

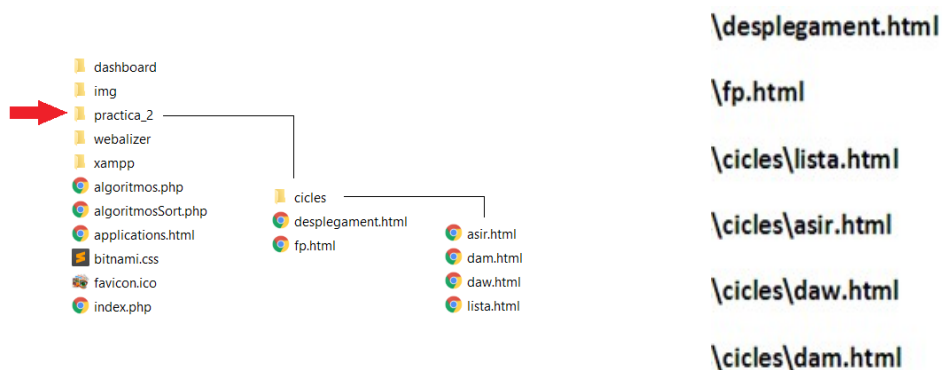


Práctica 2 – Administración de servidores web.

El principal objetivo de esta práctica es conocer las principales directivas de configuración del servidor Web Apache, ampliando la funcionalidad del servidor mediante la activación y configuración de módulos, además de crear y configurar hosts virtuales.

Como siempre, realiza un tutorial explicando cómo has realizado cada uno de los pasos de la práctica y adjunta las capturas de pantalla. Además, debes explicar lo que has aprendido (conclusiones obtenidas), al realizar cada uno de estos pasos.

1. Crea los siguientes ficheros y directorios para realizar las pruebas que comentaremos a continuación (dentro de los ficheros html debe aparecer vuestro nombre y apellidos + “nombre del fichero”): Dentro de htdocs hay una carpeta llamada practica_2 la cual contiene la estructura de carpetas y ficheros.



Estos ficheros deben estar en una carpeta sin ningún otro fichero, por tanto, para realizar mejor la práctica podemos modificar la directiva DocumentRoot, apuntando a la carpeta en la que hayáis dejado los ficheros.

- a) ¿Qué ocurre si accedemos en el navegador localhost/? Accederá a la página index.php que está dentro de la carpeta htdocs.
- b) ¿Y si accedemos directamente a localhost/cicles? En mi caso debería acceder a localhost/practica_2/cicles. Al no tener página index la carpeta cicles redirecciona a una página mostrando el árbol de directorios o ficheros que se encuentren ahí.
- c) Configura el servidor de manera que cuando pongamos localhost, nos sirva por defecto la página desplegements.html (¿te suena verdad?) En la carpeta C:\xampp\apache\conf encontraremos el archivo httpd.conf el cual editamos la directiva DocumentRoot de esta manera:

```
DocumentRoot "D:/xampp/htdocs/practica_2/desplegement.html"
<Directory "D:/xampp/htdocs">
```

- d) ¿qué ocurre si añadimos la siguiente directiva? Denegará para todos el acceso a dicha ruta (**C:xampp/htdocs/practica2**).

```
<Directory "C:/xampp/htdocs/practica2">
    Order Allow,Deny
    Deny from all
</Directory>
```

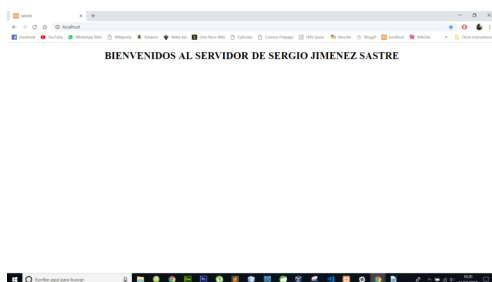
- e) ¿Qué significa la directiva Order? Explica su uso y posibles aplicaciones. La directiva Order controla el orden en el cual las directivas allow y deny son evaluadas.

Directiva Allow	Directiva Deny
Allow especifica cual cliente pueda acceder a un directorio dado. El solicitante puede ser all, un nombre de dominio, una dirección IP, una dirección IP parcial, un par de red/máscara de la red, etc. El directorio DocumentRoot esta configurado para permitir (Allow) peticiones desde todos (all), es decir, que todos tienen acceso.	Deny funciona igual que Allow, excepto que especifica a quién se le niega el acceso. DocumentRoot no es configurado para negar (Deny) peticiones a ninguno por defecto.

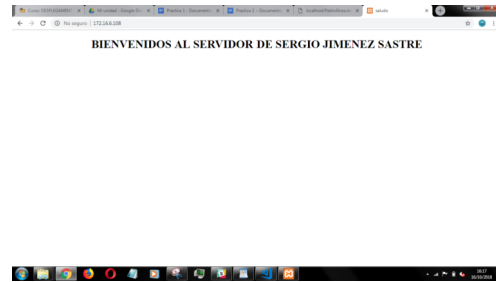
El servidor es configurado para evaluar las directivas Allow antes de las directivas Deny para el directorio DocumentRoot.

- f) Permite el acceso a todos, y comprueba que tu compañero puede acceder a tus carpetas, y haz capturas que efectivamente es así.

Mi PC ↓



PC compañero ↓



- g) Modifica la directiva Order para que sólo sea accesible la carpeta “cicles” por tu equipo. Comprueba con tu compañero que efectivamente es así, y no puede acceder. En la carpeta **C:xampp\apache\conf** encontraremos el archivo **httpd.conf** el cual editamos la directiva DocumentRoot de esta manera:

```
<Directory "D:/xampp/htdocs/practica_2/cicles">
    Order Allow,Deny
    Allow from all
</Directory>

<Directory "D:/xampp/htdocs/">
    Order Allow,Deny
    Deny from all
</Directory>
```



- h) Si tu compañero escribe completamente la ruta de la carpeta localhost/cicles/lista.html ¿qué ocurre? [Accederá al fichero el cual se encuentra en la ruta seleccionada.](#)
- i) Y si queremos restringir el acceso a la carpeta cicles y a todo su contenido, excepto al fichero daw.html. Para ello, podemos utilizar las directivas "Directory" y "DirectoryMatch".
- j) Investiga para qué sirve la directiva Alias, y haz uso de la misma con un ejemplo práctico. [La directiva Alias permite que el servidor abra páginas situadas en cualquier directorio, aunque no estén dentro del directorio establecido mediante la directiva DocumentRoot.](#)

Para que Apache pueda acceder a esos directorios, en el archivo httpd.conf debe estar cargado el módulo alias mediante la directiva: →

LoadModule alias_module modules/mod_alias.so

Ejemplo:

Supongamos que un usuario está siguiendo un curso de PHP y al mismo tiempo en su trabajo también escribe programas en PHP. Este usuario quiere guardar los archivos en diferentes carpetas de Mis Documentos, pero al mismo tiempo acceder fácilmente a los archivos sin tener que escribir direcciones muy largas:

Programas	Ruta en el navegador	Carpeta en el disco
Programas escritos en el curso de PHP	http://localhost/clase	Mis documentos/Cursos/PHP
Programas escritos para el trabajo	http://localhost/curro	Mis documentos/Trabajo/PHP

Para ello, hay que añadir en el archivo httpd.conf dos directivas Alias dentro de la directiva <IfModule alias_module>. En esta directiva se pueden crear redirecciones, alias y alias de script. Para mantener el archivo bien organizado, se recomienda escribir los alias a continuación de las líneas de comentario relativas a los alias.

El ejemplo siguiente muestra los dos alias del ejemplo anterior. →

```
<IfModule alias_module>
# ...
# 2018-09-01. He añadido este Alias para los programas escritos en el curso de PHP.
Alias /clase "C:/Users/HombreDelUsuario/Documents/Cursos/PHP"
<Directory "C:/Users/HombreDelUsuario/Documents/Cursos/PHP">
    Options Indexes FollowSymLinks Includes ExecCGI
    AllowOverride All
    Require all granted
</Directory>

# 2018-09-01. He añadido este Alias para los programas escritos para el trabajo.
Alias /curro "C:/Users/HombreDelUsuario/Documents/Trabajo/PHP"
<Directory "C:/Users/HombreDelUsuario/Documents/Trabajo/PHP">
    Options Indexes FollowSymLinks Includes ExecCGI
    AllowOverride All
    Require all granted
</Directory>
# ...
</IfModule>
```

- k) Investiga para qué sirve la directiva Options –indexes, y comprueba su uso.
- l) [La directiva Order quedará obsoleta](#) en futuras versiones de Apache: https://httpd.apache.org/docs/trunk/es/mod/mod_access_compat.html

Por ello, nos aconseja, el uso de la directiva REQUIRE. Explica cómo se utilizan las directivas RequireAll, RequireAny y RequireNone, y realiza varias pruebas de aplicación de las mismas.

2. Configura apache para crear dos servidores virtuales basados en nombre, éstos se llaman `tunombre+apellido.com` y `apellido+nombre.com`. A continuación rearemos un servidor virtual el cual tendrá como nombre **`sergio_jimenez.com`**. La manera de hacerlo es la siguiente:

Cada vez que queremos abrir nuestro proyecto tendremos que introducir **`localhost/ruta_proyecto`**. Hay dos formas de solventar esto, una de ellas es configurando el fichero **`httpd.conf`** de Apache cambiando la directiva `DocumentRoot` (directorio principal de nuestro servidor) y la otra es crear un `virtualhost` (servidor virtual).

Importante: Después cada paso debemos reiniciar el Apache.

Paso 1: Verificación

Primero debemos verificar que tenemos descomentadas ciertas líneas en el archivo **`httpd.conf`** de Apache, las líneas a descomentar (si estuvieran comentadas, cosa que sucede si trabajamos con una versión Antigua de XAMPP) son las siguientes: En primer lugar descomentar la línea que dice **`LoadModule rewrite_module modules/mod_rewrite.so`** y la línea la cual dice **`Include conf/extra/httpd-vhosts.conf`**.

Paso 2: Editando el fichero `httpd-vhosts.conf`

Debemos editar el fichero llamado **`httpd-vhost.conf`** (el cual se encuentra en `xampp\apache\conf\extra`). Al final del archivo deberemos agregar unas líneas.

La línea que dice **`DocumentRoot "C:/xampp/apps/wordpress/htdocs"`** hay que editarla por la ruta de tu proyecto y la línea que dice `ServerName` debemos colocar nuestro nombre de dominio deseado. De esta manera →

```
<VirtualHost *:80>
    DocumentRoot "D:/xampp/htdocs/"
    ServerName localhost
</VirtualHost>
<VirtualHost *:80>
    DocumentRoot "D:/xampp/htdocs/practica_2"
    ServerName sergio_jimenez.com
</VirtualHost>
```

Paso 3: Editando el fichero `hosts`

Para terminar con la configuración nos queda entrar con permisos de administrador a **`C:\windows\system32\drivers\etc\hosts`** y editarlo.

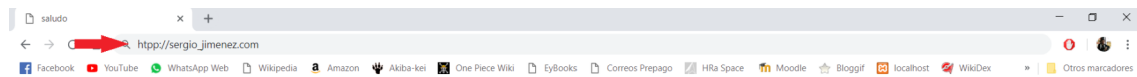
Justo al lado de la dirección `127.0.0.1` (referente a nuestro host local) debemos escribir nuestro nombre de dominio anteriormente dado.

Un ejemplo de como debería quedar esta línea →

```
# Nombre de dominio del servidor
127.0.0.1      sergio_jimenez.com
```

Paso 4: Probándolo

Para probarlo deberemos abrir un navegador y en la barra de direcciones escribir el nombre de dominio el cual le hemos asignado, en mi caso es **`sergio_jimenez.com`**. La primera vez que accedemos debemos introducir `http://` detrás de nuestro nombre de dominio (para que el navegador no crea que se trata de una búsqueda en internet) de esta forma **`http://sergio_jimenez.com`**.



BIENVENIDOS AL SERVIDOR DE SERGIO JIMENEZ SASTRE

No es necesario tener nuestro proyecto ubicado en **C:\xampp\htdocs**, podríamos crear una carpeta alternativa en otra ruta ajena a la carpeta xampp, pero sin olvidar añadir dicha ruta en los ficheros de configuración anteriormente mencionados.

Ayuda:

<http://httpd.apache.org/docs/current/vhosts/name-based.html>

<https://styde.net/creando-virtual-hosts-con-apache-en-windows-para-wamp-o-xampp/>