



Pt1a - Instal·lació i configuració d'apache

Estudiant (COGNOMS, NOM): **Daniel Májer**

Professors: **Dora Yenny
Muñoz
Córdoba**

DAWBIO-2
27 de noviembre, 2022

Format del lliurament de la pràctica

- Format del lliurament
1erCOGNOM_2onCOGNOM_NOM_DAW-M08-UF1-NF1-Pt1a.pdf

Objectius generals de la pràctica

Instal·lació i configuració del servidor web apache.

Desenvolupament de la pràctica

A cada pas heu de mostrar les ordres executades i el resultat. Afegiu captures de pantalla quan calgui (navegador..).

En aquesta pràctica s'utilitza el servidor web HTTP Apache instal·lat en un Ubuntu Server, haureu d'instal·lar-lo de forma remota i al llarg de la pràctica configurareu algunes de les diferents opcions que presenta, des de la utilització d'índex, usuaris, grups, passwords, fins a la creació de host virtuals.

Heu de treballar sempre **de forma remota**, quan heu de modificar fitxers o afegir fitxers podeu utilitzar SFTP (modo línia o Filezilla).

Recordeu algunes ordres que podeu necessitar:

- per veure la ip: **ip address show**
o **ip a**
- per veure els ports (sockets) tcp «listening»: **sudo ss -tlnp**)



Instal·lació

Treballarem a una VM amb Ubuntu Server. Configureu la Xarxa de la VM de forma que podeu accedir per SSH des de l'amfitrió, amb 'Adaptador pont' (també es pot fer amb NAT i reenviament de ports.)

Instal·lació del servidor Apache a Ubuntu Server de forma remota (SSH).

- a) Des de l'amfitrió, accedint per SSH (modo línia o putty) a la VM: instal·leu el servidor web Apache. A <https://ubuntu.com/server/docs/web-servers-apache> trobareu informació per fer la instal·lació (i informació sobre la configuració que necessitareu més endavant)

1) Conectamos a nuestro servidor via SSH:

- `$ ssh -p 8000 dama@192.168.128.200`

```
(venv) dama@pop-os:~  
$ ssh -p 8000 dama@192.168.128.200  
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-50-generic x86_64)  
  
 * Documentation:  https://help.ubuntu.com  
 * Management:    https://landscape.canonical.com  
 * Support:       https://ubuntu.com/advantage  
  
System information as of Thu Nov 17 08:12:05 PM UTC 2022  
  
System load:  0.0078125      Processes:            135  
Usage of /:   26.2% of 14.38GB Users logged in:      1  
Memory usage: 2%            IPv4 address for enp0s3: 192.168.128.200  
Swap usage:   0%  
  
39 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
Last login: Thu Nov 17 19:45:43 2022 from 192.168.128.122  
dama@dama-server:~$
```

2) Instalamos apache2:

- `sudo apt install apache2`

```
dama@dama-server:~$ sudo apt install apache2  
[sudo] password for dama:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  apache2-bin apache2-data apache2-utils  
Suggested packages:  
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser  
The following packages will be upgraded:  
  apache2 apache2-bin apache2-data apache2-utils  
4 upgraded, 0 newly installed, 0 to remove and 80 not upgraded.  
Need to get 1,696 kB of archives.  
After this operation, 4,096 B of additional disk space will be used.  
Do you want to continue? [Y/n] y
```



3) Confirmar que el servicio de apache2 está activo:

- `sudo systemctl status apache2`

```
dama@damaserver:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-11-17 20:13:31 UTC; 1min 0s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2212 (apache2)
    Tasks: 55 (limit: 11214)
   Memory: 5.0M
      CPU: 29ms
   CGroup: /system.slice/apache2.service
           └─2212 /usr/sbin/apache2 -k start
             └─2214 /usr/sbin/apache2 -k start
               └─2215 /usr/sbin/apache2 -k start

Nov 17 20:13:31 damaserver systemd[1]: Starting The Apache HTTP Server...
Nov 17 20:13:31 damaserver apachectl[2211]: AH00558: apache2: Could not reliably determine the server's
Nov 17 20:13:31 damaserver systemd[1]: Started The Apache HTTP Server.
dama@damaserver:~$
```

b) Comproveu amb el navegador (a l'amfitrió) que el vostre Servidor Web funciona.

1) Miraremos el IP, que tiene el servidor en la red local:

- `ifconfig`

```
Nov 17 20:13:31 damaserver systemd[1]: Started The Apache HTTP Server.
dama@damaserver:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.128.200 netmask 255.255.255.0 broadcast 192.168.128.255
    inet6 fe80::a00:27ff:fe26:9a25 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:26:9a:25 txqueuelen 1000 (Ethernet)
    RX packets 111169 bytes 165400939 (165.4 MB)
    RX errors 0 dropped 947 overruns 0 frame 0
    TX packets 37242 bytes 3138360 (3.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

2) Abrimos este mismo IP en el navegador:

Apache2 Default Page

Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/  
|-- apache2.conf  
|  
|-- ports.conf  
|-- mods-enabled  
|   |-- *.load  
|   |-- *.conf  
|-- conf-enabled  
|   |-- *.conf  
|-- sites-enabled  
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations respectively.

Accés als fitxers `/var/www/html/index.html`

1. Prova de modificar aquest arxiu HTML de la següent manera: com usuari root i amb un editor (o sigui amb sudo). Posa-hi un missatge de benvinguda personalitzat.

Utilizar sudo por acaso sí necesita permiso:

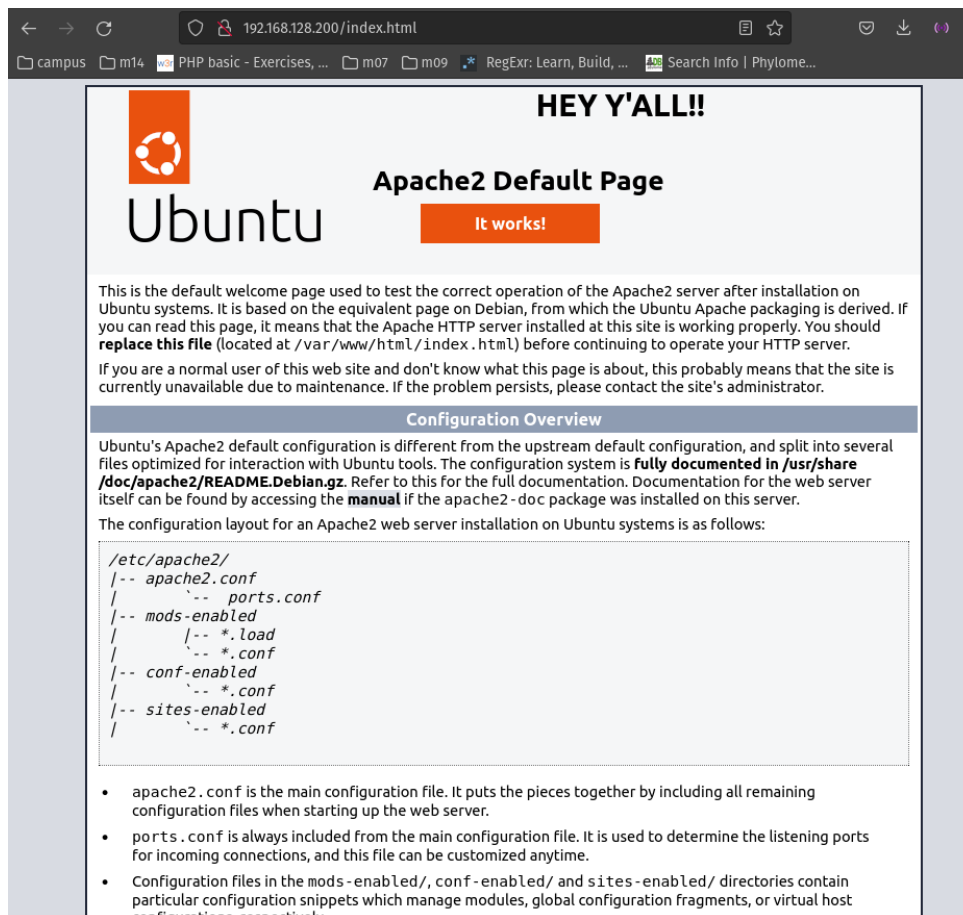
- `sudo vim /etc/www/html/index.html`

Buscamos un lugar en el tope del `<body>`, y escribimos nuestro mensaje de bienvenido.

Guardamos el fichero modificado y refrescamos la página web (sí no vamos a ver el cambio directamente, vamos a intentar acceder explícitamente al `index.html`,

<http://192.168.128.200/index.html>).

```
        style="width:184px;height:146px;" class="floating_element" />
<div>
<h1> HEY Y'ALL!! </h1>
    <span style="margin-top: 1.5em;" class="floating_element">
        Apache2 Default Page
    </span>
</div>
<div class="banner">
```





2. Busca una manera de modificar aquest arxiu i d'afegir-ne de nous sense actuar com a root (o sigui, sense sudo). La idea és que donis permisos adequats (MAI 777!) i que la carpeta sigui d'un grup, per exemple "web".

Crea el grup "web" amb la comanda `addgroup`. Si no estàs segur de com va, ja saps, utilitza el manual:

```
$ man addgroup
```

Dona els permisos i propietari (recordeu! **usuari:grup**) adequats a la carpeta

`/var/www/html` per tal que pertanyi al grup **web**

Recordeu que cal utilitzar les comandes **chmod** i **chown** per realitzar-ho.

Afegeix l'usuari principal a aquest grup: es pot fer amb diverses comandes com **adduser**, **usermod** o directament modificant l'arxiu `/etc/groups`.

Surt de la sessió de l'usuari principal i torna a entrar, i crea una nova pàgina HTML amb algun text. Visualitza-la amb el navegador.

Recorda explicar a l'informe les comandes exactes que has utilitzat i justifica els permisos assignats.

- 1.) Creo el grupo "web" (aunque ya existe de una práctica anterior):
(`sudo addgroup web`)

```
dama@dama-server:~$ sudo addgroup web
[sudo] password for dama:
addgroup: The group `web' already exists.
dama@dama-server:~$
```



- 2.) Cambiamos los dueños (usuario y grupo) del directorio “/var/www/html” y también cambiamos los permisos, para que todos los usuarios en el grupo web podrán editar el fichero.

```
("sudo chown $USER:web /var/www/html")  
("sudo chown $USER:$USER /var/www/html/index.html")  
("sudo chmod -R 775 /var/www/html/index.html")
```

Después los pasos anteriores añadimos nuestro usuario al grupo “web”.
("sudo usermod -a -G web dama")

```
dama@damaserver:/var/www/html$ sudo chown $USER:web /var/www/html  
dama@damaserver:/var/www/html$ sudo chown $USER:$USER /var/www/html/index.html  
dama@damaserver:/var/www/html$ sudo chmod -R 775 /var/www/html/index.html  
dama@damaserver:/var/www/html$ sudo usermod -a -G web dama  
dama@damaserver:/var/www/html$ ls -la /var/www  
total 12  
drwxr-xr-x  3 root root 4096 Oct 20 18:29 .  
drwxr-xr-x 15 root root 4096 Nov  3 19:37 ..  
drwxr-xr-x  3 dama web  4096 Nov 24 19:33 html  
dama@damaserver:/var/www/html$ ls -la /var/www/html  
total 24  
drwxr-xr-x 3 dama web  4096 Nov 24 19:33 .  
drwxr-xr-x 3 root root 4096 Oct 20 18:29 ..  
-rwxrwxr-x 1 dama dama 10695 Nov 17 20:26 index.html  
drwxrwx--x 3 root web  4096 Nov 10 19:58 shared  
dama@damaserver:/var/www/html$ groups dama  
dama : dama adm cdrom sudo dip plugdev lxd web  
dama@damaserver:/var/www/html$
```

3.) Como último paso confirmamos sí sin permisos de *sudo* podemos editar el fichero *index.html*.

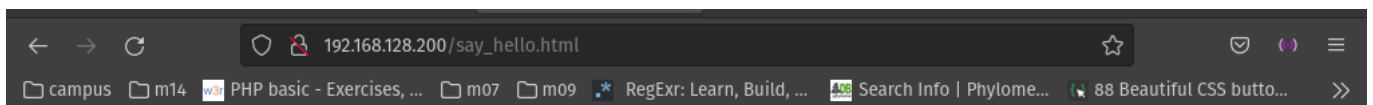
(Y le añadimos el siguiente texto a la página: "<h1> THIS VERSION IS AFTER CHANGING PERMISSIONS!! </h1>")

```

<div>
<h1> HEY Y'ALL!! </h1>
<h1> THIS VERSION IS AFTER CHANGING PERMISSIONS!! </h1>
<span style="margin-top: 1.5em;" class="floating_element">
  Apache2 Default Page
```



4.) Creamos otra página con algún contenido y visualizamos en el navegador.



Daniel told me that I need to say "Hello" to all of you...

3. Crea un arxiu **phpinfo.php** amb el següent contingut. **Instal·la i configura el mòdul de PHP** per a que es mostri la informació del servidor. A <https://help.ubuntu.com/its/serverguide/php.html> trobareu informació per fer la instal·lació. Visualitzeu-lo amb el navegador. Recorda posar una captura de pantalla a l'informe. I que es vegi l'escriptori, res de retallades.

```
<html>
  <body>
    <?php
      phpinfo();
    ?>
  </body>
</html>
```

- 1.) Primero instalamos todas las dependencias de PHP para que podamos interpretar con Apache los scripts de PHP.

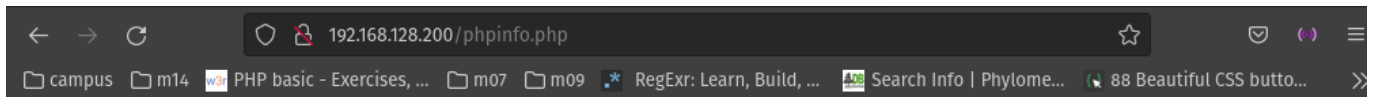
```
("sudo apt install -y php8.1 php8.1-mysql libapache2-mod-php8.1
php8.1-curl php8.1-xml")
```

```
dama@damaserver:~/setup_lamp_stack$ sudo apt install -y php8.1 php8.1-mysql libapache2-mod-php8.1 php8.1-curl
php8.1-xml
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  php-common php8.1-cli php8.1-common php8.1-openssl php8.1-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php8.1 php-common php8.1 php8.1-cli php8.1-common php8.1-curl php8.1-mysql php8.1-openssl
  php8.1-readline php8.1-xml
0 upgraded, 10 newly installed, 0 to remove and 60 not upgraded.
Need to get 5,411 kB of archives.
After this operation, 22.4 MB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu jammy/main amd64 php-common all 2:92ubuntu1 [12.4 kB]
Get:2 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-common amd64 8.1.2-1ubuntu2.9 [1,126
```

2.) Reiniciamos el servicio Apache2 y creamos el fichero "**phpinfo.php**" para comprobar si funcionaba.

```
("sudo systemctl restart apache2")  
("echo "<html><body><?php phpinfo(); ?></body></html>" >  
/var/www/html/phpinfo.php")
```

Visitamos la página nueva de PHP ("<http://localhost/phpinfo.php>")



PHP Version 8.1.2-1ubuntu2.9



System	Linux damaserver 5.15.0-50-generic #56-Ubuntu SMP Tue Sep 20 13:23:26 UTC 2022 x86_64
Build Date	Oct 19 2022 14:58:09
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-mysqld.ini, /etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/15-xml.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-curl.ini, /etc/php/8.1/apache2/conf.d/20-dom.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-ffi.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-mysqli.ini, /etc/php/8.1/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-simplexml.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini, /etc/php/8.1/apache2/conf.d/20-xmlreader.ini, /etc/php/8.1/apache2/conf.d/20-xmlwriter.ini, /etc/php/8.1/apache2/conf.d/20-xsl.ini

Indexes

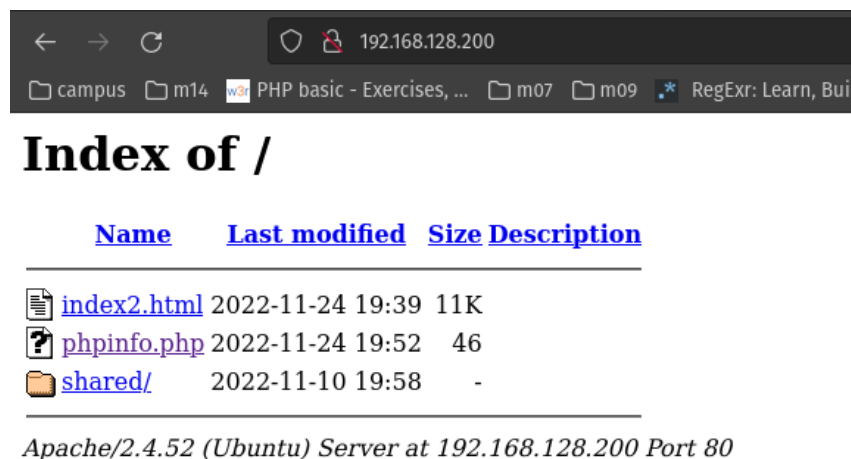
4. La configuració per defecte de l'apache permet visualitzar els arxius que hi ha a una determinada carpeta (és la Opció *Indexes* configurada pel directori `/var/www` al fitxer `apache2.conf`).

Per comprovar-ho, canvia el nom de `index.html` a `index2.html`.

RECORDA: A PARTIR D'ARA NO TREBALLIS COM A ROOT, SINÓ COM A USUARI PRINCIPAL

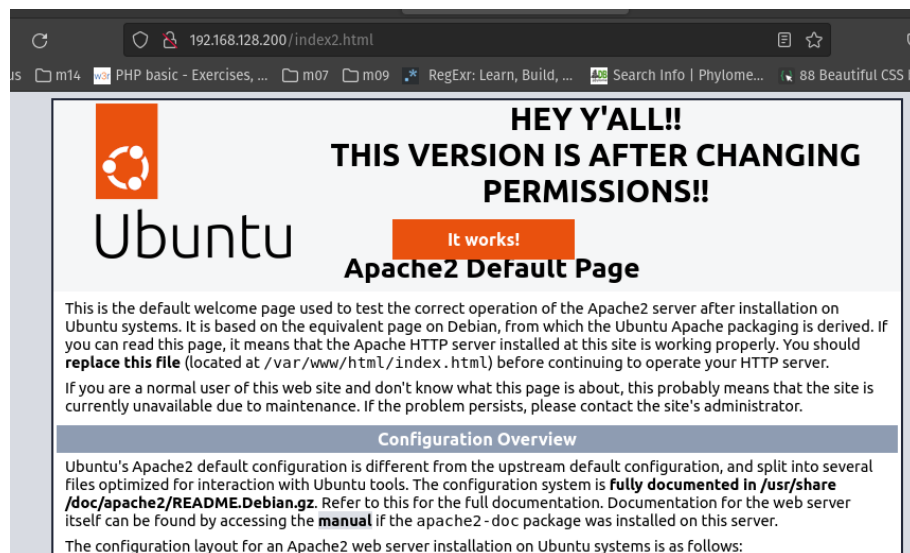
Quin efecte té? (Explica-ho a l'informe i inclou pantallada).

- 1.) Renombremos el fichero "*index.html*" a "*index2.html*".
(`"mv /var/www/html/index.html /var/www/html/index2.html"`)
- 2.) Visitamos la página:
(`"http://192.168.128.200/"`)



Apache ya no nos sirve el contenido de la página principal, porque por defecto en el directorio root (`/var/www/html`) busca por un fichero "*index.html*".

Sí hacemos click al "*index2.html*", entonces ya sale la página.



5. La gent veu els nostres arxius si falta l'index.html, això pot resultar un forat de seguretat ja per un descuit nostre podrien conèixer la estructura de la web i atacar arxius que contenen codi executable. Si ho volem evitar cal desactivar els "Indexes".

Exercici

Busca articles on expliqui com desactivar els indexes de les següents maneres:

1. Desactivar indexes per tota la web (PISTA: busca els arxius de configuració dels directoris virtuals).

Para hacerlo, primero tenemos que abrir "/etc/apache2/apache2.conf", luego buscamos la línea: "Options Indexes FollowSymLinks" y cambiamos esa línea a: "Options FollowSymLinks"
Por último, guardamos y salimos del archivo, y reiniciamos el servidor apache:
("sudo systemctl reiniciar apache2")

```
AllowOverride None
Require all granted
</Directory>

<Directory /var/www/>
Options Indexes FollowSymLinks
AllowOverride None
Require all granted
</Directory>

#<Directory /srv/>
# Options Indexes FollowSymLinks
# AllowOverride None
# Require all granted
#</Directory>
```

=>

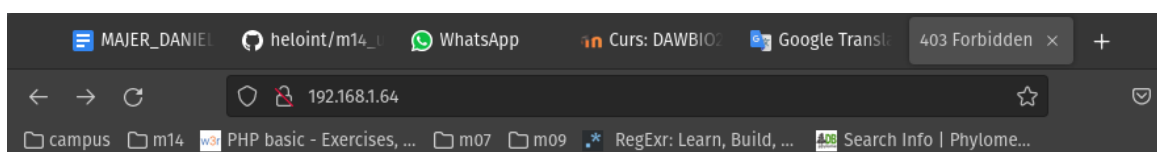
```
AllowOverride None
Require all granted
</Directory>

<Directory /var/www/>
Options FollowSymLinks
AllowOverride None
Require all granted
</Directory>

#<Directory /srv/>
# Options Indexes FollowSymLinks
# AllowOverride None
# Require all granted
#</Directory>
```

```
dama@dama-server:~$ nvim /etc/apache2/apache2.conf
dama@dama-server:~$ sudo !!
sudo nvim /etc/apache2/apache2.conf
[sudo] password for dama:
dama@dama-server:~$ sudo systemctl restart apache2
dama@dama-server:~$
```

Ahora sí podemos ver que restringe a los usuarios para acceder a los contenidos de los directorios.



Forbidden

You don't have permission to access this resource.

Apache/2.4.52 (Ubuntu) Server at 192.168.1.64 Port 80

2. Desactivar *Indexes* per un directori concret. Per realitzar això caldrà utilitzar un arxiu *.htaccess* dins de la carpeta que vols anul·lar els *Indexes*. També et caldrà activar l'ús d'aquests arxius al fitxer de configuració del directori virtual, en particular cal tocar el "*AllowOverride*", però abans de fer-ho documenteu-vos).

Resultat final:

Deixa els *Indexes* activats per la web sencera.

Prohibeix-los de la carpeta *images*, on pujaràs 3 fotos d'exemple amb aquests noms: *1.png*, *2.png*, *3.png*

1.) Activamos los indexes en toda la web.

("sudo vim /etc/apache2/apache2.conf")

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>
```

=>

```
<Directory />
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>
```

2.) Creamos un directorio "images" en el directorio "/var/www/html".

("mkdir -p /var/www/html/images")

Confirmamos:

("ls -la /var/www/html")

```
dama@dama-server:~$ mkdir -p /var/www/html/images
dama@dama-server:~$ ls -la /var/www/html
total 36
drwxr-xr-x 4 dama web 4096 Nov 27 11:54 .
drwxr-xr-x 3 root root 4096 Oct 20 18:29 ..
drwxrwxr-x 2 dama dama 4096 Nov 27 11:54 images
-rwxrwxr-x 1 dama dama 10752 Nov 24 19:39 index2.html
-rw-rw-r-- 1 dama dama 46 Nov 24 19:52 phpinfo.php
-rw-rw-r-- 1 dama dama 348 Nov 24 20:21 say_hello.html
drwxrwx--x 3 root web 4096 Nov 10 19:58 shared
dama@dama-server:~$
```



3.) Desde nuestro local subimos los imagenes.

("for i in {1..3}; do scp -P 8000 /home/dama/Pictures/\$i.png
dama@192.168.1.64:/var/www/html/images/; done")

```
(venv) dama@pop-os:~  
$ for i in {1..3}; do scp -P 8000 /home/dama/Pictures/$i.png dama@192.168.1.64:/var/www/html/images/; done  
1.png 100% 27KB 13.2MB/s 00:00  
2.png 100% 44KB 19.7MB/s 00:00  
3.png 100% 34KB 14.3MB/s 00:00  
(venv) dama@pop-os:~
```

4.) Declaramos el directorio en el fichero "/etc/apache2/apache2.conf".

("sudo vim /etc/apache2/apache2.conf")

```
(  
    "<Directory /var/www/html/images>  
        AllowOverride All  
        Require all granted  
    </Directory>"  
)
```

```
<Directory /var/www/html/images>  
    AllowOverride All  
    Require all granted  
</Directory>
```

5.) Añadimos el fichero .htaccess al directorio "/var/www/html/images/" con el siguiente contenido dentro:

- "Options FollowSymLinks"

Commando: **echo Options -Indexes > /var/www/html/images/.htaccess**

```
dama@damaserver: ~  
dama@damaserver:~$ echo Options -Indexes > /var/www/html/images/.htaccess  
dama@damaserver:~$ cat !$  
cat /var/www/html/images/.htaccess  
Options -Indexes  
dama@damaserver:~$
```

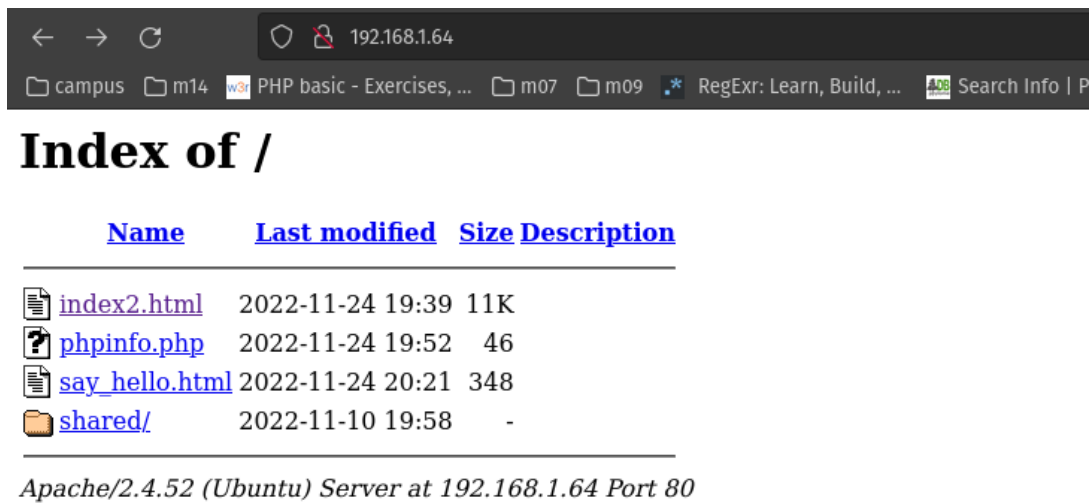
6.) Reiniciamos apache

("sudo systemctl restart apache2")

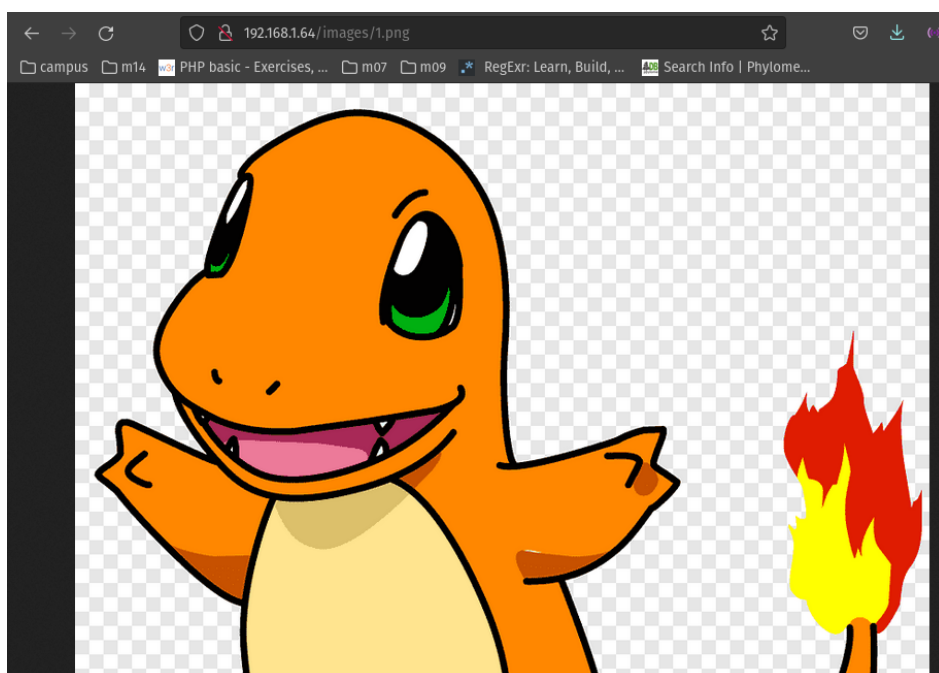
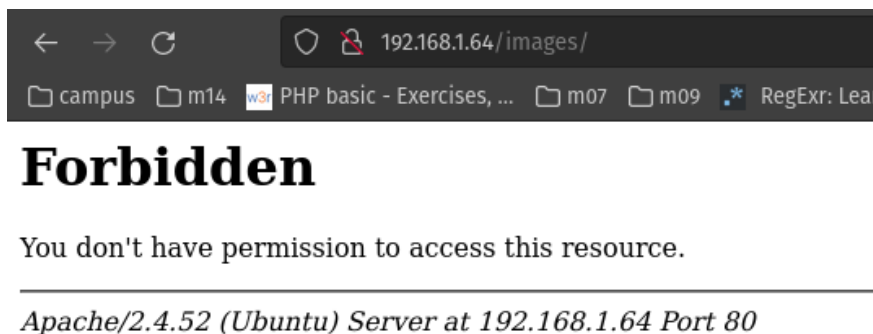
```
dama@damaserver:~$ sudo systemctl restart apache2  
[sudo] password for dama:  
dama@damaserver:~$
```

7.) Confirmamos sí la configuración nueva funciona.

- El Index funciona con la página “principal”.



- Pero como podemos ver, al directorio “images” ya no podemos acceder sin especificar el fichero de .png.





userdir

6. És molt útil per un servidor que cada usuari pugui tenir el seu espai web i que no interfereixi al dels altres usuaris. Apache ho té contemplat amb el mòdul "userdir". Per activar-lo és tan senzill com:

```
$ sudo a2enmod userdir
```

El directori de *mods-enabled* (son links simbòlics) ara hi figura userdir, també:

```
zeus@zeusserver:~$ ls /etc/apache2/mods-enabled/
```

```
alias.conf authz_host.load deflate.load negotiation.conf rewrite.load userdir.load alias.load  
authz_user.load dir.conf negotiation.load setenvif.conf  
auth_basic.load autoindex.conf dir.load php7.0.conf setenvif.load  
authn_file.load autoindex.load env.load php7.0.load status.conf  
authz_default.load cgi.load mime.conf reqtimeout.conf status.load authz_groupfile.load  
deflate.conf mime.load reqtimeout.load userdir.conf
```

Si tenim l'usuari zeus (per exemple), podem accedir a la carpeta **/home/zeus/public_html** (ULL! S'ha de crear "a mà") apuntant el navegador a:

```
http://<ip_del_server>/~zeus
```

La "tilde" de la lletra "ñ" s'aconsegueix amb [AltGr]+[ñ]. Si esteu en Windows i se us resisteix, premeu [Alt]+126 en el teclat numèric.

Activa els **userdir**. Comprova que estan activats llistant els mòduls activats d'apache.

Crea la carpeta de l'usuari principal (zeus, funky, super o el què sigui).

Afegeix una pàgina HTML. Captureu pantalla visualitzant-la.

- 1.) Creamos el directorio ~/public_html para nuestro usuario, sí no existe aún.

```
("mkdir -p ~/public_html")
```

Añadimos los permisos:

```
("chmod 755 ~/public_html")
```

Añadimos el usuario "www-data" al grupo de nuestro usuario.

```
("sudo adduser www-data $(whoami)")
```

- 2.) Activamos el "userdir" mod en Apache2.

```
("sudo a2enmod userdir")
```

- 3.) Reiniciamos el servicio de Apache.

```
("sudo systemctl restart apache2")
```



```
dama@dama-server: ~  
dama@dama-server:~$ mkdir -p ~/public_html  
dama@dama-server:~$ chmod 755 ~/public_html  
dama@dama-server:~$ sudo adduser www-data $(whoami)  
The user `www-data' is already a member of `dama'.  
dama@dama-server:~$ sudo a2enmod userdir  
Module userdir already enabled  
dama@dama-server:~$ sudo systemctl restart apache2  
dama@dama-server:~$
```

4.) Añadimos un fichero index.html al directorio “~/public_html” y comprobamos en el navegador.
 (“echo \<p> Hello \</p> > ~/public_html/index.html”)



Hosts virtuals

7. L'objectiu d'aquest apartat és tenir 2 dominis diferents al mateix servidor. Cadascun d'aquests dominis ha d'anar a parar a una web d'un usuari diferent:

www.zeus.local -> /home/zeus/public_html
www.mercuri.local -> /home/mercuri/public_html

Aquest apartat no el podreu fer si no heu entès i activat correctament el mòdul "userdir"

- a. **Arxiu de virtual host:** cada site (domini) ha de tenir un arxiu a **/etc/apache2/sites-available**. Crea una nova site copiant l'arxiu **000-default.conf** a **mercuri.conf** (has de tenir, doncs, 2 arxius) a "sites-available".

1.) Primero creamos el usuario "mercuri".
("sudo adduser mercuri")

Y le añadimos al grupo de "sudo".
("sudo usermod -a -G sudo mercuri")

```
dama@dama-server:/etc/apache2/sites-available$ sudo adduser mercuri
Adding user `mercuri' ...
Adding new group `mercuri' (1006) ...
Adding new user `mercuri' (1005) with group `mercuri' ...
Creating home directory `/home/mercuri' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for mercuri
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n]
dama@dama-server:/etc/apache2/sites-available$
```

```
dama@dama-server:~$ sudo usermod -a -G sudo mercuri
dama@dama-server:~$ groups mercuri
mercuri : mercuri sudo
dama@dama-server:~$
```

2.) Entramos al usuario "mercuri" y realizamos los pasos de la pregunta anterior.

```
dama@dama-server:~$ su mercuri
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

mercuri@dama-server:/home/dama$ cd
mercuri@dama-server:~$ mkdir -p ~/public_html
mercuri@dama-server:~$ chmod 755 ~/public_html
mercuri@dama-server:~$ sudo adduser www-data $(whoami)
[sudo] password for mercuri:
Adding user `www-data' to group `mercuri' ...
Adding user www-data to group mercuri
Done.
mercuri@dama-server:~$ sudo a2enmod userdir
Module userdir already enabled
mercuri@dama-server:~$ sudo systemctl restart apache2
mercuri@dama-server:~$
```



3.) Hacemos una copia del fichero `/etc/apache2/sites-available/000-default.conf` como `mercuri.conf` .

("sudo cp /etc/apache2/sites-available/000-default.conf
/etc/apache2/sites_available/mercuri.conf")

```
mercuri@damaserver: ~  
mercuri@damaserver:~$ sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/mercuri.conf  
mercuri@damaserver:~$ cat !$  
cat /etc/apache2/sites-available/mercuri.conf  
<VirtualHost *:80>  
    # The ServerName directive sets the request scheme, hostname and port that  
    # the server uses to identify itself. This is used when creating  
    # redirection URLs. In the context of virtual hosts, the ServerName  
    # specifies what hostname must appear in the request's Host: header to  
    # match this virtual host. For the default virtual host (this file) this  
    # value is not decisive as it is used as a last resort host regardless.  
    # However, you must set it for any further virtual host explicitly.  
    #ServerName www.example.com  
  
    ServerAdmin webmaster@localhost  
    DocumentRoot /var/www/html  
  
    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,  
    # error, crit, alert, emerg.  
    # It is also possible to configure the loglevel for particular  
    # modules, e.g.  
    #LogLevel info ssl:warn  
  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
  
    # For most configuration files from conf-available/, which are  
    # enabled or disabled at a global level, it is possible to  
    # include a line for only one particular virtual host. For example the  
    # following line enables the CGI configuration for this host only  
    # after it has been globally disabled with "a2disconf".  
    #Include conf-available/serve-cgi-bin.conf  
</VirtualHost>  
  
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet  
mercuri@damaserver:~$
```

b. **Configura** el teu nou VirtualHost. Utilitza el següent article:

- 🌐 <http://httpd.apache.org/docs/2.2/vhosts/name-based.html> : la millor font sempre és la pàgina oficial.
- 🌐 En l'article està tot en el mateix arxiu, i nosaltres el tenim repartit en 2.
- 🌐 Bàsicament es tracta de modificar el nou arxiu mercuri.conf amb:
 - Afegir la directiva **ServerName** (el default no la té ja que serveix a tothom qui ho demani).
 - Modificar **DocumentRoot** i el **Directory** per tal que apuntin a la carpeta de l'usuari mercuri (public_html).
 -

1.) Abrimos el fichero “mercuri.conf” con permiso de sudo y modificamos las líneas indicadas. (“sudo vim mercuri.conf”)

- Cambiamos el **ServerName** para nuestro “domain” (www.mercuri.local).
- Cambiamos el **DocumentRoot** para “/home/mercuri/public_html”
- No está pedido, pero es muy recomendado crear error y access logs separados para cada sitio web.

El fichero final ha quedado así:

```
dama@dama-server: ~  
NameVirtualHost *:80  
  
<VirtualHost *:80>  
    # The ServerName directive sets the request scheme, hostname and port that  
    # the server uses to identify itself. This is used when creating  
    # redirection URLs. In the context of virtual hosts, the ServerName  
    # specifies what hostname must appear in the request's Host: header to  
    # match this virtual host. For the default virtual host (this file) this  
    # value is not decisive as it is used as a last resort host regardless.  
    # However, you must set it for any further virtual host explicitly.  
    #ServerName www.example.com  
  
    ServerAdmin webmaster@localhost  
    ServerName www.mercuri.local  
    ServerAlias mercuri.local *.mercuri.local mercuri  
    DocumentRoot /home/mercuri/public_html  
  
    <Directory /home/mercuri/public_html>  
        Options FollowSymLinks  
        AllowOverride All  
        Require all granted  
    </Directory>  
  
    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,  
    # error, crit, alert, emerg.  
    # It is also possible to configure the loglevel for particular  
    # modules, e.g.  
    #LogLevel info ssl:warn  
  
    ErrorLog ${APACHE_LOG_DIR}/mercuri.error.log  
    CustomLog ${APACHE_LOG_DIR}/mercuri.access.log combined  
  
    # For most configuration files from conf-available/, which are  
    # enabled or disabled at a global level, it is possible to  
    # include a line for only one particular virtual host. For example the  
    # following line enables the CGI configuration for this host only  
    # after it has been globally disabled with "a2disconf".  
    #Include conf-available/serve-cgi-bin.conf  
</VirtualHost>
```



c. **Activar la site** amb la comanda a2ensite. Esbrina com fer-ho utilitzant el manual: **\$ man a2ensite**

1. Activamos nuestro website con el siguiente comando:
("sudo a2ensite mercuri")

Y después reiniciamos el servicio Apache2.
("sudo systemctl restart apache2")

```
dama@dama-server:~$ sudo a2ensite mercuri
Site mercuri already enabled
dama@dama-server:~$ sudo systemctl restart apache2
dama@dama-server:~$ ls -la /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 Nov 27 15:10 .
drwxr-xr-x 8 root root 4096 Nov 27 12:49 ..
lrwxrwxrwx 1 root root 35 Nov 27 15:10 000-default.conf -> ../sites-available/000-default.conf
lrwxrwxrwx 1 root root 31 Nov 27 14:04 mercuri.conf -> ../sites-available/mercuri.conf
dama@dama-server:~$
```

2. En nuestra machina de cliente tenemos que añadir explícitamente el "domain", porque no hemos dado la alta al nuestro domain en nuestro proveedor de internet.

("echo 192.168.1.64 www.mercuri.local") => (En nuestro caso este el IP local de servidor que tenemos que usar.)

```
(venv) dama@pop-os:~$ cat !$
cat /etc/hosts
127.0.0.1 localhost
::1 localhost
127.0.1.1 pop-os.localdomain pop-os

192.168.1.64 www.mercuri.local
(venv) dama@pop-os:~$
```



- d. Comprova que s'ha activat bé: ha d'haver un link simbòlic a **/etc/apache2/sites-enabled** cap al nou fitxer "mercuri.conf".

```
zeus@zeusserver:/etc/apache2/sites-available$ ls /etc/apache2/sites-available/  
000-default.conf default-ssl.conf mercuri.conf  
zeus@zeusserver:/etc/apache2/sites-available$ ll /etc/apache2/sites-enabled/  
total 8  
drwxr-xr-x 2 root root 4096 2012-01-18 06:10 ./  
drwxr-xr-x 7 root root 4096 2012-01-17 23:17 ../  
lrwxrwxrwx 1 root root 26 2012-01-17 18:15 000-default.conf ->  
../sites-available/000-default.conf  
lrwxrwxrwx 1 root root 26 2012-01-18 06:10 mercuri.conf ->  
../sites-available/mercuri.conf
```

(**"ls -la /etc/apache2/sites-enabled"**)

```
dama@damaerver:~$ sudo a2ensite mercuri  
Site mercuri already enabled  
dama@damaerver:~$ sudo systemctl restart apache2  
dama@damaerver:~$ ls -la /etc/apache2/sites-enabled/  
total 8  
drwxr-xr-x 2 root root 4096 Nov 27 15:10 .  
drwxr-xr-x 8 root root 4096 Nov 27 12:49 ..  
lrwxrwxrwx 1 root root 35 Nov 27 15:10 000-default.conf -> ../sites-available/000-default.conf  
lrwxrwxrwx 1 root root 31 Nov 27 14:04 mercuri.conf -> ../sites-available/mercuri.conf  
dama@damaerver:~$
```

- e. **Recarrega (ull, no reinicia) el servei Apache2:**

\$ sudo service apache2 reload

-El sitio web funciona en local.

```
dama@damaerver:~$ sudo service apache2 reload  
dama@damaerver:~$
```



Index of /

[Name](#) [Last modified](#) [Size](#) [Description](#)

Apache/2.4.52 (Ubuntu) Server at www.mercuri.local Port 80

.htaccess

8. Dintre del teu host virtual crea una carpeta anomenada private i restringeix l'accés mitjançant un arxiu `.htaccess`

1.) Creamos el fichero de ".htaccess".

`echo Options -Indexes > ~/public_html/.htaccess`

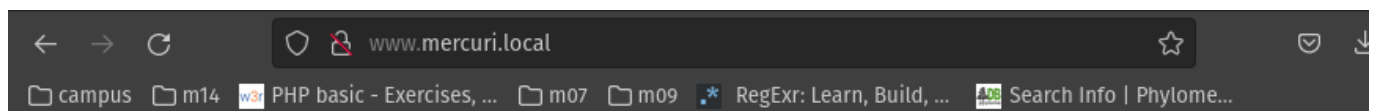
```
dama@damaserver:~/public_html$ echo Options -Indexes > ~/public_html/.htaccess
dama@damaserver:~/public_html$ cat !$
cat ~/public_html/.htaccess
Options -Indexes
dama@damaserver:~/public_html$
```

2.) Verificamos que la línea de "AllowOverride" tiene el parámetro "All".

```
<Directory /home/mercuri/public_html>
    Options FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

3.) Reiniciamos el servicio y comprobamos el url en el navegador de nuestro cliente.

```
dama@damaserver:~$ sudo systemctl restart apache2
dama@damaserver:~$
```

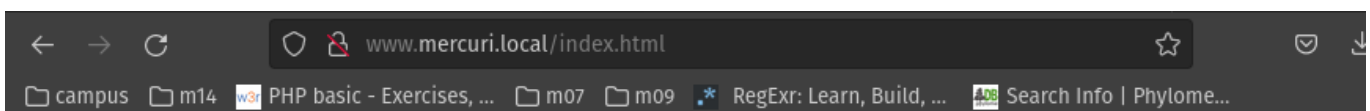


Forbidden

You don't have permission to access this resource.

Apache/2.4.52 (Ubuntu) Server at www.mercuri.local Port 80

- La página ya no nos deja listar el directorio gracias a los cambios.
- Pero sí añadimos una página de index.html, entonces eso ya nos sirve el servidor.



Hello hello, I am here



Exercici FINAL

9. Realitza el mateix exercici però amb un usuari amb el teu nom (per exemple **mercedes**) i la site amb el domini **www.elteunom.local** (per exemple www.dorayenny.local).
Penja una pàgina en PHP al teu domini.

- 1.) Creamos el usuario "mercedes".
("sudo adduser mercedes")

```
dama@dama-server:~$ sudo adduser mercedes
[sudo] password for dama:
Adding user `mercedes' ...
Adding new group `mercedes' (1007) ...
Adding new user `mercedes' (1006) with group `mercedes' ...
Creating home directory `/home/mercedes' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for mercedes
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
dama@dama-server:~$
```

- 2.) Entramos al "mercedes" y creamos el directorio "~/public_html" en su home.
("su mercedes")
("mkdir -p ~/public_html")

```
mercedes@dama-server: ~
dama@dama-server:~$ su mercedes
Password:
su: Authentication failure
dama@dama-server:~$ su mercedes
Password:
mercedes@dama-server:/home/dama$ cd
mercedes@dama-server:~$ mkdir -p ~/public_html
mercedes@dama-server:~$ ls -la
total 28
drwxr-x--- 3 mercedes mercedes 4096 Nov 27 18:04 .
drwxr-xr-x 9 root      root    4096 Nov 27 18:03 ..
-rw----- 1 mercedes mercedes   39 Nov 27 18:04 .bash_history
-rw-r--r-- 1 mercedes mercedes  220 Nov 27 18:03 .bash_logout
-rw-r--r-- 1 mercedes mercedes 3771 Nov 27 18:03 .bashrc
-rw-r--r-- 1 mercedes mercedes  807 Nov 27 18:03 .profile
drwxrwxr-x 2 mercedes mercedes 4096 Nov 27 18:04 public_html
mercedes@dama-server:~$
```




3.) Añadir los permisos al directorio "public_html".

```
-rw-r--r-- 1 mercedes mercedes 807 Nov 27 18:03 .profile
drwxrwxr-x 2 mercedes mercedes 4096 Nov 27 18:04 public_html
mercedes@damaaserver:~$ chmod -R 755 public_html
mercedes@damaaserver:~$ ll
total 28
drwxr-x--- 3 mercedes mercedes 4096 Nov 27 18:04 ./
drwxr-xr-x 9 root      root    4096 Nov 27 18:03 ../
-rw----- 1 mercedes mercedes   39 Nov 27 18:04 .bash_history
-rw-r--r-- 1 mercedes mercedes  220 Nov 27 18:03 .bash_logout
-rw-r--r-- 1 mercedes mercedes 3771 Nov 27 18:03 .bashrc
-rw-r--r-- 1 mercedes mercedes  807 Nov 27 18:03 .profile
drwxr-xr-x 2 mercedes mercedes 4096 Nov 27 18:04 public_html/
mercedes@damaaserver:~$
```

4.) Añadimos "mercedes" al sudoer.

Fuera de mercedes: ("sudo usermod -a -G sudo mercedes")

*Volver a mercedes

```
dama@damaaserver:~$ sudo usermod -a -G sudo mercedes
dama@damaaserver:~$ su mercedes
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
mercedes@damaaserver:/home/dama$ cd
mercedes@damaaserver:~$ sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/danielmajer.conf
```



5.) Crear virtualhost para "mercedes".

("sudo cp /etc/apache2/sites-available/000-default.conf
/etc/apache2/sites-available/danielmajer.conf")

```
mercedes@damaserver: ~  
dama@pop-os: ~/m14_  
<VirtualHost *:80>  
# The ServerName directive sets the request scheme, hostname and port that  
# the server uses to identify itself. This is used when creating  
# redirection URLs. In the context of virtual hosts, the ServerName  
# specifies what hostname must appear in the request's Host: header to  
# match this virtual host. For the default virtual host (this file) this  
# value is not decisive as it is used as a last resort host regardless.  
# However, you must set it for any further virtual host explicitly.  
#ServerName www.example.com  
  
ServerAdmin webmaster@localhost  
ServerName www.danielmajer.local  
ServerAlias danielmajer.local *.danielmajer.local danielmajer  
DocumentRoot /home/mercedes/public_html  
DirectoryIndex index.php  
  
<Directory /home/mercedes/public_html>  
    Options FollowSymLinks  
    AllowOverride All  
    Require all granted  
</Directory>  
  
# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,  
# error, crit, alert, emerg.  
# It is also possible to configure the loglevel for particular  
# modules, e.g.  
#LogLevel info ssl:warn  
  
ErrorLog ${APACHE_LOG_DIR}/danielmajer.error.log  
CustomLog ${APACHE_LOG_DIR}/danielmajer.access.log combined  
  
# For most configuration files from conf-available/, which are  
# enabled or disabled at a global level, it is possible to  
# include a line for only one particular virtual host. For example the  
# following line enables the CGI configuration for this host only  
# after it has been globally disabled with "a2disconf".  
#Include conf-available/serve-cgi-bin.conf  
</VirtualHost>  
  
# vim: syntax=apache ts=4 sw=4 sts=4 sr=poet
```

- 6.) Crear el fichero “.htaccess” en el dir. “~/public_html” con el contenido:
 (“echo Options -Indexes > ~/public_html/.htaccess”)

```
mercedes@damaserver: ~  
mercedes@damaserver:~$ echo Options -Indexes > ~/public_html/.htaccess  
mercedes@damaserver:~$ cat !$  
cat ~/public_html/.htaccess  
Options -Indexes  
mercedes@damaserver:~$
```

- 7.) Activamos el nuevo virtualhost con a2ensite y reiniciamos el servicio Apache.
 (“sudo a2ensite danielmajer”)
 (“sudo systemctl reload apache2”)
 (“ls -la /etc/apache2/sites-enabled”)

```
mercedes@damaserver: ~  
mercedes@damaserver:~$ sudo a2ensite danielmajer  
[sudo] password for mercedes:  
Enabling site danielmajer.  
To activate the new configuration, you need to run:  
  systemctl reload apache2  
mercedes@damaserver:~$ sudo systemctl reload apache2  
mercedes@damaserver:~$ ls -la /etc/apache2/sites-enabled  
total 8  
drwxr-xr-x 2 root root 4096 Nov 27 18:40 .  
drwxr-xr-x 8 root root 4096 Nov 27 12:49 ..  
lrwxrwxrwx 1 root root 35 Nov 27 15:10 000-default.conf -> ../sites-available/000-default.conf  
lrwxrwxrwx 1 root root 35 Nov 27 18:40 danielmajer.conf -> ../sites-available/danielmajer.conf  
lrwxrwxrwx 1 root root 31 Nov 27 14:04 mercuri.conf -> ../sites-available/mercuri.conf  
mercedes@damaserver:~$
```

- 8.) Añadir el domain “www.danielmajer.local” al /etc/hosts en nuestro cliente.
 (“sudo nvim /etc/hosts”)
Contenido: (“192.168.1.64 www.danielmajer.local”)

```
mercedes@damaserver: ~  
127.0.0.1 localhost  
::1 localhost  
127.0.1.1 pop-os.localdomain pop-os  
192.168.1.64 www.mercuri.local  
192.168.1.64 www.danielmajer.local
```

9.) Comentar los siguientes lineas en el “/etc/apache2/mods-available/php8.1.conf”:

```
require all denied
</FilesMatch>

# Running PHP scripts in user directories is disabled by default
#
# To re-enable PHP in user directories comment the following lines
# (from <IfModule ...> to </IfModule>.) Do NOT set it to On as it
# prevents .htaccess files from disabling it.
# <IfModule mod_userdir.c>
#     <Directory /home/*/public_html>
#         php_admin_flag engine Off
#     </Directory>
# </IfModule>
~
~
```

10.) Crear una página en PHP en el fichero “~/public_html/index.php”

```
mercedes@damaserver: ~
dama@pop-os: ~/m14

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="author" content="Dániel Májer">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>M08 UF1 NF1 Pt1a - Dániel Májer</title>
</head>
<body>

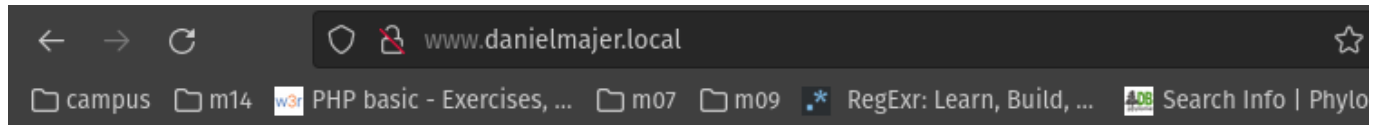
    <?php

        echo "<p><b><u> Author of this awesome PHP page is: Dániel Májer!</u></b></p>";
        for($i = 0; $i < 10; $i++) {
            echo "<p> This is my awesome PHP page... Just kidding. </p>";
        }

    ?>

</body>
</html>"'
```

11.) [Comprobar sí funciona la página.](#)



Author of this awesome PHP page is: Dániel Májer!

This is my awesome PHP page... Just kidding.

This is my awesome PHP page... Just kidding.

This is my awesome PHP page... Just kidding.

This is my awesome PHP page... Just kidding.

This is my awesome PHP page... Just kidding.

This is my awesome PHP page... Just kidding.

This is my awesome PHP page... Just kidding.

This is my awesome PHP page... Just kidding.

This is my awesome PHP page... Just kidding.

This is my awesome PHP page... Just kidding.

'''