativas durante las comparaciones)

cis: cadena con mayúsculas y minúsculas ignoradas (las mayúsculas y minúsculas no son significativas durante las comparaciones)

tel: cadena de número de teléfono (como «cis», pero durante las comparaciones se ignoran los espacios en blanco y los guiones "-")

dn: "distinguished name" («nombre distintivo»)

```
defaultaccess { none | compare | search | read | write }
```

Esta opción especifica el acceso por defecto que se concederá a los solicitantes que no coincidan con ninguna otra línea de acceso (véanse los ejemplos de control de acceso más abajo). Nótese que un nivel de acceso implica en sí también los niveles de acceso inferiores a él. Por ejemplo, el acceso a escritura implica el acceso a lectura, el acceso a búsqueda y el acceso a comparación.

Valor predeterminado: defaultaccess read

`include <nombrefichero>`

Esta opción ordena a *slapd* que lea información adicional de configuración desde el fichero especificado, antes de continuar con la línea siguiente del fichero actual. El fichero que se especifica ha de seguir el formato normal de configuración de *slapd*. Utilice esta opción para incluir ficheros que contengan las clases de objetos (*objectclass*) y definiciones de atributo (*attribute definitions*) de su base de datos de segundo plano o *backend*. El paquete de software de LDAP viene con los ficheros `slapd.oc.conf y slapd.at.conf.`

Nota: Tenga cuidado al utilizar esta opción. No hay límite mínimo en el número de opciones include anidadas, ni tampoco se hace una detección de anidamiento para el caso de bucles sin fin.

`loglevel <numeroentero>`

Esta opción especifica el nivel de detalle con el que el sistema debe registrar en un archivo de registro (syslog) las informaciones de depuración y las estadísticas de funcionamiento (en este caso se registran mediante el servicio LOCAL4 de syslogd(8). Para que esta característica esté habilitada, es necesario haber compilado *slapd* con la opción de compilación -DLDAP_DEBUG, excepto para los dos niveles de estadísticas, que están siempre habilitados. Los niveles de registro son acumulativos. Para visualizar qué números corresponden a cada tipo de depuración, ejecute slapd con la opción -? o consulte la tabla de más abajo. Los valores posibles para <numeroentero> son:

1 trazado de llamadas a función

2 manejo de paquetes de depurado

4 depurado de trazado intensivo *heavy trace*

8 gestión de conexiones

16 mostrar los paquetes enviados y recibidos

32 procesado del filtro de búsqueda

64 procesado de ficheros de configuración

128 procesado de listas de control de acceso

256 estadísticas de registro de conexiones/operaciones/resultados

512 enviar las entradas de registro de estadísticas

1024 imprimir los *backends* de comunicación con el intérprete de órdenes

2048 imprimir el análisis completo de depuración

Ejemplo: loglevel 255 hará que grandes cantidades de información vayan a un archivo de registro a través de syslog.

Valor predeterminado: loglevel 256

`objectclass <nombre>`

`objectclass <nombre>`

`[ requires <atributos> ]`

`[ allows <atributos> ]`

Esta opción define las reglas de estructura o esquema para la clase de objetos (*objectclass*) especificada. Se usa junto con la opción schemacheck.

`referral <url>`

Esta opción especifica la autoridad en la que basarse cuando *slapd* no pueda hallar una base de datos local para gestionar una petición.

Ejemplo: referral ldap://ldap.itd.umich.edu

Esto remitirá las consultas no locales al servidor LDAP de la Universidad de Michigan. Algunos clientes LDAP con capacidades inteligentes podrán redirigir su consulta a dicho servidor, pero tenga en cuenta que la mayoría de esos clientes no gestionarán URLs sencillas de LDAP que contengan una parte de nombre de máquina y, como opción, una parte de nombre distintivo (dn).

`schemacheck { on | off }`

Esta opción activa (on) o desactiva (off) la verificación de estructura. Si la verificación de estructura está activada, se comprobarán las entradas que se añadan o modifiquen, con el fin de garantizar que obedecen a las reglas de estructura o esquema que implica la clase de objetos (*objectclass*) a la que pertenecen, tal y como las definen las correspondientes opciones de la clase de objetos correspondiente. Si la verificación de estructura está desactivada (off), esta verificación no se realiza.

Valor predeterminado: schemacheck off

`sizelimit <numeroentero>`

Esta opción especifica el número máximo de entradas que hay que devolver de una operación de búsqueda.

Valor predeterminado: sizelimit 500

`srvtab <nombrefichero>`

Esta opción especifica el fichero srvtab en el que *slapd* puede encontrar las claves *kerberos* necesarias para autentificar a los clientes que usen *kerberos*. Esto opción es significativa únicamente si Usted utiliza autentificación por *kerberos*, que ha de activarse en el momento de la compilación incluyendo las definiciones apropiadas en el fichero Make-common.

Valor predeterminado: srvtab /etc/srvtab

`timelimit <numeroentero>`

Esta opción especifica el número máximo de segundos (en tiempo real) que *slapd* pasará contestando una petición de búsqueda. Si pasado ese tiempo no se ha contestado una petición, se devolverá un resultado que indicará *exceeding time*, «tiempo sobrepasado».

Valor predeterminado: timelimit 3600

5.1.4.2.3 Opciones generales del backend

Las opciones de esta sección sólo se aplican al *backend* en el que estén definidas. Estas opciones están soportadas para todos los tipos de *backend*.

`database <tipobasededatos>`

Esta opción marca el comienzo de la definición de una nueva instancia de base de datos. <tipobasededatos> debe ser una de las siguientes: ldbm, shell, o passwd dependiendo del *backend* sobre el que servirá la base de datos.

Ejemplo: database ldbm marca el comienzo de la definición de una nueva instancia de una base de datos con *backend* LDBM.

`lastmod { on | off }`

Esta opción controla si *slapd* mantendrá automáticamente para cada entrada los atributos modifiersName, modifyTimestamp, creatorsName y createTimestamp.

Valor predeterminado: lastmod off

`readonly { on | off }`

Esta opción pone a la base de datos en modo «sólo lectura». Cualquier intento de modificar la base de datos devolverá un error de "unwilling to perform" («no se llevará a cabo la operación»).

Valor predeterminado: readonly off

```
replica host=<nombredemaquina>[:<puerto>]
```

```
replica host=<nombredemaquina>[:<puerto>]
```

```
"binddn=<DN>"
```

```
bindmethod={ simple | kerberos }
```

```
[credentials=<contraseña>]
```

```
[srvtab=<nombrefichero>]
```

Esta opción especifica una dirección para la duplicación o réplica de esta base de datos. El parámetro host= especifica en qué máquina (y opcionalmente, en qué puerto) puede encontrarse la instancia del *slapd* esclavo. Para <nombremáquina> puede usarse lo mismo un nombre que una dirección IP. Si no se proporciona el parámetro <puerto> se usará el puerto estándar de LDAP, el 389.

El parámetro binddn proporciona el DN al que se vinculará el *slapd* esclavo para sus actualizaciones. Ha de tratarse de un DN que tenga acceso de lectura y escritura a la base de datos del esclavo, que normalmente aparece como rootdn en el fichero de configuración del esclavo. También tiene que coincidir con la opción updatedn en el fichero de configuración del *slapd* esclavo. Puesto que los DN son proclives a contener espacios incrustados, la cadena completa "binddn=<DN>" ha de estar encerrada entre comillas.

bindmethod puede ser o bien simple o bien kerberos, dependiendo de si se usa autentificación sencilla basada en contraseñas o bien kerberos cuando se conecte con el *slapd* esclavo. La autentificación sencilla precisa que se proporcione una contraseña válida. La autentificación mediante kerberos precisa de un fichero srvtab válido.

El parámetro credentials=, que sólo se precisa si se usa autentificación sencilla, proporciona la contraseña para binddn en el *slapd* esclavo.

El parámetro srvtab=, que sólo se precisa si se usa autentificación mediante kerberos, especifica el nombre de fichero que aloja la llave kerberos para el *slapd* esclavo. Si se omite, se utiliza el fichero /etc/srvtab.

```
replogfile <nombrefichero>
```

Esta opción especifica el nombre del fichero de registro de duplicación (registro de réplica) en el cual *slapd* registrará los cambios. El registro de duplicación generalmente lo escribe *slapd* y lo lee *slurpd*. Esta opción normalmente sólo tiene efecto si se usa *slurpd* para duplicar la base de datos. Sin embargo, puede utilizarla también para generar un registro de transacciones, si *slurpd* no se está ejecutando. En este caso, necesitará truncar periódicamente el fichero, pues de otra manera crecería indefinidamente.

`rootdn <dn>`

Esta opción identifica al DN de una entrada no sujeta a control de acceso o a restricciones en los permisos de administración para las operaciones en esta base de datos.

Ejemplo: rootdn "cn=Manager, o=U of M, c=US"

`rootkrbname <nombrekerberos>`

Esta opción especifica un nombre kerberos que funcionará en todos los casos para el DN dado anteriormente, con independencia de que exista una entrada con el DN especificado o de que tenga el atributo krbName. Esta opción es útil al crear una base de datos y también cuando se utilice *slurpd* para proporcionar servicios de duplicación (servicios de réplica).

Ejemplo: rootkrbname admin@umich.edu

`rootpw <password>`

Esta opción especifica una contraseña, que funcionará en todos los casos, para el DN dado anteriormente, con independencia de que el DN en cuestión exista o ya tenga contraseña. Esta opción es útil al crear una base de datos y también cuando se utilice *slurpd* para proporcionar servicios de duplicación (servicios de réplica). Evite tener una contraseña de texto sencillo acompañando a esta opción. Proporcione una contraseña cifrada (puede usar una entrada del fichero de Unix /etc/passwd/). *slapd* soporta también otros métodos de cifrado.

Ejemplos: rootpw secret rootpw {crypto}contraseña_cifrada_va_aquí

`suffix <dn sufijo>`

Esta opción especifica el sufijo DN de consultas que se le pasará a la base de datos de *backend*. Pueden proporcionarse múltiples líneas de sufijo, y se requiere al menos una para cada definición de base de datos.

Ejemplo: suffix "o=University of Michigan, c=US"

Las consultas que tengan un DN terminado en "o=University of Michigan, c=US" se le pasarán a este *backend* de base de datos.

Nota: cuando se selecciona el *backend* al que hay que pasarle la consulta, *slapd* examina la línea o líneas de sufijo en cada definición de base de datos en el orden en que aparecen en el fichero. De esta manera, si el sufijo de una base de datos es el prefijo de otra, dicho sufijo debe aparecer después que el prefijo en el fichero de configuración.

`updatedn <dn>`

Esta opción sólo se aplica a un *slapd* esclavo. Especifica el DN al que se le permite hacer cambios en la duplicación. Generalmente se trata del DN al que *slurpd* se vincula cuando hace cambios a la duplicación o réplica.

5.1.4.2.4 Opciones específicas del *backend* LDBM

Las opciones de esta categoría sólo se aplican a la base de datos de backend LDBM. Es decir, tienen que ir después de una línea "database ldbm" y antes de otra línea de "database".

`cachesize <numeroentero>`

Esta opción especifica a instancia de la base de datos de *backend* LDBM el número de entradas en la memoria caché interna que ha de mantener.

Valor predeterminado: cachesize 1000

`dbcachesize <numeroentero>`

Esta opción especifica el tamaño en bytes de la memoria caché interna asociada con cada fichero de índice abierto. En caso de no estar soportada por el método de base de datos subyacente, esta opción se ignora sin mayores avisos. El incremento de este número utilizará más memoria, pero también causará un aumento espectacular del rendimiento, especialmente durante las modificaciones o a la hora de construir los índices.

Valor predeterminado: dbcachesize 100000

`directory <directorio>`

Esta opción especifica el directorio donde residen los ficheros LDBM que contienen la base de datos y sus ficheros asociados.

Valor predeterminado: directory usr/tmp /

```
index {<listadeatributos> | default} [pres,eq,approx,sub,none]
```

Esta opción especifica qué índices hay que mantener para un atributo especificado. Si se proporciona únicamente una <listadeatributos> se mantendrán todos los índices posibles.

Ejemplos: index cn index sn,uid eq,sub,approx index default none

Este ejemplo hará que se dé mantenimiento a todos los índices para el atributo cn: que se mantengan índices de igualdad, subcadenas y cadenas aproximadas en el caso de los atributos sn y uid; y que no se mantengan índices para todos los demás atributos.

```
mode <numeroentero>
```

Esta opción especifica qué permisos de ficheros (modo de protección) debe tener el índice de la base de datos recién creada.

Valor predeterminado: mode 0600

5.1.4.2.5 Access Control Examples

La característica de control de acceso presentada atrás es bastante potente. En esta sección se muestran varios ejemplos de su uso. Primeramente, algunos ejemplos sencillos:

```
access to * by * read
```

Esta directiva de acceso concede acceso de lectura a todo el mundo. Si aparece en solitario tiene el mismo efecto que la siguiente línea de defaultaccess:

```
defaultaccess read
```

El siguiente ejemplo muestra el uso de una expresión regular para seleccionar las entradas por DN en dos directivas de acceso en las que el orden es significativo.

```
access to dn=".*, o=U of M, c=US"

by * search

access to dn=".*, c=US"

by * read
```

El acceso en modo lectura se concede a las entradas que están bajo el sub-árbol "o=University of Michigan, c=US", al que se permite el acceso en modo lectura. Si se hubiera invertido el orden de las directivas de acceso, la directiva específica de la

Universidad de Michigan nunca hubiera coincidido, puesto que todas las entradas de la U. de M. son también entradas de c=US.

El próximo ejemplo vuelve a mostrar la importancia del orden, tanto en lo que se refiere a las directivas de acceso como a las cláusulas "by". También muestra el uso de un selector de atributos para conceder acceso a un atributo específico y a varios selectores <quién> (<who>).

```
access to dn=".*, o=U of M, c=US" attr=homePhone

by self write

by dn=".*, o=U of M, c=US" search

by domain=.*\.umich\.edu read

by * compare

access to dn=".*, o=U of M, c=US"

by self write

by dn=".*, o=U of M, c=US" search

by * none
```

Este ejemplo se aplica a las entradas en el sub-árbol "o=U of M, c=US". La entrada tiene permisos de escritura sobre el atributo homePhone, permisos de búsqueda para otras entradas de la "U of M", permisos de lectura para otros clientes que conecten desde algún lugar del dominio umich.edu, y permisos de comparación para el resto del mundo.

A veces es útil permitir que un DN particular se añada o elimine a sí mismo de un atributo. Por ejemplo, si se desea crear un grupo y permitir que los usuarios lo añadan y eliminen de su atributo member en su propio DN, puede lograrse con una directiva de acceso como la siguiente:

```
access to attr=member,entry

by dnattr=member selfwrite
```

El selector dnattr <quién> nos dice que el acceso se aplica a entradas listadas en el atributo member. El selector selfwrite access especifica que tales miembros sólo pueden añadir o eliminar del atributo su propio DN y no otros valores. El añadido del atributo entry es necesario, ya que se requiere el access de la entrada para acceder a alguno de los atributos de esa entrada.

Observe que la construcción attr=member en la cláusula <qué> es un atajo para la cláusula "dn=* attr=member" (es decir, que coincide con el atributo member en todas las entradas).

5.1.4.3 Ejecución del servidor LDAP

Slapd puede ejecutarse de dos maneras diferentes, como demonio o servicio permanente, o bien desde inetd(8). Se recomienda la ejecución como demonio permanente, sobre todo si usa el *backend* de LDBM. Ello permitirá al backend beneficiarse del uso de memoria de almacenamiento intermedio (*caché*) y evita problemas de acceso compartido a los ficheros de índices de LDBM. Si únicamente ejecuta un *backend* de tipo SHELL o PASSWD, entonces sí puede considerar la opción de ejecutar *slapd* desde inetd.

5.1.4.3.1 Opciones desde la línea de órdenes

Slapd soporta las siguientes opciones de línea de órdenes:

`-d <nivel> | ?`

Esta opción fija el nivel de depuración de *slapd* en <nivel>. Cuando el nivel es un carácter `?', se muestran los distintos niveles de depuración y *slapd* termina, con independencia de cualquier otra opción que se introduzca. Los niveles de depuración existentes, son:

1 trazado de llamadas a función

2 manejo de paquetes de depurado

4 depurado de trazado intensivo *heavy trace*

8 gestión de conexiones

16 mostrar los paquetes enviados y recibidos

32 procesado del filtro de búsqueda

64 procesado de ficheros de configuración

128 procesado de listas de control de acceso

256 estadísticas de registro de conexiones/operaciones/resultados

512 enviar las entradas de registro de estadísticas

1024 imprimir los *backends* de comunicación con el intérprete de órdenes

2048 imprimir el análisis completo de depuración

65535 activar depuración completa

Los niveles de depuración son acumulativos. Si desea trazar llamadas a funciones y observar qué fichero de configuración se está procesando, fije el nivel de depuración al resultado de la suma de estos dos niveles (en este caso, 65). Consulte el fichero `<ldap.h> para más detalles.`

Observe que *slapd* se tendrá que haber compilado con la opción -DLDAP_DEBUG definida, si se desea un nivel de depuración superior a los dos niveles de estadísticas disponibles.

`-f <nombrefichero>`

Esta opción especifica un fichero de configuración alternativo para *slapd*

-i

Esta opción le especifica a *slapd* que se ejecute desde inetd en vez de hacerlo como un demonio o servicio independiente. En la próxima sección encontrará más detalles sobre la ejecución de *slapd* desde inetd.

`-p <puerto>`

Esta opción especifica un puerto TCP alternativo en el que *slapd* se mantendrá a la escucha para las conexiones. El puerto por defecto es el 389.

5.1.4.3.2 Ejecución de *slapd* como demonio o servicio independiente

Como norma general, *slapd* se ejecuta de la siguiente manera:

`# $(ETCDIR)/slapd [<opción>]*`

donde ETCDIR tiene el valor que le haya asignado en el fichero Make-common o en el guión ./configure durante la configuración previa a la compilación, y <opción> es una de las opciones descritas más arriba. A menos que haya especificado un nivel de depuración, *slapd* se desvinculará automáticamente del terminal desde el que lo lanzó, y se ejecutará en segundo plano, en modo demonio o servicio. Cualquiera de las opciones de más arriba pueden darse en la línea de órdenes para hacer que *slapd* cargue un fichero de configuración diferente, o que escuche en otro puerto, etcétera.

Véase el siguiente ejemplo de comienzo de *slapd*:

`$(ETCDIR)/slapd -f /home/malere/mi_slapd.conf -d 255`

5.1.4.3.3 Ejecución de slapd desde inetd

En primer lugar, asegúrese de que sea una buena idea ejecutar *slapd* desde inetd. Si está usando el *backend* LDBM, entonces no es buena idea. Si está en un entorno de mucho servicio, entonces la sobrecarga que supone ejecutarlo desde inetd también lo convierte en una mala idea. Si no es su caso, puede seguir adelante con los dos pasos necesarios.

El primer paso es añadir a /etc/services/ una línea como la siguiente:

```
ldap 389 # ldap directory service
```

El segundo paso es añadir una línea como la siguiente a su fichero /etc/inetd.conf:

```
ldap stream tcp nowait nobody $(ETCDIR)/slapd slapd -i
```

donde ETCDIR tiene el valor que le haya asignado en el fichero Make-common o en el guión ./configure durante la configuración previa a la compilación. Finalmente envíele a inetd una señal -HUP y ya tendrá su configuración. (N. del T.: pruebe con killall -HUP inetd o mejor todavía kill -HUP $(pidof inetd) o kill -TERM `cat $(ETCDIR)/inetd.pid`)

5.1.4.4 Creación y mantenimiento de bases de datos

Esta sección le explica cómo crear una base de datos de *slapd* empezando desde cero. Hay dos maneras de crear una base de datos: la primera, puede crear la base de datos en línea, usando LDAP. Con este método, sólo tiene que ejecutar *slapd* y añadir entradas usando el cliente LDAP de su elección. Este método es adecuado para bases de datos relativamente pequeñas (algunos cientos o miles de entradas, dependiendo de los requerimientos).

El segundo método de creación de bases de datos es no hacerlo en línea, sino mediante herramientas de generación de índices. Este es el mejor método si tiene que crear muchos miles de entradas, que si se introdujeran con el método LDAP llevarían un tiempo intolerablemente largo. También es útil si desea asegurarse de que no se accederá a la base de datos durante su creación.

5.1.4.4.1 Creación de una base de datos en línea

El paquete de software *OpenLDAP* viene con una utilidad llamada ldapadd, que se utiliza para añadir entradas mientras el servidor LDAP se ejecuta. Si decide crear en línea la base de datos, puede utilizar la herramienta ldapadd para añadir las entradas. Tras de añadir las primeras entradas, puede seguir usando posteriormente ldapadd para añadir más entradas. Antes de iniciar *slapd*, asegúrese de que activa las siguientes opciones de configuración en su fichero sladp.conf:

```
suffix <dn>
```

Tal como se explicó anteriormente, esta opción describe qué entradas se mantendrán en esta base de datos. Deberá dar a esta opción el valor del DN de la raíz del sub-árbol que va a crear. Por ejemplo:

```
suffix "o=TUDelft, c=NL"
```

Asegúrese de especificar en qué directorio se crearán los archivos de configuración:

directory <directorio>

Por ejemplo:

```
directory /usr/local/tudelft
```

Tendrá que realizar el siguiente paso para poder conectar con *slapd* como usuario con permisos para añadir entradas. Se lleva a cabo añadiendo las dos siguientes opciones en la definición de la base de datos:

```
rootdn <dn>
```

```
rootpw <contraseña> /* ¡Recuerde usar contraseña «cripto» aquí ! */
```

Estas opciones especifican un DN y una contraseña que pueden usarse para autentificarse como la entrada «superusuario» de una base de datos (es decir, la entrada que tiene permisos para realizar cualquier tarea). El DN y la contraseña especificados aquí funcionarán siempre, con independencia de que la entrada de DN exista realmente o tenga una contraseña válida igual a la especificada. Así se arregla del problema del qué va primero, si el huevo o la gallina, a la hora de autentificarse y añadir entradas antes de que las propias entradas existan siquiera.

Finalmente, debe asegurarse de que la definición de base de datos contiene las definiciones de índices que usted desea:

```
index {<attrlist> | default} [pres,eq,approx,sub,none]
```

Para indexar los atributos de clase de objetos (objectclass) cn, sn y uid se pueden usar por ejemplo las siguientes líneas de configuración:

```
index cn,sn,uid

index objectclass pres,eq

index default none
```

Una vez que haya configurado las cuestiones a su gusto, arranque su cliente LDAP y comience a añadir entradas. Por ejemplo, para añadir la entrada TUDelft seguida de una entrada Postmaster utilizando la herramienta ldapadd, puede crear un fichero llamado /tmp/entradanueva con el contenido:

```
o=TUDelft, c=NL

objectClass=organization

o=TUDelft

description=Technical University of Delft Netherlands

cn=Postmaster, o=TUDelft, c=NL

objectClass=organizationalRole

cn=Postmaster

description= TUDelft postmaster - postmaster@tudelft.nl
```

y luego utilizar una orden como la siguiente para crear de verdad la entrada:

```
ldapadd -f /tmp/entradanueva -D "cn=Manager, o=TUDelft, c=NL" -w
secret
```

Esta orden supone que se ha configurado rootdn como "cn=Manager, o=TUDelft, c=NL" y que la contraseña rootpw es «secret». Si no se desea escribir la contraseña en la línea de órdenes, se utiliza la opción -W de la orden ldapadd, en lugar de -w "contraseña". Será necesario introducir interactivamente la contraseña:

```
ldapadd -f /tmp/entradanueva -D "cn=Manager, o=TUDelft, c=NL" -W
```

Enter LDAP Password :

5.1.4.4.2 Creación de una base de datos sin estar en línea

El segundo método de creación de una base de datos es hacerlo sin estar en línea, usando las herramientas de generación de índices descritas más abajo. Este es el mejor método si tiene que crear muchos miles de entradas, y crearlas con el método

interactivo de LDAP descrito arriba llevaría mucho tiempo. Estas herramientas leen el fichero de configuración de *slapd* y un fichero de entrada LDIF que contiene una representación de las entradas que hay que añadir en formato de texto. Estas herramientas generan los ficheros de índices LDBM directamente. Deberá asegurarse de la activación de varias opciones de configuración importantes, de manera que se encuentren primero en la definición de base de datos del fichero de configuración:

```
suffix <dn>
```

Tal y como se ha descrito en la sección anterior, esta opción especifica qué entradas se mantendrán mediante esta base de datos. Debe Usted ajustarla al DN de la raíz del sub-árbol que intenta crear, por ejemplo:

```
suffix "o=TUDelft, c=NL"
```

Asegúrese de especificar un directorio en el que crear los ficheros de índice:

```
directory <directorio>
```

Por ejemplo:

```
directory /usr/local/tudelft
```

Después querrá seguramente aumentar el tamaño de la memoria de almacenamiento intermedio (*caché*) interna de la aplicación, y que se utiliza para cada fichero de índice abierto. Para un mejor rendimiento durante la creación del fichero de índices, la situación de rendimiento ideal es aquella en la que el fichero completo se carga en memoria. Si el volumen de sus datos no permite esta operación, o bien si no dispone de mucha memoria, todavía podrá darle un valor alto a este parámetro y dejar que haga su trabajo la paginación a disco del sistema. El tamaño se configura con la opción siguiente:

```
dbcachesize <númeroentero>
```

Por ejemplo:

```
dbcachesize 50000000
```

Esto creará una memoria de almacenamiento intermedio (*caché*) de un tamaño de 50 MB, que ya es bastante grande (en la Universidad de Michigan la base de datos tiene cerca de 125.000 entradas, y el mayor fichero de índices ocupa aproximadamente 45 MB). Experimente Usted varias veces con este parámetro y con el grado de paralelismo (ver más abajo), con el fin de ver qué combinación funciona mejor en su sistema. Acuérdese de devolver a este número su valor original después de crear los ficheros de índices y antes de ejecutar *slapd*.

Finalmente tendrá que especificar concretamente qué índices desea construir. Esto se consigue con una o más opciones que se aplican a los índices.

```
index {<listaatributos> | default} [pres,eq,approx,sub,none]
```

Por ejemplo:

```
index cn,sn,uid pres,eq,approx
```

```
index default none
```

Esto creará índices de presencia (*presence*), igualdad (*equality*) y aproximación (*approximate*) de los atributos cn, sn y uid, y no creará ningún índice para ningún otro atributo.

Una vez que haya configurado estos parámetros según sus preferencias, cree los índices ejecutando el programa ldif2ldbm:

```
ldif2ldbm -i <ficheroentrada> -f <ficheroconfigslapd> [-d <niveldepuración>] [-j <númeroentero>] [-n <numerodebasedatos>] [-e <directorioetc>]
```

Los argumentos tienen estos significados:

```
-i <ficheroentrada>
```

Especifica el fichero de entrada LDIF que contiene en formato de texto las entradas por añadir.

```
-f <ficheroconfigslapd>
```

Especifica el fichero de configuración de *slapd* que indica dónde crear los índices, qué índices hay que crear, etc.

```
-d <niveldepuración>
```

Activa la depuración, según se especifique en <niveldepuración>. Los niveles de depuración son los mismos que para *slapd*.

```
-j <númeroentero>
```

Es un argumento opcional que especifica que, como mínimo la cantidad <númeroentero> de procesos han de iniciarse en paralelo en el momento de construir los índices. El valor por defecto es 1. Si se especifica un valor superior a 1, ldif2ldbm creará como mínimo ese número de subprocesos a la hora de construir los índices. Para construir cada índice de atributos se crea un subproceso separado. La ejecución en paralelo de estos procesos puede acelerar grandemente el resultado, pero tenga

cuidado de no crear demasiados procesos que compitan todos a la vez por los recursos de disco y memoria.

```
-n <númerodebasedatos>
```

Es un argumento opcional que especifica la base de datos del fichero de configuración para la cual hay que construir los índices. La primera base de datos aparece listada como «1», la segunda como «2», etc. Se usa por defecto la primera base de datos LDBM que aparezca en el fichero de configuración.

```
-e <etcdir>
```

Es un argumento opcional que especifica el directorio en donde ldif2ldbm podrá hallar las otras herramientas de conversión de bases de datos que precisa para ejecutarse (ldif2index y similares). El valor por defecto es el directorio de instalación que se especificó en el guión de instalación. Véase el siguiente ejemplo del uso de la orden ldif2ldbm:

```
/usr/local/sbin/ldif2ldbm -i nuevas_entradas -f myslapd.conf
```

5.1.4.4.3 Cuestiones adicionales sobre el formato LDIF

El archivo que se va a utilizar para añadir la entrada original será el siguiente:

auteco.ldif:

```
####################################

dn: dc=auteco,dc=local

objectclass: dcObject

objectclass: organization

o: auteco

dc: auteco

dn: cn=Admin,dc=auteco,dc=local

objectclass: organizationalRole

cn: Admin

####################################
```

La forma de agregarlo al directorio es con el siguiente comando, el cual pedirá la clave del “admin.”:

```
$ ldapadd –x –D "cn=Admin,dc=auteco,dc=local" –W –f auteco.ldif
```

Una vez hecho esto, ya se pueden añadir el resto de datos al LDAP. Una vez añadidos los datos, se pueden comprobar de la siguiente manera:

```
$ ldapsearch –x –b 'dc=autecog,dc=local' '(objectclass=*)'
```

Lo cual nos debe devolver una salida en formato LDIF con todos los datos que están en el directorio. Un fichero LDIF corriente tiene este aspecto:

```
dn: o=auteco, c=ES

o: auteco

objectclass: organization

dn: cn=David Rios, o=auteco, c=ES

cn: David Rios

sn: Rios

mail: drios@auteco.local

objectclass: person
```

Como se puede ver, cada entrada está identificada unívocamente por un nombre distintivo (DN, *"distinguished name"*). El DN (nombre distintivo) está compuesto por el nombre de la entrada en cuestión, más la ruta de nombres que permiten rastrear la entrada hacia atrás hasta la parte superior de la jerarquía del directorio.

En LDAP, una clase de objetos define la colección de atributos que pueden usarse para definir una entrada. El estándar LDAP proporciona estos tipos básicos para las clases de objetos:

- Grupos en el directorio, entre ellos listas no ordenadas de objetos individuales o de grupos de objetos.
- Emplazamientos, como por ejemplo el nombre del país y su descripción.
- Organizaciones que están en el directorio.
- Personas que están en el directorio.

Una entrada determinada puede pertenecer a más de una clase de objetos. Por ejemplo, la entrada para personas se define mediante la clase de objetos person, pero también puede definirse mediante atributos en las clases de objetos inetOrgPerson, groupOfNames y organization. La estructura de clases de objetos del servidor (su

esquema) determina la lista total de atributos requeridos y permitidos para una entrada concreta.

Los datos del directorio se representan mediante pares de atributo y su valor. Cualquier pieza de información específica se asocia con un atributo descriptivo.

Por ejemplo, el atributo commonName, o cn («nombre de pila»), se usa para almacenar el nombre de una persona. Puede representarse en el directorio a una persona llamada Juan Ruiz, mediante

```
cn: Juan Ruiz
```

Cada persona que se introduzca en el directorio se define mediante la colección de atributos que hay en la clase de objetos *person*. Otros atributos que se usan para definir esta entrada, serán:

```
givenname: Juan
```

```
surname: Ruiz
```

```
mail: jruiz@auteco.local
```

Los atributos requeridos son aquellos que deben estar presentes en las entradas que utilicen la clase de objetos. Todas las entradas precisan del atributo objectClass, que lista las clases de objeto a las que pertenece una entrada.

Los atributos permitidos son aquellos que pueden estar presentes en las entradas que utilicen la clase de objetos. Por ejemplo, en la clase de objetos person, se requieren los atributos cn y sn. Los atributos description («descripción»), telephoneNumber («número de teléfono»), seeAlso («véase también»), y userpassword («contraseña del usuario») se permiten pero no se requieren.

Cada atributo tiene la definición de sintaxis que le corresponde. La definición de sintaxis describe el tipo de información que proporciona ese atributo:

`bin` binario

`ces` cadena con mayúsculas y minúsculas exactas (las mayúsculas y minúsculas son significativas durante las comparaciones)

`cis` cadena con mayúsculas y minúsculas ignoradas (las mayúsculas y minúsculas no son significativas durante las comparaciones)

`tel` cadena de número de teléfono (como cis, pero durante las comparaciones se ignoran los espacios en blanco y los guiones "-")

`dn` "distinguished name" («nombre distintivo»)

Para conocer en qué lugar de su sistema se remplazan las definiciones de clases de objetos y de atributos, véase el primer párrafo de la sección 5.1.4.2.

Nota: en un fichero LDIF, los espacios finales no se eliminan de los valores, ni tampoco se comprimen múltiples espacios internos. Si no los quiere en sus datos, no debe incluirlos.

5.1.4.4.4 Las utilidades ldapsearch, ldapdelete y ldapmodify

ldapsearch - *ldapsearch* es una interfaz, accesible desde la línea de órdenes, para la llamada a biblioteca *ldap_search(3)*. Use esta utilidad para buscar entradas en el dorsal (*backend*) de nuestra base de datos LDAP.

La sinopsis de las opciones de *ldapsearch* es la siguiente (véase la página de manual de ldapsearch(1) para conocer el significado de cada opción):

```
ldapsearch  [-n]  [-u]  [-v]

[-k]  [-K]  [-t]  [-A]

[-B] [-L] [-R]

[-d niveldepuración] [-F separador]

[-f fichero] [-D dn_deacceso]  [-W]

[-w contraseña_acceso]  [-h servidorldap]

[-p puertoldap]   [-b basebúsqueda]

[-s base|one|sub]

[-a never|always|search|find]

[-l límitetiempo] [-z límitetamaño] filtro

[atributos...]
```

ldapsearch abre una conexión a un servidor LDAP, se «engancha» a él y lleva a cabo una búsqueda utilizando el filtro filtro. Este filtro debe ajustarse a la representación de cadenas de texto para filtros LDAP, tal y como se definen en el Request for Comments 1558 (RFC 1558). Si *ldapsearch* encuentra una o más entradas, se obtienen los atributos especificados en atributos y se imprimen por salida estándar las entradas y sus valores. Si no se listan atributos, se devuelven todos los atributos.

He aquí algunos ejemplos del uso de *ldapsearch*:

```
ldapsearch -b 'o=TUDelft,c=NL' 'objectclass=*'
```

```
ldapsearch -b 'o=TUDelft,c=NL' 'cn=Rene van Leuken'

ldasearch -u -b 'o=TUDelft,c=NL' 'cn=Luiz Malere' sn mail
```

La opción -b representa la base de búsqueda (el punto inicial de la búsqueda) y la opción -u representa la opción «amigable para el usuario», refiriéndose a la información de salida.

ldapdelete - *ldapdelete* es una interfaz, accesible desde la línea de órdenes, para la llamada a biblioteca *ldap_delete(3)*. Use esta utilidad para buscar entradas en el dorsal (*backend*) de nuestra base de datos LDAP.

La sinopsis de las opciones de *ldapdelete* es la siguiente (véase la página de manual de ldapdelete(1) para conocer el significado de cada opción):

```
ldapdelete   [-n]   [-v]  [-k]

[-K]  [-c]  [-d nivel_depuración]

[-f fichero]  [-D dn_de_enganche]  [-W]

[-w contraseña]  [-h servidorldap]

[-p puertoldap] [dn]...
```

ldapdelete abre una conexión con un servidor LDAP, se «engancha» a él, y borra una o más entradas. Si se proporcionan uno o más argumentos de dn, se borrarán las entradas con estos Distinguished Names. Cada dn debe se la representación mediante una cadena de un DN de la forma especificada en el RFC 1779. Si no se proporcionan argumentos de dn, entonces se leerá la lista de DNs desde la entrada estándar (o bien desde un fichero si se utiliza la opción -f).

He aquí algunos ejemplos del uso de *ldapdelete*:

```
ldapdelete 'cn=Luiz Malere,o=TUDelft,c=NL'

ldapdelete -v 'cn=Rene van Leuken,o=TUDelft,c=NL' -D 'cn=Luiz Malere,o=TUDelft,c=NL' -W
```

La opción -v representa el modo verboso. La opción -D representa el DN de enganche (*Binddn*) y la opción -W sirve para que se nos pregunte interactivamente por la contraseña.

ldapmodify - *ldapmodify* es una interfaz, accesible desde la línea de órdenes, para la llamada a biblioteca *ldap_modify(3)* y *ldap_add(3)*. Use esta utilidad para modificar entradas en el dorsal (*backend*) de nuestra base de datos LDAP.

La sinopsis de las opciones de *ldapmodify* es la siguiente (véase la página de manual de ldapmodify(1) para conocer el significado de cada opción):

```
ldapmodify   [-a]  [-b]  [-c]

[-r]  [-n]  [-v]  [-k]

[-d nivelde_depuración]  [-D dn_de_enganche]

[-W]  [-w contraseña]  [-h servidorldap]

[-p puertoldap] [-f fichero]

ldapadd [-b] [-c] [-r] [-n]

[-v]  [-k]  [-K]

[-d nivelde_depuración]  [-D dn_de_enganche]

[-w contraseña]  [-h servidorldap]

[-p puertoldap] [-f fichero]
```

ldapadd está implementado en forma de enlace no simbólico (*hard link*) a la utilidad *ldapmodify*. Cuando se llama a la utilidad en la forma ldapadd, la opción -a (añadir nueva entrada) se activa automáticamente.

ldapmodify abre una conexión con un servidor LDAP, se «engancha» a él, y modifica o añade entradas. La información sobre la entrada que hay que modificar se lee desde la entrada estándar o desde un fichero a través de la opción -f.

He aquí algunos ejemplos del uso de ldapmodify:

Suponiendo que el fichero /tmp/entrymods exista y tenga un contenido:

```
dn: cn=Modify Me, o=University of Michigan, c=US

changetype: modify

replace: mail

mail: modme@terminator.rs.itd.umich.edu

-

add: title

title: Grand Poobah

-

add: jpegPhoto
```

```
jpegPhoto: /tmp/modme.jpeg

-

delete: description

-
```

La orden:

```
ldapmodify -b -r -f /tmp/modif_entrada
```

sustituirá el contenido del atributo mail de la entrada "Modify Me" con el valor "modme@terminator.rs.itd.umich.edu", le añadirá el título de "Grand Poobah", el contenido del fichero */tmp/modme.jpeg* como jpegPhoto, y eliminará completamente el atributo description.

Pueden llevarse a cabo las mismas modificaciones usando el formato de introducción más antiguo de *ldapmodify*:

```
cn=Modify Me, o=University of Michigan, c=US

mail=modme@terminator.rs.itd.umich.edu

+title=Grand Poobah

+jpegPhoto=/tmp/modme.jpeg

-description
```

Esto, más la orden siguiente:

```
ldapmodify -b -r -f /tmp/modif_entrada
```

Eso, suponiendo que el fichero /tmp/newentry exista y tenga el siguiente contenido:

```
dn: cn=Barbara Jensen, o=University of Michigan, c=US

objectClass: person

cn: Barbara Jensen

cn: Babs Jensen

sn: Jensen

title: the world's most famous manager

mail: bjensen@terminator.rs.itd.umich.edu

uid: bjensen
```

La orden:

```
ldapadd -f /tmp/modif_entrada
```

Suponiendo que el fichero /tmp/modif_entrada exista y tenga el contenido:

```
dn: cn=Barbara Jensen, o=University of Michigan, c=US
changetype: delete
```

La orden:

```
ldapmodify -f /tmp/modif_entrada
```

eliminará la entrada de Babs Jensen.

La opción -f representa «fichero» (leer las modificaciones desde un fichero en vez de hacerlo desde la entrada estándar). La opción -b representa a «binario», es decir, cualquier tipo de valores que empiecen por '/' en el fichero de entrada se interpretan como binarios, y la -r representa «reemplazar» (sustituir los valores existentes de forma predeterminada).

5.2 Seguridad del Servidor Web Apache

La seguridad, ya sea en Apache o en cualquier otro servidor, es un tema bastante amplio y muy serio. Así es que a continuación se verá una forma de configurar la seguridad, pero hay muchas otras. Cada uno debe buscar la que más se ajuste a sus necesidades.

5.2.1 Comprobación del entorno

Antes de empezar debemos comprobar que nuestro Apache está compilado con los módulos de seguridad que vamos a utilizar.

Para ver listar los módulos, podemos ejecutar:

```
# cd /usr/local/apache2/bin
# ./httpd -l
Compiled in modules:
core.c
mod_access.c
mod_auth.c
mod_auth_digest.c
mod_include.c
mod_log_config.c
```

```
mod_env.c
mod_setenvif.c
mod_ssl.c
prefork.c
http_core.c
mod_mime.c
mod_status.c
mod_autoindex.c
mod_asis.c
mod_cgi.c
mod_negotiation.c
mod_dir.c
mod_imap.c
mod_actions.c
mod_userdir.c
mod_alias.c
mod_so.c
```

En la lista deben aparecer:

_ mod_auth.c: para autenticación básica.

_ mod_auth_digest.c: para autenticación con el método digest (usuario y password encriptados).

_ mod_ssl.c: para poder activar SSL (https).

mod_auth.c debería aparecer, ya que se trata de una opción por defecto. Si los otros dos módulos no aparecen tendremos que recompilar Apache.

Para compilar Apache con soporte SSL es necesario tener instalado los fuentes de openssl, ya que al compilar se necesitan los ficheros de cabecera (.h). También necesitamos tener instalado openssl para poder trabajar con los certificados.

Para instalar ambos paquetes basta con ejecutar:

apt-get install openssl libssl-dev

Ahora ya estamos listos para recompilar nuestro Apache. Nos situamos en el directorio donde tentemos los fuentes de Apache y usamos los siguientes comandos:

./configure --prefix=/usr/local/apache2 --with-mpm=prefork --enable-so --enable-auth-digest

--enable-ssl

make

make install

5.2.2 Autenticación HTTP

Podemos configurar Apache de forma que, para acceder a cierto directorio (y subdirectorios) sea necesario introducir un usuario y una clave.

Dentro de esta posibilidad vamos a ver dos métodos:

_ Basic: cuando en el cliente se introduce el usuario y la clave, estos viajan al servidor sin cifrar.

_ Digest: el usuario y la clave van cifrados del cliente al servidor.

Estos métodos tienen la ventaja de que es Apache quien se encarga de comprobar las correctas credenciales del cliente, por lo que no tenemos que hacer ningún tipo de comprobación en nuestra aplicación. Hay que tener en cuenta que estos dos métodos sólo sirven para autenticar a un usuario cuando intenta acceder a un determinado recurso. Es decir, Apache identifica si se trata de un usuario válido, y en tal caso le deja acceder al recurso. Pero los datos que posteriormente se envíen de cliente al servidor, o viceversa, no tienen ningún tipo de cifrado. Estos métodos sólo sirven para controlar el acceso, no para proteger los datos una vez se ha comprobado que el acceso es válido.

5.2.3 Autenticación HTTP Basic

Para este modo de autenticación se utiliza el módulo mod_auth.

Este método tiene la ventaja de que lo soportan todos los navegadores, pero tiene la desventaja de que el nombre de usuario y la clave no van cifrados del cliente al servidor. Esto hace que no sea un método recomendado para entornos donde la seguridad es clave.

Configuración de httpd.conf

En el fichero /usr/local/apache2/conf/httpd.conf basta con añadir las siguientes líneas:

<Directory "/usr/local/apache2/htdocs/private">

AuthName "privatefiles"

AuthType Basic

AuthUserFile /usr/local/apache2/conf/passwd_basic

Require valid-user

</Directory>

Veamos cada una de las directivas:

_ Directory: estamos diciendo a Apache que las directivas que vienen a continuación tiene efecto sobre el directorio /usr/local/apache2/htdocs/private, y sus subdirectorios. Es decir, que vamos a proteger este directorio y sus subdirectorios.

_ AuthName: es el nombre del dominio de autenticación. También es el texto que aparecerá en la ventana que pide el el usuario y la clave.

_AuthType: el tipo de autenticación.

_ AuthUserFile: el fichero donde están los usuarios y las claves.

_ Require: con esta directiva indicamos qué usuarios tienen acceso, tenemos varias posibilidades:

_ valid-user: cualquier usuario que esté en el fichero de claves.

_ user <lista de usurios>: lista de usuarios, separados por espacios, que pueden acceder.

_ group <lista de grupos>: lista de grupos, separados por espacios, que pueden acceder. Ojo, porque si usamos esta opción también necesitamos usar la directiva AuthGroupFile para indicar dónde se encuentra el fichero con la definición de los grupos.

5.2.3.1 Creación de un usuario

Para crear los usuarios para el método de autenticación 'Basic', usaremos la aplicación htpasswd. Por ejemplo:

cd /usr/local/apache2/bin

./htpasswd /usr/local/apache2/conf/passwd_basic admin

De esta forma estamos añadiendo el usuario 'admin' al fichero de claves /usr/local/apache2/conf/passwd_basic. El programa nos pedirá la clave y luego nos la vuelve a preguntar para confirmarla. Si el fichero no existe (para la primera vez), es necesario lanzar el comando con la opción -c:

./htpasswd -c /usr/local/apache2/conf/passwd_basic admin

También podemos especificar la clave en la línea de comandos a continuación del nombre. En este caso tendremos que utilizar la opción -b:

./htpasswd -b /usr/local/apache2/conf/passwd_basic admin laclave

Si ejecutamos el comando sin argumentos, obtendremos la ayuda de las posibles opciones.

5.2.3.2 Creación de un grupo

Para especificar los grupos basta con crear un fichero de texto con el siguiente formato para cada línea:

nombreGrupo: user1 user2 user3 ...

Luego, basta con usar la directiva AuthGroupFile para indicar la ruta completa donde se encuentra el fichero que hemos creado con la definición de los grupos.

Cada usuario del grupo lo añadiremos al fichero de claves, tal y como se describió en el apartado anterior.

5.2.4 Autenticación HTTP Digest

Para este modo de autenticación se utiliza el módulo mod_auth_digest.

Este método tiene la gran ventaja de que el usuario y la clave van cifradas del cliente al servidor. Pero tiene el inconveniente de que podemos encontrarnos con versiones antiguas de navegadores que no lo soporten.

5.2.4.1 Configuración de httpd.conf

En el fichero /usr/local/apache2/conf/httpd.conf basta con añadir las siguientes líneas:

<Directory "/usr/local/apache2/htdocs/private">

AuthName "privatefiles"

AuthType Digest

AuthDigestFile /usr/local/apache2/conf/passwd_digest

Require valid-user

</Directory>

Vemos que es prácticamente igual que cuando usábamos el método Basic. Hemos cambiado:

AuthName: esta vez indicamos que el método a usar es 'Digest'.

AuthDigestFile: esta directiva sustituye a AuthUserFile, pero tiene la misma finalidad: especificar donde se encuentra el fichero de claves. Hay que tener presente que el fichero de claves tiene que ser distinto que el que usamos con el método Basic, ya que tiene formatos distintos.

5.2.4.2 Creación de un usuario

Para crear los usuarios para el método de autenticación 'Digest', usaremos la aplicación htdigest. Por ejemplo:

cd /usr/local/apache2/bin

./htdigest /usr/local/apache2/conf/passwd_digest privatefiles admin

De esta forma estamos añadiendo el usuario 'admin' al fichero de claves /usr/local/apache2/conf/passwd_digest. El programa nos pedirá la clave y luego nos la vuelve a preguntar para confirmarla.

Vemos que en la línea de comandos hemos introducido como segundo parámetro 'privatefiles'. Este parámetro debe coincidir exactamente con el nombre del dominio de autenticación que hemos especificado con la directiva AuthName. Es decir, cuando añadimos un usuario, lo hacemos a un dominio de autenticación concreto. De forma que si con la directiva Require especificamos 'valid-user', se consideran sólo usuarios válidos los que pertenecen al dominio de autenticación especificado en AuthName.

Si el fichero no existe (para la primera vez), es necesario lanzar el comando con la opción -c:

./htpasswd -c /usr/local/apache2/conf/passwd_digest privatefiles admin

Si ejecutamos el comando sin argumentos, obtendremos la ayuda de las posibles opciones.

5.2.5 Logout automático

El único inconveniente que tiene la autenticación por HTTP es que no podemos conseguir un logout automático, pasado un determinado periodo de tiempo.

Esto es debido a que una vez nos hemos autenticado, el navegador cachea la credencial de forma que aunque fijemos un periodo de validez, una vez transcurrido este, si el navegador intenta acceder a otro recurso de la zona protegida, el servidor le devolverá un error 401 (Authorization Required), y el navegador automáticamente volverá a mandar las credenciales cacheadas al servidor.

La única forma que tenemos para hacer logout es:

_ Cerrar el navegador.

_ Sobreescribir las credenciales. Para esto podemos crear un identificador de usuario válido, pero sin privilegios, y poner una URL de logout a la que sólo puede acceder este usuario. Esta URL se convierte en un botón de logout.

En cualquiera de estos dos casos hay que instruir y convencer al usuario para que los use.

Si para nuestra aplicación es vital esta funcionalidad, necesitamos manejar la sesión e implementar nuestro propio sistema de login.

5.2.6 SSL

Trabajar con SSL nos permite que todos los datos que se transfieren entre el cliente y el servidor vayan cifrados.

Antes de hablar más sobre SSL, vamos a definir algunos términos:

_ RSA Private Keys: fichero digital que podemos usar para descifras mensajes que nos mandan. Tiene una parte pública (que distribuimos con nuestro certificado), que permite a la gente cifrar los mensajes que nos manda. Este mecanismo de clave asimétrica nos asegura que los mensajes cifrados con la clave pública (que

distribuimos a mucha gente) sólo pueden ser descifrados con la clave privada (que sólo conocemos nosotros).

_ Certificate Signing Request (CSR): es un fichero digital que contiene nuestra clave pública y nuestro nombre.

_ Certification Authority (CA): entidad de confianza encargada de firmar certificados (CSR).

_ Certificate (CRT): Una vez la CA ha firmado el CSR, obtenemos un CRT. Este fichero contiene nuestra clave pública, nuestro nombre, el nombre del CA, y está firmado digitalmente por la CA. De esta forma otras entidades pueden verificar esta firma para comprobar la veracidad del certificado. Es decir, si obtenemos un certificado que está firmado por una CA que consideramos de confianza, podemos confiar también en la autenticidad del certificado.

Ahora que ya tenemos un poco más claros los conceptos, podemos ver que hay varias posibilidades para configurar SSL. Por ejemplo:

_ Cualquier cliente puede conectarse a una URL determinada, usando https. En este caso el servidor enviará su certificado al cliente para que este pueda descifrar la información que le llega del servidor y cifrar la que envía hacia el servidor.

_ También podríamos hacer que sólo los clientes que tengan un determinado certificado puedan conectarse a una determinada URL.

_ Otra posibilidad (la que vamos a ver en este tutorial) es combinar el primer ejemplo con las técnicas de autenticación que hemos visto antes. De forma que cuando intentemos acceder a una determinada URL usando https, tendremos que autenticarnos primero.

5.2.7 Creando nuestra propia CA

Existen varias CAs que, previo pago, pueden firmar nuestro CSR. Estas CAs son mundialmente conocidas, de forma que cualquier cliente podrá conectarse con confianza a nuestro servidor.

Un ejemplo de este tipo de entidades de certificación puede ser VISA o VeriSign.

En nuestro caso, como estamos probando, nos crearemos nuestra propia CA.

Para facilitar este tipo de tareas el paquete openssl nos proporciona el script CA.sh (o CA.pl en Perl). Ejecutaremos:

cd /usr/local/apache2

/usr/lib/ssl/misc/CA.sh -newca

Nos hará las siguientes preguntas:

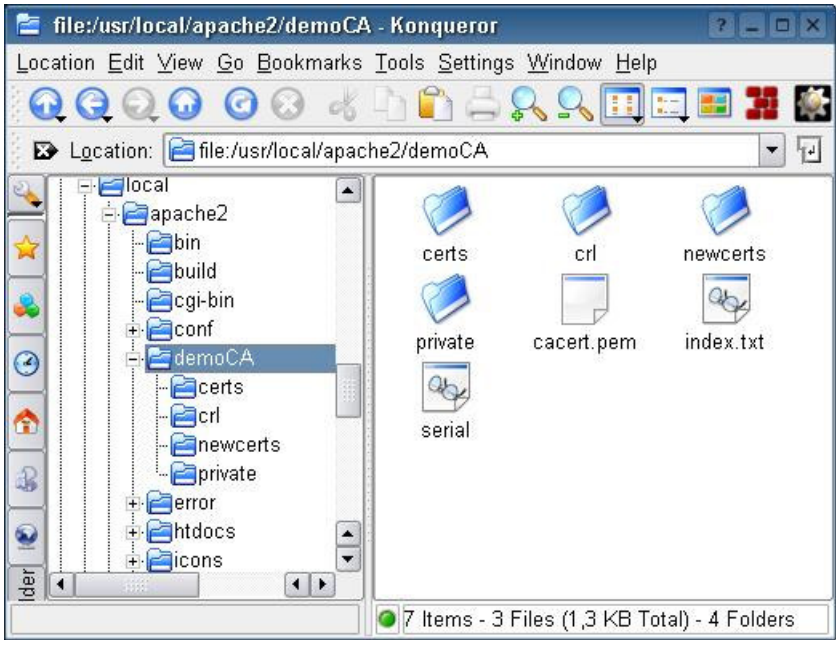
1. Nombre del certificado de la CA o que pulsemos 'enter' para crearlo. En nuestro caso pulsamos 'enter' para crear uno nuevo.

2. Una 'pass phrase', que nos vuelve a preguntar para confirmarla. Esta será la clave para acceder a la clave privada (la clave privada se guarda cifrada). Deberíamos poner más de una palabra.

3. Cierta información para añadir al certificado: código del país, provincia, localidad, nombre de la organización, de la unidad.

Una vez finaliza la ejecución del script, podemos ver que en el directorio /usr/local/apache2 nos ha aparecido un nuevo directorio:

demoCA. Este tiene la siguiente estructura:



El fichero cacer.pem es el certificado de la CA, que luego usaremos para firmar nuestro certificado.

Los subdirectorios están vacíos, a excepción de private, donde encontramos el fichero cakey.pem. Este fichero es la clave de la CA.

5.2.8 Creamos nuestro CSR

Ahora vamos a crear el CSR que debería firmar una auténtica CA (nosotros lo firmaremos con nuestra propia CA, la que hicimos en el apartado anterior). Para esto seguimos los siguientes pasos:

1. Creamos la clave para nuestro servidor Apache (la clave será triple-DES y en formato PEM):

openssl genrsa -des3 -out server.key 1024

El comando nos pedirá un 'pass phrase' y nos la vuelve a preguntar para confirmarla.

2. Creamos el CSR (estará en formato PEM), usando la clave generada en el punto anterior:

openssl req -new -key server.key -out server.csr

Lo primero que hará será pedirnos la 'pass phrase' que le pusimos a la clave en el punto anterior. Luego nos pedirá los datos que queremos añadir a nuestro certificado. Aquí es importante tener en cuenta que cuando nos pregunte por el 'Common Name' debemos poner el nombre completo del dominio del servidor, por ejemplo, si vamos a acceder usando https://www.foo.dom/, tendremos que poner www.foo.dom.

Con esto hemos conseguido generar el fichero server.csr.

5.2.9 Creamos nuestro CRT

El último paso es firmar el CSR para conseguir el CRT. Para esto volveremos a usar el script CA.sh. El problema que tiene este script es que trabaja con unos nombres fijos de ficheros, así que antes de ejecutarlo tenemos que renombrar el fichero server.csr (en mi caso he preferido hacer un enlace simbólico).

ln -s server.csr newreq.pem

CA.sh -signreq

Nos pedirá la 'pass phrase' de la clave que hemos usado para generar el certificado de la CA (ver apartado 5.1. paso 2).

Nos mostrará la información de nuestro certificado, y nos preguntará si queremos firmar el certificado. Por supuesto, contestamos que sí.

Nos preguntará si queremos hacer 'comit' de los certificados firmados, es decir si queremos confirmar la operación. De nuevo, contestamos que sí.

Nos habrá generado el fichero newcert.pem. Este fichero lo renombramos por server.crt.

mv newcert.pem server.crt

5.2.10 Configuramos Apache

_ Lo primero es poner los ficheros donde corresponde.

```
# mkdir conf/ssl.key
# mkdir conf/ssl.crt
# mv server.key conf/ssl.key/
# mv server.crt conf/ssl.crt/
```

_ Ahora añadimos al fichero /usr/local/apache2/conf/httpd.conf:

<Directory "/usr/local/apache2/htdocs/private">

SSLRequireSSL

AuthName "privatefiles"

AuthType Basic

AuthUserFile /usr/local/apache2/conf/passwd_basic

Require valid-user

</Directory>

Vemos que es la misma configuración que usamos en el apartado 4.1, salvo que hemos añadido la directiva SSLRequireSSL.

Esta directiva prohíbe el acceso a no ser que sea HTTP sobre SSL (HTTPS).

Con esto conseguimos que para acceder al directorio /usr/local/apache2/htdocs/private sea obligatoriamente usando HTTPS, y previa autenticación.

_ Por último, sólo nos queda arrancar Apache. Hay que tener en cuenta que para arrancar Apache con soporte SSL hay que utilizar:

/usr/local/apache2/bin/apachectl startssl

Vemos que al arrancar Apache nos pide una 'pass phrase'. Esta es la que usamos al crear nuestro certificado (ver apartado 5.2. punto 7.). Esto lo hace porque la clave está guardada cifrada.

Si queremos evitar tener que escribir la 'pass phrase' cada vez que arranquemos Apache, podemos guardar la clave sin cifrar, pero ojo, esto no es nada recomendable, sobre todo en producción. Podemos tener un grave problema de seguridad.

Para generar una clave sin cifrar podemos hacer.

cd /usr/local/apache2/conf/ssl.key/

openssl rsa -in server.key -out server.unencrypted.key

Luego es conveniente restringir los permisos al máximo.

chmod 400 server.unencrypted.key

Otra manera para no tener que escribir la 'pass phrase' es usar la directiva SSLPassPhraseDialog exec: ... Donde cambiaremos los puntos suspensivos por un programa que saque por la salida estándar la 'pass phrase'.

Hay que tener en cuenta que este método no tiene por qué ser más seguro que el anterior. En general no recomendaría ninguno de los dos para un sistema en producción.

5.2.11 Limpia de archivos

Los ficheros que necesita Apache sólo son:

_ Clave: /usr/local/apache2/conf/ssl.key/server.key.

_ Certificado firmado por la CA: /usr/local/apache2/conf/ssl.crt/server.crt.

Así que si queremos podemos borrar el resto de los archivos que hemos generado para firmar nuestro certificado.

cd /usr/local/apache2/

rm server.csr newreq.pem

rm -Rf demoCA/

5.3 Instalación de las páginas Web

Las páginas de toda la intranet han sido desarrolladas por medio de una herramienta especializada para el desarrollo Web y un editor de texto.

La forma de instalar el contenido de la Intranet AUTECO, es haciendo una copia de las carpetas que contienen las páginas en la ruta: **srv/www/htdocs/** en el servidor.