UNIVERSIDAD DE GRANADA

Tecnologías Web

3º Grado en Ingeniería Informática

Guión de prácticas
Instalación de un Servidor Web

Instalación y configuración de un servidor web Apache con PHP y de un servidor MySQL

© Prof. Javier Martínez Baena
Dpto. Ciencias de la Computación e I. A.
Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial

Instalación de un Servidor Web

1. Objetivo

El objetivo de la práctica es instalar un servidor Web en una máquina con Sistema Operativo GNU/Linux. Para esta práctica el alumno puede llevar su propio ordenador o bien utilizar un PC del aula de prácticas. El alumno también podrá instalar el software en otros Sistemas Operativos aunque en ese caso deberá buscar los pasos equivalentes a los explicados en el guión. En cualquier caso se recomienda que el alumno se familiarice con el S.O. GNU/Linux ya que será el que utiliza el host en el que se van a hacer las entregas de prácticas de toda la asignatura.

2. Aula de prácticas

En caso de que se realice la instalación en el aula de prácticas, deberá usar el siguiente código:

```
twebinstala
```

Se cargará una máquina virtual (Virtual Box) con la distribución LUbuntu 16.04.3 de GNU/Linux. Es una distribución básica que permitirá hacer una instalación limpia del servidor. El usuario existente en dicho sistema es:

Usuario: `tweb`
Clave: `tweb`

Recuerde que para hacer tareas de administración deberá acceder con privilegios de `root`. Lo más cómodo será abrir una shell con privilegios de administrador mediante el siguiente comando:

```
sudo bash
```

La máquina virtual tiene redirigidos los siguientes puertos:

Servicio	Guest	Host
HTTP	80	8080
HTTPS	443	4443
SSH	22	2222

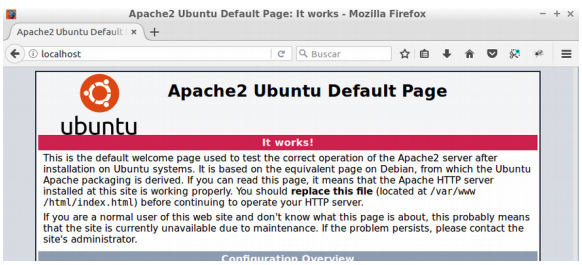
Esta redirección permitirá acceder desde otros ordenadores al servidor web instalado en cada ordenador del aula, de esta forma los alumnos puedan conectarse a los servidores de sus compañeros para poder probarlos.

Las direcciones IP de los PC de las aulas de prácticas son de la forma 172.18.XXX.YYY donde XXX.YYY se puede consultar en la pegatina que hay en cada torre.

Por ejemplo, para poder acceder a una página alojada en el PC 172.18.142.18, deberá escribir: http://172.18.142.18:8080. La petición llegará al PC y este, a su vez, la redirigirá al puerto 80 de la máquina virtual, en donde supuestamente debería estar funcionando el servidor Apache.

3. Instalación del servidor web Apache

Dependiendo del S.O. y de la distribución que se esté usando, podría variar el nombre de este paquete. En la página web https://httpd.apache.org dispone de los fuentes del servidor y de enlaces de descarga para distintas plataformas. En el caso de Ubuntu, el paquete que contiene el servidor se llama apache2 y se puede instalar desde los repositorios oficiales:




```
apt-get install apache2
```

Una vez instalado se puede probar que funciona accediendo a la dirección http://localhost. Si la instalación ha sido correcta verá una página como la de la figura.

3.1. Configuración

El servidor hace uso de dos directorios en el sistema: uno para alojar las páginas web y otro ara mantener su configuración global. Las páginas se alojan, por defecto, en `/var/www/html` y la configuración está en `/etc/apache2`.

Dentro de la carpeta de configuración también pueden variar los nombres de los ficheros o los subdirectorios en función de la distribución. En Ubuntu los principales archivos se organizan así:

```
etc/apache2/
├── apache2.conf        Fichero principal de configuración
│
├── ports.conf          Configuración de los puertos que escucha el servidor
│
├── mods-available      Configuraciones de módulos disponibles
├── mods-enabled        Configuración de los módulos de apache activos
│   ├── *.load
│   └── *.conf
│
├── conf-available      Ficheros de configuración disponibles
├── conf-enabled        Ficheros de configuración activos
│   └── *.conf
│
├── sites-available     Configuraciones de hosts virtuales disponibles
└── sites-enabled       Configuraciones de hosts virtuales activos
    └── *.conf
```

Los sistemas Debian (Ubuntu está basado en Debian) fragmentan la configuración del servidor el múltiples ficheros para facilitar las tareas de administración. En otros sistemas podría encontrar un único fichero con todos los parámetros de configuración (se suele llamar `httpd.conf`). A continuación se comentan algunos aspectos de la configuración.

Se puede ver que existen dos ficheros globales de configuración:

- `apache2.conf`. Este fichero es el que carga el servidor web cuando se inicia, en él se encuentran algunas directivas globales y otras para cargar el resto de ficheros de configuración.
- `ports.conf`. Este fichero contiene algunas directivas para indicarle al servidor qué puertos debe escuchar. Normalmente escuchará el puerto 80 para las peticiones HTTP sin cifrado y el 443 si está habilitado el protocolo HTTPS.

y tres grupos de directorios con diversos ficheros de configuración organizados como:

- `conf-`XXX (configuración general). Siguen siendo parámetros globales de la configuración. Sin embargo, con esta organización, se pueden mantener distintos aspectos de la configuración en distintos ficheros para facilitar la administración del servidor.
- `mods-`XXX (configuración de módulos). Ficheros de configuración de módulos del servidor web instalados en el sistema.
- `sites-`XXX (configuración de hosts virtuales). Le permite al servidor comportarse como si hubiese instalados varios servidores web en una misma máquina. Esto puede ocurrir cuando una misma IP tiene asignados varios nombres de dominio y queremos que las páginas servidas sean diferentes según el nombre de dominio. Otro caso habitual de uso es disponer de distintos servicios web dependiendo del puerto por el que accedemos al servidor (por defecto 80).

Cada uno de esos grupos de directorios está formado por dos directorios:

- XXX`-available`. Ficheros de configuración disponibles en el sistema pero que no tienen porqué estar en uso. Estos ficheros no se usan al iniciar el servidor web.
- XXX`-enabled`. Enlaces simbólicos a los ficheros en XXX`-available` que sí están activos. Estos ficheros son cargados al iniciar el servidor web.

3.1.1. Configuración principal

El fichero de configuración principal es `/etc/apache2/apache2.conf`. En él se definen algunos parámetros y se hace la inclusión del resto de ficheros de configuración. El propio fichero incluye instrucciones y documentación detallada. Se describen a continuación algunos parámetros:

Parámetro	**Descripción**
`ServerRoot`	Directorio en el que se almacenan: los ficheros de configuración, los ficheros con mensajes de error y los log del servidor. Por defecto vale `/etc/apache2`
`Timeout`	Número de segundos para agotar tiempo de espera de peticiones
`KeepAlive`	Permitir o no conexiones persistentes (varias peticiones en una misma conexión)
`AccessFileName`	Nombre del fichero que buscará en cada directorio por si hay directrices específicas. Por defecto vale `.htaccess`

Es posible definir directivas del servidor para directorios concretos. La forma de uso es:

```
<Directory ...>
    ...
</Directory>
```

y disponemos de diversas opciones. Se indican a continuación algunas habituales:

Parámetro	**Descripción**
`DirectoryIndex <F1 F2 ...>`	Al acceder a un directorio, se busca alguno de los ficheros F1, F2, ... en ese orden. Si existe se sirve como resultado de le petición.

`Options Indexes`
`Options +Indexes`
`Options -Indexes`
	Al solicitar un directorio, en caso de no haber indicado nada en `DirectoryIndex`, se muestra un listado de ficheros
	Si se indica `-Indexes` no muestra el listado

`Options FollowSymLinks`
`Options +FollowSymLinks`
`Options -FollowSymLinks`
	Permite seguir enlaces simbólicos. Necesario si se hace enlace a algún directorio de nuestro home para ser servido a través de `/var/www/html`
	Si se indica `-FollowSymLinks` no sigue los enlaces

`AllowOverride None`
`AllowOverride All`
`AllowOverride <DIR>`
	Indica si se puede o no ignorar alguna opción en caso de que exista un fichero `.htaccess`
	None: Se ignora por completo el fichero `.htaccess`
	`All`: Se permiten todas las directivas en `.htaccess`
	`<DIR>`: se pueden permitir algunas concretas

Ejemplo:

```
# Denegar acceso al sistema de ficheros raíz.
# Denegar el uso de ficheros .htaccess
<Directory />
       Options FollowSymLinks
       AllowOverride None
       Require all denied
</Directory>

# Permitir al servidor el acceso al directorio en donde se alojan las páginas
# Permitir el uso de ficheros .htaccess
# Mostrar listados de directorios
<Directory /var/www/>
       Options +Indexes +FollowSymLinks
       AllowOverride All
```

```
        Require all granted
    </Directory>
```

Otra directiva que se usa con cierta frecuencia es la que permite establecer alias de directorios:

Parámetro	Descripción
`Alias "alias" "ruta"`	Con esta directiva al acceder a http://localhost/alias, el servidor accede a http://localhost/ruta

Situaciones en las que es habitual su uso:

- Simplificar el acceso a URL muy largas. Supongamos que en un sitio web tenemos la siguiente URL:
  ```
  http://miservidor.com/listados/gestion/profesores/infraestructuras/index.html
  ```
  Si incluimos la directiva:
  ```
  Alias "/prof" "/listados/gestion/profesores/infraestructuras"
  ```
  ahora usaríamos la URL:
  ```
  http://miservidor.com/prof
  ```
- Acceso a directorios que están fuera del sistema de ficheros accesible por el servidor web. Supongamos que tenemos un directorio con imágenes fuera de /var/www/html
  ```
  Alias "/image" "/ftp/pub/image"
  <Directory "/ftp/pub/image">
      Require all granted
  </Directory>
  ```
  Observe que hemos de dar permiso al servidor para acceder al directorio.

3.1.2. Resto de ficheros de configuración global

Estos ficheros se encuentran en los directorios `conf-enabled` y `conf-available`. Se recomienda siempre hacer las modificaciones sobre los ficheros disponibles en `conf-available` para, a continuación, activar los cambios si es necesario.

Activando y desactivando ficheros de configuración

Dispone de un comando `a2enconf` y de otro `a2disconf` para activar o desactivar algún fichero de configuración concreto de entre los disponibles. Estos comandos lo que hacen es, básicamente, crear o quitar el enlace entre las carpetas `conf-available` y `conf-enabled`.

Por ejemplo, podríamos haber creado un fichero de configuración personal en el fichero `/etc/apache2/conf-available/miconfig.conf`. Para activarlo ejecutaríamos:

```
a2enconf miconfig.conf
```

y para deshabilitarlo ejecutaríamos:

```
a2disconf miconfig.conf
```

Para que los cambios tengan efecto ha de reiniciarse el servidor (se verá más adelante).

3.1.3. Configuración de hosts virtuales

Estos ficheros se encuentran en los directorios `sites-enabled` y `sites-available`. Se recomienda siempre hacer las modificaciones sobre los ficheros disponibles en `sites-available` para, a continuación, activar los cambios si es necesario.

Aquí se definen parámetros para configurar distintos hosts virtuales en un mismo servidor web. Los casos habituales de uso son:

- Disponemos de varios nombres de dominio asociados a una misma dirección IP y queremos que nuestro servidor web se comporte de forma distinta en función del nombre de dominio por el que el usuario ha llegado a él. Se definiría un host virtual para cada nombre de dominio.

- Deseamos que, dependiendo del puerto al que se conecta el usuario, el servidor web se comporte de forma diferente. El puerto por defecto para escuchar conexiones HTTP es el 80 pero podríamos hacer que nuestro servidor web respondiese también ante conexiones a otros puertos (ver fichero `ports.conf`). Al disponer de diferentes puertos de entrada al servidor web, podemos hacer que se comporte como si hubiese un servidor web distinto en cada uno de ellos.

El fichero `000-default.conf` contiene la configuración del host por defecto. La configuración inicial define un host virtual para cualquier nombre de dominio asociado a la máquina y para el puerto 80.

Por defecto también se instala otro fichero (`default-ssl.conf`) para configurar un host virtual que atiende al puerto 443 (conexiones HTTPS).

Activando y desactivando ficheros de configuración

Existen los comandos `a2ensite` y `a2dissite` para habilitar o deshabilitar un host virtual y funcionan de forma análoga a `a2enconf` y `a2disconf`.

3.1.4. Configuración de módulos

Estos ficheros se encuentran en los directorios `mods-enabled` y `mods-available`. Se recomienda siempre hacer las modificaciones sobre los ficheros disponibles en `mods-available` para, a continuación, activar los cambios si es necesario.

Configuración de los distintos módulos instalados en el servidor, como por ejemplo:

`dir.conf`	Secuencia de búsqueda de ficheros cuando se solitica una carpeta.
`userdir.conf`	Configuración de directorios `public_html` de los usuarios del sistema.

Activando y desactivando ficheros de configuración

Existen los comandos `a2enmod` y `a2dismod` para habilitar o deshabilitar un módulo y funcionan de forma análoga a `a2enconf` y `a2disconf`.

3.2. Iniciar y detener el servidor web

Un S.O. de tipo Unix puede encontrarse en diferentes niveles de ejecución (runlevel):

- Nivel de ejecución 0: Apagado.
- Nivel de ejecución 1: Monousuario (sólo usuario root; no es necesaria la contraseña). Se suele usar para analizar y reparar problemas.
- Nivel de ejecución 2: Multiusuario sin soporte de red.
- Nivel de ejecución 3: Multiusuario con soporte de red.
- Nivel de ejecución 4: Como el runlevel 3, pero no se suele usar.
- Nivel de ejecución 5: Multiusuario en modo gráfico (X Windows).
- Nivel de ejecución 6: Reinicio.

Dependiendo del runlevel, se ejecutan unos servicios u otros durante el arranque de la máquina. Habitualmente un PC de escritorio levanta un servidor gráfico por lo que su runlevel se establece en 5. Un servidor no necesita levantar entorno gráfico y su runlevel puede ser 3.

En la carpeta `/etc` verá que existen varias carpetas nombradas `rcX.d` en las que se incluyen los scripts que permiten iniciar los distintos servicios de la máquina dependiendo del nivel de ejecución en el que se encuentre. Normalmente, puesto que el mismo servicio puede ser levantado en distintos niveles de ejecución, y para evitar duplicar dicho script múltiples veces, el contenido de las carpetas es un enlace al script, que se encuentra realmente en `/etc/init.d`.

En esta práctica no debe hacer nada relacionado con este subsistema dado que la instalación del software del servidor Apache ya lo ha configurado adecuadamente, sin embargo, es posible que cuando tenga que hacer una instalación de un servidor en producción sí tenga que tener en cuenta aspectos relativos a la forma en que se inicia el servidor durante el arranque de la máquina.

Puede usar los siguientes comandos para iniciar, detener o ver el estado de un servicio desde un

terminal. En nuestro caso, el servicio se llama apache2:

```
service apache2 status      Muestra el estado del servicio
service apache2 stop        Detiene el servicio
service apache2 start       Inicia el servicio
service apache2 restart     Reinicia el servicio (lo detiene y lo inicia)
```

3.2.1. El sistema systemd

"service" en realidad es un wrapper para facilitar a los administradores las tareas habituales para levantar y tirar servicios. Este sistema permite trabajar de forma transparente con distintos entornos para la gestión de servicios. La mayor parte de los sistemas GNU/Linux actuales usan systemd, que es un conjunto de demonios, bibliotecas y herramientas para la gestión de servicios.

La forma de trabajar con systemd, en lugar de con service, para las tareas indicadas antes es la siguiente:

```
systemctl status apache2      Muestra el estado del servicio
systemctl stop apache2        Detiene el servicio
systemctl start apache2       Inicia el servicio
systemctl restart apache2     Reinicia el servicio (lo detiene y lo inicia)
```

3.3. Abrir puertos en el servidor

Es muy habitual que el ordenador que ejecuta el servidor web esté protegido frente a intrusiones con un firewall. Recuerde que debe abrir los puertos en los que está escuchando el servidor Apache. Por defecto esos puertos son el 80 (HTTP) y el 443 (HTTPS).

En Ubuntu puede usar el comando ufw (que es un front-end para iptables) para abrir o cerrar puertos así como habilitar o deshabilitar el firewall:

```
ufw disable                 Deshabilita el cortafuegos por completo
ufw enable                  Habilita el cortafuegos
ufw status                  Muestra el estado del cortafuegos
ufw status numbered         Muestra el estado del cortafuegos numerando las reglas
ufw allow http              Habilita las conexiones al puerto del servicio http
ufw allow 80                Habilita las conexiones al puerto 80
ufw deny http               Deniega conexiones al servicio http (silencioso)
ufw reject http             Deniega conexiones al servicio http (devuelve error)
ufw delete <nºregla>        Borra una regla del firewall
```

Pruebe a abrir y cerrar los puertos y acceda desde otras máquinas para ver el comportamiento. Recuerde que en las máquinas virtuales del aula de prácticas los puertos están redirigidos.

3.4. Ficheros de log

En `/var/log/apache2` se encuentran los ficheros de log del servidor Apache. Son ficheros de texto en los que el servidor Apache va anotando todo lo que hace (esencialmente las peticiones de páginas que recibe el servidor y la respuesta que este da).

Ejecute la siguiente orden en un terminal y a continuación acceda al servidor desde un navegador para ver los datos que se registran:

```
tail -f /var/log/apache2/access.log
```

4. Instalación de PHP

El servidor que se ha instalado permite servir páginas web (ficheros `.html`). Si se necesita que el servidor pueda ejecutar aplicaciones PHP, es necesario tener instalado, además:

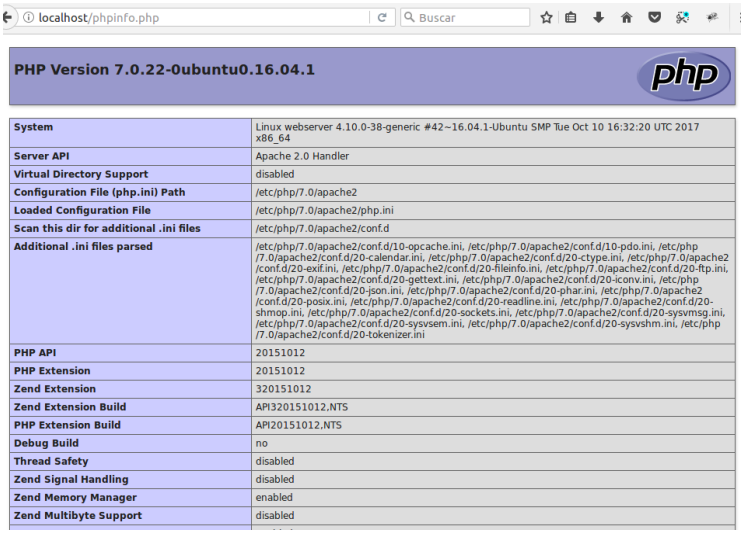
- Un intérprete PHP.
- El módulo de Apache para que este pueda usar el intérprete PHP.

Para comprobar si ya dispone de este software haga lo siguiente:

1. Cree un fichero llamado `/var/www/html/phpinfo.php` con el contenido:

```
<?php phpinfo(); ?>
```

2. Abra un navegador y pruebe a acceder a la URL `http://localhost/phpinfo.php`



Si ve una página en blanco se debe a que no tiene instalado el software para ejecutar scripts PHP. En este caso si mira el código fuente de la página podrá ver el contenido del fichero `phpinfo.php`.

En caso de que tenga instalado el módulo de PHP debería ver una página similar a la de la figura, en donde el resultado de la ejecución de `phpinfo.php` muestra distintos aspectos de la configuración del módulo de PHP.

De nuevo, dependiendo del S.O. que tengamos, el proceso de instalación puede variar. En Ubuntu este software está disponible en los repositorios oficiales:

```
apt-get install php
apt-get install libapache2-mod-php
```

Una vez instalados los paquetes, ya deberíamos poder ejecutar scripts PHP desde Apache.

4.1. Configuración del módulo de PHP

En Ubuntu, la configuración de este módulo está en `/etc/php/VERSION/apache2/php.ini` (donde `VERSION` dependerá de la versión del intérprete instalada, en nuestro caso es la `7.0`). El fichero está autodocumentado por lo que es muy sencillo hacer cambios.

Algunos parámetros que pueden resultar de interés:

- Controlar el tipo de mensajes de error o avisos que se muestran al ajecutar código PHP desde el navegador. Dependiendo de si estamos en un servidor de desarrollo o de un servidor en producción deberemos tener una configuración u otra.
  Variables relevantes: `error_reporting, display_errors, display_startup_errors, log_errors`
- Tiempo máximo de ejecución permitido para un script.
  Variables: `max_execution_time, max_input_time`
- Limitación de la memoria que puede consumir un script.
  Variables: `memory_limit`
- Si se permite o no subir ficheros desde un navegador.
  Variables: `file_uploads, upload_tmp_dir, max_file_uploads`
- Tamaño máximo de los ficheros que se pueden subir desde el navegador.
  Variables: `upload_max_filesize`
- Cantidad máxima de información que se puede transferir en un POST.
  Variables: `post_max_size`.
- Parámetros para configurar módulos de PHP (acceso a BBDD, ...)
- ...

En este curso es importante tener activa la visualización de todo tipo de mensajes de error que puedan ser de ayuda durante el proceso de desarrollo de software.

Ejercicio: activar mensajes de error

1. Copie el fichero que ha creado antes (`phpinfo.php`) en otro llamado `phperror.php` y modifíquelo para que la sintaxis del código PHP sea incorrecta.
2. Edite el fichero de configuración de PHP para que no muestre ningún tipo de mensaje de error (suponga que esta es la configuración básica de un servidor en producción).

3. Cargue el fichero `phperror.php` desde un navegador y mire el resultado.
4. Edite de nuevo el fichero de configuración de PHP para que muestre todo tipo de mensajes de error (servidor de desarrollo).
5. Cargue de nuevo el fichero `phperror.php` y observe el comportamiento.

5. Ejercicio: configuración básica de Apache

Lo primero que debería hacer después de la instalación de Apache es configurarlo para adecuarlo a su entorno.

En este ejercicio debe crear esta jerarquía de páginas en el servidor (el contenido de cada fichero lo puede consultar en el apéndice de este documento):

```
├── info
│   ├── index.html      (este es el fichero index.html de Apache)
│   └── phpinfo.php
├── privado
│   ├── claves.html
│   └── nuevasclaves.php
└── publico
    ├── index.html
    ├── index.php
    ├── protegido
    │   ├── claves.html
    │   └── nuevasclaves.php
    └── usuarios.html
```

En donde (ejemplos de ficheros los puede consultar en el Apéndice):

- `info/` En este directorio se han puesto los ficheros
  - `index.html` Este es el fichero de ejemplo que viene instalado con Apache, que se ha movido aquí
  - `phpinfo.php` Este es el fichero que se ha usado en una sección previa para comprobar el funcionamiento de PHP
- `privado/` En este directorio se van a poner ficheros que no deseamos que sean accesibles desde el exterior, es decir, que aunque los ficheros estén dentro de la jerarquía de directorios de `/var/www/html`, el servidor no los servirá en ningún caso.
  - `claves.html` Este fichero simula mostrar las claves de los usuarios del sistema
  - `nuevasclaves.php` Este fichero simula generar nuevas claves para los usuarios del sistema
- `publico/` En este directorio se alojan ficheros para servir desde el servidor web accesibles por cualquier usuario
  - `index.html` Un fichero de ejemplo HTML
  - `index.php` Un fichero de ejemplo PHP
  - `usuarios.html` Un fichero de ejemplo HTML que simula listar los usuarios del sistema
  - `protegido/` Este directorio, aunque es accesible desde el exterior, estará protegido por usuario y clave de forma que solo algunos usuarios puedan acceder a él.
    - `claves.html` (Mismo fichero que en `privado/`)
    - `nuevasclaves.php` (Mismo fichero que en `privado/`)

Ejercicio: Diferencia de comportamiento

Acceda a las URL que se listan a continuación

1. `http://localhost`
2. `http://localhost/publico`
3. `http://localhost/publico/protegido`

Puede comprobar que en algún caso se muestra una página web y en otro un listado de ficheros y directorios. ¿A qué se debe la diferencia de comportamiento?

Ejercicio: Desactivar listado de directorios

Cuando decida personalizar la configuración de un servidor Apache, puede optar por dos métodos. El primero es modificar alguno de los ficheros de configuración ya existentes y el segundo es crear un nuevo fichero de configuración con sus preferencias. Se recomienda, en la medida de lo posible, crear un fichero de configuración propia para, de esta forma, facilitar el mantenimiento posterior. Será más fácil localizar sus opciones personalizadas o cambiar por completo el comportamiento del servidor de una forma sencilla.

Para este ejercicio cree un fichero llamado `/etc/apache2/conf-available/miconfig.conf` y, a continuación edítelo para realizar el ejercicio propuesto a continuación.

Si accede a `http://localhost/publico/protegido` podrá ver un listado con los ficheros incluidos en el directorio. Modifique la configuración para impedir que se muestren listados en ese directorio del servidor. Es habitual impedir que se muestren los listados ya que los usuarios de los sitios web normalmente navegan a través de las páginas web y no a través del sistema de ficheros del servidor. Consulte la sección 3.1.1 si tiene dudas.

Si ha funcionado bien debería obtener un error 403-Forbidden al acceder a la URL en lugar del listado del directorio.

Si no ha funcionado puede deberse a que:

- No ha usado las opciones adecuadas. Solución: busque otras (consulte los ejemplos de la sección 3.1.1)
- La sintaxis es errónea. Solución: escriba bien
- No ha reiniciado el servidor web. Solución: consulte la sección 3.2 y reinicie el servidor web
- No está activo el fichero de configuración. Solución: consulte la sección 3.1.2 y active su fichero de configuración personalizada.

Observe que cada vez que haga alguna modificación de la configuración debe reiniciar el servidor web para que la cargue.

Ejercicio: Cambiar documento mostrado por defecto

Si accede a `http://localhost/publico`, el navegador le mostrará el documento `index.html`. Modifique la configuración para que, en lugar de ese documento, se muestre el documento `usuarios.html`. Consulte la sección 3.1.1 si tiene dudas.

Ejercicio: Cree un alias de un directorio

Añada un alias a la configuración para que cuando se acceda a la URL `http://localhost/pp` se muestre el contenido de `http://localhost/publico/protegido`. Consulte la sección 3.1.1 si tiene dudas.

Ejercicio: Restaure la configuración inicial temporalmente

Anule la configuración que ha estado creando y restaure la configuración inicial. Use los comandos vistos en la sección 3.1.2. Vuelva a activarla a continuación.

5.1. Protección de directorios

Los directorios que se están sirviendo pueden protegerse para permitir el acceso desde determinadas direcciones IP o nombres de dominio o para que puedan acceder solo determinados usuarios. Para ello se utiliza la directiva `Require`:

`Require all granted`	Permite el acceso desde cualquier lugar y a cualquier usuario
`Require all denied`	Deniega el acceso
`Require ip <dirección>`	Permite el acceso solo desde determinadas direcciones IP (separadas por espacios en blanco)
`Require not ip <dirección>`	Deniega el acceso desde determinadas direcciones IP

`Require host <nombre>`	Permite el acceso solo desde determinados nombres de dominio (separados por espacios en blanco)
`Require not host <nombre>`	Deniega el acceso desde determinados nombres de dominio

Ejercicio: Proteger el directorio privado/

Continue con la configuración del servidor e impida que los usuarios tengan acceso al directorio `privado/` evitando así que las claves queden expuestas o que se puedan modificar por usuarios no autorizados.

Para comprobar el resultado cargue de nuevo la URL `http://localhost/privado` y compruebe si ha funcionado.

Si ha funcionado debería obtener un error 403-Forbidden al acceder a la URL

5.1.1. Protección con usuario y clave

Para proteger un directorio con usuario y clave, primero debe crear un fichero de usuarios y claves. Este fichero se debe almacenar fuera de la estructura de directorios que exporta el servidor web por motivos de seguridad. Para ello use el comando `htpasswd`:

```
htpasswd -c /var/www/claves.ht usuario
```

Sea cuidadoso porque si el fichero `claves.ht` ya existía, con la opción `-c` lo borra y crea uno nuevo. Si lo que desea es añadir nuevos usuarios a ese fichero ejecute:

```
htpasswd /var/www/claves.ht otrousuario
```

Para borrar un usuario de ese fichero use la opción `-D`:

```
htpasswd -D /var/www/claves.ht usuario
```

La configuración de acceso por clave también puede hacerse a nivel global en el servidor. Para proteger una carpeta, añada a su directiva `Directory` el siguiente contenido:

```
AuthType Basic
AuthName "Acceso restringido"
AuthUserFile "/var/www/claves.ht"
Require user usuario
```

Con ello se da permiso únicamente al usuario indicado. Puede incluir múltiples usuarios, grupos de usuarios, etc. También se puede restringir el acceso por IP o por dominio. Si desea dar acceso a cualquiera de los usuarios que estén almacenados en el fichero de claves puede usar:

```
Require valid-user
```

La opción `AuthName` se conoce como "`Realm`" y es un texto que tiene dos funciones:

- Mostrarlo al usuario en la ventana que solicita las credenciales.
- Si el navegador accede a diferentes carpetas protegidas en las que coincide su realm, la clave solo la pedirá la primera vez. Por tanto, es conveniente que use un texto distinto para aplicaciones diferentes evitando así que usuarios autenticados en otra aplicación tengan acceso a la suya.

Debe tener en cuenta que con este método de autenticación, el usuario y la clave se transmiten en texto plano (sin cifrado).

Ejercicio: Protección de directorio

Proteja el directorio `/publico/protegido` de forma que únicamente pueda acceder a él un usuario llamado "`curioso`". Obverse que una vez autenticado el usuario, no se vuelve a pedir la clave hasta que se cierra y se abre de nuevo el navegador.

5.2. Configuración a nivel de usuario

A veces los desarrolladores de aplicaciones web no tienen acceso a la configuración global del sistema. Apache permite que estos usuarios (sin privilegios de administrador) puedan definir sus

propios ficheros de configuración. Para ello, es necesario que la configuración global de Apache lo permita. Si desea que el usuario de una determinada carpeta (y subcarpetas) pueda añadir sus propios ficheros de configuración, debe añadir la opción "`AllowOverride All`" a la directiva `Directory` correspondiente en la configuración global. Si añadiese "`AllowOverride None`" estaría indicando justo lo contrario: que el usuario no tiene permiso para establecer su propia configuración. `AllowOverride` también permite indicar, de forma individualizada, las opciones que podría modificar un usuario.

Las opciones de usuario se almacenan en la carpeta en la que se desea que tengan efecto en un fichero llamado `.htaccess` (también afecta a subcarpetas), cuya sintaxis es la misma que la de los ficheros de configuración global.

Ejercicio: añadir la opción de personalizar la configuración por directorios

Cree un fichero llamado `.htaccess` en la carpeta `/info` y añádale una directiva que impida el acceso a ella desde el servicio web (`Require all denied`). Compruebe si tiene efecto accediendo a la URL `http:/localhost/info` y viendo que ya no puede ver su contenido (que en este caso era el documento `index.html` que trae Apache por defecto).

Si no nota que tenga efecto, compruebe el fichero de configuración global de Apache (`apache2.conf`) y mire qué directivas `AllowOverride` tiene en él. En caso de que esas directivas impidan de forma global el acceso a los ficheros `.htaccess`, deberá añadir a su fichero de configuración particular (`miconf.conf`) la correspondiente directiva que permita acceder a `.htaccess` desde el directorio `/info`. No modifique los ficheros de configuración global de Apache.

Observe que el contenido de los ficheros `.htaccess` tiene efecto aun sin reiniciar el servidor web. El uso de esta forma de configuración es menos eficiente que el uso de ficheros de configuración globales, aunque en situaciones en las que el desarrollador web no es administrador del sistema es la única forma de personalizar la configuración del servidor web.

6. Instalación de MySQL

Un gran porcentaje de aplicaciones web almacenan la información con la que trabajan en Bases de Datos. En esos casos es necesario disponer de un servidor de BBDD, que puede estar en la misma máquina que el servidor web o en una máquina distinta. En esta asignatura se va a usar MySQL, un DBMS abierto y muy usado desarrollado en la actualidad por ORACLE.

La instalación en Ubuntu se realiza con el comando:

```
apt-get install mysql-server
```

Además, la instalación del servidor implica la instalación de `mysql-client`, que son los programas para acceder al servidor MySQL desde la línea de órdenes. La configuración de MySQL se encuentra en `/etc/mysql/my.cnf`.

Durante el proceso de instalación se preguntará cual es la clave del administrador del DBMS. En caso de que por algún motivo no tenga la clave de administrador (olvido, etc.) y necesite restaurarla, puede consultar la documentación de MySQL[1].

La administración del servidor de BBDD se puede hacer de varias formas:

- Desde la línea de órdenes con el cliente `mysql`.
- Con alguna aplicación que tenga interfaz gráfica.
- Con alguna aplicación web.

A continuación se muestran dos aplicaciones muy completas para administrar la BBDD: un cliente web y un cliente gráfico.

6.1. Instalación de phpMyAdmin

Esta es una aplicación web muy completa de administración de MySQL. La instalación en Ubuntu se hace con el comando:

```
apt-get install phpmyadmin
```

1 https://dev.mysql.com/doc/refman/8.0/en/resetting-permissions.html

Durante la instalación le preguntará la clave del DBMS MySQL y también la clave para acceder a phpMyAdmin. El usuario administradfor creado por defecto es "`root`".

Para usarla debe acceder a la dirección `http://localhost/phpmyadmin`.

Como una aplicación web que es, el acceso a phpMyAdmin se realiza a través del servidor Apache. Eso quiere decir que todos los ficheros de la aplicación son páginas .html, .php o similares accesibles desde un navegador.

Ejercicio: Localizar ficheros de phpMyAdmin

Sin realizar búsquedas por internet, averigue dónde se ha instalado phpMyAdmin y dónde se encuentra su configuración. Podrá comprobar que no está en `/var/www/html`, que es donde se almacenan por defecto las páginas servidas por Apache. Analice los ficheros de configuración del servidor para responder a la pregunta.

6.2. Instalación de mysql-workbench

Este es un cliente de escritorio para conectarse a MySQL desarrollado por ORACLE. Para instalarlo ejecute:

```
apt-get install mysql-workbench
```

Observe que no siempre es factible usar un cliente de escritorio o, en general, conectarse de forma remota a un servidor de BBDD. Es muy habitual proteger los sevidores de BBDD de forma que únicamente acepten conexiones desde la máquina que aloja las aplicaciones web (a veces es la misma para ambos servicios). Por tanto, si instala este cliente en un ordenador diferente al que aloja la BBDD, es posible que no tenga acceso a ella. Si es usted el administrador del sistema puede dar los permisos correspondientes pero esta no tiene porqué ser la situación habitual.

Por contra, un cliente web como phpMyAdmin sí podrá administrar la BBDD remota ya que todas las conexiones que se hacen llegan desde el ordenador que tiene alojado el servicio web, no desde su ordenador de escritorio.

7. Configuración de un servidor web seguro (HTTPS)

Si desea configurar el servidor Apache para que tenga habilitado el protocolo HTTPS, puede seguir las instrucciones de esta página web:

http://blog.hostdime.com.co/como-crear-un-certificado-ssl-en-apache-para-ubuntu-14-04/

que, resumidas, son estas:

- Activar el módulo SSL
  - `a2enmod ssl`
  - `service apache2 restart`
- Crear un certificado SSL autofirmado
  - `mkdir /etc/apache2/ssl`
  - `openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt`
- Modificar configuración de apache creando un virtualhost que escuche en el puerto 443
  - Editar `/etc/apache2/sites-available/default-ssl.conf`
  - `a2ensite default-ssl.conf`
  - `service apache2 restart`

Con esta configuración, al acceder con HTTPS toda la información entre el cliente y el servidor se transmite cifrada. Tenga en cuenta que el certificado no está verificado por ninguna autoridad por lo que la seguridad aún puede estar comprometida.

8. El directorio public_html de los usuarios

En ocasiones es posible que una organización permita que sus miembros dispongan de sus propias páginas web y, para ello, se instala un servidor web de forma que cada usuario de dicha máquina

disponga de un directorio en su home en donde aloje su página personal. Dicho directorio se suele llamar `public_html`.

Si en esa máquina existiese un usuario llamado "`curioso`", este podría crear una carpeta en su home llamada `public_html` y alojar en ella sus páginas web. Así, cualquiera podría visitarlas escribiendo la URL http://nombredelservidor/~curioso

Para que el servidor web pueda servir este directorio, que por defecto no está entre los servidor, debe modificar la configuración. En /etc/apache2/mods-available hay un fichero llamado userdir.conf que contiene las directivas necesarias para activar esta posibilidad.

Ejercicio: Active el public_html del usuario tweb

Para comenzar el ejercicio compruebe que esta funcionalidad no está activa. Acceda a la URL `http://localhost/~tweb` y obtendrá un error 404-Not Found.

Active el directorio public_html del usuario tweb siguiendo estos pasos:

1. Cree el directorio `public_html` en el home del usuario `tweb`.
2. Active la configuración de Apache que permite servir dicho directorio. Consulte la sección 3.1.4.

Ahora debería funcionar la URL `http://localhost/~tweb`

9. Entrega de la práctica

No hay entrega.

Se recomienda que el alumno haga la instalación de todas las herramientas en su ordenador personal.

Apéndice

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Fichero index.html</title>
  </head>
  <body>
      <h1>Está viendo la página index.html</h1>
      <p>Se trata de código HTML sencillos</p>
  </body>
</html>
```

index.php

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Fichero index.php</title>
  </head>
  <body>
    <?php echo "<h1>Está viendo la página index.php</h1>";
          echo "<p>En este caso lo que ve es el resultado de ejecutar un
script en PHP cuyo resultado es código HTML</p>";
    ?>
  </body>
</html>
```

claves.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Claves de acceso</title>
  </head>
  <body>
      <h1>Claves de acceso al servidor:</h1>
      <p>Jesús: duR5kI76D2</p>
      <p>José : hfc45F45Dg</p>
      <p>María: 98hGh6Fhr5</p>
  </body>
</html>
```

nuevasclaves.php

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Claves de acceso</title>
  </head>
  <body>
      <h1>Claves de acceso al servidor:</h1>
      <p>Jesús: <?php echo
substr(str_shuffle("0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTU
VWXYZ"), 0, 10);?></p>
      <p>José : <?php echo
substr(str_shuffle("0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTU
VWXYZ"), 0, 10);?></p>
      <p>María: <?php echo
substr(str_shuffle("0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTU
VWXYZ"), 0, 10);?></p>
  </body>
</html>
```

usuarios.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Usuarios</title>
  </head>
  <body>
      <h1>Usuarios del sistema:</h1>
      <p>Jesús</p>
      <p>José</p>
      <p>María</p>
  </body>
</html>
```

phpinfo.php

```
<?php phpinfo(); ?>
```