

Instalación, administración y gestión de LDAP

**Antonio Luis Román-Naranjo Varela
Administración de sistemas informáticos
Gonzalo Nazareno, a 21 de junio del 2009**

Índice

1. Introducción del proyecto integrado	-	pág. 3
1.1. ¿Qué vamos a hacer?		
1.2. ¿Qué vamos a usar?		
1.3. Finalidad del proyecto.		
2. KVM	-	pág. 4
2.1. Introducción y características.		
2.2. Instalación.		
2.3. Optimización del esquema de red.		
3. GIT	-	pág. 6
3.1. Introducción.		
3.2. Creación y uso de repositorios.		
4. LDAP	-	pág. 9
4.1. Introducción.		
4.2. Instalación y configuración.		
4.3. Explicación de la estructura usada.		
4.4. ACL		
5. PHP	-	pág. 15
5.1. Introducción.		
5.2. Visión general.		
5.3. Introducción a las librerías usadas.		
6. Moodle	-	pág. 17
6.1. Introducción y características del CMS.		
6.2. Configurar Moodle con LDAP.		
7. MediaWiki	-	pág. 22
7.1. Introducción y características del CMS		
7.2. Configurar MediaWiki con LDAP.		
8. Aplicación de gestión de usuarios	-	pág. 25
8.1. ¿Qué hace la aplicación?		
8.2. Manual de usuario		
8.2.1. Pantalla principal.		
8.2.2. Pantalla principal del profesor.		
8.2.3. Pantalla principal del alumno.		
8.2.4. Añadir usuario.		
8.2.5. Listado completo.		
8.2.6. Exportar estructura.		
8.2.7. Importar		
8.2.8. Tu perfil		
8.3. Documentación del programa.		
8.3.1. Funcionamiento de las clases utilizadas.		
8.3.1.1. LDAP Class		
8.3.1.2. LDAP2LDIF		
8.3.1.3. Upload		
8.3.2. Explicación de cada módulo. (en el código)		

9. Conclusión del proyecto

1.- Introducción al proyecto integrado

1.1.- ¿Qué vamos a hacer?

El objetivo de este proyecto es el de realizar autenticaciones y almacenamiento de información por medio de un sistema centralizado de directorios, en nuestro caso *OpenLDAP*, y hacer que esta información sea útil para distintos CMS. Además, se elaborará una aplicación PHP con la que administrar el servidor, así como realizar exportaciones de archivos LDIF e importaciones a partir de un esquema especificado.

1.2.- ¿Qué vamos a usar?

Para esta tarea nos ayudaremos de máquinas virtuales. En lugar de usar *Xen*, como se dijo en el anteproyecto, usaremos KVM, un sistema de virtualización más limpio (no requiere modificaciones del kernel), de momento, está dando unos resultados muy buenos.

También usaremos un sistema de control de versiones llamado GIT. A pesar de lo relativamente nuevo que es y la dificultad que aparenta en un principio, es muy fácil de usar, además de que guarda un log completo de los cambios con comentarios, las diferencias entre la versión actual y las anteriores, etc.

El sistema operativo ha sido una distribución Linux, ya que desde un principio me planteé el proyecto integrado usando únicamente tanto herramientas como un SO libre. Empecé en *Ubuntu Intrepid*, pero al actualizar a *Jaunty* empezaron a surgir bastantes problemas de dependencias, así que me pasé a *Debian Lenny* y, un tiempo después, a *Debian Squeeze*. Sobra decir que en el servidor (máquina virtual con KVM) siempre he tenido *Debian Lenny*.

En cuanto al IDE me decidí por *Eclipse* con el plugin de PHP, después de pasar por *Netbeans* (que funcionaba bien, pero dejaba unos espacios demasiado grandes entre las líneas del código, y parecía que eso no se podía cambiar...) y *Bluefish* (bastante limitado).

1.3.- Finalidad del proyecto.

La finalidad del proyecto es centralizar la autenticación de diferentes servicios o aplicaciones utilizando un servicio de directorios como es LDAP.

2.- KVM

- Problemas con la virtualización durante el proyecto.

Durante el transcurso de este proyecto se ha pasado por *VirtualBox*, *VMWare* antes que *KVM*. En un principio *VirtualBox* mostró unos resultados decentes, hasta que empezó a fallar lo suficiente como para decidir dejarlo. Aprovechando que el formato que usa VB para sus máquinas virtuales era fácil de convertir al formato de *VMWare*, me pase a este programa de virtualización. Al ver que el funcionamiento no era el adecuado (lentitud, fallos al arrancar) tuve que cambiarlo.

Después de convertir el HD de *VirtualBox* y después de *VMWare* a IMG (el formato que necesitaba) conseguí arrancar la máquina virtual en *KVM*.

2.1.- Introducción y características.

Kernel-based Virtual Machine , o KVM, es una solución para implementar [virtualización completa](#) con [Linux](#) sobre hardware [x86](#). Está formada por un módulo del núcleo (con el nombre `kvm.ko`) y herramientas en el espacio de usuario, siendo en su totalidad [software libre](#). El componente KVM para el núcleo está incluido en Linux desde la versión 2.6.20.

KVM permite ejecutar [máquinas virtuales](#) utilizando imágenes de disco que contienen sistemas operativos sin modificar. Cada máquina virtual tiene su propio [hardware virtualizado](#): una [tarjeta de red](#), [discos duros](#), [tarjeta gráfica](#), etc.

KVM fue creado, y está mantenido, por [Qumranet](#).

(Fuente: http://es.wikipedia.org/wiki/Kernel-based_Virtual_Machine)

Existe un estudio comparativo entre distintos sistemas de virtualización. Lo puedes encontrar [tras este hipervínculo](#).

Requisitos para su uso:

- Procesador x86 con soporte para *Virtualization Technology*
- SO Linux

Puede ejecutar huéspedes Linux (32 y 64 bits) y Windows (32 bits)

2.2.- Instalación.

El proceso de instalación de KVM es muy sencillo. En unos pocos pasos se puede tener lista una máquina virtual sin mayores complicaciones. Explico el proceso:

1. Instalamos el paquete de KVM

```
# aptitude install kvm
```

2. Creamos **un archivo IMG de 4GB en bloques de uno**. Lo usaremos para instalar nuestra máquina virtual.

```
$ dd if=/dev/zero of=maquina-virtual.img bs=1G count=4
```

3. **Arrancamos la máquina virtual diciéndole que arranque la ISO especificada en al instrucción.** (La opción “-m” especifica cuanta memoria dedicaremos a la máquina virtual).

```
# kvm -hda maquina-virtual.img -cdrom  
/home/ark/Desktop/imagen.iso -m 512 -boot d
```

4. Una vez terminada la instalación, **ejecutaremos la máquina virtual con la siguiente instrucción:**

```
#kvm -hda disk1.img -m 512 -boot c
```

2.3.- Optimización del esquema de red.

Aunque con los pasos anteriormente dados la máquina virtual sería totalmente funcional, KVM ponía a la máquina por defecto como NAT, cosa que no me servía para lo que pretendía hacer... Por esta razón, procedo a explicar como hacer que KVM funcione en modo *bridge*.

- Ponemos lo siguiente en el archivo *interfaces*:

```
auto br0  
iface br0 inet static  
address 192.168.1.2  
netmask 255.255.255.0  
gateway 192.168.1.1  
bridge_ports eth0  
bridge_fd 9  
bridge_hello 2  
bridge_maxage 12  
bridge_stp off
```

En *bridge_ports* ponemos la interfaz que usemos para conectarnos a la red y en *address* la misma IP que la puesta en *bridge_ports* (en mi caso, eth0).

- Modificamos el archivo */etc/kvm/kvm-ifup* y lo dejamos con este contenido:

```
#!/bin/sh  
ifconfig $1 0.0.0.0 promisc up  
brctl addif br0 $1  
exit 0
```

- Levantamos la máquina virtual con el siguiente comando

```
# kvm -hda VM/debian-server.img -net nic -net tap -m 512
```

Con estos 3 pasos ya tenemos lista la máquina virtual para funcionar en modo *bridge*.

3.- GIT

3.1.- Introducción.

GIT es un software de sistema de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de código fuente. Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir el interfaz de usuario o front end como Cogito o StGIT. Sin embargo, GIT se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan GIT, en particular, el grupo de programación de núcleo del sistema operativo Linux.

(Fuente: <http://es.wikipedia.org/wiki/Git>)

3.2.- Creación y uso del repositorio.

Lo primero y fundamental es instalar el paquete *git-core*, que es el que tiene todo lo necesario para la creación y gestión de los repositorios. Este paquete debe ser instalado tanto en el servidor como en los clientes.

```
# aptitude install git-core
```

Ahora, explicaré los pasos para poner GIT a punto, distinguiendo cliente de servidor.

- Servidor

- Creamos una carpeta en el lugar que queramos con el formato *carpeta.git*. En mi caso, he creado la carpeta en */var/git/ldap-php5.git*

```
# mkdir /var/git/ldap-php5.git
```

- Movemos dentro de esa carpeta los archivos de nuestro proyecto.
- Ahora crearemos dentro de esa carpeta un repositorio vacío. Nos colocamos dentro de la carpeta *carpeta.git* que habíamos creado y ejecutaremos la siguiente línea:

```
$ git init
```

Una vez hecho esto nos aparecerá el siguiente mensaje:

```
Initialized empty Git repository in ruta_git_init
```

- Añadimos los archivos.

```
$ git add nombearchivo/s
```

Con esto añadimos los archivos que queramos a la cola de nuestro repositorio. Yo, normalmente, suelo usar *git add .*, para añadir todos los archivos que se hayan cambiado.

- Confirmamos los archivos añadidos. Esto se hace con la instrucción *commit*.


```
$ git commit -a -m "Aquí ponemos un comentario"
```

Con esta línea hacemos que los cambios añadidos con `git add` se den por buenos, por decirlo de alguna forma.

- Creamos un usuario llamado `git`. Todos los cambios que hagamos los haremos con este usuario, para que no haya problema de permisos. Toda la gente que colabore en el proyecto debe saber la contraseña del usuario `git`.

```
# adduser git
```

- Damos al usuario `git` permisos sobre el repositorio

```
# chown -R git:git /var/git/ldap-php5.git
```

Con esto ya tendremos listo el lado del servidor. También existe una página donde se te permite tener un repositorio de manera gratuita, [GitHub](#), pero he optado por tenerlo todo dentro de una máquina virtual. Ahora la parte del cliente.

▪ Cliente

- Clonamos el repositorio del servidor.

```
$ git clone git@192.168.1.100:/var/git/ldap-php5.git
ldap-php5
```

En mi caso, la instrucción queda así. `git clone` es el comando, el `git` antes de la arroba es con el usuario que nos conectamos para clonar el repositorio (esto es muy importante, los clone siempre con el usuario `git`), y lo de después de la arroba es la IP del servidor y la ruta a la carpeta donde creamos el repositorio. Con esto ya tendremos una copia exacta de todo lo que añadimos en el repositorio del servidor.

Así ya tenemos listo el repositorio local para poder hacer cambios y subirlos al repositorio remoto. El procedimiento de uso de Git es el siguiente:

1. Hacemos un **`git pull`**. Este comando actualizará nuestro repositorio local con los nuevos cambios que hayan ocurrido.
2. Editamos cualquier archivo dentro del repositorio local y guardamos los cambios.
3. Hacemos un **`git status`**. Esto nos mostrará los cambios realizados hasta ahora.
4. Ejecutamos el comando **`git add`** . para añadir los cambios a la cola del repositorio local. Ahora, si hacemos un `git status`, veremos que los cambios están añadidos y listos para hacer un `commit`, o lo que es lo mismo, confirmar los cambios añadidos con `git add` .
5. Confirmamos los cambios. Para ellos usaremos la línea **`git commit -a -m "comentario"`**.
6. Ahora solo falta subir los cambios al repositorio remoto. Esto se hace con el comando `git push`.

Con esto ya sabemos la rutina a realizar cuando se hagan cambios, pero antes de terminar, un par de anotaciones más...

- **Antes de hacer cualquier cambio en nuestro repositorio local** hay que actualizar los datos con los del repositorio remoto usando el comando *git pull*.
- Git guarda un *log* con todos los cambios hechos en el repositorio. Para ver estos cambios, podemos usar dos comandos distintos:
 - `git show`: muestra las diferencias entre las últimas dos versiones (la actual y la anterior).
 - `git log -p`: el resultado es el mismo que el de `git show`, pero en lugar de mostrar los dos últimos, muestra el historial completo.

4.- LDAP

4.1.- Introducción.

LDAP (Protocolo Ligero de Acceso a Directorios) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas.

Un directorio es un conjunto de objetos con atributos organizados en una manera lógica y jerárquica. El ejemplo más común es el directorio telefónico, que consiste en una serie de nombres (personas u organizaciones) que están ordenados alfabéticamente, con cada nombre teniendo una dirección y un número de teléfono adjuntos.

Un árbol de directorio LDAP a veces refleja varios límites políticos, geográficos y/o organizacionales, dependiendo del modelo elegido. Los despliegues actuales de LDAP tienden a usar nombres de Sistema de Nombres de Dominio (DNS por sus siglas en inglés) para estructurar los niveles más altos de la jerarquía. Conforme se desciende en el directorio pueden aparecer entradas que representan personas, unidades organizacionales, impresoras, documentos, grupos de personas o cualquier cosa que representa una entrada dada en el árbol (o múltiples entradas).

(Fuente: <http://es.wikipedia.org/wiki/LDAP>)

Concretamente usaremos la implementación OpenLDAP. Se trata de una implementación libre del protocolo que soporta múltiples esquemas por lo que puede utilizarse para conectarse a cualquier otro LDAP.

Tiene su propia licencia, la OpenLDAP Public License. Al ser un protocolo independiente de la plataforma, varias distribuciones GNU/Linux y BSD lo incluyen, al igual que AIX, HP-UX, Mac OS X, Solaris, Windows (2000/XP) y z/OS.

OpenLDAP tiene cuatro componentes principales:

- slapd: demonio LDAP autónomo.
- Slurpd: demonio de replicación de actualizaciones LDAP autónomo.
- Rutinas de biblioteca del protocolo LDAP.
- Utilidades, herramientas y clientes.

(Fuente: <http://es.wikipedia.org/wiki/LDAP#OpenLDAP>)

4.2.- Instalación y configuración.

1. **aptitude install slapd**

Después de esto, nos pedirá la contraseña para el administrador del directorio LDAP.

2. **dpkg-reconfigure slapd.**

Ya que la información pedida durante la instalación es poca y modificar los ficheros a mano puede resultar una tarea un poco incómoda, usamos el comando puesto arriba para que nos pida más información.

Los datos a introducir son estos:

- **Nombre de dominio:** ldap.gonzalonazareno.org

Esto acabará reflejado de la siguiente forma en el archivo de configuración:

```
dc=ldap,dc=gonzalonazareno,dc=org
```

- **Nombre de la organización:** Proyecto Integrado
- **Contraseña del administrador:** igual que la anterior.
- **Modo de base de datos a utilizar:** BDB
- **Permitir el protocolo LDAPv2:** No

Si pide algo más, dejad la respuesta por defecto.

3. Creando la estructura del directorio.

Entramos en /etc/ldap/ y creamos el fichero base.ldif (el nombre del fichero puede ser cambiado) con el siguiente contenido:

```
dn: ou=People,dc=ldap,dc=gonzalonazareno,dc=org
ou: People
objectClass: top
objectClass: organizationalUnit

dn: ou=Group,dc=ldap,dc=gonzalonazareno,dc=org
ou: Group
objectClass: top
objectClass: organizationalUnit

dn: cn=profesores,ou=Group,dc=ldap,dc=gonzalonazareno,dc=org
objectClass: posixGroup
objectClass: top
cn: profesores
gidNumber: 2000

dn: cn=alumnos,ou=Group,dc=ldap,dc=gonzalonazareno,dc=org
objectClass: posixGroup
objectClass: top
cn: alumnos
gidNumber: 2001
```

Una vez hecho esto, paramos el servicio de slapd (/etc/init.d/slapd stop) y procedemos a aplicar el archivo base.ldif sobre el sistema LDAP. Para ello usamos el comando slapadd -l base.ldif.

4. Continuamos configurando.

Editamos el fichero /etc/ldap/ldap.conf y nos aseguramos de que están estas dos líneas sin comentar y de la siguiente forma.

```
BASE dc=ldap,dc=gonzalonazareno,dc=org
URI ldap://127.0.0.1
```

5. Terminando la configuración del servidor.

Instalamos el paquete `libpam-ldap`, que nos servirá para poder hacer consultas del directorio. Se nos pedirá que rellenemos lo siguiente:

- **Make local root Database admin:** Si
- **Does the LDAP database require login?** : No
- **Cuenta del administrador:** `cn=admin, dc=ldap, dc=gonzalonazareno, dc=org`
- **Contraseña del administrador:** la que quieras (pero igual a las anteriores).

Ahora hacemos un `dpkg-reconfigure libpam-ldap` para que nos pida más información:

- **Identificador del servidor LDAP:** `ldap://127.0.0.1/`
- **Nombre distinguido (dn):** `dc=ldap, dc=gonzalonazareno, dc=org`
- **Versión de LDAP:** 3
- **Local crypt to use when changing password:** `exop`

Con estos pasos tendremos el servidor listo para este proyecto.

4.3.- Explicación de la estructura usada.

Aunque con un simple vistazo al archivo `base.ldif` se puede deducir la estructura, se explicará por partes a continuación.



- 1er párrafo: crea la rama *People*. Dentro de ella se colocarán a los usuarios que se vayan dando de alta dentro del servidor LDAP.

```
dn: ou=People,dc=ldap,dc=gonzalonazareno,dc=org
ou: People
objectClass: top
objectClass: organizationalUnit
```

- 2º párrafo: crea la rama *Group*. Dentro de ella se colocarán los distintos grupos para los usuarios. Es el método que usará la aplicación en PHP para la asignación de funcionalidad. Los grupos son definidos en los dos últimos párrafos.

```
dn: ou=Group,dc=ldap,dc=gonzalonazareno,dc=org
ou: Group
objectClass: top
objectClass: organizationalUnit
```

- 3er párrafo: crea el grupo *profesores* dentro de la rama *Group*. Los pertenecientes a este grupo son los que tendrán acceso total a la aplicación PHP. Este grupo estará identificado por el *gid* 2000 (el *gid* es algo así como el número identificador de los grupos).

```
dn: cn=profesores,ou=Group,dc=ldap,dc=gonzalonazareno,dc=org
objectClass: posixGroup
objectClass: top
cn: profesores
gidNumber: 2000
```

- 4º párrafo: crea el grupo *alumnos* dentro de la rama *Group*. Los pertenecientes a este grupo tiene una funcionalidad muy limitada en la aplicación PHP. Este grupo está identificado por el *gid* 2001.

```
dn: cn=alumnos,ou=Group,dc=ldap,dc=gonzalonazareno,dc=org
objectClass: posixGroup
objectClass: top
cn: alumnos
gidNumber: 2001
```

4.4.- ACL

Una **Lista de Control de Acceso** o **ACL** (del [inglés](#), *Access Control List*) es un concepto de [seguridad informática](#) usado para fomentar la [separación de privilegios](#). Es una forma de determinar los [permisos de acceso](#) apropiados a un determinado objeto, dependiendo de ciertos aspectos del [proceso](#) que hace el pedido.

(Fuente: <http://es.wikipedia.org/wiki/ACL>)

Gracias a las ACL, podremos indicar restricciones desde el mismo LDAP, en lugar de realizar dichas restricciones a nivel de código, aumentando así la seguridad. El archivo usado para especificar las restricciones es el archivo `slapd.conf`. Se puede tanto modificar las entradas existentes como añadir entradas nuevas. En mi caso, he tenido que modificar dos entradas:

- `access to attrs=userPassword,shadowLastChange`

Este campo indica sobre que atributos se tendrán permisos y que tipo de permisos se tendrán. Dentro de ese campo se indican los usuarios y el tipo de permisos que tendrán. Si queremos ampliarlo y añadir más atributos, tenemos que hacerlo escribiéndolo después del signo de igual (yo lo tuve que hacer con el “cn”). Después de realizar los cambios, el campo quedó de la siguiente forma:

```
access to attrs=userPassword,shadowLastChange,cn
by dn="cn=admin,dc=ldap,dc=gonzalonazareno,dc=org" write
by dn="uid=paco,ou=People,dc=ldap,dc=gonzalonazareno,dc=org" write
by anonymous auth
by self write
by * none
```

Explicaré línea por línea la función de cada una de ellas:

- o `by dn="cn=admin,dc=ldap,dc=gonzalonazareno,dc=org" write`

Hacemos que el usuario “cn=admin” tenga permisos de lectura y escritura sobre los atributos dados en la línea “access to ...” de todos los usuarios.

- o `by dn="uid=paco,ou=People,dc=ldap,dc=gonzalonazareno,dc=org" write`

Hacemos que el usuario “Paco” tenga permisos de lectura y escritura sobre los atributos dados en la línea “access to ...” de todos los usuarios. Al igual que hice con “Paco”, tenemos que añadir a todos y cada uno de los profesores dentro de esta sección, siguiendo el mismo esquema.

- o `by anonymous auth`

Permite que los usuarios anónimos tengan capacidad de lectura de esos datos.

- o `by self write`

Nos da la capacidad de poder leer y escribir nuestros propios datos.

- o `by * none`

A los no nombrados arriba no les permitimos nada.

▪ `access to *`

En este campo si incluirán que usuarios tendrán privilegios de administración. Es necesario incluir aquí a los usuarios a los que se les quiera dar privilegios para añadir usuarios y realizar importaciones.

Este campo es un complemento del anterior. No vale de nada poner un usuario en este campo sin ponerlo en el de arriba.

Tras las modificaciones necesarias, el campo quedó de la siguiente manera:

```
access to *
by dn="cn=admin,dc=ldap,dc=gonzalonazareno,dc=org" write
by dn="uid=paco,ou=People,dc=ldap,dc=gonzalonazareno,dc=org" write
by * read
```

En resumen:

- Si ponemos un usuario en el campo de arriba con valor `write` tendrá permiso de escritura sobre todos los atributos descritos en la primera, sean o no los suyos propios. No tendrá permisos para añadir usuarios o realizar importaciones masivas.
- Si ponemos un usuario en el campo de abajo con valor `write` no tendrá ningún permiso, solo los que le concede el `by self write` del primer campo.

- Si ponemos un usuario en ambos campos conseguiremos un usuario con totales permisos sobre los elementos de la aplicación en PHP.

5.- PHP

5.1.- Introducción.

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

(Fuente: <http://es.wikipedia.org/wiki/Php>)

5.2.- Visión general

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, aunque el número de sitios en PHP ha compartido algo de su preponderante sitio con otros nuevos lenguajes no tan poderosos desde agosto de 2005. Este mismo sitio web de Wikipedia está desarrollado en PHP. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web. La versión más reciente de PHP es la 5.2.9-1 (for Windows) del 10 de Marzo de 2009.

El gran parecido que posee PHP con los lenguajes más comunes de [programación estructurada](#), como [C](#) y [Perl](#), permiten a la mayoría de los [programadores](#) crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Aunque todo en su diseño está orientado a facilitar la creación de página web, es posible crear aplicaciones con una [interfaz gráfica para el usuario](#), utilizando la extensión [PHP-Qt](#) o [PHP-GTK](#). También puede ser usado desde la [línea de órdenes](#), de la misma manera como [Perl](#) o [Python](#) pueden hacerlo, a esta versión de PHP se la llama PHP CLI (*Command Line Interface*).

Cuando el cliente hace una petición al servidor para que le envíe una [página web](#), el servidor ejecuta el intérprete de PHP. Éste procesa el [script](#) solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos [PDF](#), [Flash](#), así como imágenes en diferentes formatos.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como [MySQL](#), [Postgres](#), [Oracle](#), [ODBC](#), [DB2](#), [Microsoft SQL Server](#), [Firebird](#) y [SQLite](#).

PHP también tiene la capacidad de ser ejecutado en la mayoría de los [sistemas operativos](#), tales como [UNIX](#) (y de ese tipo, como [Linux](#) o [Mac OS X](#)) y [Windows](#), y puede interactuar con los [servidores de web](#) más populares ya que existe en versión [CGI](#), módulo para [Apache](#), e [ISAPI](#).

PHP es una alternativa a las tecnologías de [Microsoft ASP](#) y [ASP.NET](#) (que utiliza [C#](#) [VB.NET](#) como lenguajes), a [ColdFusion](#) de la compañía [Adobe](#) (antes [Macromedia](#)), a [JSP/Java](#) de [Sun Microsystems](#), y a [CGI/Perl](#). Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia [GNU](#), existe además un IDE ([entorno de desarrollo integrado](#)) comercial llamado [Zend Studio](#). Recientemente, [CodeGear](#) (la división de lenguajes de programación de [Borland](#)) ha sacado al mercado un [entorno integrado de desarrollo](#) para PHP, denominado **Delphi for PHP**. Existe un módulo para [Eclipse](#), uno de los IDE más populares, el cual ha sido el utilizado para este proyecto.

(Fuente: http://es.wikipedia.org/wiki/Php#Visi.C3.B3n_general)

5.3.- Introducción a las librerías usadas.

Durante el transcurso de este proyecto se han usado varias clases, intentando así una implementación, por pequeña que sea, de la estructura orientada a objetos en PHP. Las 3 clases usadas son las siguientes: *class_ldap*, *ldap2ldif* y *upload* (todos incluidos en la carpeta “include”).

- **Class_ldap:** clase empleada para hacer más cómodo el uso de las funciones de control de ldap incluidas en el núcleo de PHP. Será la clase que más empleemos en el código, ya que se usa tanto para realizar conexiones al servidor LDAP como para las modificaciones, adiciones, importaciones , etc.
- **Ldap2ldif:** clase empleada para la exportación del script de LDAP. Nos permitirá realizar exportaciones totales, parciales y usando filtros (por ejemplo, “gid=2001”).
- **Upload:** clase que nos permite la subida de archivos al servidor. Permite el reconocimiento de extensiones, puede limitar el peso del archivo que se vaya a subir, permite renombrar el archivo subido, reconoce el nombre del archivo, comprueba si el archivo se ha subido correctamente, redimensiona imágenes... La clase se usará en la función de importación, ya que para leer el contenido del archivo era necesario subirlo con anterioridad.

Esto no es más que una breve introducción de lo que se explicará más adelante.

6.- Moodle

6.1.- Introducción y características del CMS

Moodle es un [sistema de gestión de cursos](#), de [distribución libre](#), que ayuda a los educadores a crear comunidades de aprendizaje en línea. Este tipo de plataformas tecnológicas también se conocen como [LMS \(Learning Management System\)](#).

En términos de arquitectura, Moodle es una [aplicación web](#) que se ejecuta sin modificaciones en Unix, [GNU/Linux](#), [OpenSolaris](#), [FreeBSD](#), Windows, Mac OS X, NetWare y otros sistemas que soportan [PHP](#), incluyendo la mayoría de proveedores de hosting web.

Los datos son almacenados en una sola base de datos [SQL](#): la versión 1.7 (publicada en noviembre de 2006), hace uso total de abstracción de base de datos para que los instaladores puedan elegir entre alguno de los diversos tipos de servidores de bases de datos (Oracle y Microsoft SQL Server son dos objetivos específicos de sistemas administradores de bases de datos). La versión actual de Moodle (1.9) fue publicada en marzo de 2008. [MySQL](#) y [PostgreSQL](#) fueron las únicas opciones en Moodle 1.6.

6.2.- Configurar Moodle con LDAP.

Al contrario que la configuración en MediaWiki (que veremos más adelante), la de Moodle para que funcione con LDAP es mediante un formulario dentro de la misma plataforma. Los pasos dados para la configuración son los siguientes.

1. Nos autenticamos como administrador en moodle. En la parte izquierda nos saldrá un menú en vertical. Pulsamos sobre “users” y nos quedará algo como esto.



- Pulsamos sobre “Authentication” y nos aparecerán más opciones. Elegimos “Manage authentication”



- Nos aparecerá un listado de métodos de autenticación. Debemos activar la opción “Usar un servidor LDAP” pulsando sobre el ojo. Acto seguido pulsaremos en el botón “Configuración” de la línea de LDAP.

Nombre	Permitir	Arriba/Abajo	Configuración
Crear cuentas solo de forma manual			Configuración
No login			Configuración
Autenticación basada en Email		↓	Configuración
Usar un servidor LDAP		↑	Configuración
CAS server (SSO)			Configuración
Usar una base de datos externa			Configuración
FirstClass server			Configuración
Usar un servidor IMAP			Configuración
Moodle Network authentication			Configuración
Usar un servidor NNTP			Configuración
Sin autenticación			Configuración
PAM (Pluggable Authentication Modules)			Configuración
Usar un servidor POP3			Configuración
RADIUS server			Configuración
Shibboleth			Configuración

4. Rellenamos los campos con los datos de la imagen. La explicación a cada campo viene en el mismo formulario.

LDAP server settings

Host URL	<input type="text" value="ldap://127.0.0.1"/>	Especificar el host LDAP en forma de URL como 'ldap://ldap.myorg.com/' o 'ldaps://ldap.myorg.com/'
Version	<input type="text" value="3"/>	The version of the LDAP protocol your server is using.
LDAP encoding	<input type="text" value="utf-8"/>	Specify encoding used by LDAP server. Most probably utf-8, MS AD v2 uses default platform encoding such as cp1252, cp1250, etc.

Bind settings

Hide passwords	<input type="text" value="Si"/>	Select yes to prevent passwords from being stored in Moodle's DB.
Distinguished Name	<input type="text" value="cn=admin,dc=ldap,dc=gonzalonazareno"/>	Si quiere usar 'bind-user' para buscar usuarios, especifíquelo aquí. Algo como 'cn=ldapuser,ou=public,o=org'
Password	<input type="password" value="....."/>	Contraseña para bind-user.

User lookup settings

User type	<input type="text" value="Default"/>	Select how users are stored in LDAP. This setting also specifies how login expiration, grace logins and user creation will work.
Contexts	<input type="text" value="ou=People,dc=ldap,dc=gonzalonazareno"/>	Lista de contextos donde están localizados los usuarios. Separar contextos diferentes con ';'. Por ejemplo: 'ou=usuarios,o=org; ou=otros,o=org'
Search subcontexts	<input type="text" value="No"/>	Ponga el valor <> 0 si quiere buscar usuarios desde subcontextos.
Dereference aliases	<input type="text" value="No"/>	Determines how aliases are handled during search. Select one of the following values: "No" (LDAP_DEREF_NEVER) or "Yes" (LDAP_DEREF_ALWAYS)
User attribute	<input type="text" value="uid"/>	El atributo usado para nombrar/buscar usuarios. Normalmente 'cn'.
Member attribute	<input type="text"/>	Especificar el atributo para nombre de usuario, cuando los usuarios se integran en un grupo. Normalmente 'miembro'
Member attribute uses dn	<input type="text"/>	Optional: Overrides handling of member attribute values, either 0 or 1
Object class	<input type="text"/>	Optional: Overrides objectClass used to name/search users on ldap_user_type. Usually you dont need to chage this.

5. Continuamos rellenando. En esta parte solo cambiaremos una cosa: el tipo de encriptación en el caso de que la contraseña se modifique o se añada desde Moodle. Lo demás lo dejaremos por defecto.

Force change password	
Force change password	<input type="button" value="No"/>
Force users to change password on their first login to Moodle.	
Use standard Change Password Page	<input type="button" value="No"/>
If the external authentication system allows password changes through Moodle, switch this to Yes. This setting overrides 'Change Password URL'.	
NOTE: It is recommended that you use LDAP over an SSL encrypted tunnel (ldaps://) if the LDAP server is remote.	
Password format	<input type="button" value="Encriptación MS"/>
Specify the format of new or changed passwords in LDAP server.	
Password-change URL	<input type="text"/>
Aquí puede especificar dónde pueden sus usuarios recuperar o cambiar su nombre de usuario/contraseña si lo han olvidado. Para ello, aparecerá un botón en la página de entrada. Si deja esto en blanco, este botón no se mostrará.	
LDAP password expiration settings.	
Expiration	<input type="button" value="no"/>
Select No to disable expired password checking or LDAP to read passwordexpiration time directly from LDAP	
Expiration warning	<input type="text" value="10"/>
Number of days before password expiration warning is issued.	
Expiration attribute	<input type="text"/>
Optional: overrides ldap-attribute that stores password expiration time	
Grace logins	<input type="button" value="No"/>
Enable LDAP gracelogin support. After password has expired user can login until gracelogin count is 0. Enabling this setting displays grace login message if password is expired.	
Grace login attribute	<input type="text"/>
Optional: Overrides gracelogin attribute	
Habilitar creación por parte del usuario	
Create users externally	<input type="button" value="No"/>
Los nuevos usuarios (anónimos) pueden crear cuentas de usuario sobre el código externo de autenticación y confirmar vía correo electrónico. Si usted habilita esto, recuerde también configurar las opciones del módulo específico para la creación de usuario.	
Context for new users	<input type="text"/>
Si habilita la creación de usuario con confirmación por medio de correo electrónico, especifique el contexto en el que se crean los usuarios. Este contexto debe ser diferente de otros usuarios para prevenir problemas de seguridad. No es necesario añadir este contexto a ldap_context-variable, Moodle buscará automáticamente los usuarios de este contexto.	

6. Por último, rellenamos la sección de “Mapa de datos”. En esta sección indicaremos a que campo LDAP se corresponde cada dato. Hay más datos a parte de los que se muestran en la imagen, pero para nuestras necesidades no deberemos rellenarlos.

Data mapping

Nombre	<input style="width: 100%;" type="text" value="givenName"/> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Update local On creation </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Update external Never </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Lock value Unlocked </div>	<p>Estos campos son opcionales. Usted puede elegir pre-rellenar algunos campos de usuario en Moodle con información de los campos LDAP que especifique aquí.</p> <p>Si deja estos campos en blanco, entonces no se transferirá nada desde LDAP y se usará el sistema predeterminado en Moodle.</p>
Apellidos	<input style="width: 100%;" type="text" value="sn"/> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Update local On creation </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Update external Never </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Lock value Unlocked </div>	<p>En ambos casos, los usuarios podrán editar todos estos campos después de entrar.</p> <p>Update local: If enabled, the field will be updated (from external auth) every time the user logs in or there is a user synchronization. Fields set to update locally should be locked.</p>
Correo electrónico	<input style="width: 100%;" type="text" value="mail"/> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Update local On creation </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Update external Never </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Lock value Unlocked </div>	<p>Lock value: If enabled, will prevent Moodle users and admins from editing the field directly. Use this option if you are maintaining this data in the external auth system.</p> <p>Update external: If enabled, the external auth will be updated when the user record is updated. Fields should be unlocked to allow edits.</p>
Ciudad	<input style="width: 100%;" type="text" value="l"/> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Update local On creation </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Update external Never </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Lock value Unlocked </div>	<p>Note: Updating external LDAP data requires that you set binddn and bindpw to a bind-user with editing privileges to all the user records. It currently does not preserve multi-valued attributes, and will remove extra values on update.</p>
Pais	<input style="width: 100%;" type="text" value="c"/> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Update local On creation </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Update external Never </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Lock value Unlocked </div>	

7.- MediaWiki

7.1.- Introducción y características del CMS

MediaWiki es un [motor](#) para [wikis](#) bajo licencia [GNU](#), programado en [PHP](#). A pesar de haber sido creado y desarrollado para [Wikipedia](#) y otros proyectos de la fundación [Wikimedia](#) ([Wikcionario](#), [Wikilibros](#), ...), ha tenido una gran expansión desde el año 2005, existiendo un gran número de wikis basados en este software que nada tienen que ver con dicha fundación, aunque si comparten la idea de la generación de contenidos de manera colaborativa. La mayoría de ellos se dedican a la [documentación de software](#) o a temas especializados.

También se llama MediaWiki al [espacio de nombres](#) (ver más abajo) donde se hallan los mensajes de su interfaz listos para su traducción a la lengua local de cada wiki.

Puede ser instalado sobre servidores Web [Apache](#) o [IIS](#) y puede usar como motor de base de datos [MySQL](#) o [PostgreSQL](#).

Características

- A diferencia de los wikis clásicos, los nombres de las páginas no tienen porqué estar en "[CamelCase](#)", lo que permite tener nombres más naturales.
- Espacios de nombres: permiten separar páginas de distintos tipos. Así, se puede tener un espacio de nombres para artículos, otro para plantillas, otro para debates, etc. que el software trata de distinta forma.
- Páginas de discusión: cada página del wiki tiene una página de discusión propia, dedicada a hablar de su mejora u otros fines.
- Soporte de [TeX](#), para visualizar fórmulas matemáticas. Las fórmulas pueden mostrarse de varias formas, según las capacidades del navegador.
- Listas de seguimiento, de tal forma que cada usuario pueda seguir los cambios en los artículos de su interés.
- Sistema de plugins que permite extender fácilmente el software. Los plugins instalados se listan automáticamente en "Páginas especiales".
- Capacidad de bloquear temporalmente usuarios o páginas.
- Soporte de plantillas personalizadas con parámetros.
- Creación de líneas de tiempos a través de código wiki.
- Sistema de categorías jerárquico, que permite crear listados de artículos o de "thumbnails" de imágenes.

- Admite varios niveles de usuario, así como la posibilidad de que sólo los usuarios registrados puedan editar, o de impedir el registro de más usuarios. Así, puede utilizarse como [CMS](#) o [Groupware](#).
- Soporte para [memcached](#) y el sistema de caché [Squid](#).
- Pieles ("skins") personalizables por cada usuario.

(Fuente: <http://es.wikipedia.org/wiki/MediaWiki>)

7.2.- Configurar MediaWiki con LDAP.

El proceso para hacer que MediaWiki funcione con LDAP es el siguiente:

- Descargamos la [extensión necesaria para que MediaWiki funcione con LDAP](#).
- Pegamos el archivo dentro de `/var/www/carpeta_mediawiki/extensions`
- Ahora editaremos el archivo `LocalSettings.php`, que se encuentra dentro de la carpeta de MediaWiki y, al final del documento, pondremos lo siguiente:

```
#####
##  LDAP  ##
#####

require_once("$IP/extensions/LdapAuthentication.php");
$wgAuth = new LdapAuthenticationPlugin();

//Nombre de dominios utilizados
$wgLDAPDomainNames = array("gonzalonazareno.org");

//Asociación entre nombres de dominio y nombre dns
$wgLDAPServerNames =
array("gonzalonazareno.org"=>"localhost");

//Autenticacion de WIKI con LDAP ON!
$wgLDAPUseLocal = true;

//Encriptacion en las solicitudes LDAP
$wgLDAPEncryptionType = array("gonzalonazareno.org"=>"clear");

//Elegimos la base de las consultas
```

```
$wgLDAPBaseDNs =  
array("gonzalonazareno.org"=>"ou=People,dc=ldap,dc=gonzalonaza  
reno,dc=org");  
$wgLDAPSearchAttributes = array("gonzalonazareno.org"=>"uid");  
  
//utilizamos los grupos para las directivas de grupo  
$wgLDAPGroupsPrevail = array("gonzalonazareno.org"=>true);  
$wgLDAPGroupNameAttribute =  
array("gonzalonazareno.org"=>"cn");  
  
## Validacion y control de acceso para visitantes.  
$wgGroupPermissions['*']['createaccount'] = false;  
$wgGroupPermissions['*']['read'] = true;  
$wgGroupPermissions['*']['edit'] = false;  
$wgGroupPermissions['*']['createpage'] = false;  
$wgGroupPermissions['*']['createtalk'] = false;
```

8.- Aplicación de gestión de usuarios

8.1.- ¿Qué hace la aplicación?

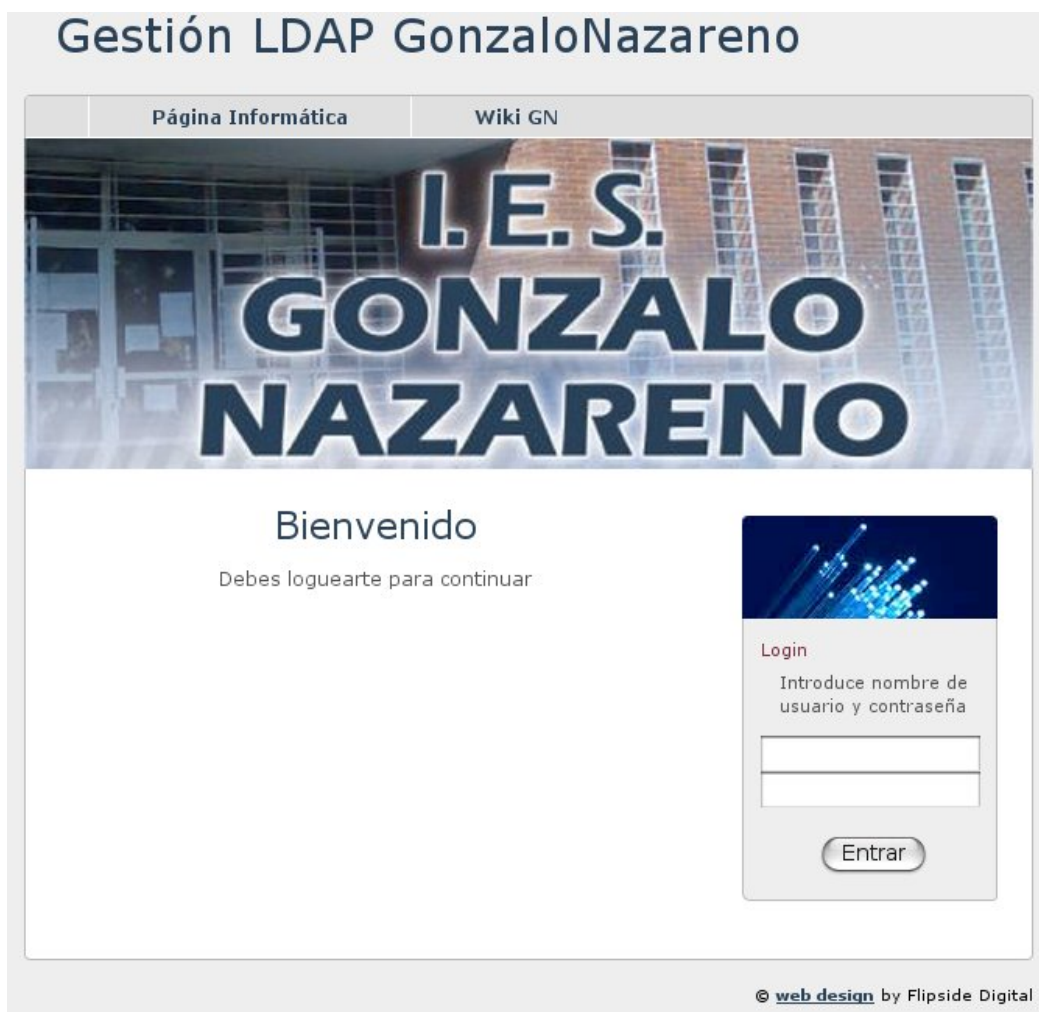
La función de esta aplicación es la de gestionar la información contenida en el servidor LDAP. Operaciones como el borrado, añadido o importado de usuarios se realizará de forma sencilla mediante la aplicación. Todo será explicado con mayor detalle en las próximas secciones.

8.2.- Manual de usuario

En esta sección se explicará cada una de las funciones que se pueden usar en la aplicación y el modo de usarlas a nivel de usuario. Aspectos más técnicos serán tratados en las próximas secciones.

8.2.1.- Pantalla principal

La pantalla principal está compuesta por un formulario de autenticación y una pantalla de bienvenida. Una vez que nos loguemos, tanto el formulario como la pantalla de bienvenida cambiarán.



Gestión LDAP GonzaloNazareno

Página Informática Wiki GN

I.E.S. GONZALO NAZARENO

Bienvenido

Debes loguearte para continuar

Login

Introduce nombre de usuario y contraseña

Entrar

© web design by Flipside Digital

8.2.2.- Pantalla principal del profesor.

Una vez que nos autentiquemos, el formulario de *login* cambiará según seamos profesor o alumno. En el caso del profesor, la pantalla principal será como esta:



Las funciones del administrador son las siguientes:

- **Añadir usuario:** nos permite añadir un nuevo usuario a partir de un formulario.
- **Listado completo:** nos hace un listado de todos los usuarios que se encuentren en el servidor LDAP. Se nos permitirá realizar modificaciones de esos usuarios o borrarlos.
- **Exportar estructura:** nos exporta la estructura del servidor LDAP en formato LDIF.
- **Importar:** nos permite importar alumnos a partir de un archivo CSV.


- **Tu perfil:** te permite modificar los datos de tu perfil.
- **Desconectar:** cierra la sesión.

8.2.3.- Pantalla principal de alumno

La funcionalidad de un alumno en esta aplicación es limitada. Como podemos observar, las opciones del alumno quedan reducidas a dos: **modificar su perfil** y **desconectarse**.

Gestión LDAP GonzaloNazareno

[Página Informática](#)[Wiki GN](#)



Bienvenido

Se ha identificado como ark

Perteneces al grupo "Alumnos"



Login

Bienvenido ark

--Menu--

[Tu perfil](#)

[Desconectar](#)

© [web design](#) by Flipside Digital

8.2.4.- Añadir usuario

Con esta función podemos añadir nuevos usuarios rellenando el formulario. Podremos registrar usuarios como profesores o alumnos, y dispondremos de un botón de generación aleatoria de contraseñas. El número de usuario es asignado automáticamente.

Gestión LDAP GonzaloNazareno

Página Informática

Wiki GN



I.E.S. GONZALO NAZARENO

Nombre de usuario

Contraseña

Generar

Nombre

Apellidos

Correo electrónico

Ciudad

Grupo

Seleccione un grupo ...

Modificar



Login

Bienvenido paco

--Menu--

[Añadir usuario](#)

[Listado completo](#)

[Exportar estructura](#)

[Importar](#)

[Tu perfil](#)

[Desconectar](#)

© [web design](#) by Flipside Digital

8.2.5.- Listado completo

Nos mostrará un listado completo de usuarios que estén registrados en el servidor LDAP, así como algunos datos significativos.

Gestión LDAP GonzaloNazareno

[Página Informática](#)[Wiki GN](#)



Listado de usuarios en el servidor

Nombre completo (cn)	Paco Admin Poderoso
Nombre de usuario (uid)	paco
Número de grupo (gidnumber)	2000
ID de usuario (uidnumber)	2024

[Modificar](#)[Borrar](#)



Login

Bienvenido paco

--Menu--

[Añadir usuario](#)

[Listado completo](#)

[Exportar estructura](#)

[Importar](#)

[Tu perfil](#)

[Desconectar](#)

Nombre completo (cn)

Nombre de usuario (uid)

Ismael Márquez Díaz

bampi

En cada uno de los usuarios disponemos de dos botones, uno para modificar y otro para borrar. Pulsando sobre el botón *modificar* encontraremos una pantalla como la siguiente:

Gestión LDAP GonzaloNazareno

[Página Informática](#)[Wiki GN](#)



Modificando usuario

Nombre de usuario	<input type="text" value="bampi"/>
Nombre	<input type="text" value="Ismael"/>
Apellidos	<input type="text" value="Márquez Díaz"/>
Correo electrónico	<input type="text" value="bampidh18@gmail.com"/>
Ciudad	<input type="text" value="Dos Hermanas"/>

Modificar



Login
Bienvenido paco
--Menu--
[Añadir usuario](#)
[Listado completo](#)
[Exportar estructura](#)
[Importar](#)
[Tu perfil](#)
[Desconectar](#)

© [web design](#) by Flipside Digital

Los cambios realizados serán confirmados una vez pulsemos sobre *modificar*.

8.2.6.- Exportar estructura

Pulsando en esa opción nos aparecerá una ventana para guardar un archivo en formato LDIF (según que navegadores, se nos pedirá guardarlo o se abrirá directamente en el navegador).

Fragmento de ejemplo del resultado:


```
# LDIF generated by ldap2ldif (1.0)
# Author: Antonio Luis Román-Naranjo Varela
(arkpokalipsis@gmail.com)
# Server: Proyecto integrado
# Total Entries: 126

# LDIF Entry 1: dc=ldap,dc=gonzalonazareno,dc=org
dn:dc=ldap,dc=gonzalonazareno,dc=org
objectclass: top
objectclass: dcObject
objectclass: organization
o: Gonzalo Nazareno
dc: ldap

# LDIF Entry 2: cn=admin,dc=ldap,dc=gonzalonazareno,dc=org
dn:cn=admin,dc=ldap,dc=gonzalonazareno,dc=org
objectclass: simpleSecurityObject
objectclass: organizationalRole
description: LDAP administrator

# LDIF Entry 3: ou=People,dc=ldap,dc=gonzalonazareno,dc=org
dn:ou=People,dc=ldap,dc=gonzalonazareno,dc=org
ou: People
objectclass: top
objectclass: organizationalUnit

# LDIF Entry 4: ou=Group,dc=ldap,dc=gonzalonazareno,dc=org
dn:ou=Group,dc=ldap,dc=gonzalonazareno,dc=org
ou: Group
objectclass: top
objectclass: organizationalUnit

# LDIF Entry 5:
cn=profesores,ou=Group,dc=ldap,dc=gonzalonazareno,dc=org
dn:cn=profesores,ou=Group,dc=ldap,dc=gonzalonazareno,dc=org
objectclass: posixGroup
objectclass: top
gidnumber: 2000

# LDIF Entry 6:
cn=alumnos,ou=Group,dc=ldap,dc=gonzalonazareno,dc=org
dn:cn=alumnos,ou=Group,dc=ldap,dc=gonzalonazareno,dc=org
objectclass: posixGroup
objectclass: top
gidnumber: 2001

# LDIF Entry 7:
uid=paco,ou=People,dc=ldap,dc=gonzalonazareno,dc=org
dn:uid=paco,ou=People,dc=ldap,dc=gonzalonazareno,dc=org
gidnumber: 2000
homedirectory: /home/paco
objectclass: inetOrgPerson
objectclass: posixAccount
objectclass: top
uidnumber: 2024
uid: paco
```

```
sn: Admin Poderoso
givenname: Paco
mail: pacorl@gmail.com
l: Dos Hermanas
```


```
# LDIF Entry 8:
uid=bampi,ou=People,dc=ldap,dc=gonzalonazareno,dc=org
dn:uid=bampi,ou=People,dc=ldap,dc=gonzalonazareno,dc=org
uid: bampi
objectclass: top
objectclass: posixAccount
objectclass: person
objectclass: inetOrgPerson
uidnumber: 2097
gidnumber: 2001
homedirectory: /home/bampi
givenname: Ismael
sn: Márquez Díaz
mail: bampidh18@gmail.com
l: Dos Hermanas
```

8.2.7.- Importar

Podremos importar una lista de usuarios a partir de un archivo CSV. Solo se podrán importar alumnos, los profesores deberán ser importados a mano.

Gestión LDAP GonzaloNazareno

[Página Informática](#)[Wiki GN](#)




Importar

Elija el fichero desde el que desea importar los datos. Debe ser un archivo csv separado por comas. El orden debe ser (aquí irá el orden)

[Examinar...](#)

[Importar](#)



Login

Bienvenido paco

--Menu--

[Añadir usuario](#)

[Listado completo](#)

[Exportar estructura](#)

[Importar](#)

[Tu perfil](#)

[Desconectar](#)

© [web design](#) by Flipside Digital

El archivo CSV debe seguir el siguiente orden: "Nombre", "1er apellido", "2º apellido", "DNI", "correo", "Localidad". El DNI se emplea como contraseña. Ej.:

"Antonio", "Román-N", "Varela", "11111111-K", asdf@gmail.com, "Dos Hermanas"

8.2.8.- Tu perfil

Con esta función podremos editar nuestros datos personales. Se usa exactamente el mismo formulario que para el *modificar*.

Gestión LDAP GonzaloNazareno

[Página Informática](#)[Wiki GN](#)



Modificando usuario

Nombre de usuario	<input type="text" value="paco"/>
Nombre	<input type="text" value="Paco"/>
Apellidos	<input type="text" value="Administrador"/>
Correo electrónico	<input type="text" value="pacorl@gmail.com"/>
Ciudad	<input type="text" value="Dos Hermanas"/>

Modificar



Login
Bienvenido paco
--Menu--
[Añadir usuario](#)
[Listado completo](#)
[Exportar estructura](#)
[Importar](#)
[Tu perfil](#)
[Desconectar](#)

© [web design](#) by Flipside Digital

8.3.- Documentación del programa

En esta sección se entrará en aspectos de programación del funcionamiento del programa, tanto de las clases como del mismo código, así como explicaciones de ciertas partes del código.

8.3.1.- Funcionamiento de las clases utilizadas

En esta sección se hará una descripción de cada una de las clases. Los ejemplos de uso están dentro de la misma clase o en un archivo separado en la misma carpeta, según el caso.

8.3.1.1.- LDAP Class

Esta clase ha sido creada específicamente desde 0 para este proyecto con el objetivo de entrar un poco más en el campo de la estructura orientada a objetos en PHP. (Información más completa dentro de la misma clase).

- **Variables globales**

- `$servidor`: aquí se guardará la IP del servidor LDAP. Toma su valor al crear el objeto.
- `$conexion`: aquí se guardará la ID de conexión al realizar el `ldap_connect`.
- `$busqueda`: aquí se guardará el resultado del `ldap_search`.
- `$resultado_buscar`: aquí se guardará el resultado del `ldap_get_entries`.

- **Funciones**

- `__construct ($ip_servidor)`: constructor de la clase.
- `conectar ()`: usa la variable dada al constructor para realizar una conexión al servidor LDAP.
- `autenticar($usuario,$cad_conexion,$pass)`: hace uso de los 3 datos de entrada para realizar una autenticación contra el servidor LDAP.
- `anon_auth()`: realiza una autenticación anónima al servidor LDAP.
- `add ($uid,$cad_conexion, $array)`: sirve para añadir registros con nombre de usuario `$uid` y los datos contenidos en `$array`.
- `Buscar($dn_busqueda,$filtro)`: realiza búsquedas en `$dn_busqueda` usando del uid `$filtro`.
- `obtener_entradas ()`: Obtiene las entradas resultado de la búsqueda anterior.
- `entrada ($dato,$linea)`: Obtiene el dato del lugar del array especificado.
- `modificar ($uid,$cadena_conecx,$array)`: Funcion que modifica una entrada a partir del uid, cogiendo como datos los introducidos en un array.

- `borrar ($uid,$cadena_conexion)`: Funcion para borrar registros. Solo borra por uid. Con esto nos sobra para lo que vamos a hacer.
- `logout()`: nos desloguea de la sesión actual.
- `Desconectar()`: nos desconecta del servidor LDAP.

8.3.1.2.- LDAP2LDIF

De esta clase solo se comentará una parte, ya que el resto no se ha usado o no es de importancia para nuestro proyecto. (Información más completa dentro de *Manual.txt*)

▪ Funciones

- `init("usuario=NULL". "Contraseña=NULL")`: función para autenticarse contra el servidor LDAP. Si no se especifican usuario y contraseña se realizará una autenticación anónima.
- `search("rama", "criterio de búsqueda")`: realiza una búsqueda en el servidor LDAP con un criterio dado (Ej: uid="antonio").
- `export()`: exporta en formato LDIF el resultado de la búsqueda.

8.3.1.3.- Upload

Al igual que en la clase anterior, de esta solo se comentará lo necesario, ya que hay funcionalidades (redimensión de imágenes, por ejemplo) que no se han empleado. (Información más completa dentro de *manual.php*).

▪ Funciones

- `set_max_size("num_bytes")`: nos permite especificar el tamaño máximo de la subida (en bytes).
- `set_directory("directorio")`: nos permite especificar el directorio donde se subirán los ficheros.
- `set_tmp_name($_FILES['file']['tmp_name'])`: guarda en una variable el nombre del archivo subido.
- `set_file_size($_FILES['file']['size'])`: guarda en una variable el tamaño del archivo subido.
- `set_file_type($_FILES['file']['type'])`: guarda en una variable el tipo de fichero.
- `set_file_name($_FILES['file']['nombre_nuevo'])`: renombra el archivo.
- `start_copy()`: comienza la copia del archivo.

- `is_ok()`: devuelve *true* si la subida se hizo correctamente, y *false* si no lo hizo.

9.- Conclusión del proyecto

Ahora que he terminado el proyecto, puedo decir que la mayoría de los objetivos del proyecto se han cumplido, aunque también es cierto que esperaba que el tiempo que se dio para el proyecto diera algo más de sí. Son muchas las cosas que se han quedado en el tintero y que me gustaría haber tratado, pero por falta de tiempo no han podido ser...Aún así, se han conseguido hacer bastantes, a pesar de los muchos problemas surgidos, sobre todo en el campo de la programación.

Así pues, he conseguido realizar la aplicación de gestión de LDAP y he aprendido bastantes cosas que antes del proyecto desconocía (ACL, GIT, KVM, algunas cosas de programación...). Pero, sobre todo, este proyecto me ha servido para darme cuenta de lo tantísimo que he aprendido en tan poco tiempo, y también para ver que soy capaz de hacer cosas que no sabía hacer en un principio sin ayuda constante de nadie.

En cuanto a mi apreciación del proyecto en sí, creo que he conseguido mi propósito, cumpliendo la mayoría de los objetivos dados en el anteproyecto, haciendo que sea un trabajo más que aceptable para el tiempo que he tenido.