

## Ejercicio 1

### Instalación básica de Apache 2.4 en Linux.

Actualizar los repositorios e instalar Apache en tu ordenador con sistema operativo Linux. Si no dispones de él, utiliza una máquina virtual. Comprobar que apache funciona y acceder a él desde el navegador.

## Ejercicio 2

### Iniciar, parar y reiniciar el servidor apache

Indica las diferentes formas haciendo uso de comandos de Iniciar, parar y reiniciar el servidor apache.

## Ejercicio 3

### Creación de una web estática

Genera una página web (daw\_ejemplo.html) y ubicarla en el directorio principal del servidor web. Localiza el archivo principal de configuración de apache y realiza los cambios oportunos para que cargue por defecto la página daw\_ejemplo.html. (ojo con la caché del navegador y recuerda que hay que reiniciar el servidor si has cambiado los archivos de configuración).

## Ejercicio 4

### Uso de directivas básicas. Listado de directorio

Crea una carpeta dentro de tu servidor web e incluye varios documentos, imágenes, ...

Modifica el archivo de configuración de Apache para que puedas a través del navegador visualizar el contenido de dicho directorio, pero en el directorio principal no permita que se listen los archivos.

## Ejercicio 5

### Cambio de puertos

Modifica apache para que pueda atender peticiones tanto por el puerto 80 como por el 8080.

## Ejercicio 6

### Denegar permisos en un directorio

Modifica apache para que nadie pueda acceder a un directorio determinado, mostrando el error 403.

## Ejercicio 7

### Permitir el uso de archivos .htaccess

Crea un nuevo directorio en tu servidor web (con algunos archivos html) y permite crear archivos .htaccess en él. Genera un archivo .htaccess e impide que se puedan listar el contenido de los directorios.

## Ejercicio 7.5

Crear una página html la cual se mostrará en caso de que una página solicitada no sea encontrada (error 404), cambiar posteriormente la página a una url

Mostrar un mensaje personalizado en el caso de no poder acceder a un directorio (permiso denegado error 403)

## Ejercicio 8

### Creación de dos hosts virtuales basados en nombres.

Queremos crear dos hosts virtuales en un mismo servidor web Apache, de forma que cuando llamen a la url **daw1.com** acceda a una web determinada y a **daw2.com** acceda a otra.

**Comprobaciones previas:**

- Apache2 está instalado
- Puertos de escucha por defecto 80
- Sitio por defecto /var/www/html
- Directivas del sitio por defecto

Queremos construir en nuestro servidor web apache dos sitios web con las siguientes características:

1. El nombre de dominio del primero será **www.daw1.com**, su directorio base será **/var/www/daw1** y contendrá una página llamada **index.html**, donde sólo se verá una bienvenida a la página web con un enlace que rediriga a la web del IES Cristóbal de Monroy. Además:
  - a. Este servidor utilizará el puerto por defecto (80).
  - b. No mostrará a los usuarios el contenido de su directorio.
  - c. Cuando no encuentre una página debe devolver el texto "Página no encontrada en el servidor [www.daw1.com](http://www.daw1.com)" incluida en la web 404.html
  - d. Generar un error específico cuando no tenga permiso para acceder a un directorio (err403)
  - e. Redirigir el directorio del servidor /apuntes a una carpeta personal.(alias)
  - f. Redireccionar la carpeta /fp a <http://www.todofp.es>
2. En el segundo sitio vamos a crear una página donde se pondrán enlaces a la web del instituto, moodle del departamento de informática, etc., el nombre de este sitio será **www.daw2.com**, y su directorio base será **/var/www/daw2**. En este sitio sólo tendremos una página inicial **index.html**, con los enlaces específicos. Además:
  - a. Este servidor utilizará el puerto (80) para conectarse.
  - b. La página por defecto de este sitio será **index.html**.
  - c. Sí permitirá mostrar el contenido de su directorio a los usuarios

**Nota:** La página por defecto del servidor web apache será **inicio.html**

## Ejercicio 9

### Creación de dos hosts virtuales basados en ips.

*Para crear dos hosts virtuales basados en ip, necesitaríamos en nuestro servidor dos tarjetas de red. (Podemos usar una máquina virtual para el ejemplo).*

*Otra opción que permiten los sistemas operativos modernos es crear un Interfaz de red virtual, llamado "ip alias"*

Queremos crear dos hosts virtuales en un mismo servidor web Apache, de forma que cuando llamen a la url **ip1.com** acceda a una web determinada a través de una ip y a **ip2.com** acceda a otra a través de otra ip diferente.

## Ejercicio 10

*En este ejercicio vamos a ver algunas directivas básicas:*

*DocumentRoot, DirectoryIndex, Directory, FollowSymLinks, File, FileMatch, Option, Indexes, AllowOverride, Require*

Crea la ruta **/var/www/ejercicio3**

1. Crea un enlace directo con el comando de Linux **ln** dentro de la carpeta configurada en **DocumentRoot** y comprueba si es posible seguirlo al activar la opción **FollowSymLinks**. Cambia las opciones del directorio para que no siga los enlaces simbólicos.
2. Deshabilita la opción de que se listen los archivos existentes en la carpeta cuando no existe el fichero de índice definido en la directiva **DirectoryIndex**. Recuerda que por defecto **DirectoryIndex** vale **index.html**
3. Usando la directiva **Redirect** realiza una redirección, que permita que cuando entre a tu servidor

[http://nombre\\_servidor](http://nombre_servidor), salte a [http://nombre\\_servidor/web](http://nombre_servidor/web). Prueba también a hacer que al entrar en /buscador aparezca <http://www.google.es>

4. Con la directiva **ErrorDocument** se puede crear Respuesta de error personalizadas.
  - a. Cuando no se encuentre una página (error 404) por un mensaje de error.
  - b. Crea un alias llamado error que corresponda a /srv/mapeo/error. Dentro de ese directorio crea páginas personalizadas para visualizar cuando se produzca un error 404 y cuando se tenga un forbidden (403). Configura el sistema para que se redirija a estas páginas cuando se produce un error.

## Ejercicio 11

Un fichero **.htaccess** (hypertext access), también conocido como archivo de configuración distribuida, es un fichero especial de Apache que nos permite definir diferentes directivas de configuración para cada directorio (con sus respectivos subdirectorios) sin necesidad de editar el archivo de configuración principal de Apache.

Para permitir el uso de los ficheros **.htaccess** o restringir las directivas que se puedan aplicar usamos la directiva **AllowOverride**, que puede ir acompañada de una o varias opciones: All, None, AuthConfig, FileInfo, Indexes, Limit, ... Estudia para que sirve cada una de las opciones.

Crea la ruta **/var/www/personal** y pon dentro de personal varios ficheros. En esta carpeta tienes que crear un fichero **.htaccess** que nos permita:

1. Deshabilitar la opción de listar los ficheros en ese directorio.
2. Hacer que la página entrada.html se visualice por defecto ( directiva DirectoryIndex ).
3. Hacer que los ficheros txt no sean accesibles. Utiliza un bloque similar al siguiente:

```
<FilesMatch "\.(bak|1|old|swp)$">
  Order Allow,Deny
  Deny from all
</FilesMatch>
```

4. Redireccionar a una página (por ejemplo la web del instituto)
5. Crear una lista de IPs prohibidas para la ruta /personal/misdatos. Utiliza algo similar a:

```
<Directory /carpeta>
  Order Deny,Allow
  Deny from ....
  Allow from all
</Directory>
```

6. Crear una página personalizada para cada tipo de error
7. Crea una redirección permanente: cuando entremos en una ruta concreta que salte a [www.google.es](http://www.google.es)

## Ejercicio 12

- 1) Crea un host virtual que se llame [www.miweb.com](http://www.miweb.com) cuyo directorio base sea /srv/web.
- 2) Cuando se entre a la dirección [www.miweb.com](http://www.miweb.com) se redireccionará automáticamente a [www.miweb.com/principal](http://www.miweb.com/principal), donde se mostrará un mensaje de bienvenida. En el directorio principal no se permite ver la lista de los ficheros, no se permite que se siga los enlaces simbólicos.
- 3) Debe existir una URL que sea [www.miweb.com/principal/miscos](http://www.miweb.com/principal/miscos), que visualice el directorio home del usuario.

NO→ Por lo tanto se permitirá el listado de ficheros y el seguimiento de enlaces simbólicos siempre que sean a ficheros o directorios cuyo dueño sea el usuario.

4) En todo el host virtual se debe redefinir los mensajes de error de objeto no encontrado y no permitido. Para ello se crearán dos ficheros html dentro del directorio error.

5) En el url `www.miweb.com/images` deben aparecer los ficheros que se encuentran en `/usr/share/apache2/icons/`

## Ejercicios sobre control de acceso a usuarios.

### Ejercicio 13

Configura la autenticación **HTTP Basic** sobre el directorio `/var/www/html/privado` para que sólo puedan acceder los usuarios **dawuser1** y **dawuser2**

Pasos a seguir:

1. Habilitar los módulos **auth\_basic**, **authn\_file**, **authz\_user** si no están habilitados
2. Crear un **fichero** con usuarios y contraseñas accesibles por apache usando **htpasswd**
3. Modificar el host por defecto para sólo puedan acceder los usuarios **dawuser1** y **dawuser2**
4. Comprobar que sólo pueden acceder al directorio `/var/www/html/privado` los usuarios **dawuser1** y **dawuser2**

### Ejercicio 14

Configura la autenticación **HTTP Digest** sobre un directorio personal de un usuario usando un fichero **.htaccess**

Pasos a seguir:

1. Habilitar el módulo **auth\_digest**
2. Habilitar **.htaccess** sobre el directorio `/home/dawdigest/web` para que el propio usuario **dawdigest** pueda controlar cómo sirve Apache los contenidos de ese directorio.
3. Configurar el Alias **/web** que permita acceder al directorio `/home/dawdigest/web`
4. Acceder con el usuario **dawdigest** y crear **.htaccess** sobre el directorio `/home/dawdigest/web` de forma que:
  - a. sólo se pueda acceder desde el propio ordenador donde está instalado Apache
  - b. autenticación **HTTP Digest** para que sólo pueda acceder el usuario **dawdigest**
5. Comprobar la configuración de acceso a la web y al directorio.

Enlace resumen <http://chernando.eu/doc/apache/>

### Ejercicio 15

#### Instala un certificado de seguridad en Apache2

Pasos a seguir:

1. Instalar **OpenSSL**
2. Crear una **llave privada**
3. Crear un **CSR** (Certificate Signing Request)
4. Generar el **certificado SSL**
5. **Configurar** el certificado SSL en **Apache**
6. Verificar el certificado SSL desde un navegador

### Ejercicio 16

### Instalación de módulos en el servidor Apache2

En este ejercicio vamos a instalar los módulos **mod\_status** y **mod\_info** a modo de ejemplo

1. **mod\_status**: Permite monitorizar el rendimiento del servidor Apache2. Genera un documento en HTML con información sobre el estado actual del servidor
2. **mod\_info**: Proporciona una vista resumida de la configuración del servidor.

## Ejercicio 17

### Instalación de webalizer

Webalizer es un pequeño programa hecho en C el cual nos permite generar reportes de alguna página web. Gracias a esos reportes, podemos observar el número de personas que han entrado en la web, etc...

1. Instalación de webalizer y configuración en el servidor Apache
2. Visualización de logs a través del navegador

## Ejercicio 18

Instala apache2 en un servidor linux con el SO Ubuntu 16.04.

Deshabilita el host virtual por defecto y crea dos Host virtuales con las siguientes características detalladas en los siguientes apartados:

#### **Host Virtual 1:**

- Atenderá a la URL `www.webpublica.local` por el puerto por defecto
- El directorio de la web por defecto es `/var/www/webpublica`
- La web dispondrá de un directorio llamado **imagenes** en el que cualquier usuario en un futuro podrá subir sus propias imagenes, este debe servir el tipo mime por defecto jpg. (Incluye a modo de pruebas algunos archivos jpg y otras imágenes sin extensión o con una extensión cualquiera.
- No se permitirá en todo el site mostrar el contenido del directorio ni enlaces simbólicos ni archivos `.htaccess`, salvo el directorio **imagenes** que sí se podrá mostrar únicamente su contenido
- Elabora algún contenido html para poderlo visualizar en el cliente a modo de pruebas.
- Habilita el site y comprueba que funciona correctamente.

#### **Host Virtual 2:**

- Atenderá a la URL `www.webprivada.local` por el puerto 443 (seguro)
- El directorio de la web por defecto es `/opt/webprivada`
- La web privada mostrará una página inicial con un enlace a la web privada que se encuentra en la carpeta `/opt/webprivada/privada`.
- Para poder acceder a esa web, usaremos la autenticación básica de forma que únicamente:
  - los usuarios indicados en el archivo `/etc/apache2/.htpasswdprivado` podrán tener acceso y
  - si nos conectamos desde la IP local con cualquier usuario.

## Ejercicio 19

Crea un host virtual de forma que escuche las peticiones por el puerto 80 por defecto y que cuando accedamos a un directorio en concreto del host, cambie a https (puerto 443) modo seguro. O lo que es lo mismo habilitar https en un único directorio de un site.

ayuda: <https://wiki.apache.org/httpd/RedirectSSL>

## Ejercicio 20

Instala en un servidor LAMP en ubuntu 16.04 (Apache2, MySQL server y Php). Una vez comprobado que funciona, instala el CMS Joomla o Drupal y configurarlo para crear una web.

<https://openwebinars.net/blog/como-instalar-linux-apache-mysql-y-php-lamp-en-ubuntu-1404/>

<https://www.digitalocean.com/community/tutorials/como-instalar-linux-apache-mysql-php-lamp-en-ubuntu-16-04-es>