



23 DE NOVIEMBRE DE 2022

UNIDAD 2: SERVICIO WEB


PRÁCTICA 2.1: SERVIDOR WEB

ALBERTO GARCÍA NAVARRO

DESPLIEGUE DE APLICACIONES WEB

2º CFGS DAW

Curso 2022-2023



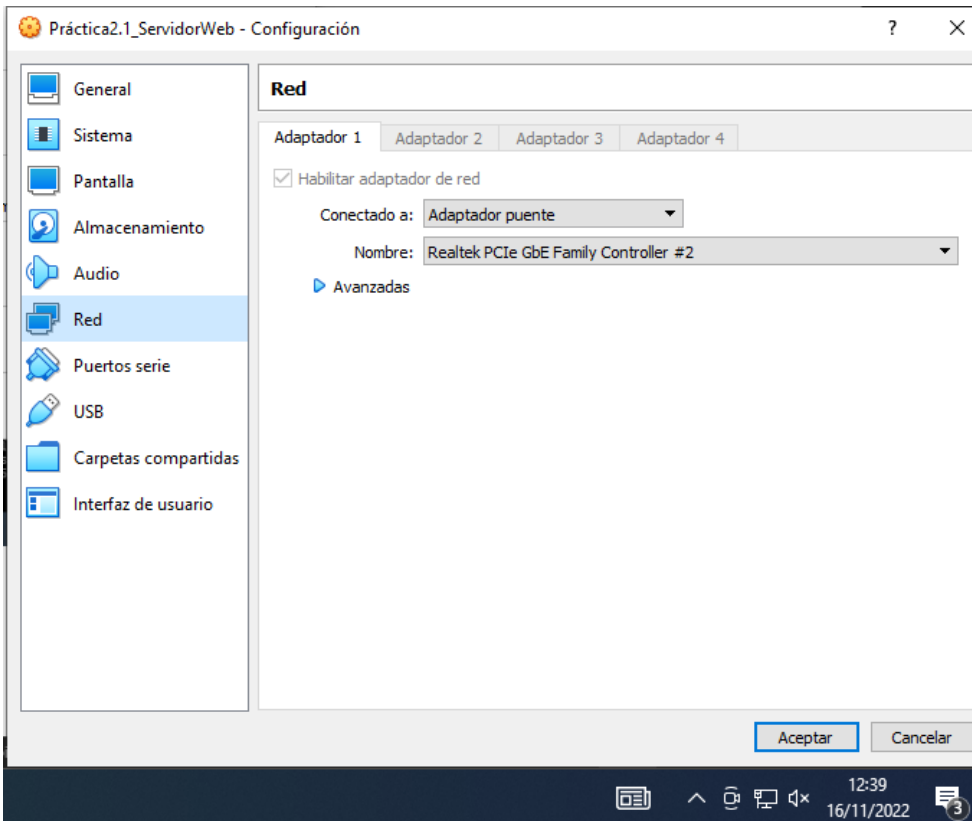
ÍNDICE

A) Instalación del servidor web Apache.....	pág. 2
A.1) Instalación apache en Linux.....	pág. 2
B) Ficheros de configuración y directivas en Apache para Linux.....	pág. 4
B.1) Análisis de los ficheros apache2.conf, port.conf y 000-default.conf.....	pág. 4
C) Configuración básica en Apache para Linux.....	pág. 7
C.1) Ficheros y directorios de prueba.....	pág. 7
C.2) Archivos a servir por defecto (Directory Index).....	pág. 8
C.3) Opciones sobre directorios (Options Indexes).....	pág. 9
C.4) Códigos de error (ErrorDocument).....	pág. 10
C.5) Directorios virtuales (Alias).....	pág. 11
C.6) Redirecciones (Redirect).....	pág. 12
D) GitHub.....	pág. 13

A) Instalación del servidor web Apache.

A.1) Instalación apache en Linux.

Antes de nada, deberemos configurar la red de nuestra máquina virtual como Adaptador puente.



PASO 1) Actualizamos nuestro sistema Linux con “apt-get update” e instalamos la versión 2 de apache con “apt-get install apache2”.

Comprobamos que existe la carpeta “/var/www/html” donde se alojarán las páginas de nuestro servidor.

```
agn@servidoragn:~$ cd /var/www/html/
agn@servidoragn:/var/www/html$ ls
index.html
agn@servidoragn:/var/www/html$ _
```

PASO 2) Con la instalación de apache se nos ha creado el usuario “www-data”, lo comprobamos con el comando “ps -ef | grep apache”. Ahora el servidor está en escucha, lo comprobamos con “netstat -ltn” y observamos que su **puerto de escucha es el 80**.

```
agn@servidoragn:~$ ps -ef | grep apache
root      728      1    0 11:36 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  733     728    0 11:36 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  734     728    0 11:36 ?        00:00:00 /usr/sbin/apache2 -k start
agn       1095    1084    0 11:41 tty1    00:00:00 grep --color=auto apache

agn@servidoragn:~$ netstat -ltn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:53            0.0.0.0:*                LISTEN
tcp6       0      0 :::80                   :::*                     LISTEN
agn@servidoragn:~$ _
```

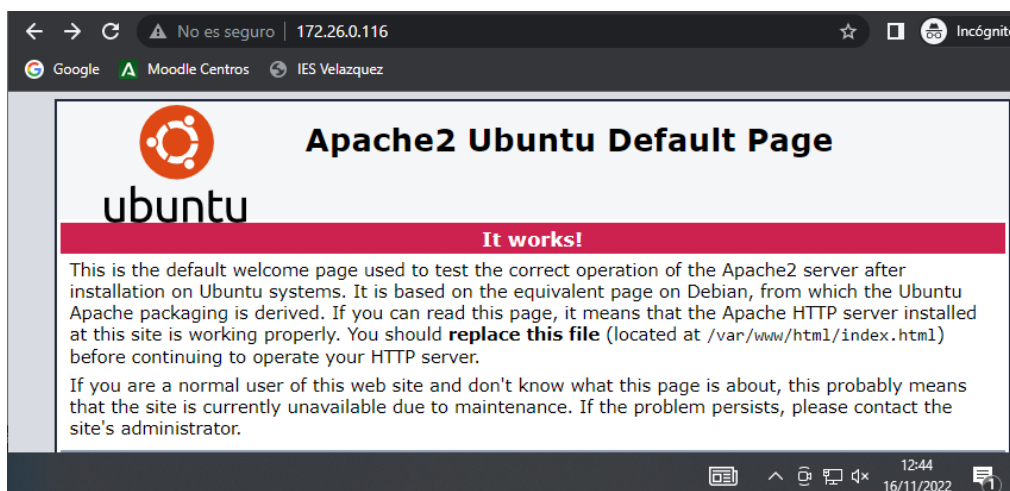
PASO 3) Verificamos la versión instalada de apache con el comando “`apache2 -v`” y vemos que tenemos la **versión 2.4.41**. Al comprobar el archivo que está dentro de la carpeta “`/var/www/html`” con el comando “`ls /var/www/html`” vemos que se nos ha creado un html por defecto para nuestro servidor.

```
agn@servidoragn:~$ apache2 -v
Server version: Apache/2.4.41 (Ubuntu)
Server built: 2022-06-14T13:30:55
agn@servidoragn:~$ ls /var/www/html
index.html
agn@servidoragn:~$ _
```

```
<div class="section_header section_header_red">
  <div id="about"></div>
  It works!
</div>
<div class="content_section_text">
  <p>
    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 <b>replace this file</b> (located at
    <tt>/var/www/html/index.html</tt>) before continuing to operate your HTTP server
  </p>

  <p>
    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.
  </p>
</div>
<div class="section_header">
  <div id="changes"></div>
  Configuration Overview
</div>
<div class="content_section_text">
  <p>
    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
    <b>fully documented in
    /usr/share/doc/apache2/README.Debian.gz</b>. Refer to this for the full
    documentation. Documentation for the web server itself can be
  </p>
--More--
```

PASO 4) Al introducir la IP de nuestra máquina Linux (172.26.0.116) se nos abre el index.html creado por defecto al instalar el servidor apache2.



B) Ficheros de configuración y directivas en Apache para Linux.

B.1) Análisis de los ficheros apache2.conf, port.conf y 000-default.conf.

PASO 1 Y 2) Al hacer un listado del directorio “/etc/apache2/” vemos todos los archivos de configuración disponibles.

```
root@servidoragn:/etc/apache2# ls -l
total 80
-rw-r--r-- 1 root root 7224 jun 14 13:30 apache2.conf
drwxr-xr-x 2 root root 4096 nov 15 09:16 conf-available
drwxr-xr-x 2 root root 4096 nov 15 09:16 conf-enabled
-rw-r--r-- 1 root root 1782 feb 23 2021 envvars
-rw-r--r-- 1 root root 31063 feb 23 2021 magic
drwxr-xr-x 2 root root 12288 nov 15 09:16 mods-available
drwxr-xr-x 2 root root 4096 nov 15 09:16 mods-enabled
-rw-r--r-- 1 root root 320 feb 23 2021 ports.conf
drwxr-xr-x 2 root root 4096 nov 15 09:16 sites-available
drwxr-xr-x 2 root root 4096 nov 15 09:16 sites-enabled
root@servidoragn:/etc/apache2#
```

PASO 3) Abrimos el archivo “apache2.conf” y analizamos las directivas:

El Path de la directiva **ServerRoot** es “/etc/apache2”.

```
# Do NOT add a slash at the end of the directory path.
#
#ServerRoot "/etc/apache2"
```

Observamos que la directiva que decide permitir o denegar conexiones persistentes es **KeepAlive**, que **por defecto está en ON (permite)**. Esta directiva se conjunta con **MaxKeepAliveRequests** y **KeepAliveTimeout** para **configurar las conexiones persistentes**.

```
#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On

#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100

#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 5
```

El tiempo de espera para enviar/recibir peticiones del servidor lo marca la directiva **Timeout**, siendo por defecto **300 segundos**.

```
#  
# Timeout: The number of seconds before receives and sends time out.  
#  
Timeout 300
```

El fichero para registrar los errores está definido por la directiva **ErrorLog**, que por defecto se registra en el fichero **"error.log"**.

```
# ErrorLog: The location of the error log file.  
# If you do not specify an ErrorLog directive within a <VirtualHost>  
# container, error messages relating to that virtual host will be  
# logged here.  If you *do* define an error logfile for a <VirtualHost>  
# container, that host's errors will be logged there and not here.  
#  
ErrorLog ${APACHE_LOG_DIR}/error.log
```

Por último, la directiva **LogLevel** sirve para **controlar qué tipo de errores vamos a guardar en el fichero**. Por defecto está en **Warn**, que **guarda eventos inesperados que no hacen que se detenga su ejecución**.

```
#  
# LogLevel: Control the severity of messages logged to the error_log.  
# Available values: trace8, ..., trace1, debug, info, notice, warn,  
# error, crit, alert, emerg.  
# It is also possible to configure the log level for particular modules, e.g.  
# "LogLevel info ssl:warn"  
#  
LogLevel warn
```

PASO 4) Comprobamos que está creado el archivo “/etc/apache2/sites-available/000-default.conf” para la configuración del servidor virtual. Al abrirlo, observamos que la directiva que indica el directorio donde se sirven los contenidos del servidor virtual es **DocumentRoot**, siendo la **ruta por defecto “/var/www/html”**.

```
<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
```

PASO 5) Accedemos al archivo “/etc/apache2/sites-enabled/000-default.conf” y verificamos que es un **enlace simbólico al archivo anterior**.

```
root@servidoragn:/etc/apache2/sites-enabled# ls -l
total 0
lrwxrwxrwx 1 root root 35 nov 15 09:16 000-default.conf -> ../sites-available/000-default.conf
root@servidoragn:/etc/apache2/sites-enabled#
```

PASO 6) Ahora abrimos el archivo “/etc/apache2/ports.conf” y vemos que su **puerto de escucha es el 80** y si usamos **conexión segura usaría el 443**.

```
root@servidoragn:/etc/apache2# cat ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
root@servidoragn:/etc/apache2# _
```

PASO 7) Por último, observamos en el fichero “/etc/apache2/apache2.conf” y nos vamos al **Directory /var/www/** para ver como Apache sirve el contenido.

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

C) Configuración básica en Apache para Linux.

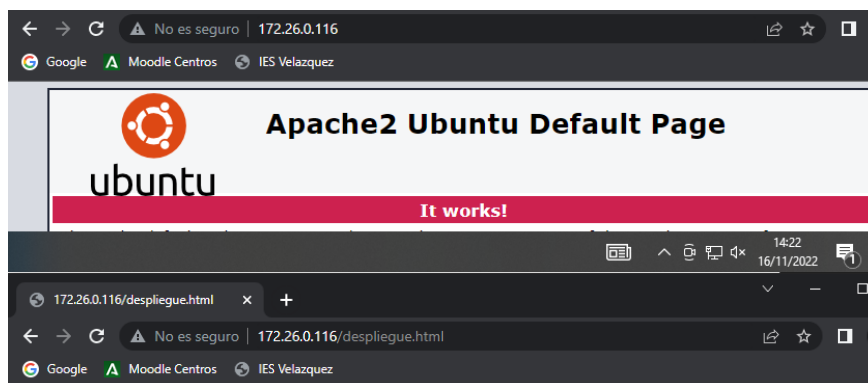
C.1) Ficheros y directorios de prueba.

PASO 1) Creamos los directorios y archivos pedidos con un encabezado para cada archivo.

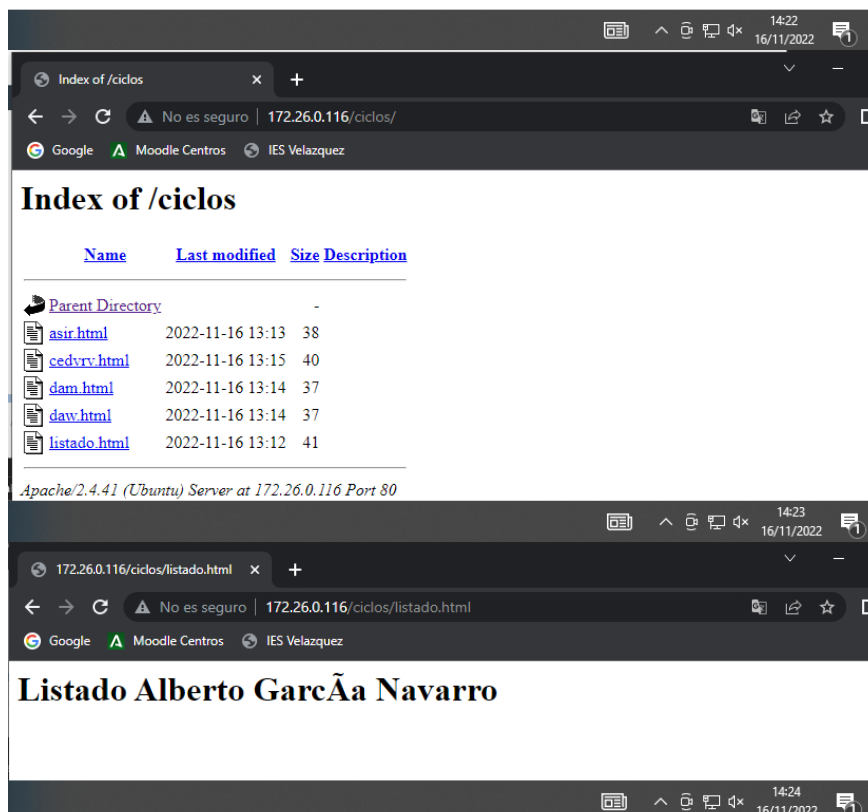
```
agn@servidoragn:/var/www/html$ tree
.
├── ciclos
│   ├── asir.html
│   ├── cedvrv.html
│   ├── dam.html
│   ├── daw.html
│   └── listado.html
├── despliegue.html
├── fp.html
└── index.html

1 directory, 8 files
agn@servidoragn:/var/www/html$
```

PASO 2) Comprobamos en el navegador que aparecen los archivos html que hemos creado.

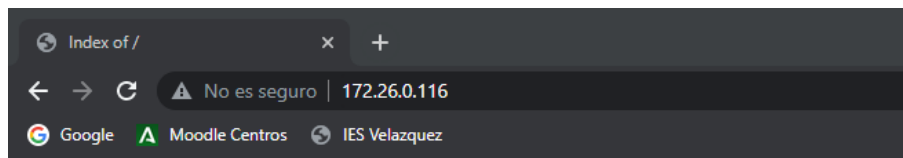


Despliegue Alberto Garc a Navarro



C.2) Archivos a servir por defecto (Directory Index).

PASO 1 Y 2) Vamos a renombrar el fichero “index.html” por defecto de apache2 a “índice.html”. Observamos que al acceder a la dirección IP del servidor apache ya no nos muestra este archivo “index.html”, sino que se muestra un listado de recursos de la carpeta “/var/www/html”.



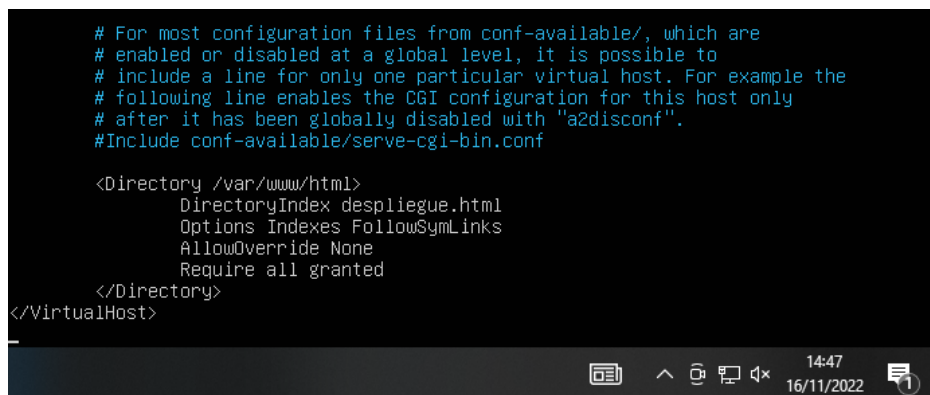
Index of /

Name	Last modified	Size	Description
ciclos/	2022-11-16 13:15	-	
despliegue.html	2022-11-16 13:11	44	
fp.html	2022-11-16 13:11	36	
indice.html	2022-11-15 09:16	11K	

Apache/2.4.41 (Ubuntu) Server at 172.26.0.116 Port 80

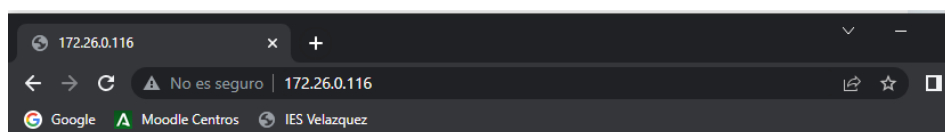


PASO 3) Realizamos la copia del archivo “/etc/apache2/sites-available/000-default.conf” antes de modificarlo. Añadimos la siguiente directiva:



PASO 4) Reiniciamos el servidor Apache con “sudo service apache2 restart” y verificamos que está todo correcto al no aparecernos ningún error.

PASO 5) Ahora al acceder a la dirección IP de nuestro servidor desde el navegador, se abrirá la página de “despliegue.html” como hemos definido en el anterior paso.



Despliegue Alberto García Navarro



C.3) Opciones sobre directorios (Options Indexes).

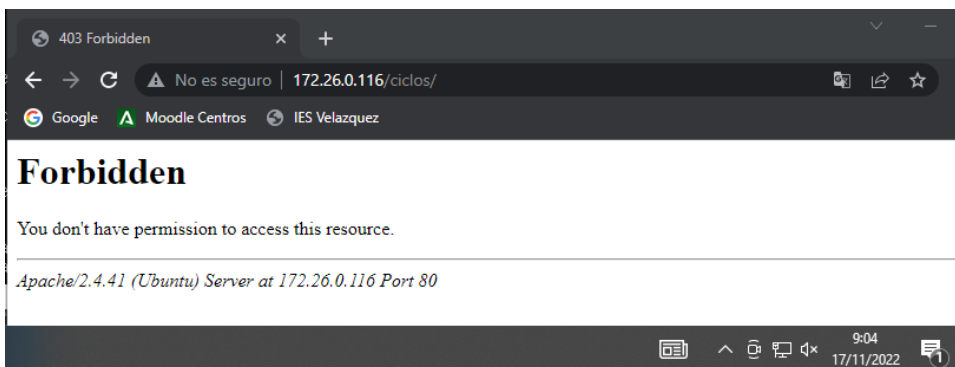
PASO 1 Y 2) Vamos a crear una nueva directiva “Directory” para la ruta “/var/www/html/ciclos/”. Modificamos entonces el archivo “/etc/apache2/sites-available/000-default.conf”.

```
<Directory /var/www/html>
    DirectoryIndex despliegue.html
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

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

PASO 3) Reiniciamos el servidor Apache con “sudo service apache2 restart” y verificamos que está todo correcto al no aparecernos ningún error.

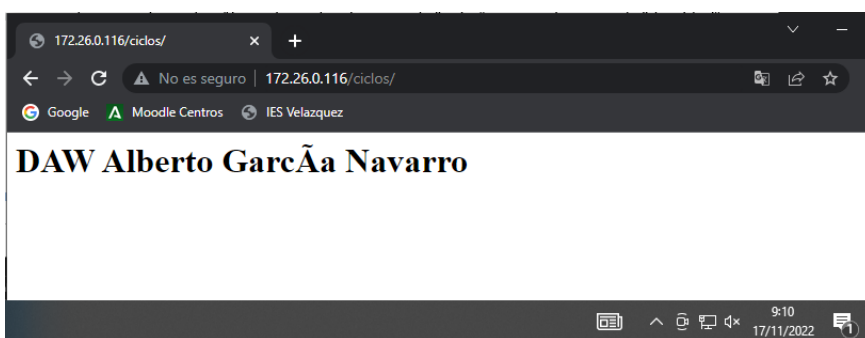
PASO 4) Ahora al acceder al recurso de “ciclos” vemos como el acceso está prohibido al no haber ningún index.



PASO 5) Ahora añadimos “DirectoryIndex” a la directiva anterior creada para que nos muestre el archivo “daw.html” por defecto.

```
<Directory /var/www/html/ciclos>
    DirectoryIndex daw.html_
    Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
</VirtualHost>
```

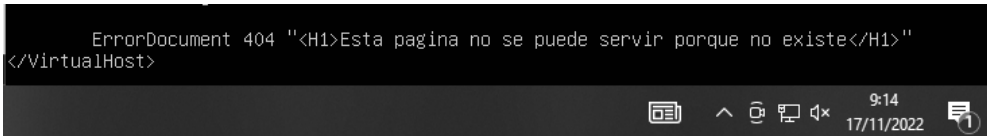
PASO 6 Y 7) Reiniciamos de nuevo el servidor. Ahora al acceder al recurso de “ciclos” veremos la página de “daw.html”.



C.4) Códigos de error (ErrorDocument).

PASO 1) Se va a crear con la directiva “ErrorDocument” un mensaje de error cuando se intente buscar un contenido que no está disponible en nuestro servidor. Modificamos el fichero “/etc/apache2/sites-available/000-default.conf” y añadimos esta directiva junto al mensaje de error.

```
ErrorDocument 404 "<H1>Esta pagina no se puede servir porque no existe</H1>"  
</VirtualHost>
```



PASO 2 Y 3) Reiniciamos el servidor con “sudo service apache2 restart” y comprobamos introduciendo en el navegador un recurso no existente en nuestro servidor. Observamos que aparece el mensaje de error que hemos definido.



PASO 4) En vez de un mensaje de error podemos definir que abra un fichero. Cambiamos la directiva para que vaya al fichero “404.html”.

```
</Directory>  
  
ErrorDocument 404 /404.html  
</VirtualHost>
```

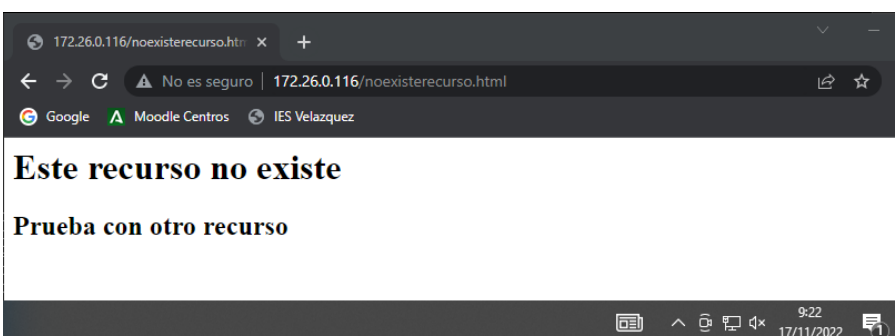


PASO 5) Creamos el fichero en la raíz de nuestro servidor “/var/www/html/404.html” con el mensaje de error.

```
root@servidoragn:/var/www/html# cat 404.html  
<h1>Este recurso no existe</h1>  
<h2>Prueba con otro recurso</h2>  
root@servidoragn:/var/www/html#
```



PASO 6 Y 7) Reiniciamos el servidor apache y comprobamos en el navegador que sale el mensaje de error al buscar un recurso que no existe.



C.5) Directorios virtuales (Alias).

PASO 1) Vamos a crear un directorio “/apuntes” dentro de nuestra carpeta de usuario con un fichero llamado “apuntesTema1.html”.

```
agn@servidoragn:~/apuntes$ cat apuntesTema1.html
<h1>Apuntes Tema1 Alberto Garcia Navarro</h1>
agn@servidoragn:~/apuntes$ _
```

PASO 2) Una vez creado, nos vamos al fichero “/etc/apache2/sites-available/000-default.conf” y usamos la directiva “Alias” para crear un directorio virtual llamado “/apuntes” que haga referencia a la carpeta que acabamos de crear “/home/agn/apuntes”. Y configuramos su directiva “Directory”.

```
Alias /apuntes /home/agn/apuntes
<Directory /home/agn/apuntes>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

ErrorDocument 404 /404.html
</VirtualHost>_
```

PASO 3 Y 4) Reiniciamos el servidor con “sudo service apache2 restart” y comprobamos en nuestro navegador que al buscar el directorio “apuntes” aparecerá el contenido de la carpeta que hemos creado.

Index of /apuntes

No es seguro | 172.26.0.116/apuntes/

Google Moodle Centros IES Velazquez

Index of /apuntes

Name	Last modified	Size	Description
Parent Directory	-		
apuntesTema1.html	2022-11-17 08:32	46	

Apache/2.4.41 (Ubuntu) Server at 172.26.0.116 Port 80

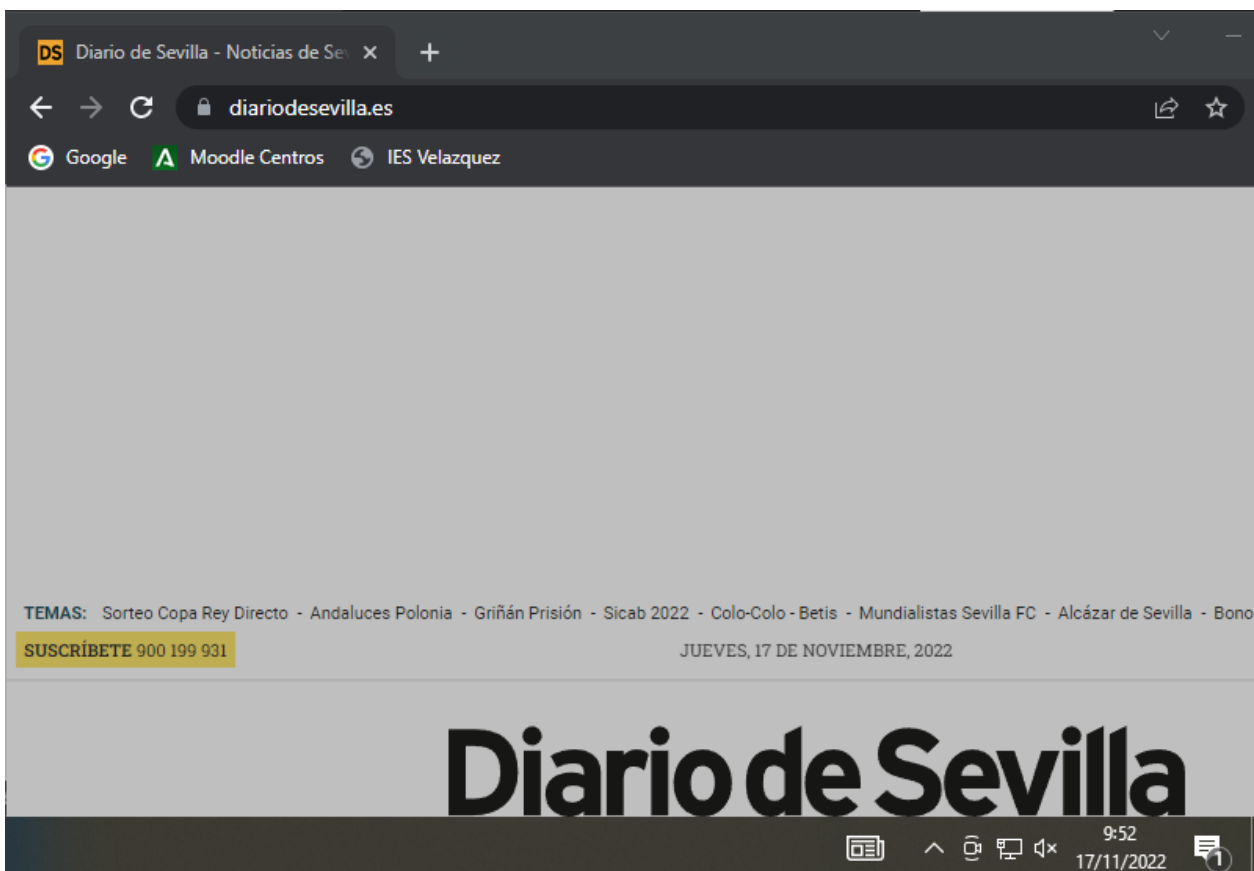
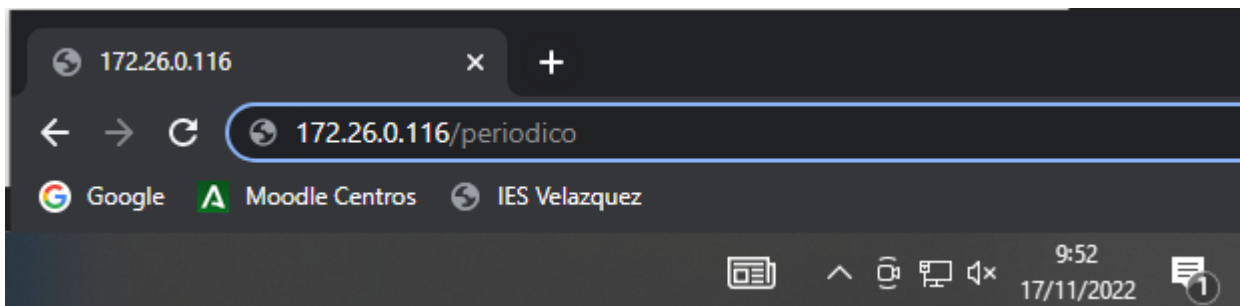
C.6) Redirecciones (Redirect).

PASO 1) Modificamos el archivo “/etc/apache2/sites-available/000-default.conf” para añadir la directiva “Redirect” que nos va a permitir redireccionar a un recurso ficticio de una página de internet.

```
Redirect /periodico http://www.diariodesevilla.es

ErrorDocument 404 /404.html
</VirtualHost>
```

PASO 2 Y 3) Reiniciamos el servidor con “sudo service apache2 restart” y al poner en el navegador la IP de nuestro servidor seguido del recurso “/periodico” que hemos definido anteriormente, nos redirecciona a la url que hemos guardado.



D) GitHub.

PASO 1) Añadimos la ruta de nuestro github junto al token que nos permite conectarnos. Siendo esta: `https://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@github.com/agarnav98/Despliegue-git.git`

```
Marmotín@DESKTOP-35COFHG MINGW64 ~/repositorio/Despliegue-git
$ git remote add github https://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@github.com/agarnav98/Despliegue-git.git
```

PASO 2) Una vez establecido la ruta, creamos una carpeta con nuestro archivo que se va a subir al repositorio. Hacemos “git add .” para añadir el documento a la zona de intercambio temporal, luego “git commit -m” para subir el documento a nuestro repositorio local y añadiéndole un mensaje. Y, por último, hacemos “git push github master” para subirlo a nuestro repositorio remoto.

```
Marmotín@DESKTOP-35COFHG MINGW64 ~/repositorio/Despliegue-git (master)
$ git add .

Marmotín@DESKTOP-35COFHG MINGW64 ~/repositorio/Despliegue-git (master)
$ git commit -m "Practica 2.1 realizada"
[master (root-commit) aec2d4e] Practica 2.1 realizada
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Tema 2 - Servicio Web/Practica_2.1_Servidor_Web.docx
 create mode 100644 Tema 2 - Servicio Web/~$actica_2.1_Servidor_Web.docx

Marmotín@DESKTOP-35COFHG MINGW64 ~/repositorio/Despliegue-git (master)
$ git status
On branch master
nothing to commit, working tree clean

Marmotín@DESKTOP-35COFHG MINGW64 ~/repositorio/Despliegue-git (master)
$
```

PASO 3) Comprobamos que se ha completado la subida a nuestro repositorio remoto.

