GOBIERNO DE ESPAÑA
MINISTERIO DE EDUCACIÓN, POLÍTICA SOCIAL Y DEPORTE
SECRETARÍA DE ESTADO DE EDUCACIÓN Y FORMACIÓN
DIRECCIÓN GENERAL DE FORMACIÓN PROFESIONAL
INSTITUTO SUPERIOR DE FORMACIÓN Y RECURSOS EN RED PARA EL PROFESORADO

REDES DE ÁREA LOCAL: APLICACIONES Y SERVICIO EN LINUX

Servidor Web Apache

Formación en Red

Índice de contenido

Servidor Web Apache

El servidor web apache es una de las aplicaciones estrella del mundo Linux. Es el servidor web más implantado entre los distintos servidores que ofertan servicios web en Internet.

Entre las características más significativas destacamos:

- Es modular
- Capacidad para crear servidores virtuales
- Capacidad para crear servidores seguros https
- Capacidad para crear sitios privados

En este curso haremos uso de éstas y otras características de apache.

Organización del sitio web

La organización que realizaremos de nuestro servidor Apache, será la clásica en los sistemas Unix: la **página web de la intranet** se almacenará en la carpeta raíz del servidor web, las **páginas de los usuarios** se almacenarán en la carpeta home de cada usuario y para albergar las **páginas web de los distintos departamentos** didácticos del centro, lo más práctico es crear nuevos usuarios con el nombre del departamento.

Espacio web para la Intranet

Por defecto, la carpeta raiz del servidor web es la carpeta /var/www. Todos los documentos que se encuentren dentro de la carpeta raíz del servidor web, serán

accesibles vía web. Dentro del raiz de documentos crearemos la **página web de nuestra intranet.**

Carpeta principal del servidor web (DocumentRoot)

- **Carpeta raíz del servidor web:** /var/www
- **Acceso a la web principal:** http://ip-del-servidor ó http://nombre-del-servidor

Para acceder vía web a la página almacenada en la carpeta raíz del servidor, desde un navegador debemos acceder directamente con la dirección IP a: http://ip-del-servidor o bien utilizando el nombre del mismo si tenemos el DNS funcionando: http://nombre-del-servidor. Si no tenemos el DNS funcionando, podemos añadir el nombre y la IP en /etc/hosts para resolver localmente.

Espacio web para cada usuario

Cada usuario del sistema dispondrá de un espacio web que se almacena dentro de su carpeta home en una carpeta llamada 'public_html'. Si dicha carpeta no existe, el propio usuario puede crearla y copiar dentro de ella su página web. Los permisos recomendados son 644 para que el 'grupo' y el 'resto' de usuarios tengan acceso de lectura y así se puedan visualizar las páginas.

Para acceder vía web a la página de un usuario, desde un navegador debemos acceder directamente con la dirección IP a: http://ip-del-servidor/~login-usuario/

El caracter '~'comúnmente conocido como gusanillo y que se obtiene con Alt Gr + 4 sirve para indicar a apache que debe servir la página desde el home del usuario (en Linux el 'gusanillo' equivale a la carpeta home). Ejemplo, si hemos creado un usuario javier y éste ha creado la carpeta /home/javier/public_html y ha copiado en ella su página web, desde cualquier PC de la red podremos acceder a dicha carpeta yendo a la dirección http://ip-del-servidor/~javier/. Para que la página aparezca automáticamente, es necesario crear un archivo llamado index.html.

Carpetas web de los usuarios

- **Carpeta web de javier:** /home/javier/public_html
- **Acceso a la web de javier:** http://ip-del-servidor/~javier/

Espacio web para los departamentos

Para proporcionar espacio web a los departamentos, lo más sencillo es crear un usuario para cada departamento. Podemos crear los usuarios: matematicas, lengua, ingles, plastica (sin acentos), etc... Al igual que cada usuario del sistema, dispondrán de un espacio web dentro de su carpeta home en una carpeta llamada 'public_html'. Si dicha carpeta no existe, habrá que crearla y copiar dentro de ella la página web del departamento.

Para acceder vía web a la página del departamento, desde un navegador debemos acceder directamente con la dirección IP a: http://ip-del-servidor/~departamento. Ejemplo, si hemos creado un usuario matematicas y hemos creado la carpeta /home/matematicas/public_html y copiado en ella la web del departamento de

matemáticas, desde cualquier PC de la red podremos acceder a dicha web yendo a la dirección http://ip-del-servidor/~matematicas. Para que la página aparezca automáticamente, es necesario crear un archivo llamado index.html.

Carpetas web de los departamentos

- **Carpeta web del dpto. de matemáticas:** /home/matematicas/public_html
- **Acceso a la web de dpto. de matemáticas:** http://ip-del-servidor/~matematicas/

De la misma manera, se pueden crear usuarios para proporcionar espacio web a otros órganos del centro, p.ej: ccp, orientacion, equipodirectivo, conserjeria, etc... para que dispongan de su propio espacio web.

Espacio web seguro

Además crearemos un sitio web virtual seguro en el servidor web Apache para poder tener acceso vía SSL a contenidos que deseamos que sean seguros, es decir, accesibles en el navegador mediante el protocolo "https", será la carpeta /var/www/websegura

Carpeta web segura

- **Carpeta web segura:** /var/www/websegura
- **Acceso a la web segura:** https://ip-del-servidor/websegura/

Dentro de esta estructura la mayoría de los contenidos serán públicos y cualquier usuario podrá acceder a ellos. Sin embargo, algunas de las carpetas serán privadas y solo se tendrá acceso a ellas identificándose con nombre de usuario y contraseña.

Instalación de Apache2

Disponer de un servidor web en el centro nos permitirá alojar nuestras propias páginas y aplicaciones web de forma que den servicio tanto desde dentro de la intranet como desde Internet. Serán la base que facilitará el acceso a la información por parte de la comunidad educativa.

```
// Instalación de apache2
# apt-get install apache2
```

Con lo cual se instalarán los archivos necesarios para que funcione nuestro servidor web. Se instalará apache v2.

Configuración de Apache

Los archivos de configuración de apache2 se encuentran en la carpeta **/etc/apache2**. El archivo principal de configuración es **/etc/apache2/apache2.conf**. Antes de realizar cualquier cambio en este archivo, es conveniente realizar una copia de seguridad del mismo ya que si apache encuentra algún error en el archivo de configuración, no arrancará.

Se pueden configurar infinidad de parámetros. Aquí, para poner en marcha el servidor, editaremos el archivo apache2.conf y añadiremos únicamente el siguiente parámetro:

```
// Añadir en apache2.conf
ServerName www.ieslapaloma.com
```

Para que los PCs de la red local sepan que www.ieslapaloma.com es nuestro servidor web, debemos crear una entrada 'www' hacia su dirección IP en el servidor DNS, o bien editar el archivo /etc/hosts agregando la línea: '192.168.1.239 www.ieslapaloma.com' (si la IP del servidor fuera 192.168.1.239). Si no, no quedará más remedio que acceder utilizando la dirección IP del servidor.

Arranque y parada del servidor web apache

El servidor web apache2, al igual que todos los servicios en Debian, dispone de un script de arranque y parada en la carpeta /etc/init.d.

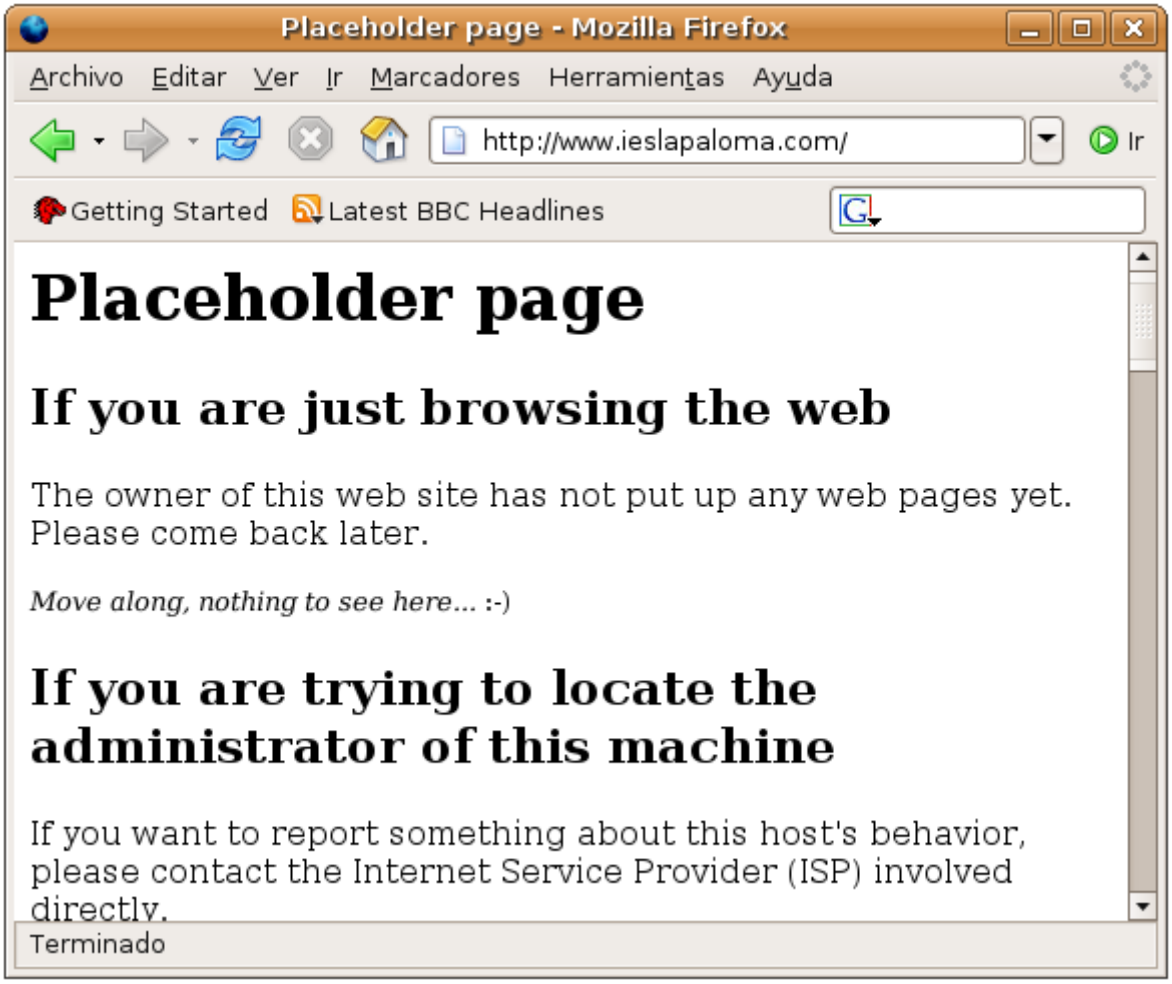
```
// Arrancar o reiniciar el servidor apache2
# /etc/init.d/apache2 restart

// Parar el servidor apache
root@cnice-desktop:/# /etc/init.d/apache stop
```

Arranque automático del servidor Web Apache al iniciar el sistema.

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado Arranque automático de servicios al iniciar el sistema.

Para comprobar que apache funciona perfectamente, desde el navegador de cualquier estación de trabajo de nuestro centro, debemos dirigirnos a 'http://ip-del-servidor'. Si tenemos el DNS funcionando, podemos acceder a 'http://www.ieslapaloma.com', visualizando la siguiente pantalla:



Lo que siempre funcionará es ir con la dirección IP. Ejemplo, si la dirección IP de nuestro servidor fuera 192.168.1.239, podemos ir con el navegador a la dirección http://192.168.1.239 y obtendremos el mismo resultado. Podemos personalizar nuestra página modificando el archivo index.html que hay dentro de la carpeta /var/www.

Como vemos en la pantalla anterior, la instalación de Apache se produjo de forma adecuada, así pues hemos completado este apartado satisfactoriamente.

Acceso a carpetas seguras

Introducción

Una página web segura o un sitio web seguro es un sitio web que utiliza el protocolo https en lugar de utilizar el protocolo http.

El protocolo https es idéntico al protocolo http con la excepción de que la transferencia de información entre el cliente (navegador web) y el servidor (servidor web) viaja a través de Internet cifrada utilizando robustos algoritmos de cifrado de datos proporcionados por el paquete OpenSSL.

Los algoritmos de cifrado utilizados reunen las características necesarias para garantizar que la información que sale desde el servidor hacia el cliente, esté cifrada y solamente pueda ser descifrada por el cliente y que la información que sale desde el cliente hacia el

servidor, esté cifrada y solamente pueda ser descifrada por el servidor. Si durante la transferencia de la información un 'hacker' hiciera copia de los paquetes de datos e intentara descifrarlos, los algoritmos garantizarían que no podría hacerlo por fuerza bruta (probando todas las claves posibles) en un plazo mínimo de varios años.

Durante la transmisión, se utilizan algotirmos de cifrado simétricos, pero para intercambiar las claves de cifrado, hay una sesión inicial de cifrado asimétrico.

Módulo ssl para apache2

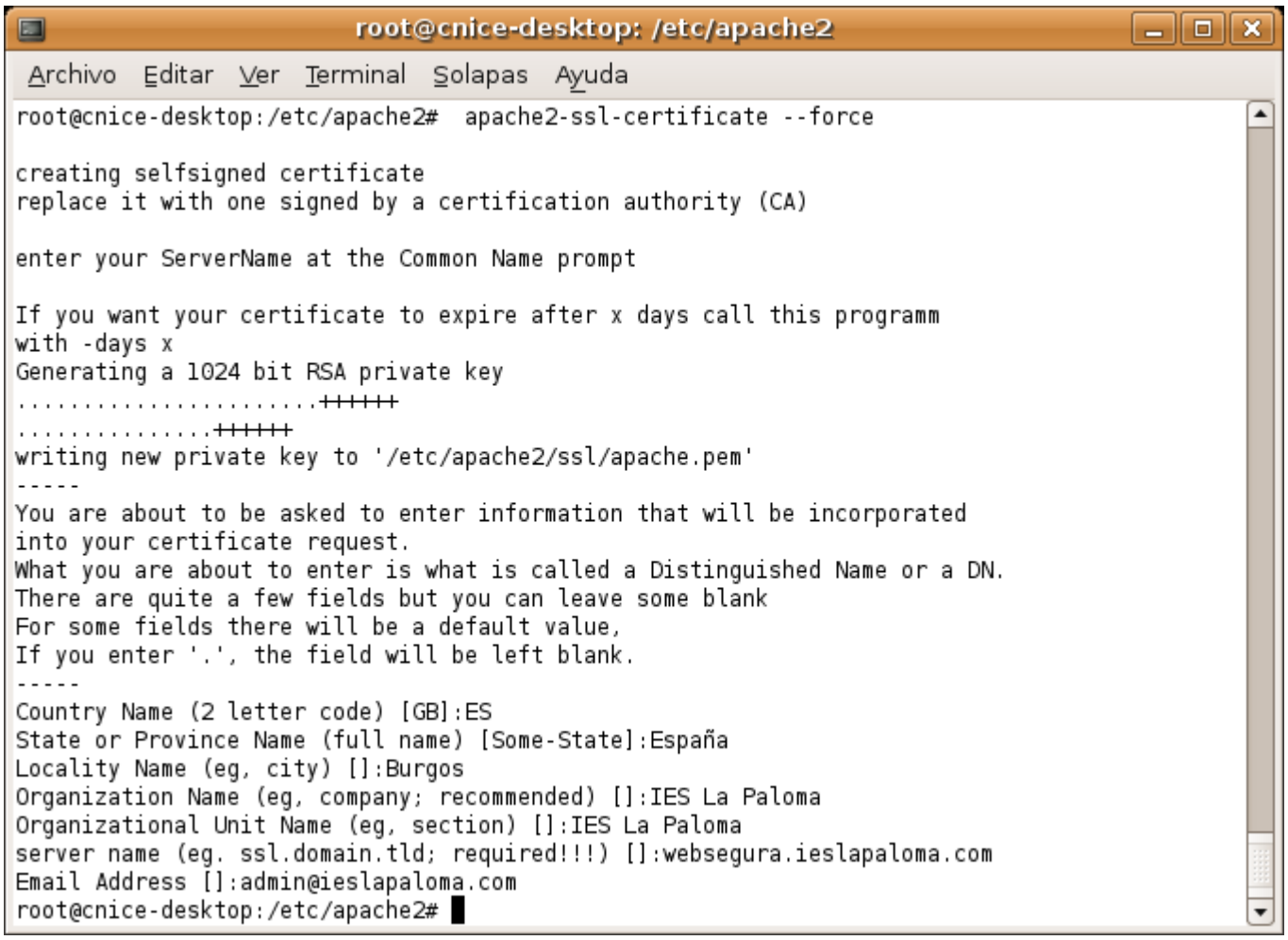
Al instalar apache2 se instala también el módulo ssl para apache2, por lo que no es necesario instalar ningún paquete adicional. Tan solo debemos generar un certificado para el servidor y activar el módulo ssl.

Generar el certificado

Para que nuestro servidor pueda servir páginas seguras con el protocolo https, necesita un certificado. Dicho certificado permitirá que nuestro servidor utilizar cifrado asimétrico para intercambiar las claves de cifrado con los clientes, antes de iniciar una trasmisión segura de información. Inicialmente, el cliente deberá aceptar el certificado del servidor, ya que generaremos un certificado autofirmado. Si queremos evitarlo, deberíamos contratar un certificado a una entidad certificadora confiable, pero tiene un coste que no merece la pena soportar en un entorno educativo. Para generar nuestro certificado autofirmado, ejecutaremos el comando:

```
// Generar certificado autofirmado
# apache2-ssl-certificate
```

Tan solo tendremos que responder a algunas preguntas sencillas sobre nuestra ubicación geográfica y el nombre de nuestra organización. A continuación vemos un ejemplo del uso del comando:



Después de la ejecución de comando apache2-ssl-certificate, se habrá generado el archivo /etc/apache2/ssl/apache.pem que contiene la claves que permitirán al servidor utilizar cifrado asimétrico. El siguiente paso será configurar un servidor virtual para que utilice dicho certificado.

Crear servidor virtual seguro en apache2

Primero crearemos una carpeta de nombre 'websegura' dentro de '/var/www'. Dicha carpeta será el raíz de documentos (DocumentRoot) de nuestro servidor virtual seguro, de modo que todo lo que coloquemos en dicha carpeta deba ser accedido vía 'https'. Eso lo indicaremos más adelante mediante el parámetro SSLRequireSSL. El protocolo https utiliza el puerto 443, por lo tanto, tendremos habilitar dicho puerto para que apache lo utilice:

```
// Habilitar puerto 443. Añadir en /etc/apache2/ports.conf
Listen 443
```

Después debemos crear el servidor virtual en apache. Dicho servidor virtual dispondrá de una url de acceso diferente a la de nuestra web principal (websegura.ieslapaloma.com en nuestro ejemplo) y será accesible mediante https, por tanto tendremos que habilitar SSL e indicar la ruta del archivo que contiene el certificado. Todo ello lo haremos editando el archivo /etc/apache2/sites-available/default:

```
// Servidor virtual seguro.

// Añadir al principio en /etc/apache2/sites-available/default
NameVirtualHost websegura.ieslapaloma.com:443



// Añadir al final en /etc/apache2/sites-available/default

<VirtualHost websegura.ieslapaloma.com:443>
        ServerName websegura.ieslapaloma.com
        DocumentRoot /var/www/websegura
                SSLEngine On
        SSLCertificateFile /etc/apache2/ssl/apache.pem
        ErrorLog /var/log/apache2/error.log
        CustomLog /var/log/apache2/access.log combined
</VirtualHost>

<Directory "/var/www/websegura">
    Options Indexes FollowSymlinks MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
    SSLRequireSSL
</Directory>
```

Posteriormente debemos habilitar el módulo ssl del servidor apache:

```
// Habilitar el módulo ssl
 # a2enmod ssl
```

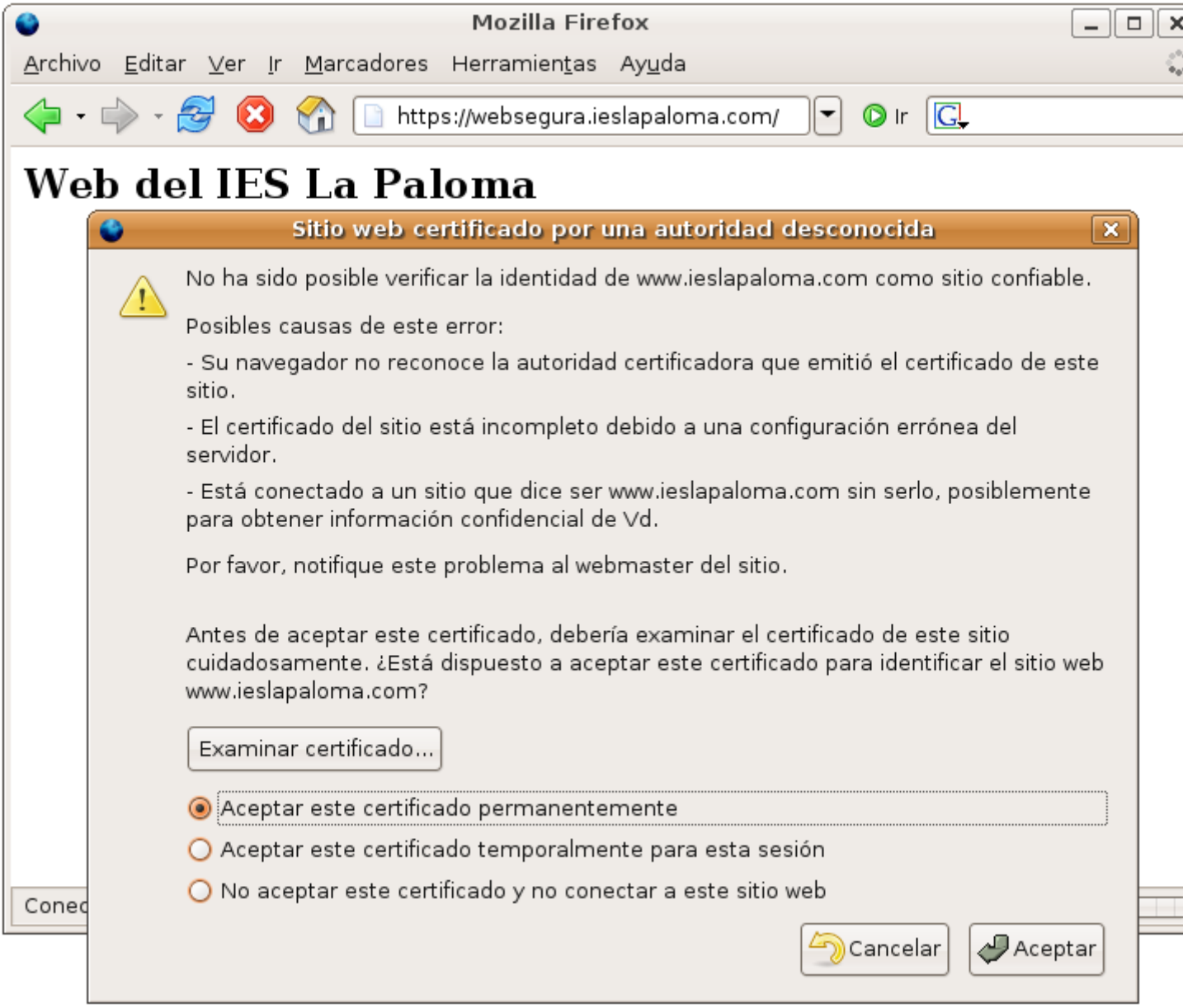
Finalmente reiniciamos el servidor apache:

```
// Reinicio de apache
 # /etc/init.d/apache2 restart
```

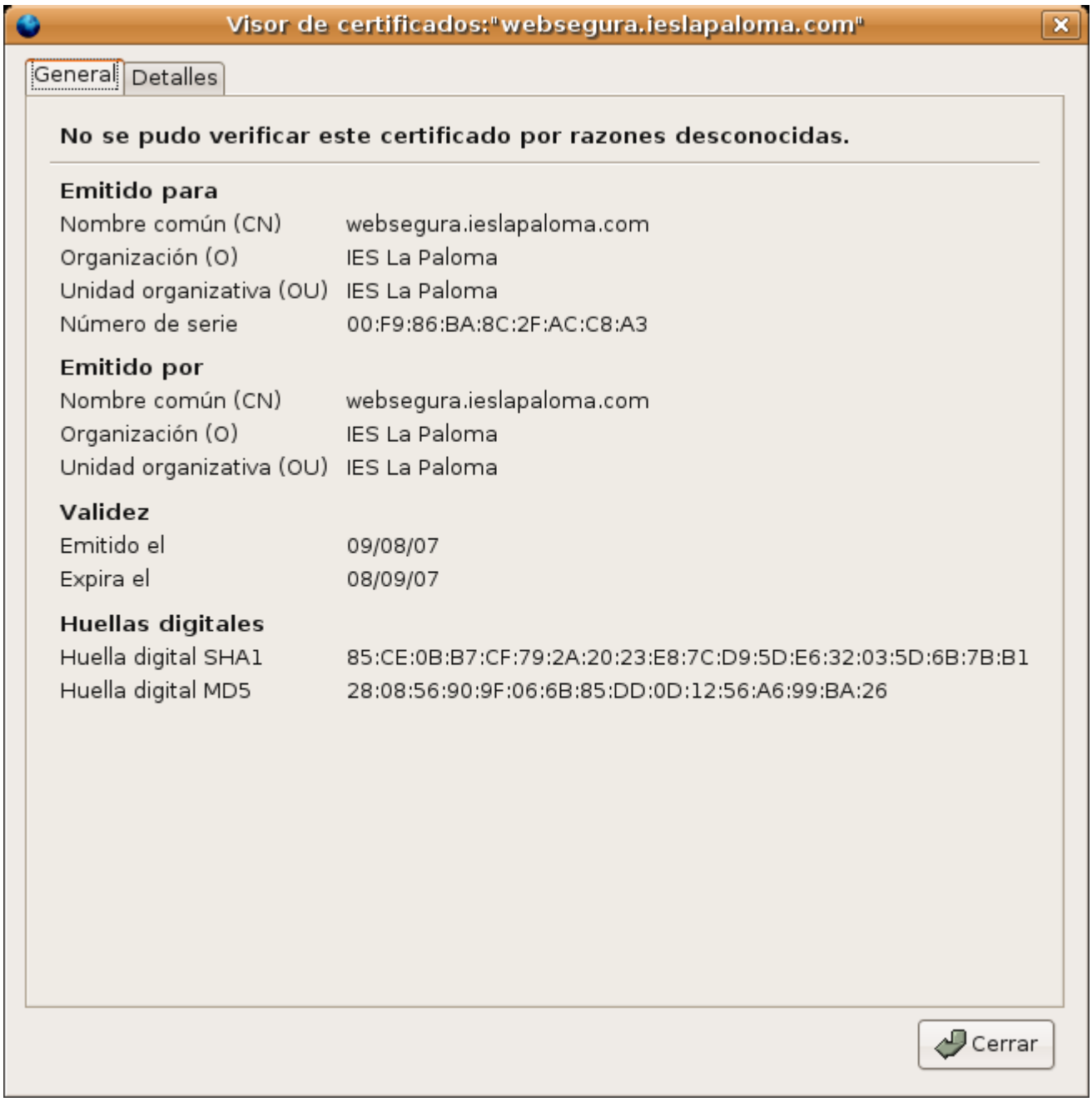
Probando el acceso a la página web segura

Nota: Si no tenemos un DNS funcionando, debemos incluir en /etc/hosts una línea para resolver localmente el nombre de nuestro servidor por su IP, ejemplo: 192.168.1.239 websegura.ieslapaloma.com, ya que en este caso, navegar con la dirección IP no funcionará.

Para acceder a las páginas seguras de nuestro servidor web, tecleamos desde el navegador 'https://websegura.ieslapaloma.com'. Lo primero que se muestra es la alerta de seguridad que nos indica que el certificado no está emitido por una CA en la que confiamos:

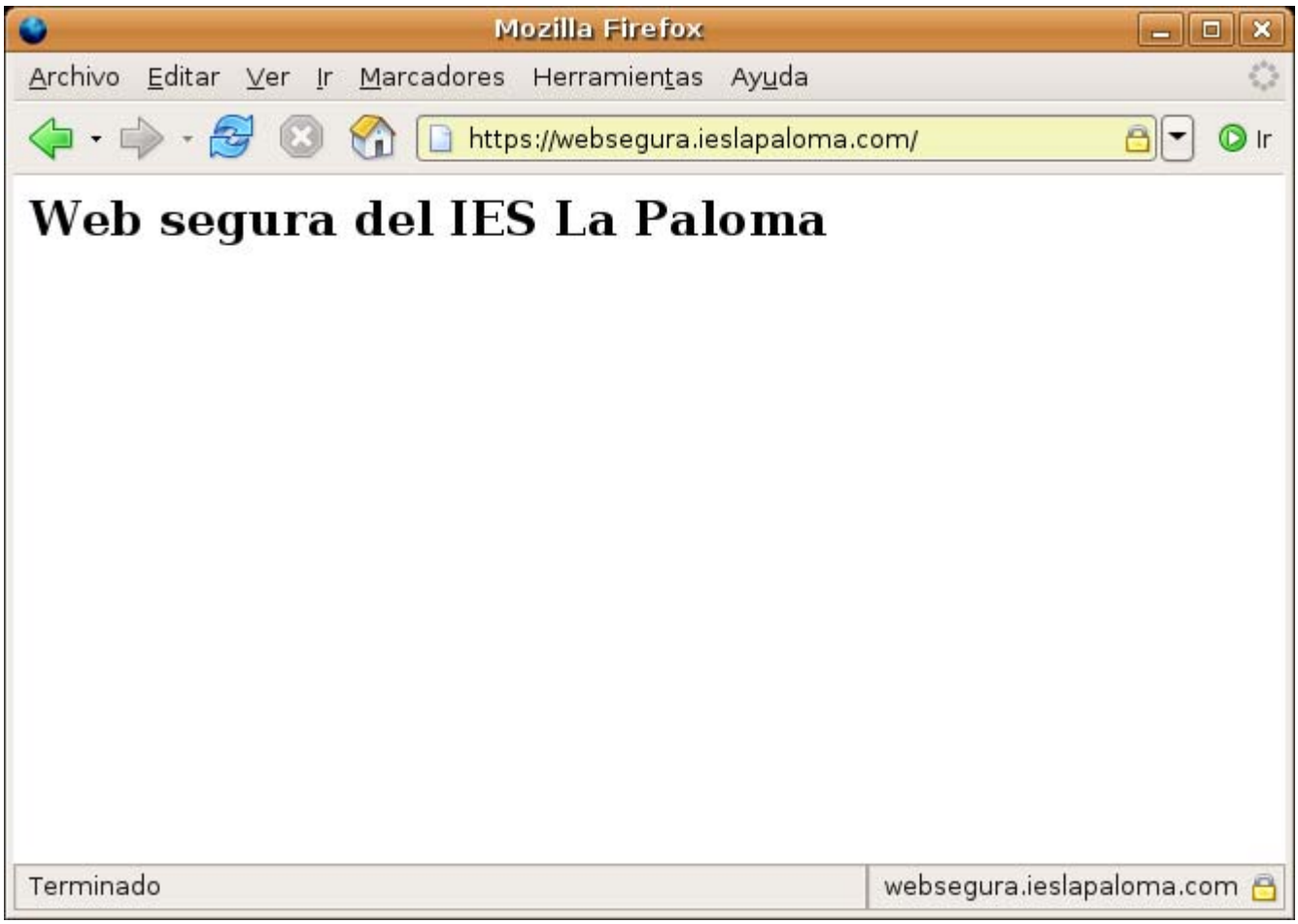


Si pulsamos sobre el botón 'Examinar certificado' veremos la información tanto del certificado como de la entidad certificadora que lo firma:

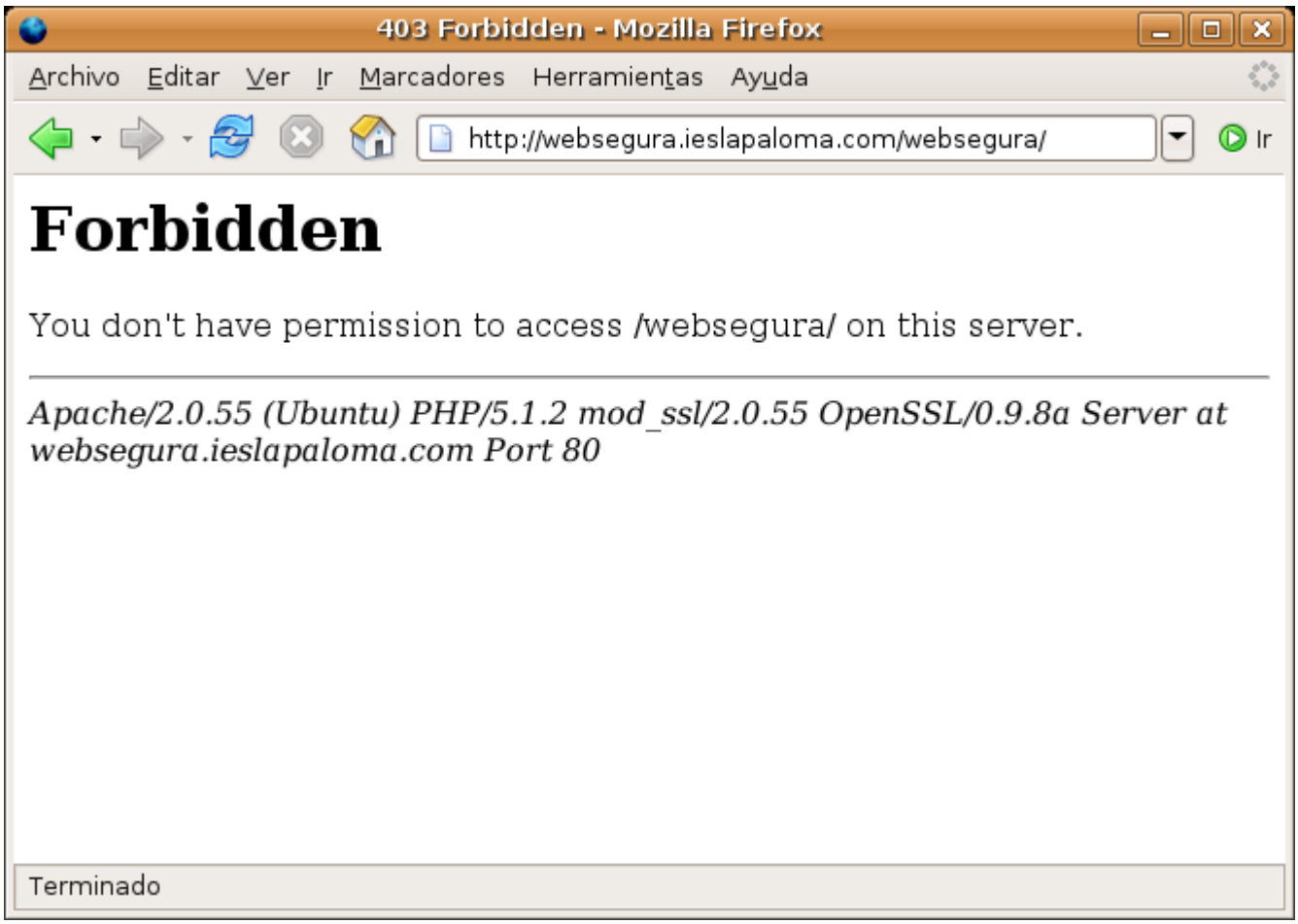


Si aceptamos el certificado significa que, a pesar de estar firmado por una entidad certificadora que no es de confianza para el navegador (lo hemos firmado nosotros mismos), lo aceptamos. Tendremos que indicar al navegador si aceptamos el certificado para siempre o solo para ahora. Como tenemos la seguridad de que el certificado es bueno porque acabamos de crearle nosotros mismos, podemos aceptarlo para siempre y así el navegador no volverá a preguntarnos más sobre él ya que hemos indicado manualmente que confiamos en este certificado:

Ahora ya tenemos acceso a la web segura mediante el protocolo https lo que nos garantiza que la información de la página segura, antes de salir del servidor, ha sido cifrada y por tanto la trasferencia de datos desde el servidor a nuestro navegador se ha producido de forma segura. Al llegar a nuestro navegador, se han descifrado los datos. El candado cerrado que aparece abajo a la derecha en el navegador, indica que la trasferencia de datos se ha realizado de forma segura.



Como sabemos la ruta de la carpeta segura, si intentamos acceder a la carpeta segura utilizando el protocolo http yendo con el navegador a 'http://www.ieslapaloma.com/websegura', apache denegará el acceso ya que en '/etc/apache2/sites-available/default' se ha especificado que la carpeta debe ser accedida mediante https:



Carpetas seguras de usuario

Si en el centro existiera la necesidad de que los profesores dispongan de una carpeta web segura donde poder colocar contenidos accesibles vía SSL, como serán casos excepcionales, una solución sencilla es crear una carpeta dentro de la carpeta '/var/www/websegura' para dicho profesor y para que éste tenga acceso de forma autónoma a subir contenidos a dicha carpeta, se le puede crear un usuario adicional cuyo home sea la carpeta correspondiente, ejemplo, para el profesor Javier podemos crear otro usuario llamado javier-s (javier-seguro) cuyo home sea /var/www/websegura/javier. Podría subir contenidos por ftp utilizando el usuario javier-s. El acceso a los contenidos desde un navegador sería yendo a la dirección https://websegura.ieslapaloma.com/javier

Este proceso habría que hacerlo para todos los profesores o departamentos de nuestro centro que requieran de carpeta segura.

Archivos log de apache

Por defecto, apache utiliza dos archivos de registro: access.log y error.log que están almacenados en la carpeta /var/log/apache2.

En el archivo **/var/log/apache2/access.log**, apache va registrando todos los accesos que los PCs hacen al servidor web y en cada línea de dicho archivo va almacenando la IP, la fecha y la hora, el comando HTTP enviado por el cliente, la url solicitada y la versión del navegador y el sistema operativo. Analizando este archivo podemos ver las veces que se ha descargado una página o un archivo, o las IPs más activas. Este archivo de registro es

utilizado por los programas que presentan estadísticas de acceso al servidor web como awstats.

En el archivo **/var/log/apache2/error.log**, apache registra todas las incidencias o errores que se van produciendo. Ejemplo, cuando un cliente solicita una página inexistente o cuando un cliente intenta entrar en una carpeta prohibida o protegida. Si estamos configurando algo en apache (carpetas privadas, carpetas seguras, servidores web virtuales, alias, etc...) y no funciona, una buena idea es hacer pruebas y examinar el archivo error.log ya que nos puede dar pistas para encontrar la solución a nuestro problema.

Acceso a carpetas privadas con autentificación por LDAP

Otra posibilidad muy interesante es que los profesores e incluso el sitio web de la Intranet de nuestro centro, puedan disponer de carpetas privadas accesibles mediante el navegador pero no por cualquier usuario; por ejemplo los profesores podrían disponer de una carpeta donde almacenar información confidencial accesible desde la web -notas, por ejemplo-. Así mismo puede ocurrir que queremos tener en el servidor web de nuestra intranet páginas a las que sólo puedan tener acceso de lectura los profesores del centro. Vamos a ver cómo conseguir todo esto.

Lo primero que hemos de tener en cuenta es que para que podamos autenticar a los usuarios en apache mediante LDAP, hemos de habilitar un módulo especial en nuestro servidor web para que apache pueda validar el acceso a las carpetas deseadas a través de la base de usuarios del servidor LDAP. Dicho módulo se habilita ejecutando el siguiente comando:

```
// Habilitar módulo de autentificación de apache con ldap
# a2enmod ldap
```

El siguiente paso es crear la carpeta "/var/www/webprivada", lugar donde ubicaremos las páginas privadas de nuestro servidor web. Dicha carpeta tendrá como grupo propietario el grupo profesores.

Posteriormente introducimos en /etc/apache2/sites-available/default textualmente las siguientes líneas, mediante las cuales logramos definir la carpeta "privada" como aquella a partir de la cual el contenido allí contenido será privado y sólo accesible por los profesores de nuestro centro y por el administrador.

```
// Carpeta privada con acceso a profesores. Añadir en /etc/apache2/sites-available/default
Alias /privada/ "/var/www/webprivada/"
<Directory "/var/www/webprivada">
        Options Indexes FollowSymLinks
        AllowOverride None
        Order allow,deny
        Allow from all
        AuthType basic
        AuthName "Identificacion LDAP ieslapaloma.com"
        AuthLDAPUrl ldap://ip-servidor-ldap:389/dc=ieslapaloma,dc=com?uid
        AuthLDAPBindDN "cn=admin,dc=ieslapaloma,dc=com"
        AuthLDAPBindPassword xxxxxx
        AuthLDAPGroupAttributeIsDN off
        AuthLDAPGroupAttribute memberUid
```

```
        require group cn=profesores,ou=groups,dc=ieslapaloma,dc=com
</Directory>
```

En el parámetro AuthLDAPUrl sustituiremos la cadena 'ip-servidor-ldap' por la dirección IP o el nombre del servidor LDAP y en el parámetro "AuthLDAPBindPassword" la cadena "xxxxxx" por la contraseña que hayamos asignado al usuario "administrador (admin) " del servidor LDAP.

En el parámetro AuthLDAPUrl vemos que al final termina con '?uid'. Significa que lo que debe de introducir el usuario es su uid (login del usuario). Podemos filtrar la entrada del usuario y poner condiciones si terminamos la url con '?uid??(atributo=valor)'. De ésta forma solamente serían válidos aquellos usuarios que tengan un atributo con un valor determinado, ejemplo '?uid??(gidNumber=1001)' solo admitiría usuarios cuyo grupo primario sea 1001.

El parámetro AuthLDAPGroupAttributeIsDN debe estar a off para que no utilice el cn (nombre común) del usuario sino el uid a la hora de comprobar la pertenencia a un grupo.

En el parámetro AuthLDAPGroupAttribute debemos indicar el campo que se analizará para comprobar la pertenencia a un grupo.

En el parámetro 'require', exigimos que pertenezca a un grupo. Otras opciones son 'require user' seguido de una lista de usuarios permitidos, ejemplo 'require user miguel joaquin jessica'. Para permitir a cualquier usuario que exista en el servidor LDAP, podemos usar 'requiere valid-user'.

Más información en: http://httpd.apache.org/docs/2.0/mod/mod_auth_ldap.html

Guardamos los cambios realizados y para completar el proceso reiniciaremos el servidor "apache"

```
// Reiniciar apache
# /etc/init.d/apache2 restart
```

Si ubicamos un fichero de nombre "prueba.html" en dicha carpeta ("/var/www/privada"), podremos acceder a ella mediante la URL "http://www.ieslapaloma.com/privada/prueba.html", mostrándose la siguiente pantalla en la cual se nos pedirá autenticación, y en la cual serán válidas las credenciales de algún profesor.

Apache+PHP+MySQL+PHPMyAdmin

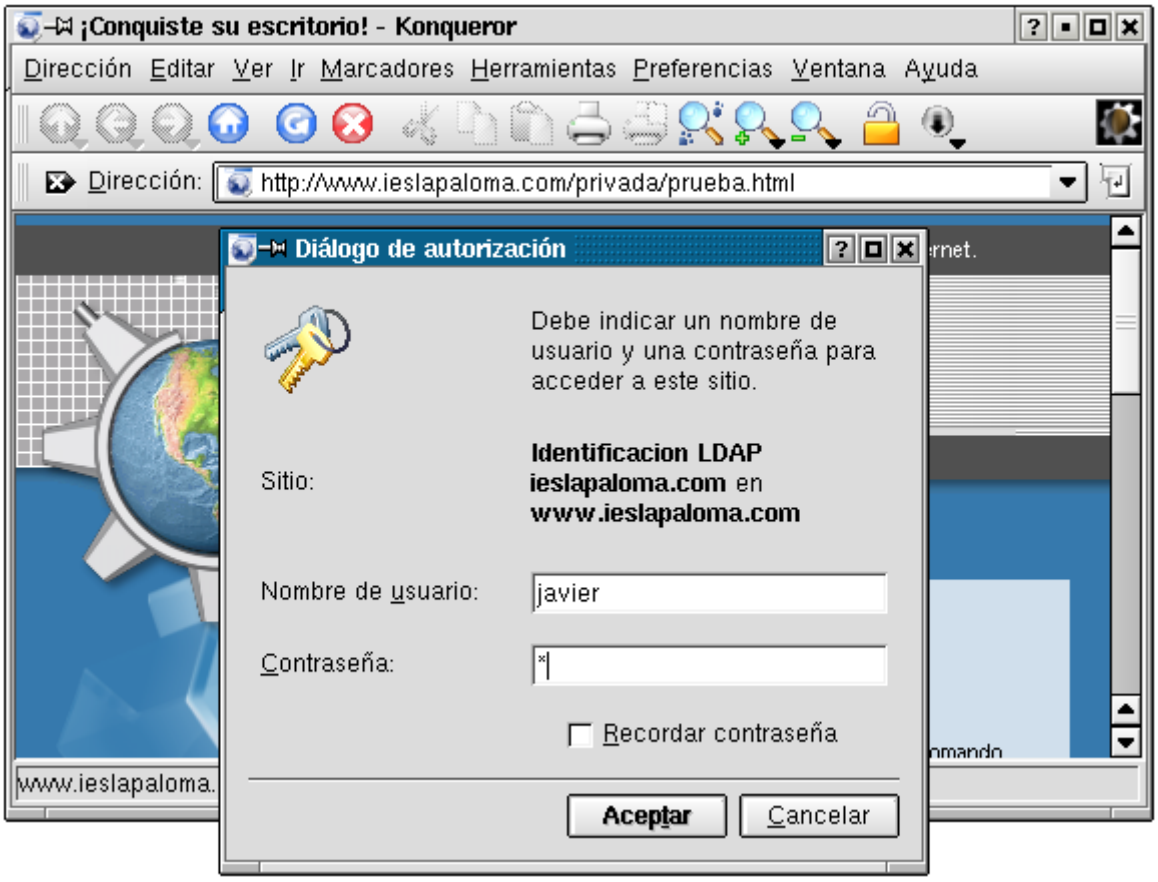
Para poder aprovechar al máximo las características del servidor web apache, es muy conveniente que pueda ejecutar scripts en servidor y pueda acceder a bases de datos.

Las aplicaciones web más interesantes como los gestores de contenidos para crear y mantener sitios web dinámicos, wikis, blogs, foros-web, repositorios de arhivos, etc... requieren de lenguaje script en servidor y sistema gestor de bases de datos.

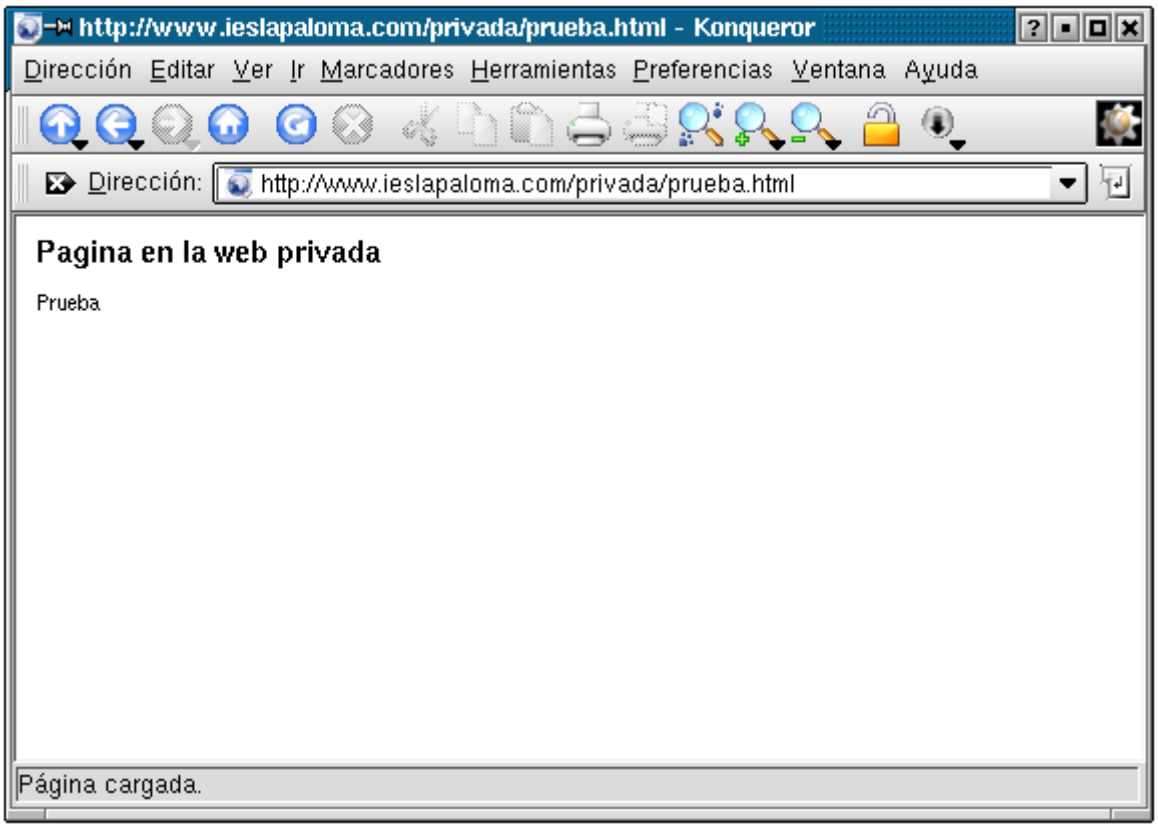
En el desarrollo web del mundo Linux el lenguaje script en servidor más utilizado es el lenguaje php y el sistema gestor de bases de datos más utilizado es mysql. Phpmyadmin es una excelente herramienta para administrar bases de datos mysql.

Más información sobre cómo instalar y configurar php, mysql y phpmyadmin en:

- Instalacion_y_configuracion_de_PHP
- Instalacion_y_configuracion_de_MySQL
- Instalacion_y_configuracion_de_PHPMyAdmin



Una vez validado adecuadamente algún usuario con permisos de acceso a los contenidos privados se mostrará la página solicitada.



Además podemos crear una carpeta privada para cada profesor, de modo que el contenido allí existente sólo fuera accesible por él mismo previa autenticación; para ello crearemos una carpeta de nombre 'privada' colgando de la carpeta personal de cada profesor (por ejemplo en el caso del profesor Javier, en '/home/javier/public-html/'). Además de la creación de dicha carpeta 'privada' en la ruta correspondiente, hemos de editar el fichero /etc/apache2/sites-available/default e incluir la siguiente entrada en el apartado correspondiente a los directorios:

```
// Carpeta privada de javier. Añadir en /etc/apache2/sites-available/default
Alias javier-p "/home/javier/public_html/privada"
<Directory "/home/javier/public_html/privada">
        Options Indexes FollowSymLinks
        AllowOverride None
        Order allow,deny
        Allow from all
        AuthType basic
        AuthName "Identificacion LDAP ieslapaloma.com"
        AuthLDAPUrl ldap://ip-servidor-ldap:389/dc=ieslapaloma,dc=com?uid
        AuthLDAPBindDN "cn=admin,dc=ieslapaloma,dc=com"
        AuthLDAPBindPassword xxxxxx
        require user javier
</Directory>
```

Igual que antes, sustituiremos las cadenas 'ip-servidor-ldap' y "xxxxxx" por sus valores correctos. Además hemos de introducir esta entrada para cada uno de los profesores del centro, sustituyendo en las rutas de las dos primeras líneas el valor "javier" por el del profesor que deseamos que tenga el acceso seguro, así como dicho valor también en la penúltima línea.

Tras almacenar los cambios en el fichero de configuración y reiniciar el servicio apache, para acceder a un fichero de nombre "prueba.html" ubicado en la carpeta privada del profesor Javier teclearemos la URL:
'http://www.ieslapaloma.com/~javier/privada/prueba.html'

Es posible hacer, y de hecho es recomendable, que las carpetas privadas sean además seguras, es decir, utilicen un canal SSL, con lo cual el acceso a las carpetas seguras sería 'https' en el puerto '443', el resto de las rutas de las URL de acceso se mantendrían estables. Para lograrlo hemos de introducir en cada una de las entradas '<Directory>' la instrucción 'SSLRequireSSL'.