

# **Práctica UD2: Servicio web Apache**

Desenvolvemento de aplicacións

**MP0614. Despregamento de aplicacións web**

## Sumario

Instrucciones .....	3
Práctica de Servicios Web .....	4
1.1. Instalación de Servidor Web Apache .....	4
1.2. Virtual Hosts .....	5
1.3. Configura Apache para que use HTTPS .....	6
1.4. Redirección de tráfico .....	7
1.5. Página 404 personalizada .....	8
1.6. Archivo .htaccess .....	9
1.7 Autenticación .....	10
1.8. Instalación de módulos .....	11
ANEXO. Información de apoyo. ....	12

## Instrucciones

- Las capturas de las máquinas virtuales deben mostrar el nombre de la máquina.
- En el nombre de la máquina virtual debe contener la inicial y el apellido del alumno/a que entrega la práctica.
  - Por ejemplo, si creo una máquina virtual llamada "vsFTPd Server", debo nombrarla "jlopez vsFTPd Server".
- Las capturas deben de tener una calidad suficiente para que su contenido pueda ser legible.
- La entrega será en la tarea de la plataforma moodle mediante un fichero pdf practica\_x\_tu\_nombre.pdf (x es número de practica y tu\_nombre es tu nombre) en el que se puedan ver en las diferentes secciones lo solicitado.

# Práctica de Servicios Web

## 1.1. Instalación de Servidor Web Apache

/\*Instala el servicio de Apache en una máquina virtual (te sugiero distro Ubuntu o Debian).

Comprueba mediante linea de comandos que el servicio está activo

Visita la página web de Apache por defecto

Aporta capturas de pantalla.\*/

Actualizamos los repositorios de la maquina usando:

**apt-get update**

una vez actualizado, instalamos apache con el siguiente comando:

**apt-get install apache2**

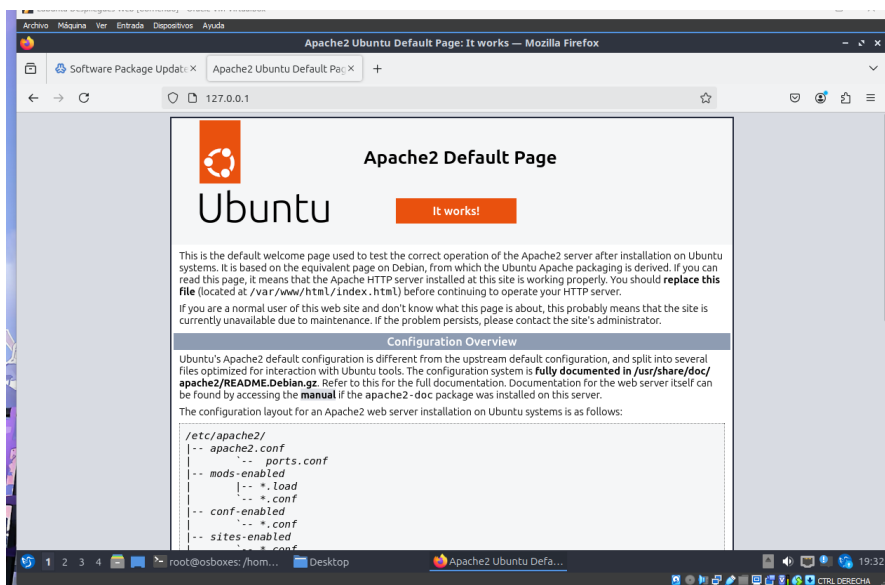
Podemos comprobar que se ha instalado viendo el estado de su servicio:

**service apache2 status**

```
root@osboxes: /home/osboxes# service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-09-13 19:29:15 EDT; 2min 16s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 22753 (apache2)
    Tasks: 55 (limit: 4602)
   Memory: 5.5M (peak: 6.1M)
      CPU: 34ms
   CGroup: /system.slice/apache2.service
           └─22753 /usr/sbin/apache2 -k start
             22756 /usr/sbin/apache2 -k start
             22757 /usr/sbin/apache2 -k start

Sep 13 19:29:15 osboxes systemd[1]: Starting apache2.service - The Apache HTTP Server...
Sep 13 19:29:15 osboxes apache2[22752]: AH00558: apache2: Could not reliably determine the
Sep 13 19:29:15 osboxes systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-16/16 (END)
```

Accedemos desde el navegador a la página por defecto de apache en 127.0.0.1:80



## 1.2. Virtual Hosts

/\*Los Virtual Hosts son un mecanismo que nos permiten que un servidor web pueda servir varias páginas web distintas.

Crea una página web un HTML básico que simplemente ponga tu nombre. Y sítela a través de un dominio tunombre.com.

Sirve una página en un puerto diferente al 80 que simplemente ponga tu apellido. Sítela a través del nombre tuapellido.com.

En ambos casos, aporta captura de pantalla de un navegador visitando la página, y de los archivos de configuración empleados para cada virtualhost.\*/

\*\*Las capturas anteriores son de documentos que ya tenía hechos.

Vamos a crear un nuevo sitio en nuestro apache para ello copiaremos el archivo de configuración por defecto de apache ubicado en /etc/apache2/sites-available/ llamado 000-default.conf y le llamaremos a la copia david.conf

```
root@debian12:/etc/apache2/sites-available# cp 000-default.conf david.conf
root@debian12:/etc/apache2/sites-available# ls
000-default.conf  david.conf  default-ssl.conf  joomla.conf
```

Modificamos el archivo de configuración creado:

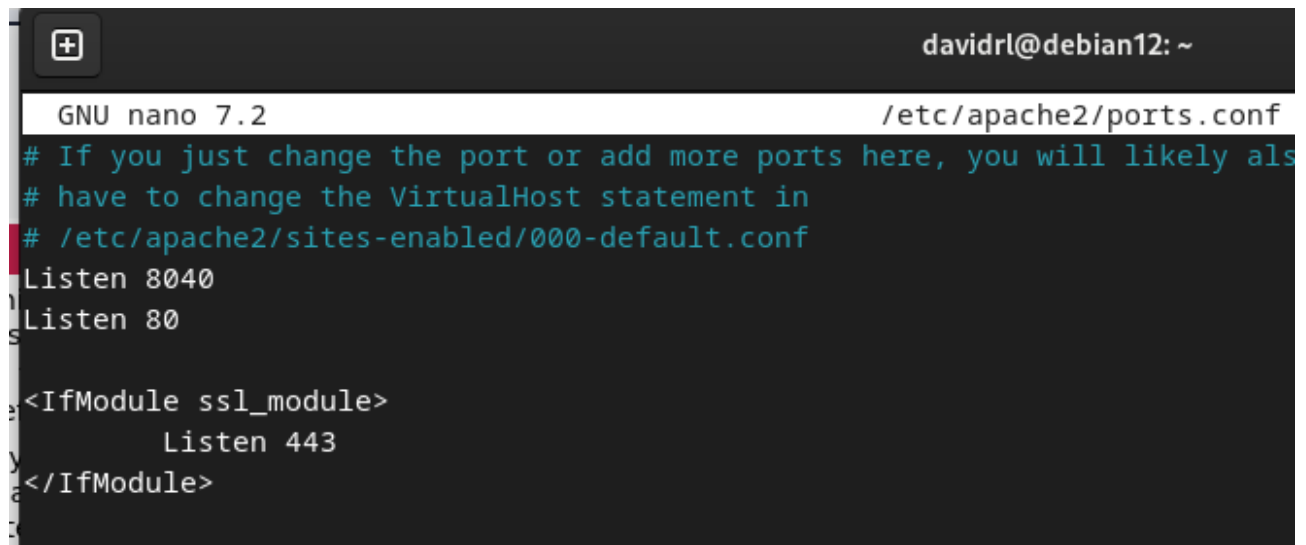
```
GNU nano 7.2                                david.conf *
```

```
<VirtualHost *:8040>
    # The ServerName directive sets the request scheme, host
    # the server uses to identify itself. This is used when
    # redirection URLs. In the context of virtual hosts, the
    # specifies what hostname must appear in the request's H
    # match this virtual host. For the default virtual host
    # value is not decisive as it is used as a last resort h
    # However, you must set it for any further virtual host
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    ServerName david.com
    DocumentRoot /var/www/david
    DirectoryIndex david.html
```

Le cambiamos el puerto por defecto por el puerto 8040, le indicamos que el dominio para este sitio es david.com , le indicamos que el directorio root de este sitio es /var/www/David y le decimos que el documento índice de este sitio es david.html

Guardamos y vamos a modificar el archivo `/etc/apache2/ports.conf` para indicarle a `apache2` que debe escuchar también por el puerto 8040



```
GNU nano 7.2 /etc/apache2/ports.conf
# If you just change the port or add more ports here, you will likely als
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf
Listen 8040
Listen 80

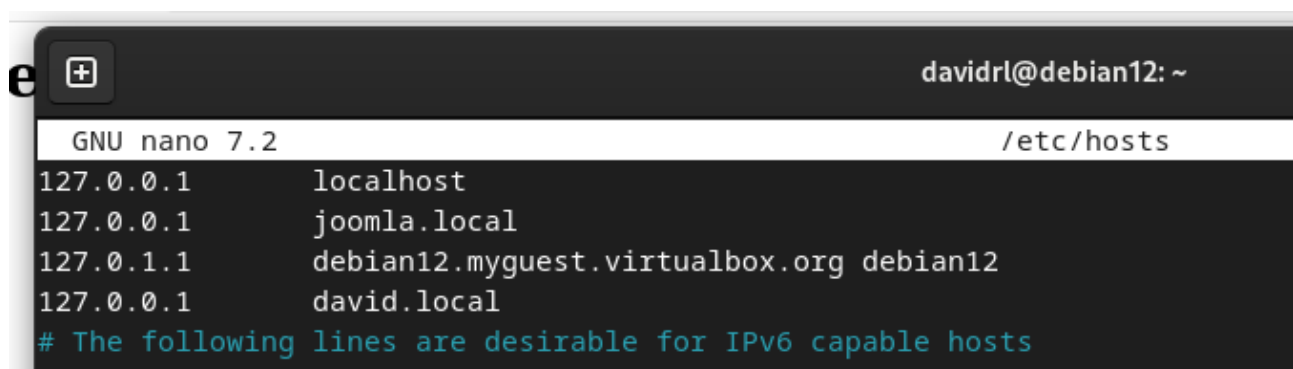
<IfModule ssl_module>
    Listen 443
</IfModule>
```

Creamos el directorio `david` y el `david.html`:



```
root@debian12:/var/www# mkdir david
root@debian12:/var/www# cd david/
root@debian12:/var/www/david# nano david.html
root@debian12:/var/www/david# cat david.html
<title>Sitio de Davidrl</title>
<h1>Este es el sitio de Davidrl</h1>
root@debian12:/var/www/david#
```

Como no tenemos un dominio realmente para nuestro sitio , vamos a indicarle a nuestra maquina que puede resolver la dirección `david.com` con `localhost`, para ello modificamos el archivo `etc/hosts`:



```
GNU nano 7.2 /etc/hosts
127.0.0.1 localhost
127.0.0.1 joomla.local
127.0.1.1 debian12.myguest.virtualbox.org debian12
127.0.0.1 david.local
# The following lines are desirable for IPv6 capable hosts
```

Una vez hecho, reiniciamos el servicio de apache y comprobamos el acceso:



Ahora repetimos los pasos para rodriguez.local, este lo colocaremos en el puerto 8050:



### 1.3. Configura Apache para que use HTTPS

/\* El protocolo HTTP sin cifrar es inseguro en aquellas páginas que requieren login, puesto que la información se transmite en texto plano.

Configura tu servidor Apache para que sirva la página con tunombre.com por HTTPS mediante un certificado autofirmado.

Aporta captura de pantalla.

\*/

Para configurar nuestro apache para usar HTTPS debemos primero debemos disponer de un certificado, en entornos reales usaríamos una entidad certificadora que nos lo de, pero también tenemos la opción de crear un certificado autofirmado, que es lo que vamos a hacer.

Antes de nada, en caso de que tuviéramos un firewall, deberíamos indicarle que permita peticiones https para el servicio de apache, para ello usamos el siguiente comando:

**ufw allow "Apache Full"**

Una vez permitido en el firewall, vamos a generar el certificado, para ello usamos el siguiente comando:

**sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout  
/etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt**

```
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:A Coruna
Locality Name (eg, city) []:Ferrol
Organization Name (eg, company) [Internet Widgits Pty Ltd]:davidrl
Organizational Unit Name (eg, section) []:Despliegue de Apps
Common Name (e.g. server FQDN or YOUR name) []:david.local
Email Address []:ejemplo@local.com
root@debian12:/etc/apache2/sites-available#
```

**\*\*Importante: en el Common Name, debemos usar el nombre que usaremos después como DNS**



Al crear el certificado, hemos generado dos archivos, la llave y el certificado, los podemos encontrar en las siguientes rutas:

Llave: /etc/ssl/private/apache-selfsigned.key

Certificado: /etc/ssl/certs/apache-selfsigned.crt

```
root@debian12:/etc/apache2/sites-available# ls /etc/ssl/private/
apache-selfsigned.key  ssl-cert-snakeoil.key
root@debian12:/etc/apache2/sites-available# ls /etc/ssl/certs/ |grep apache
apache-selfsigned.crt
root@debian12:/etc/apache2/sites-available#
```

Ahora debemos configurar el sitio para que haga uso de TLS, para ello vamos a su archivo de configuración , debemos cambiar el puerto por el 443 y añadir las siguientes entradas:

SSLEngine on

SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt

SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

```
GNU nano 7.2                                david.conf *
<VirtualHost *:443>
    # 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 david.local
    DocumentRoot /var/www/david
    DirectoryIndex david.html

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
```

Ahora debemos habilitar el ssl en nuestro apache , usamos el siguiente comando:

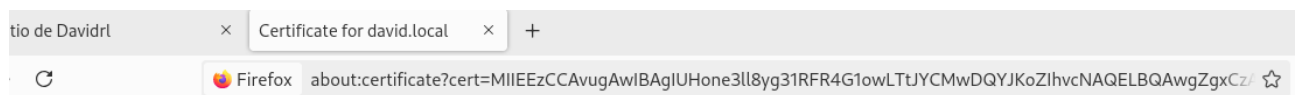
**a2enmod ssl**

```
root@debian12:/etc/apache2/sites-available# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
root@debian12:/etc/apache2/sites-available#
```

Reiniciamos el servicio de apache y comprobamos:



Vemos el certificado:



### Certificate

david.local	
<b>Subject Name</b>	
Country	ES
State/Province	A Coruna
Locality	Ferrol
Organization	davidrl
Organizational Unit	Despliegue de Apps
Common Name	david.local
Email Address	ejemplo@local.com

## 1.4. Redirección de tráfico

/\*Configura el servidor Apache para que redirija el tráfico de <http://tunombre.com> a <https://tunombre.com>.

Describe qué cambios has tenido que hacer para que esto suceda, y aporta captura de pantalla.\*/\*

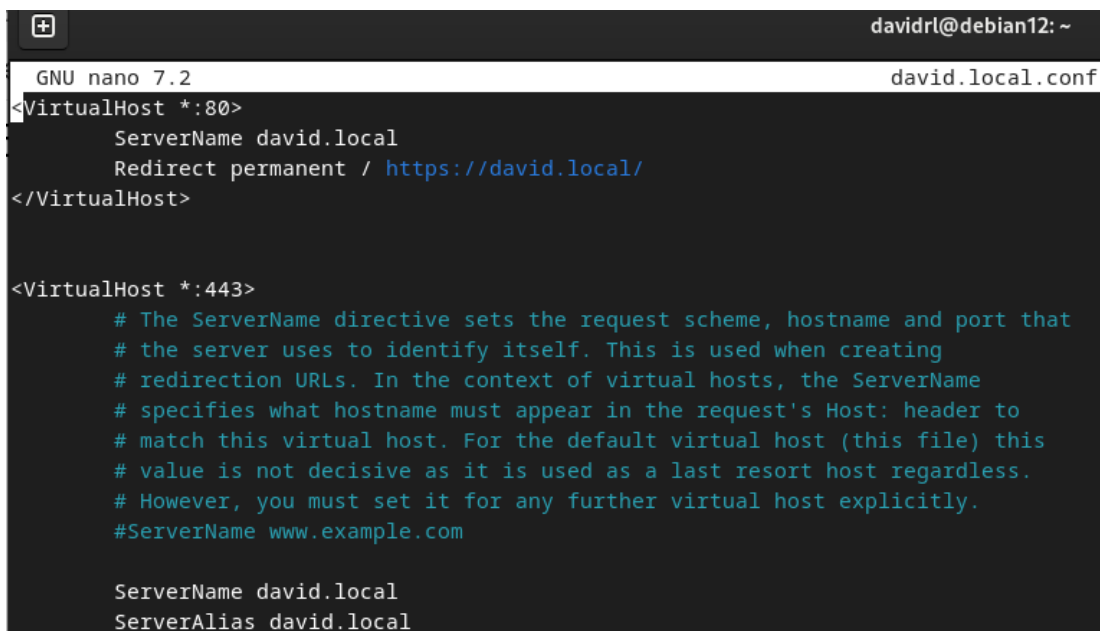
Para configurar la redirección debemos ir al archivo de configuración del sitio y añadimos la siguiente entrada:

```
<VirtualHost *:80>

    ServerName your_domain_or_ip

    Redirect permanent / https://your_domain_or_ip/

</VirtualHost>
```



```
GNU nano 7.2                                davidrl@debian12: ~
david.local.conf
<VirtualHost *:80>
    ServerName david.local
    Redirect permanent / https://david.local/
</VirtualHost>

<VirtualHost *:443>
    # 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

    ServerName david.local
    ServerAlias david.local
```

## 1.5. Página 404 personalizada

Configura tu Apache para que muestre una página 404 que hayas personalizado y aporta captura de pantalla.

Vamos al archivo de configuración de nuestro sitio, y añadimos la siguiente entrada:

**ErrorDocument** **CodigoError rutaHtml**

Queda así en nuestro sitio:

```
davidrl@debian12: ~
GNU nano 7.2 david.local.conf
# 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

ServerName david.local
ServerAlias david.local

DocumentRoot /var/www/david
DirectoryIndex "david.html"

ErrorDocument 404 /notFound.html

SSLEngine on
SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
```



## 1.6. Archivo .htaccess

Los archivos .htaccess permiten que los propios usuarios puedan adaptar las configuraciones de Apache a sus sitios web. A cambio, se pierde un poco de rendimiento, puesto que el servidor tendrá que acceder e interpretar estos ficheros.

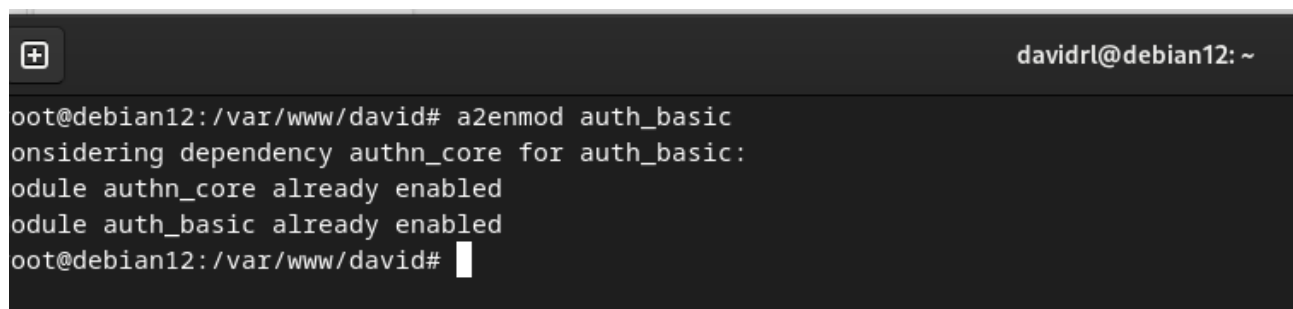
Toda configuración que se pueda poner en un .htaccess se puede también poner en la configuración general del sitio.

Crea un archivo .htaccess que impida que los archivos de un directorio se listen cuando se visitan a través de una URL. Por ejemplo, que al visitar <https://tunombre.com/directorio1> se muestren los archivos y al visitar <https://tunombre.com/directorio2> no se muestren.

Ahora voy a crear un par de directorios debajo de david.local llamada privado y otra llamada publico, configurare autenticación de htaccess para que al intentar entrar en privado nos pida autenticarnos.

Primero comprobamos que el modulo auth\_basic esta activo:

**a2enmod auth\_basic**



```
davidrl@debian12: ~  
oot@debian12:/var/www/david# a2enmod auth_basic  
Considering dependency authn_core for auth_basic:  
Module authn_core already enabled  
Module auth_basic already enabled  
oot@debian12:/var/www/david#
```

En el directorio que queremos proteger, creamos un archivo llamado .htaccess y colocamos las siguientes entradas dentro:

AuthName "Acceso restringido"

AuthType Basic

AuthUserFile "/var/www/.htpasswd"

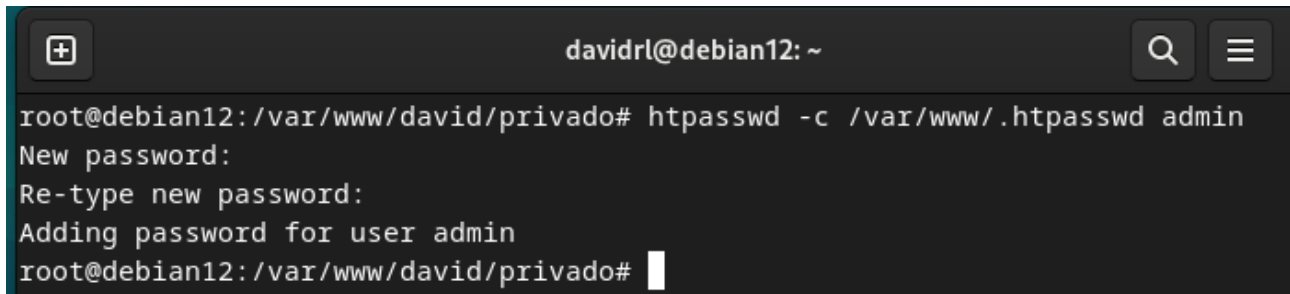
Require valid-user



```
davidrl@debian12: ~  
GNU nano 7.2 .htaccess *  
AuthName "Acceso restringido"  
AuthType Basic  
AuthUserFile "/var/www/.htpasswd"  
Require valid-user
```

Ahora creamos el archivo htpasswd, este siempre tiene que ir fuera del sitio, por seguridad, usamos el comando:

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

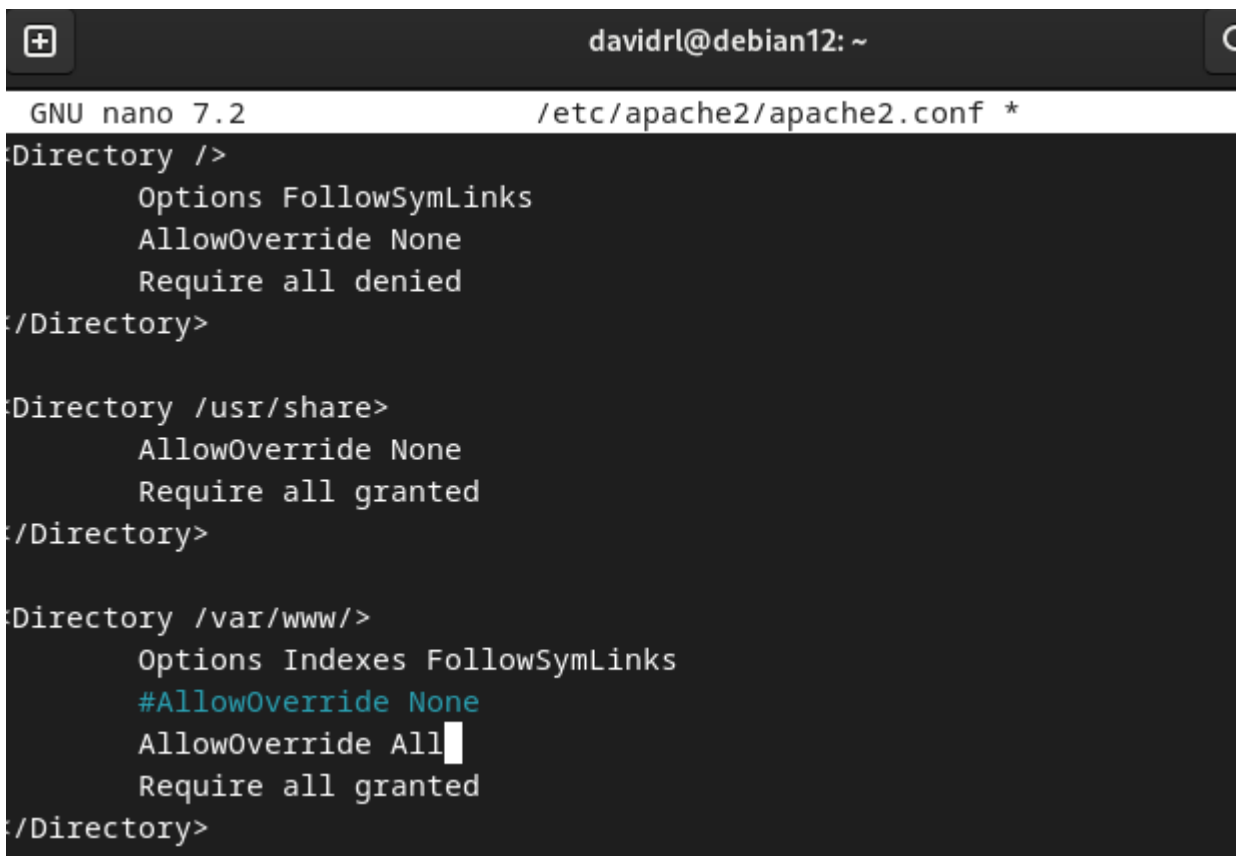
A terminal window titled 'davidrl@debian12: ~' showing the execution of the 'htpasswd' command. The user is root at debian12 in the directory /var/www/david/privado. The command 'htpasswd -c /var/www/.htpasswd admin' is entered. The prompt 'New password:' is shown, followed by 'Re-type new password:'. Then, 'Adding password for user admin' is displayed. Finally, the prompt returns to 'root@debian12:/var/www/david/privado#'.

```
davidrl@debian12: ~
root@debian12:/var/www/david/privado# htpasswd -c /var/www/.htpasswd admin
New password:
Re-type new password:
Adding password for user admin
root@debian12:/var/www/david/privado#
```

Ahora hemos creado el archivo .htpasswd en el directorio /var/www/

Una vez hecho esto, debemos ir al archivo de configuración general de apache, ubicado en /etc/apache2/apache2.conf

Debemos ir al apartado <Directory /var/www> comentar la entrada "AllowOverride None" y añadir "AllowOverride All"

A terminal window titled 'davidrl@debian12: ~' showing the nano editor editing the file /etc/apache2/apache2.conf. The editor shows the configuration for the /var/www/ directory. The line '#AllowOverride None' is highlighted in blue, and 'AllowOverride All' is being typed in. The other configuration lines for /var/www/ are 'Options Indexes FollowSymLinks' and 'Require all granted'.

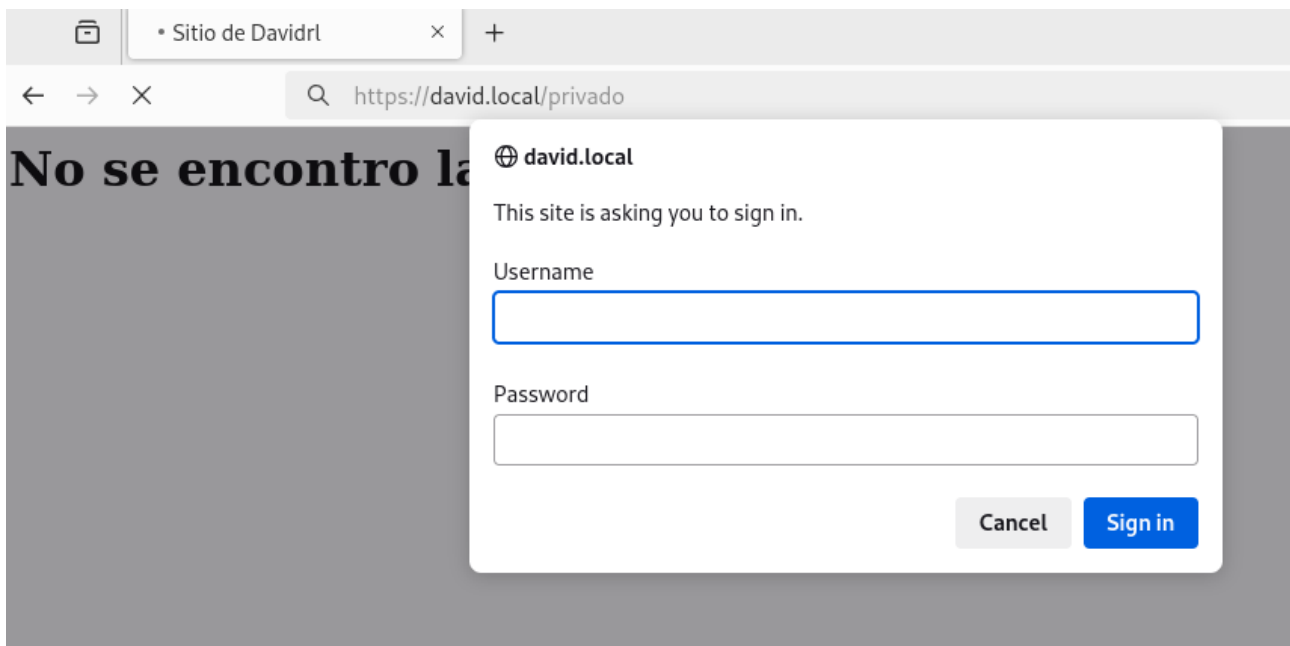
```
davidrl@debian12: ~
GNU nano 7.2 /etc/apache2/apache2.conf *
Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

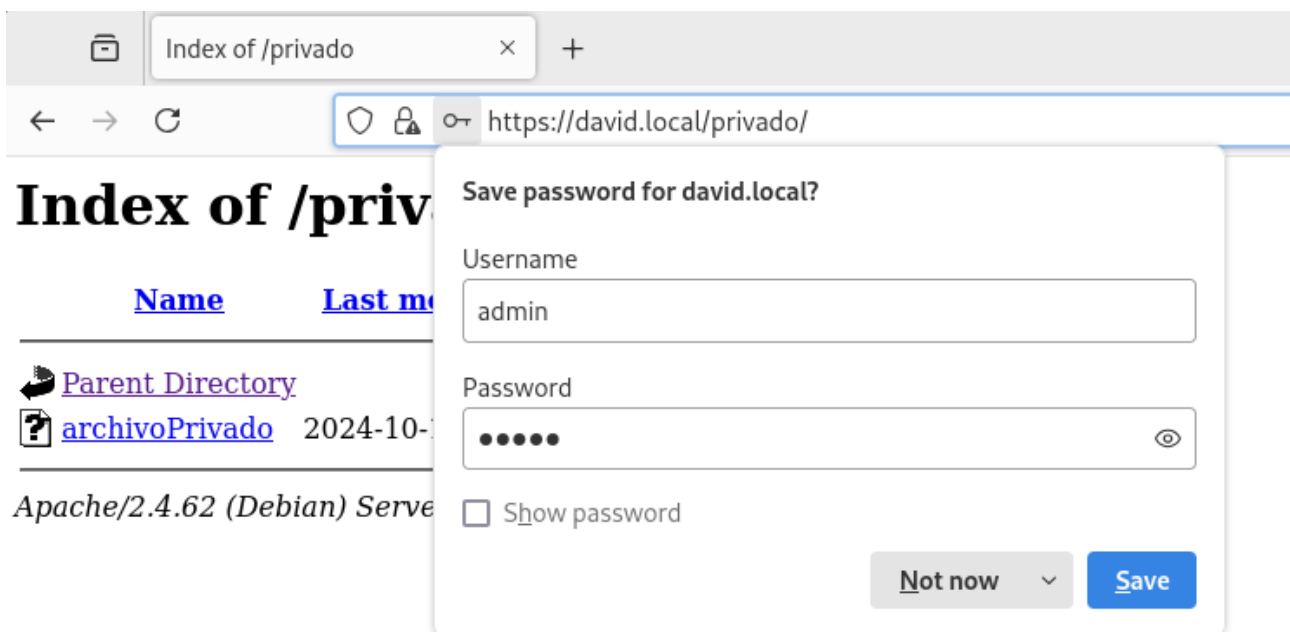
Directory /var/www/>
    Options Indexes FollowSymLinks
    #AllowOverride None
    AllowOverride All
    Require all granted
</Directory>
```

Guardamos y reiniciamos el servicio de apache2.

Ahora comprobamos:



Usamos el usuario admin para logar:



## 1.7 Autenticación

¿Qué diferencias encuentras entre la autenticación básica y la autenticación digest?

Configura con uno de los dos modos de autenticación un directorio, y muestra el correcto funcionamiento.

Para la autenticación con digest, debemos primero habilitar el modulo `auth_digest` de apache, usamos el siguiente comando:

`a2enmod auth_digest`

```
root@debian12:/var/www/david/privado# a2enmod auth_digest
Considering dependency authn_core for auth_digest:
Module authn_core already enabled
Enabling module auth_digest.
To activate the new configuration, you need to run:
    systemctl restart apache2
root@debian12:/var/www/david/privado#
```

Ahora vamos al archivo de configuración de nuestro sitio y añadimos las siguientes entradas:

`<Directory "/var/www/html/pagina1/privado">`

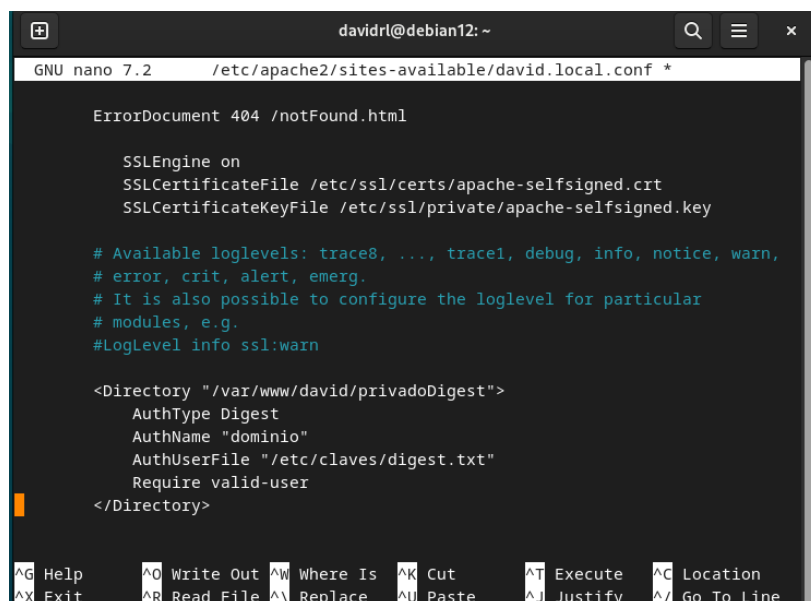
`AuthType Digest`

`AuthName "dominio"`

`AuthUserFile "/etc/claves/digest.txt"`

`Require valid-user`

`</Directory>`



```
davidrl@debian12: ~
GNU nano 7.2 /etc/apache2/sites-available/david.local.conf *

ErrorDocument 404 /notFound.html

SSLEngine on
SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

# 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

<Directory "/var/www/david/privadoDigest">
    AuthType Digest
    AuthName "dominio"
    AuthUserFile "/etc/claves/digest.txt"
    Require valid-user
</Directory>
```

Para hacer la prueba he creado otro directorio en nuestro sitio llamado `privadoDigest`.



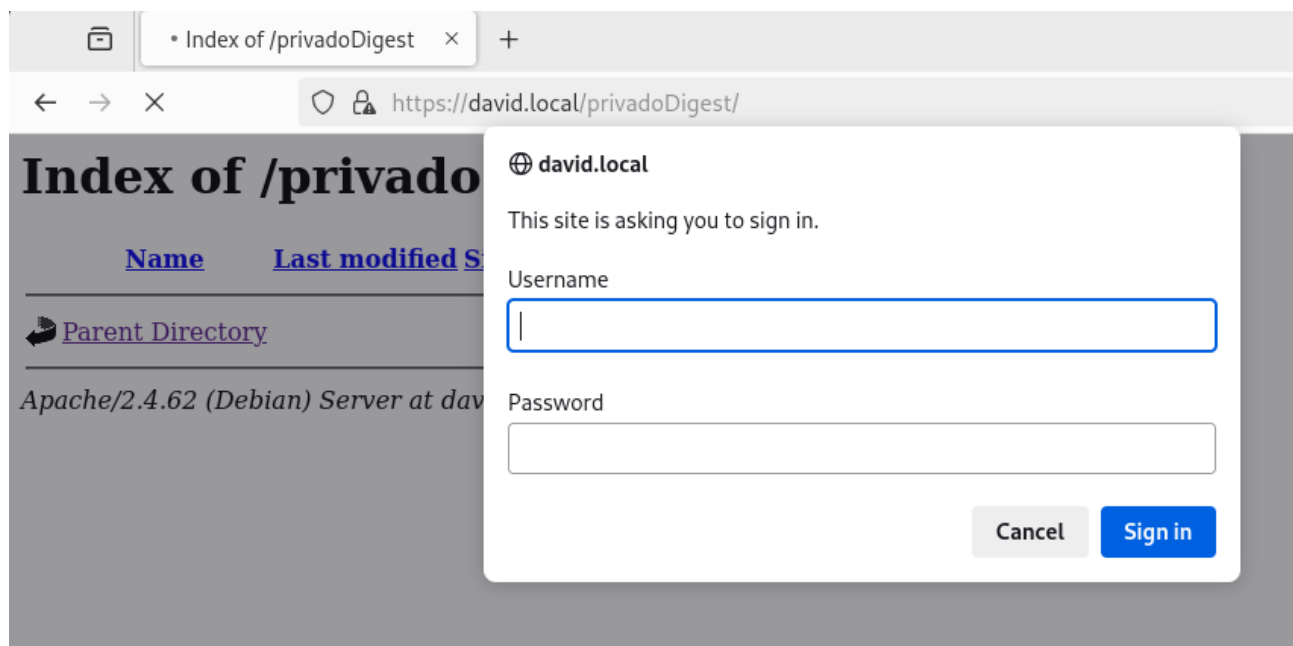
Creamos el directorio /etc/claves/ y creamos un fichero de contraseñas usando htdigest con el siguiente comando:

```
htdigest -c /etc/claves/digest.txt dominio usuario
```

**\*\*El parámetro -c se usara la primera vez, este parámetro es para crear el fichero, cuando queramos añadir mas usuarios, se hara sin dicho parametro**

```
root@debian12:/var/www/david# htdigest -c /etc/claves/digest.txt david.local admin
Adding password for admin in realm david.local.
New password:
Re-type new password:
root@debian12:/var/www/david#
```

Ahora intentamos acceder:



Ventajas: Las contraseñas son mas seguras en Digest ya que van cifradas.

Desventajas: La configuración es algo mas compleja.

## 1.8. Instalación de módulos

Instala el módulo de PHP en Apache y crea una página de prueba que demuestre que está funcionando. Por ejemplo, el módulo userdir (que hemos trabajado en clase)

Primero vamos a habilitar el modulo de userdir y ponerlo en marcha, usamos el siguiente comando para activarlo:

**a2enmod userdir**

```
root@debian12:/var/www/david# a2enmod userdir
Enabling module userdir.
To activate the new configuration, you need to run:
    systemctl restart apache2
root@debian12:/var/www/david#
```

Esto nos permitirá crear un sitio en el /home de cada usuario.

Crearemos debajo del /home del usuario davidrl el directorio public\_html , este directorio hará uso de userdir para publicar el sitio.

Ahora vamos a instalar php y posteriormente el modulo de php de Apache, usamos los siguientes comandos:

**apt-get install php**

**a2enmod proxy\_fcgi setenvif**

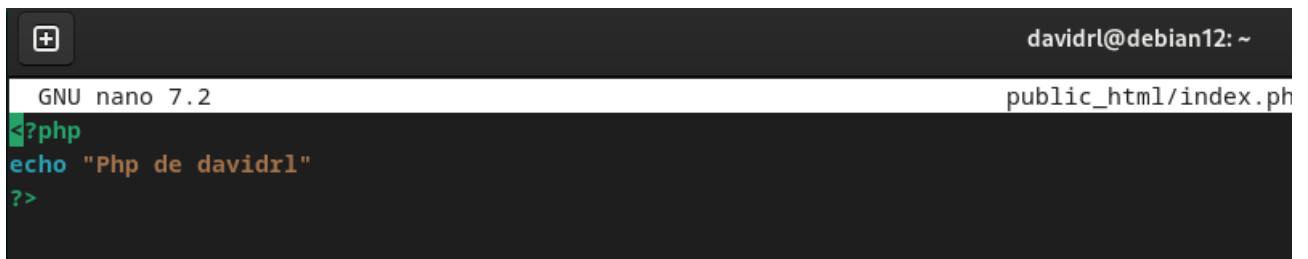
**a2enconf php8.2-fpm**

Ahora vamos a la configuración de userdir y agregamos la entrada DirectoryIndex index.php y modificamos Require a Require all granted para permitir el acceso.

```
GNU nano 7.2 /etc/apache2/m
UserDir public_html
UserDir disabled root

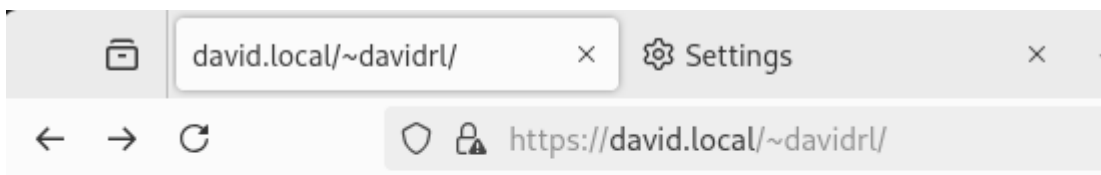
<Directory /home/*/public_html>
    AllowOverride FileInfo AuthConfig Limit Indexes
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    Require all granted
    DirectoryIndex index.php
</Directory>
```

He creado el siguiente index.php en el directorio public\_html debajo del /home del usuario davidrl



```
GNU nano 7.2 public_html/index.php
<?php
echo "Php de davidrl"
?>
```

Accedemos a la pagina del usuario en /~usuario dentro de nuestro sitio:



## **ANEXO. Información de apoyo.**

Estos enlaces te pueden servir de apoyo y guía para la realización de las tareas.

En general, para todas las tareas de este módulo, una muy buena referencia es esta:

[https://www.server-world.info/en/note?os=Ubuntu\\_22.04&p=httpd&f=1](https://www.server-world.info/en/note?os=Ubuntu_22.04&p=httpd&f=1)

Otras:

<https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-22-04>

<https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-apache-in-ubuntu-22-04>

<https://help.dreamhost.com/hc/en-us/articles/215747718-Control-directory-indexes-with-an-htaccess-file>

<https://plataforma.josedomingo.org/pledin/cursos/apache24/curso/u14/>

[https://wiki.cifprodolfoucha.es/index.php?title=Autenticación\\_en\\_Apache\\_2.4](https://wiki.cifprodolfoucha.es/index.php?title=Autenticación_en_Apache_2.4)

[https://wiki.cifprodolfoucha.es/index.php?title=Módulo\\_UserDir\\_en\\_Apache\\_2.4](https://wiki.cifprodolfoucha.es/index.php?title=Módulo_UserDir_en_Apache_2.4)