



5 DE DICIEMBRE DE 2022

UNIDAD 2: SERVICIO WEB

PRÁCTICA 2.2: ADMINISTRACIÓN DE APACHE II MÓDULOS

ALBERTO GARCÍA NAVARRO
DESPLIEGUE DE APLICACIONES WEB
2º CFGS DAW
Curso 2022-2023

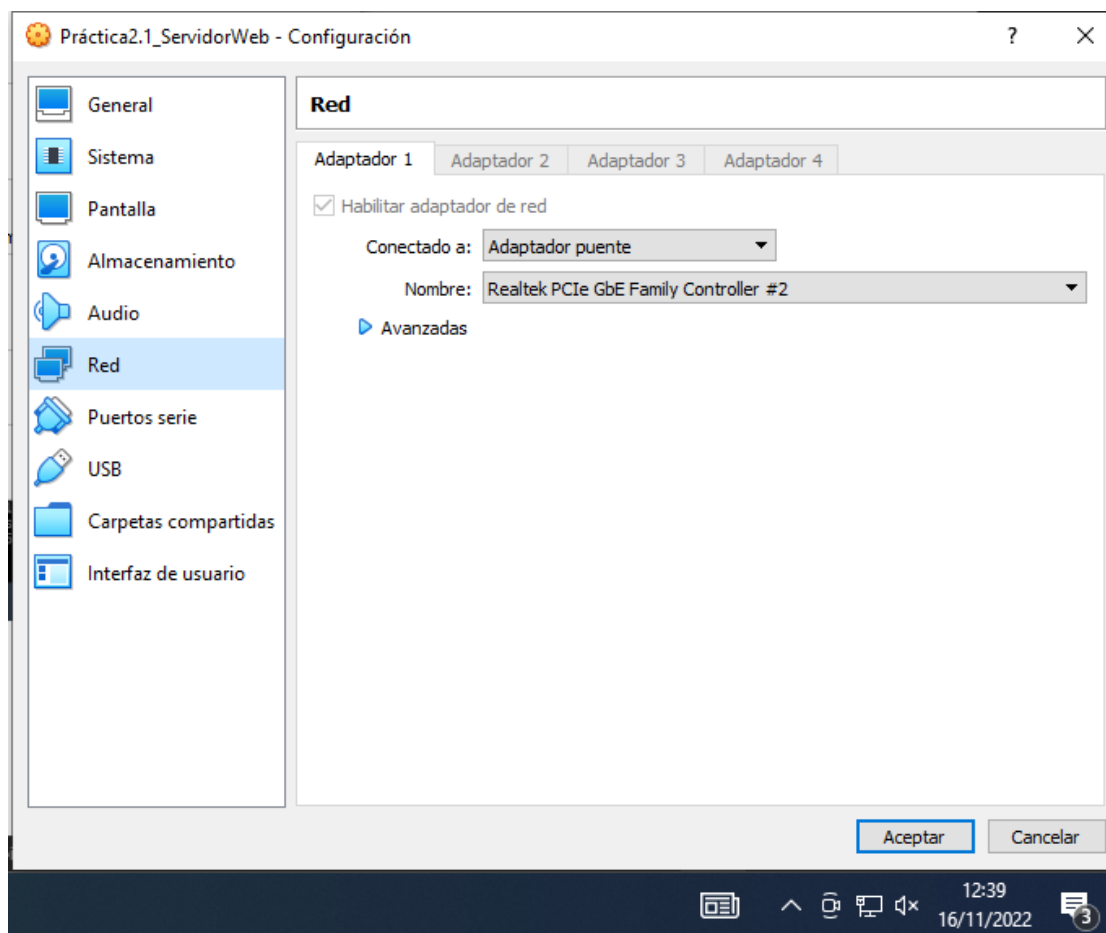


ÍNDICE

A) Módulos en Linux	pág. 2
A.1) Módulos.....	pág. 3
A.2) Módulo userdir.....	pág. 5
A.3) Módulo userdir en el servidor de clase.....	pág. 6
B) Control de acceso por IP y nombre de dominio.....	pág. 7
C) Autenticación y autorización Basic y Digest.....	pág. 9
C.1) Autenticación Basic.....	pág. 9
C.2) Autenticación Digest.....	pág. 11
D) Ficheros .htaccess.....	pág. 13
E) Ficheros de registros (logs).....	pág. 14
F) Módulos status e info.....	pág. 15
G) Webalizer	pág. 17
F) GitHub.....	pág. 18

A) Módulos en Linux.

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



La IP de nuestra máquina de Linux será en esta práctica: **172.26.217.160**



La IP de nuestra máquina física de Windows será: **172.26.0.6**



A.1) Módulos.

PASO 1) Comprobamos los módulos estáticos cargados al compilar el servidor con “sudo apache2ctl -l”.

```
agn@servidoragn:~$ sudo apache2ctl -l
Compiled in modules:
  core.c
  mod_so.c
  mod_watchdog.c
  http_core.c
  mod_log_config.c
  mod_logio.c
  mod_version.c
  mod_unixd.c
agn@servidoragn:~$ _
```

PASO 2) Comprobamos ahora los módulos que se han cargado dinámicamente que aparecen en el directorio “/etc/apache2/mods-enabled/”.

```
agn@servidoragn:~$ ls /etc/apache2/mods-enabled/
access_compat.load  authz_core.load  deflate.load  mime.load  reqtimeout.load
alias.conf          authz_host.load  dir.conf     mpm_event.conf  setenvif.conf
alias.load          authz_user.load  dir.load     mpm_event.load  setenvif.load
auth_basic.load     autoindex.conf  env.load     negotiation.conf  status.conf
authn_core.load     autoindex.load  filter.load  negotiation.load  status.load
authn_file.load     deflate.conf     mime.conf    reqtimeout.conf
agn@servidoragn:~$ _
```

PASO 3) Ahora editamos uno de los archivos “.load” de este directorio para observar la directiva “LoadModule”. La extensión que tiene **los códigos del módulo es “.so”**.

```
LoadModule alias_module /usr/lib/apache2/modules/mod_alias.so
```

PASO 4) Editamos uno de los archivos “.conf” y observamos cómo se añaden las directivas. Vemos que se usan las **etiquetas “<IfModule>”**.

```
<IfModule alias_module>
# Aliases: Add here as many aliases as you need (with no limit). The format is
# Alias fakename realname
#
# Note that if you include a trailing / on fakename then the server will
# require it to be present in the URL. So "/icons" isn't aliased in this
# example, only "/icons/". If the fakename is slash-terminated, then the
# realname must also be slash terminated, and if the fakename omits the
# trailing slash, the realname must also omit it.
#
# We include the /icons/ alias for FancyIndexed directory listings. If
# you do not use FancyIndexing, you may comment this out.

Alias /icons/ "/usr/share/apache2/icons/"

<Directory "/usr/share/apache2/icons">
    Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

PASO 5) Por último, consultamos el directorio “/usr/lib/apache2/modules/” y vemos que contiene estos archivos “.so” que aparecían en la directiva “LoadModule” de “.load”.

```
agn@servidoragn:~$ ls /usr/lib/apache2/modules
httpd.exp
mod_access_compat.so
mod_actions.so
mod_alias.so
mod_allowmethods.so
mod_asis.so
mod_auth_basic.so
mod_auth_digest.so
mod_auth_form.so
mod_authn_anon.so
mod_authn_core.so
mod_authn_dbd.so
mod_authn_dbm.so
mod_authn_file.so
mod_authn_socache.so
mod_authnz_fcgi.so
mod_authnz_ldap.so
mod_authz_core.so
mod_authz_dbd.so
mod_authz_dbm.so
mod_authz_groupfile.so
mod_authz_host.so
mod_authz_owner.so
mod_authz_user.so
mod_autoindex.so
mod_brotli.so
mod_bucketeer.so
mod_buffer.so
mod_cache_disk.so
mod_cache.so
mod_cache_socache.so
mod_case_filter_in.so
mod_case_filter.so
mod_cern_meta.so
mod_cgid.so
mod_cgi.so
mod_charset_lite.so
mod_data.so
mod_dav_fs.so
mod_dav_lock.so
mod_dav.so
mod_dbd.so
mod_deflate.so
mod_dialup.so
mod_dir.so
mod_dumpio.so
mod_echo.so
mod_env.so
mod_expires.so
mod_ext_filter.so
mod_file_cache.so
mod_filter.so
mod_headers.so
mod_heartbeat.so
mod_heartmonitor.so
mod_http2.so
mod_ident.so
mod_imagemap.so
mod_include.so
mod_info.so
mod_lbmethod_bybusyness.so
mod_lbmethod_byrequests.so
mod_lbmethod_bytraffic.so
mod_lbmethod_heartbeat.so
mod_ldap.so
mod_log_debug.so
mod_log_forensic.so
mod_lua.so
mod_macro.so
mod_md.so
mod_mime_magic.so
mod_mime.so
mod_mpm_event.so
mod_mpm_prefork.so
mod_mpm_worker.so
mod_negotiation.so
mod_proxy_ajp.so
mod_proxy_balancer.so
mod_proxy_connect.so
mod_proxy_express.so
mod_proxy_fcgi.so
mod_proxy_fdpass.so
mod_proxy_ftp.so
mod_proxy_hcheck.so
mod_proxy_html.so
mod_proxy_http2.so
mod_proxy_http.so
mod_proxy_scgi.so
mod_proxy.so
mod_proxy_uwsgi.so
mod_proxy_wstunnel.so
mod_ratelimit.so
mod_reflector.so
mod_remoteip.so
mod_reqtimeout.so
mod_request.so
mod_rewrite.so
mod_sed.so
mod_session_cookie.so
mod_session_crypto.so
mod_session_dbd.so
mod_session.so
mod_setenvif.so
mod_slotmem_plain.so
mod_slotmem_shm.so
mod_socache_dbm.so
mod_socache_memcache.so
mod_socache_redis.so
mod_socache_shmcb.so
mod_speling.so
mod_ssl.so
mod_status.so
mod_substitute.so
mod_suexec.so
mod_unique_id.so
mod_userdir.so
mod_usertrack.so
mod_vhost_alias.so
mod_xml2enc.so
```

A.2) Módulo userdir.

PASO 1) Al comprobar si el módulo “userdir” está habilitado, vemos que no está cargado en “/etc/apache2/mods-enabled/”.

PASO 2 Y 4) Para habilitar este módulo ejecutamos el comando “sudo a2enmod userdir” y reiniciamos el servidor con “systemctl restart apache2”.

PASO 3) Ya tenemos el módulo habilitado.

```
lrwxrwxrwx 1 root root 30 nov 24 08:13 userdir.conf -> ../mods-available/userdir.conf
lrwxrwxrwx 1 root root 30 nov 24 08:13 userdir.load -> ../mods-available/userdir.load
agn@servidoragn:/etc/apache2/mods-enabled$ _
```

PASO 5) Consultamos el archivo “/etc/apache2/mods-enabled/userdir.conf” y observamos que el **usuario “root” tiene deshabilitado el uso de directorios personales**. Para usar esto, el usuario deberá crear un subdirectorio “public_html” en su carpeta “home” para poner sus páginas personales.

```
<IfModule mod_userdir.c>
    UserDir public_html
    UserDir disabled root

    <Directory /home/*/public_html>
        AllowOverride FileInfo AuthConfig Limit Indexes
        Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
        Require method GET POST OPTIONS
    </Directory>
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

PASO 6) Creamos ahora este subdirectorio “public_html” en nuestro usuario con un fichero llamado “personal.html”.

PASO 7) Abrimos el navegador y accedemos a nuestro directorio raíz que es “http://172.26.217.160/~agn/”

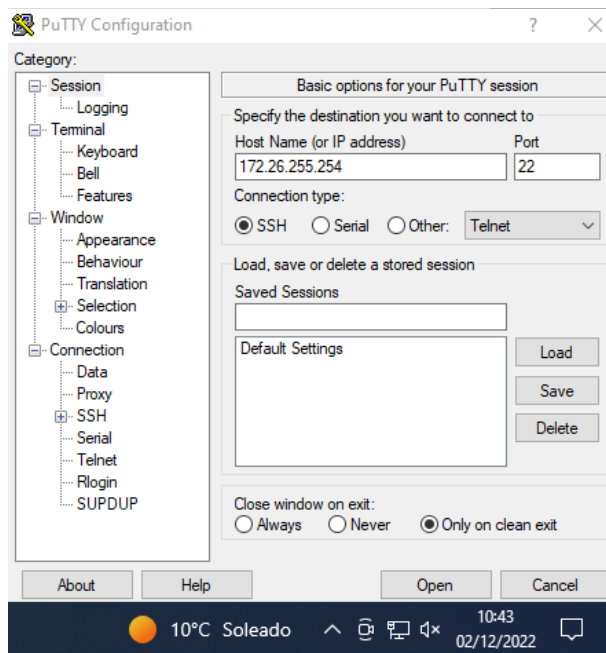
Index of /~agn

Name	Last modified	Size	Description
Parent Directory	-	-	-
personal.html	2022-11-24 08:25	65	

Apache/2.4.41 (Ubuntu) Server at 172.26.217.160 Port 80

A.3) Módulo userdir en el servidor de clase.

PASO 1) Accedemos al servidor de la clase a través de Putty introduciendo la IP (172.26.255.254).



PASO 2) Hacemos login para acceder con el usuario “agarcia” y la contraseña “alumno”.

```
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-132-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

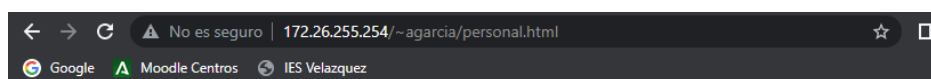
System information as of vie 02 dic 2022 09:45:32 UTC

System load:  0.0          Temperature:    28.0 C
Usage of /:   17.4% of 54.22GB Processes:      158
Memory usage: 17%          Users logged in: 3
Swap usage:   0%           IPv4 address for enp2s0: 172.26.255.254

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings
```

PASO 3) Ahora creamos el subdirectorio “public_html” donde guardaremos nuestros archivos en el servidor. Creamos un fichero llamado “personal.html” y probamos que se ve el contenido en el navegador con la IP del servidor de clase seguido del usuario “172.26.255.254/~agarcia”.



PAGINA WEB DEL USUARIO AGARCIA



B) Control de acceso por IP y nombre de dominio.

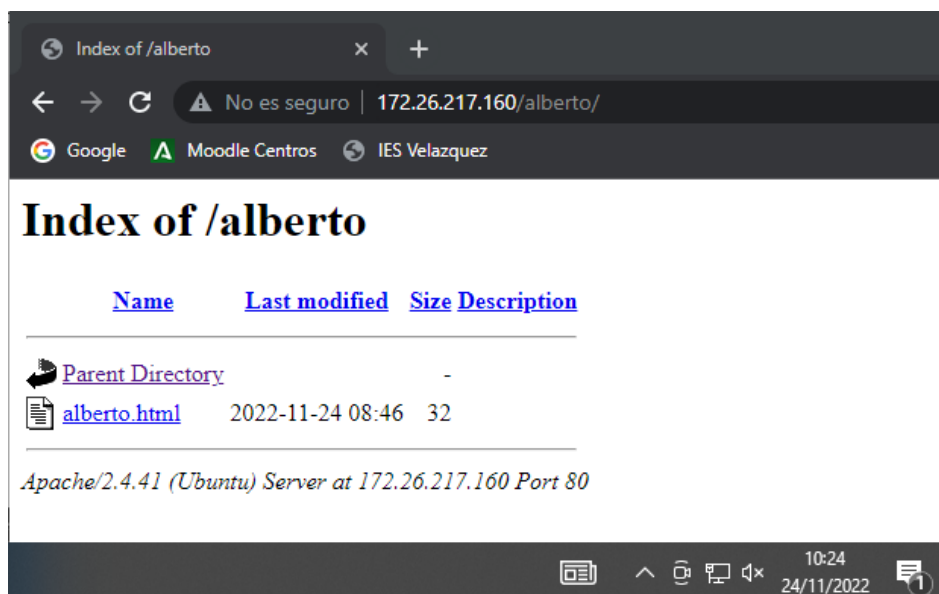
PASO 1) Comprobamos en en “/etc/apache2/mods-enabled/” que el módulo “authz.host” está habilitado.

PASO 2) Ahora creamos el directorio “/var/www/html/alberto/” con el archivo “alberto.html”.

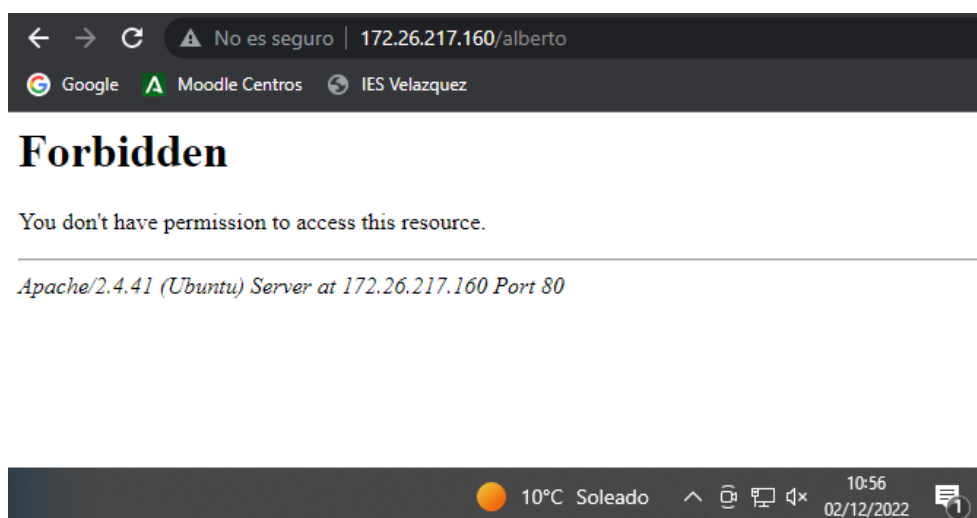
PASO 3) Editamos el fichero de configuración “/etc/apache2/sites-available/000-default.conf” con la directiva “Directory” que especifique en el “Require ip 172.26.0.6” con la IP de nuestra máquina física para que solo pueda acceder esta.

```
<Directory /var/www/html/alberto>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require ip 172.26.0.6
</Directory>
```

PASO 4 Y 5) Reiniciamos el servidor y comprobamos en el navegador que se puede acceder al recurso creado. Solo puede acceder al recurso esta IP que le hemos especificado.



PASO 6) Probamos a acceder desde otra máquina y vemos como sale la página “Forbidden” sin permisos.

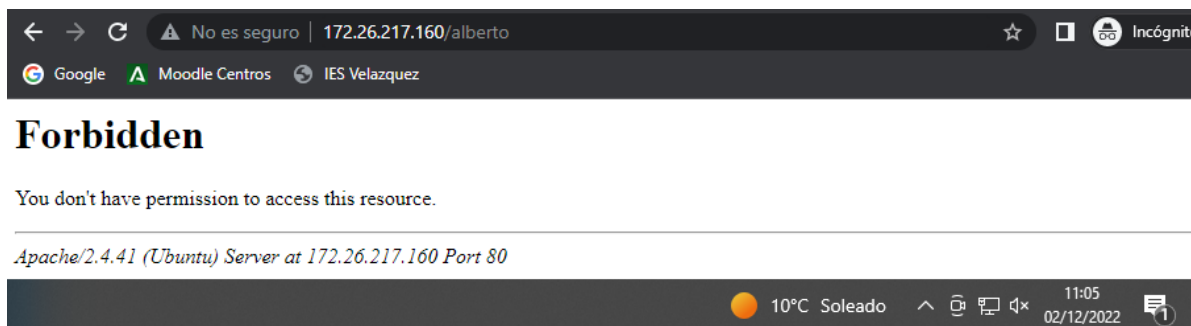


PASO 7) Añadimos en el Require el nombre de host del equipo que va a acceder, en este caso “AULA44-PC01”

```
<Directory /var/www/html/alberto>
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Require host AULA44-PC01
```



PASO 8 y 9) Reiniciamos el servidor y probamos a entrar, pero no funciona con el “hostname”.



C) Autenticación y autorización Basic y Digest.

C.1) Autenticación Basic.

PASO 1) Comprobamos en en “/etc/apache2/mods-enabled/” que el módulo “auth_basic” está habilitado.

PASO 2) Ahora creamos el directorio “/var/www/html/alberto/” con el archivo “alberto.html”.

PASO 3) Creamos el fichero que vamos a usar para la autenticación. Para ello usamos el comando “sudo htpasswd -c /etc/apache2/passwd garcia”. Esto nos pedirá una contraseña que se guardará en el archivo de la ruta elegida para el usuario “garcia”. Para añadir el siguiente usuario, ejecutamos “sudo htpasswd /etc/apache2/passwd navarro”.

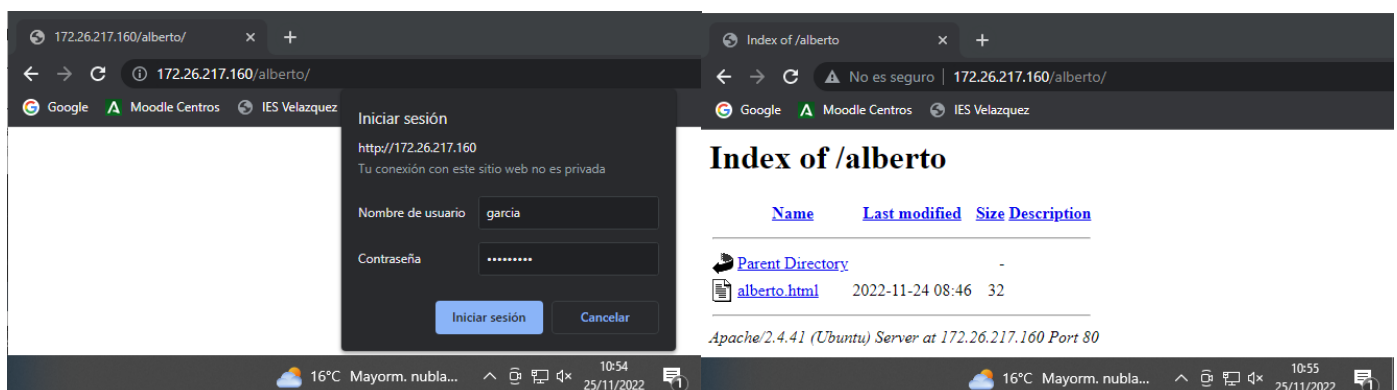
```
agn@servidoragn:/etc/apache2$ sudo htpasswd -c /etc/apache2/passwd garcia
New password:
Re-type new password:
Adding password for user garcia
agn@servidoragn:/etc/apache2$ sudo htpasswd /etc/apache2/passwd navarro
New password:
Re-type new password:
Adding password for user navarro
agn@servidoragn:/etc/apache2$

agn@servidoragn:/etc/apache2$ cat passwd
garcia:$apr1$9Aaq0U6u$Wp6NYtc.5wDQsCUvyweYF0
navarro:$apr1$5NBdasu.$Bc0UKwuzG1cgy0IzzA0iS1
agn@servidoragn:/etc/apache2$
```

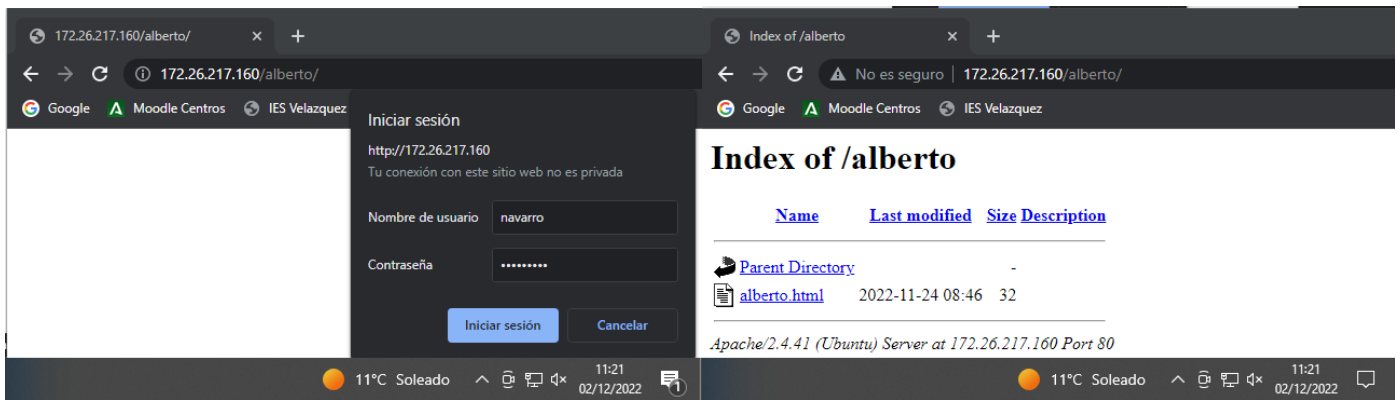
PASO 4) Es necesario editar el fichero de configuración “/etc/apache2/sites-available/000-default.conf” para crear en la directiva “Directory” los usuarios que pueden acceder al recurso, especificándolo en el “Require”.

```
<Directory /var/www/html/alberto>
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    allow from 172.26.0.6
    AuthType Basic
    AuthName "Acceso restringido"
    AuthUserFile /etc/apache2/passwd
    Require user garcia navarro_
</Directory>
```

PASO 5 Y 6) Reiniciamos el servidor y observamos que al entrar en el recurso nos pide que introduzcamos el usuario y contraseña antes definidos.



PASO 7) Hacemos lo mismo desde otra máquina y con el segundo usuario creado.



C.2) Autenticación Digest.

PASO 1) Comprobamos en “/etc/apache2/mods-enabled/” que el módulo “auth_digest” está habilitado. Nosotros no lo tenemos habilitado, entonces ejecutamos el comando “sudo a2enmod auth_digest” y reiniciamos el servidor para descargar el módulo.

PASO 2) Ahora creamos el directorio “/var/www/html/tareac2/” con el archivo “tareac2.html”.

PASO 3) Creamos el fichero que vamos a usar para la autenticación, igual que la autenticación Basic, pero ahora habrá que añadir el dominio asociado.

Para ello usamos el comando “sudo htdigest /etc/apache2/passwd informatica galberto”. Esto nos pedirá una contraseña que se guardará en el archivo de la ruta elegida en el dominio “informatica” para el usuario “galberto”. Hacemos lo mismo con el usuario “nalberto”.

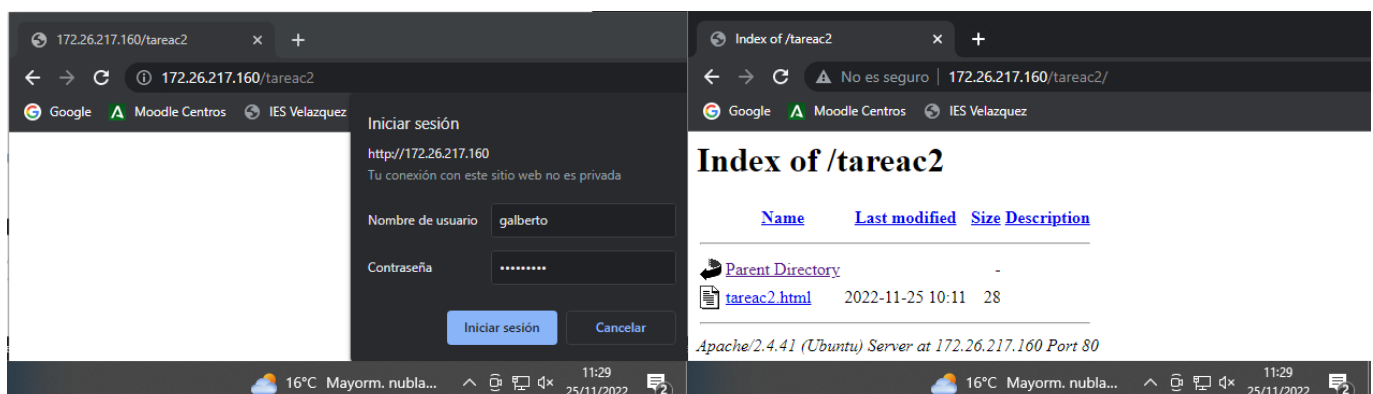
```
agn@servidoragn:~$ sudo htdigest /etc/apache2/passwd informatica galberto
Adding user galberto in realm informatica
New password:
Re-type new password:
agn@servidoragn:~$ sudo htdigest /etc/apache2/passwd informatica nalberto
Adding user nalberto in realm informatica
New password:
Re-type new password:
agn@servidoragn:~$
```

```
agn@servidoragn:~$ cat /etc/apache2/passwd
garcia:$apr1$9AaqDU6u$wp6NYtc.5wDQsCUvyweYF0
navarro:$apr1$5NBdasu.$Bc0UKwuzG1cgy0IzZA0iS1
galberto:informatica:6ba7e2beb409cf1cc5e6be99cb5e285e
nalberto:informatica:3e28303c8d060555f0a72e3fe39bbd40
agn@servidoragn:~$
```

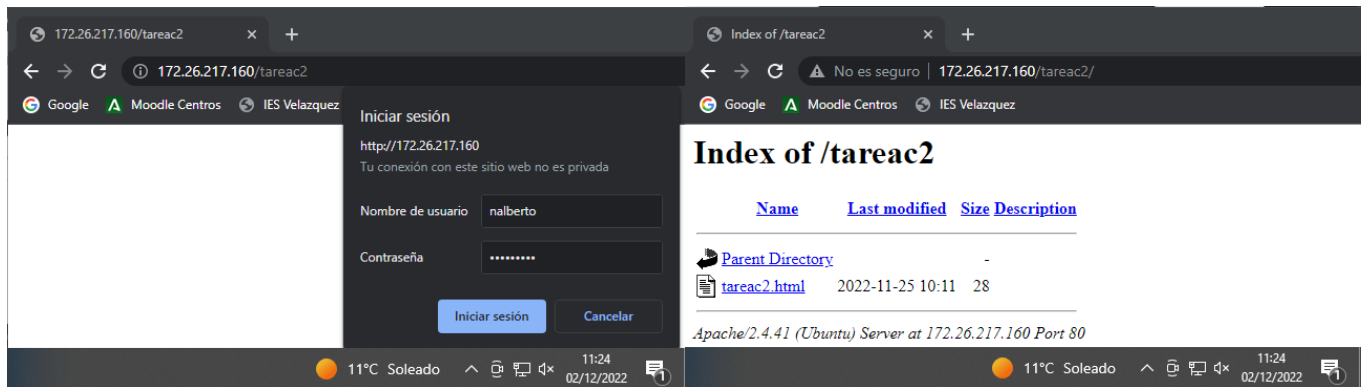
PASO 4) Es necesario editar el fichero de configuración “/etc/apache2/sites-available/000-default.conf” para crear en la directiva “Directory” los usuarios que pueden acceder al recurso, especificándolo en el “Require”. Es importante que en el “AuthName” pongamos el dominio que hemos especificado anteriormente.

```
<Directory /var/www/html/tareac2>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    AuthType Digest
    AuthName "informatica"
    AuthDigestProvider file
    AuthUserFile /etc/apache2/passwd
    Require user galberto nalberto
</Directory>
```

PASO 5 Y 6) Reiniciamos el servidor y observamos que al entrar en el recurso nos pide que introduzcamos el usuario y contraseña antes definidos.



PASO 7) Hacemos lo mismo desde otra máquina y con el segundo usuario creado.



D) Ficheros .htaccess

PASO 1) Creamos el usuario “useraccess”.

PASO 2) Abrimos el fichero con la ruta “/etc/apache2/sites-available/000-default.conf” y creamos el alias “myBlog” que apuntará a la carpeta “/home/useraccess/myBlog”. Dejaremos como única directiva “AllowOverride All” para poder hacer uso del “htaccess”.

```
Alias /myBlog /home/useraccess/myBlog
<Directory /home/useraccess/myBlog>
    AllowOverride All
</Directory>
```

PASO 3, 4 y 5) Reiniciamos el servidor, iniciamos sesión con el usuario creado “useraccess” y creamos el directorio “/home/useraccess/myBlog” con el archivo “myBlog.html”.

PASO 6) El tipo de autenticación para el acceso al recurso que se va a usar es el “Digest”. Para ello usamos el comando “sudo htdigest -c /home/useraccess/myBlog/.htdigest informatica myUserBlog”. Así crearemos el fichero “.htdigest” en la ruta del usuario.

```
agn@servidoragn:/home/useraccess/myBlog$ cat .htdigest
myUserBlog:informatica:7168ccafb201f967692b496f10a6c23f
agn@servidoragn:/home/useraccess/myBlog$
```

PASO 7) Creamos el fichero “.htaccess” en la misma ruta y lo editamos añadiendo las siguientes directivas, siendo solo accesible desde nuestra máquina física.

```
Options Indexes
Order allow,deny
allow from 172.26.0.6
AuthType Digest
AuthName "informatica"
AuthUserFile /home/useraccess/myBlog/.htdigest
Require user myUserBlog_
```

PASO 8) Accedemos al recurso “/myBlog” que hemos creado y vemos que nos pide la autenticación para acceder.

E) Ficheros de registros (logs)

PASO 1) Consultamos el fichero “/etc/apache2/sites-available/000-default.conf” y buscamos la directiva que controla los errores “log”. Esta directiva es “**ErrorLog**” que marca como ruta “/var/log/apache2/error.log”. El “**LogLevel**” que viene por defecto es “**warn**”.

La directiva que marca la ruta del archivo de los accesos es “**CustomLog**”, siendo el fichero de logs de accesos “/var/log/apache2/access.log”.

```
# 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
```

PASO 2) Consultamos el archivo “/var/log/apache2/error.log” para los errores.

```
[Fri Nov 25 11:48:00.414299 2022] [core:notice] [pid 1909:tid 140428249394240] AH00094: Command line
: '/usr/sbin/apache2'
[Fri Nov 25 11:48:12.094145 2022] [auth_digest:error] [pid 1911:tid 140428196689664] [client 172.26.
0.6:49172] AH01790: user 'galberto' in realm 'informatica' not found: /myBlog
agn@servidoragn:/etc/apache2/sites-available$
```

PASO 3) Consultamos el archivo “/var/log/apache2/error.log” para los accesos.

```
172.26.0.6 - myUserBlog [25/Nov/2022:11:49:21 +0000] "GET /myBlog/myBlog.html HTTP/1.1" 200 438 "htt
p://172.26.217.160/myBlog/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, li
ke Gecko) Chrome/107.0.0.0 Safari/537.36"
172.26.0.6 - - [25/Nov/2022:11:50:10 +0000] "-" 408 0 "-" "-"
agn@servidoragn:/etc/apache2/sites-available$
```

F) Módulos status e info

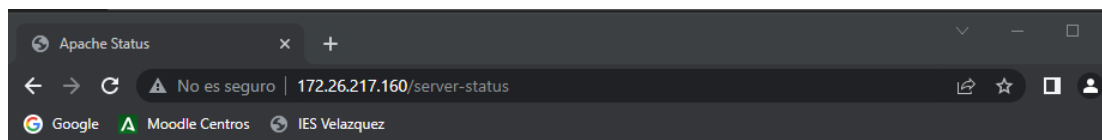
PASO 1) Verificamos que está habilitado el módulo “status” en “/etc/apache2/mods-enabled/”.

PASO 2) Editamos el fichero “status.conf” para habilitar el acceso a nuestra IP de la máquina física.

```
<IfModule mod_status.c>
    # Allow server status reports generated by mod_status,
    # with the URL of http://servername/server-status
    # Uncomment and change the "192.0.2.0/24" to allow access from other hosts.

    <Location /server-status>
        SetHandler server-status
        Require local
        Require ip 172.26.0.6_
    </Location>
```

PASO 3 Y 4) Reiniciamos el servidor y nos conectamos al recurso “server-status” desde nuestra máquina física.



Apache Server Status for 172.26.217.160 (via 172.26.217.160)

Server Version: Apache/2.4.41 (Ubuntu)
 Server MPM: event
 Server Built: 2022-06-14T13:30:55

Current Time: Friday, 25-Nov-2022 12:45:52 UTC
 Restart Time: Friday, 25-Nov-2022 12:45:26 UTC
 Parent Server Config. Generation: 1
 Parent Server MPM Generation: 0
 Server uptime: 25 seconds
 Server load: 0.00 0.12 0.12
 Total accesses: 0 - Total Traffic: 0 kB - Total Duration: 0
 CPU Usage: u0 s0 cu0 cs0
 0 requests/sec - 0 B/second
 1 requests currently being processed, 49 idle workers

Slot	PID	Stopping	Connections			Threads			Async connections		
			total	accepting	busy	idle	writing	keep-alive	closing		
0	1158	no	0	yes	0	25	0	0	0		
1	1159	no	0	yes	1	24	0	0	0		
Sum	2	0	0		1	49	0	0	0		

.....w.....

Scoreboard Key:

" " Waiting for Connection, "s" Starting up, "r" Reading Request,
 "w" Sending Reply, "k" Keepalive (read), "b" DNS Lookup,
 "c" Closing connection, "l" Logging, "g" Gracefully finishing,
 "i" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Dur	Conn	Child	Slot	Client	Protocol	VHost	Request
1-0	1159	1/0/0	W	0.00	0	0	0	0.00	0.00	0.00	172.26.0.6	http/1.1	127.0.1.1:80	GET /server-status HTTP/1.1

Srv Child Server number - generation
 PID OS process ID
 Acc Number of accesses this connection / this child / this slot
 M Mode of operation
 CPU CPU usage, number of seconds
 SS Seconds since beginning of most recent request
 Req Milliseconds required to process most recent request
 Dur Sum of milliseconds required to process all requests
 Conn Kilobytes transferred this connection
 Child Megabytes transferred this child



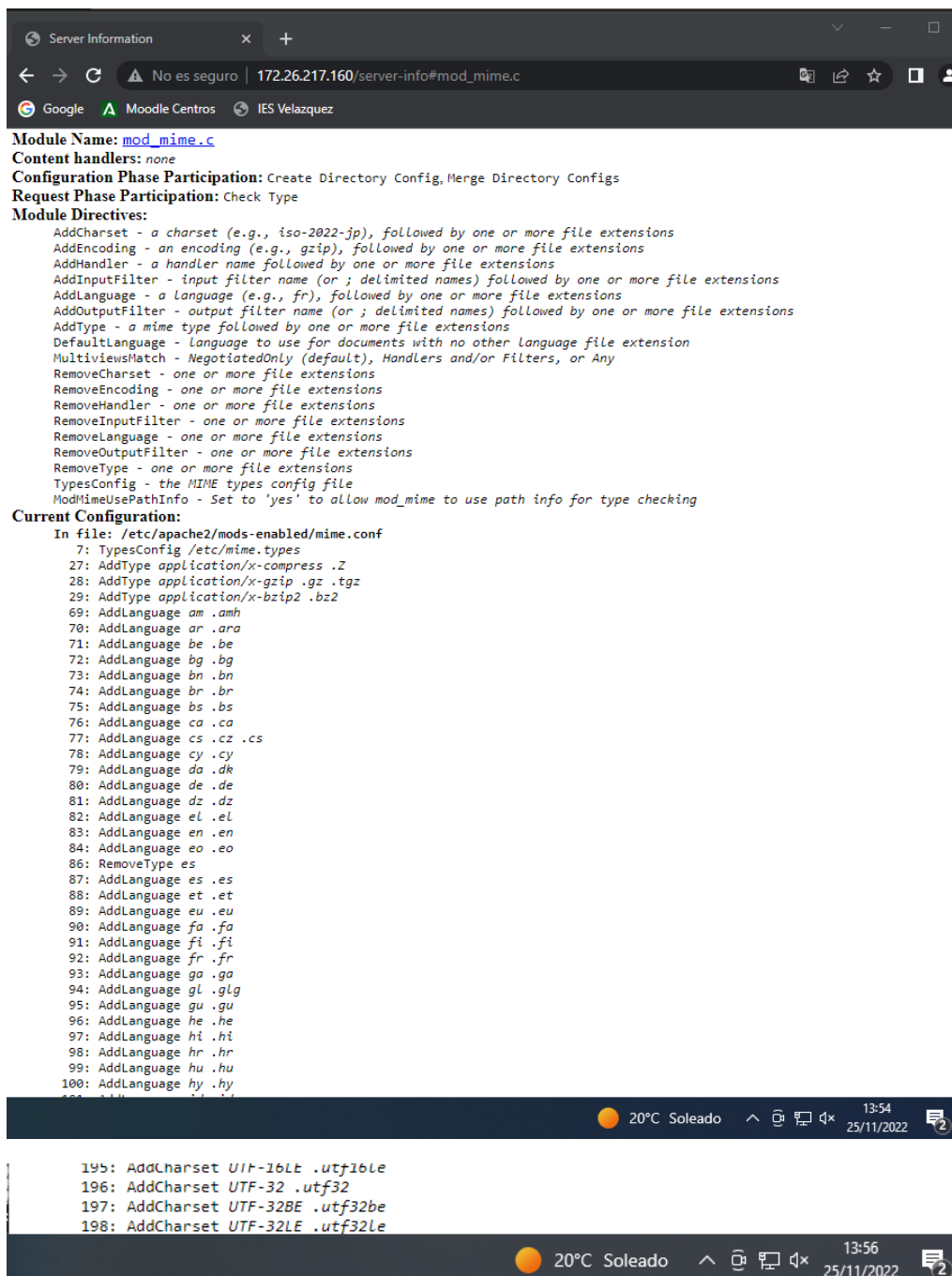
PASO 5) Verificamos que está habilitado el módulo “status” en “/etc/apache2/mods-enabled/”. En nuestro caso no lo está, así que lo descargamos con “sudo a2enmod info”.

PASO 6) Editamos el fichero “info.conf” para habilitar el acceso a nuestra IP de la máquina física.

```
<IfModule mod_info.c>

# Allow remote server configuration reports, with the URL of
# http://servername/server-info (requires that mod_info.c be
# Uncomment and change the "192.0.2.0/24" to allow access from
#
<Location /server-info>
    SetHandler server-info
    Require local
    Require ip 172.26.0.6
</Location>
```

PASO 7 Y 8) Reiniciamos el servidor y nos conectamos al recurso “server-status” desde nuestra máquina física. El módulo “mod_mime” aparece cargado con la configuración de caracteres UTF-32.



Server Information

← → ↻ No es seguro | 172.26.217.160/server-info#mod_mime.c

Google Moodle Centros IES Velazquez

Module Name: [mod_mime.c](#)

Content handlers: none

Configuration Phase Participation: Create Directory Config, Merge Directory Configs

Request Phase Participation: Check Type

Module Directives:

- AddCharset - a charset (e.g., iso-2022-jp), followed by one or more file extensions
- AddEncoding - an encoding (e.g., gzip), followed by one or more file extensions
- AddHandler - a handler name followed by one or more file extensions
- AddInputFilter - input filter name (or ; delimited names) followed by one or more file extensions
- AddLanguage - a language (e.g., fr), followed by one or more file extensions
- AddOutputFilter - output filter name (or ; delimited names) followed by one or more file extensions
- AddType - a mime type followed by one or more file extensions
- DefaultLanguage - language to use for documents with no other language file extension
- MultiviewsMatch - NegotiatedOnly (default), Handlers and/or Filters, or Any
- RemoveCharset - one or more file extensions
- RemoveEncoding - one or more file extensions
- RemoveHandler - one or more file extensions
- RemoveInputFilter - one or more file extensions
- RemoveLanguage - one or more file extensions
- RemoveOutputFilter - one or more file extensions
- RemoveType - one or more file extensions
- TypesConfig - the MIME types config file
- ModTimeUsePathInfo - Set to 'yes' to allow mod_mime to use path info for type checking

Current Configuration:

In file: /etc/apache2/mods-enabled/mime.conf

```
7: TypesConfig /etc/mime.types
27: AddType application/x-compress .Z
28: AddType application/x-gzip .gz .tgz
29: AddType application/x-bzip2 .bz2
69: AddLanguage am .amh
70: AddLanguage ar .ara
71: AddLanguage be .be
72: AddLanguage bg .bg
73: AddLanguage bn .bn
74: AddLanguage br .br
75: AddLanguage bs .bs
76: AddLanguage ca .ca
77: AddLanguage cs .cz .cs
78: AddLanguage cy .cy
79: AddLanguage da .dk
80: AddLanguage de .de
81: AddLanguage dz .dz
82: AddLanguage el .el
83: AddLanguage en .en
84: AddLanguage eo .eo
86: RemoveType es
87: AddLanguage es .es
88: AddLanguage et .et
89: AddLanguage eu .eu
90: AddLanguage fa .fa
91: AddLanguage fi .fi
92: AddLanguage fr .fr
93: AddLanguage ga .ga
94: AddLanguage gl .glg
95: AddLanguage gu .gu
96: AddLanguage he .he
97: AddLanguage ht .ht
98: AddLanguage hr .hr
99: AddLanguage hu .hu
100: AddLanguage hy .hy
195: AddCharset UTF-16LE .utf16le
196: AddCharset UTF-32 .utf32
197: AddCharset UTF-32BE .utf32be
198: AddCharset UTF-32LE .utf32le
```

G) Webalizer.

PASO 1) Instalamos la aplicación “Webalizer” usando “apt-get install webalizer”.

PASO 2) Verificamos que se ha creado el directorio “/etc/webalizer” y nos aparece el fichero “webalizer.conf”

```
# LogFile defines the web server log file to use. If not specified
# here or on the command line, input will default to STDIN. If
# the log filename ends in '.gz' (a gzip compressed file), or '.bz2'
# (bzip2 compressed file), it will be decompressed on the fly as it
# is being read.

LogFile /var/log/apache2/access.log.1
```

El “LogFile” por defecto es “/var/log/apache2/access.log.1”. Lo deberemos cambiar a nuestra ruta “/var/log/apache2/access.log”.

```
# LogFile defines the web server log file to use. If not specified
# here or on the command line, input will default to STDIN. If
# the log filename ends in '.gz' (a gzip compressed file), or '.bz2'
# (bzip2 compressed file), it will be decompressed on the fly as it
# is being read.

LogFile /var/log/apache2/access.log
```

PASO 3) Observamos que la ruta por defecto donde queda almacenado el recurso que se servirá al navegador es “/var/www/webalizer”.

```
# OutputDir is where you want to put the output files. This should
# should be a full path name, however relative ones might work as well.
# If no output directory is specified, the current directory will be used.

OutputDir /var/www/webalizer
```

Este recurso hay que moverlo a “/var/www/html/webalizer” y cambiar la directiva “OutputDir” para sincronizar nuestro servidor Apache con Webalizer.

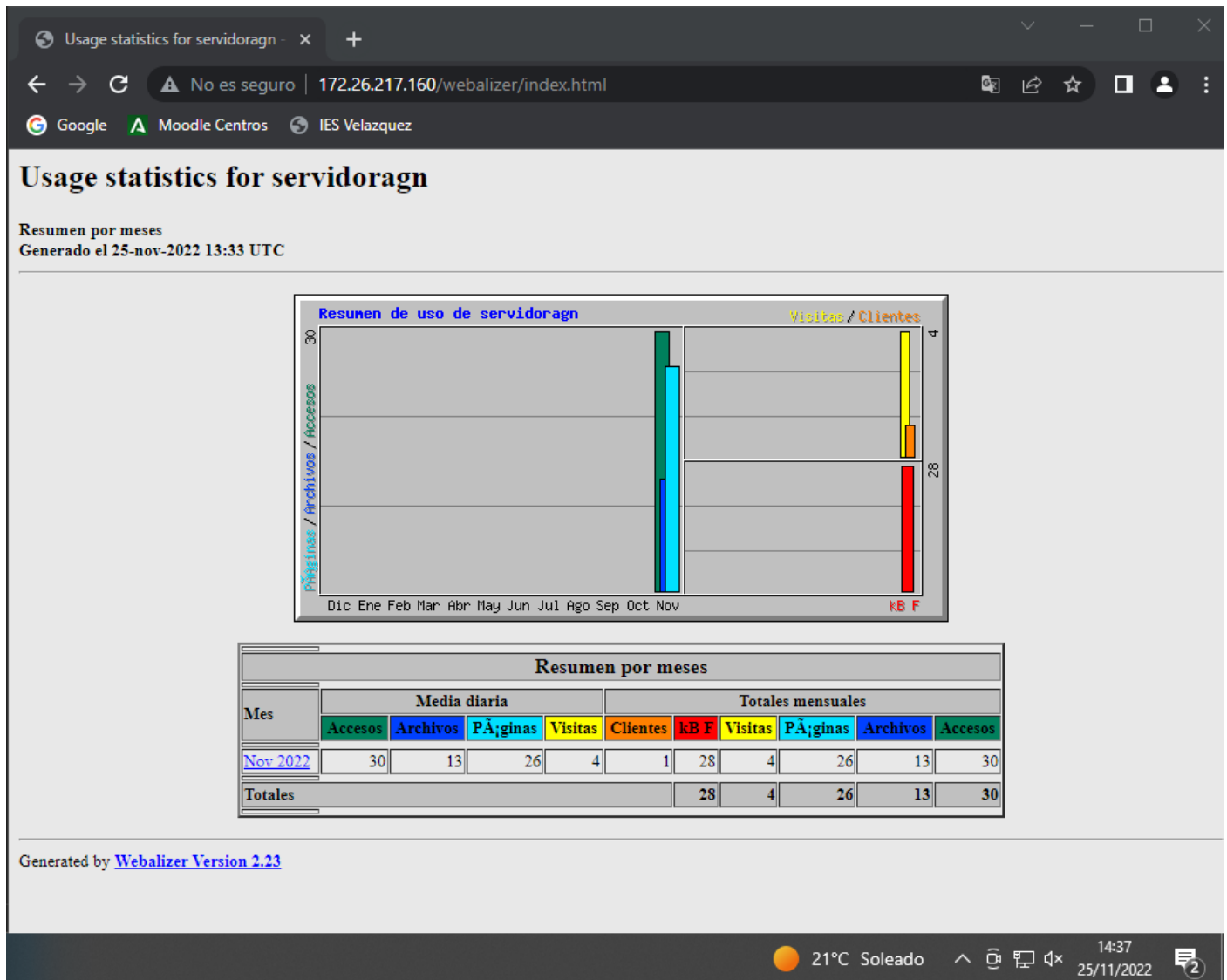
```
# OutputDir is where you want to put the output files. This should
# should be a full path name, however relative ones might work as well.
# If no output directory is specified, the current directory will be used.

OutputDir /var/www/html/webalizer
```

PASO 4) Lanzamos el programa con “sudo webalizer” para que lea el fichero log y genere el documento html con las estadísticas.

```
agn@servidoragn:~$ sudo webalizer
Webalizer V2.23-08 (Linux 5.4.0-132-generic x86_64) locale: #FcgU
Utilizando histórico /var/log/apache2/access.log (clf)
Creando informe en /var/www/html/webalizer
El nombre de máquina en el informe es 'servidoragn'
No encuentro el archivo histórico...
Generando informe de Noviembre 2022
Guardando información de archivo...
Generando informe resumido
30 registros en 1 segundos, 30/sec
agn@servidoragn:~$
```

PASO 5) Ahora accedemos al recurso “/webalizer” desde nuestro navegador y observamos la monitorización.

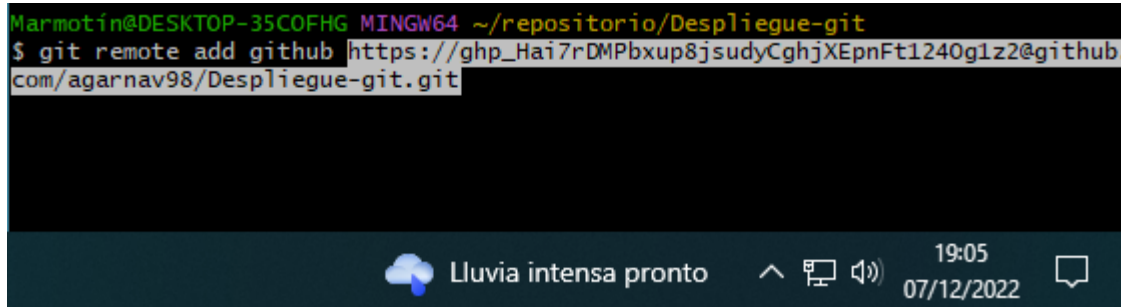


F) GitHub.

PASO 1) Añadimos con “git remote add github” la ruta de nuestro github junto al token que nos permite conectarnos ejecutando. La URL es la siguiente:

https://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@github.com/agarnav98/Despliegue-git.git
(Verificamos que está configurado de anteriores prácticas).

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

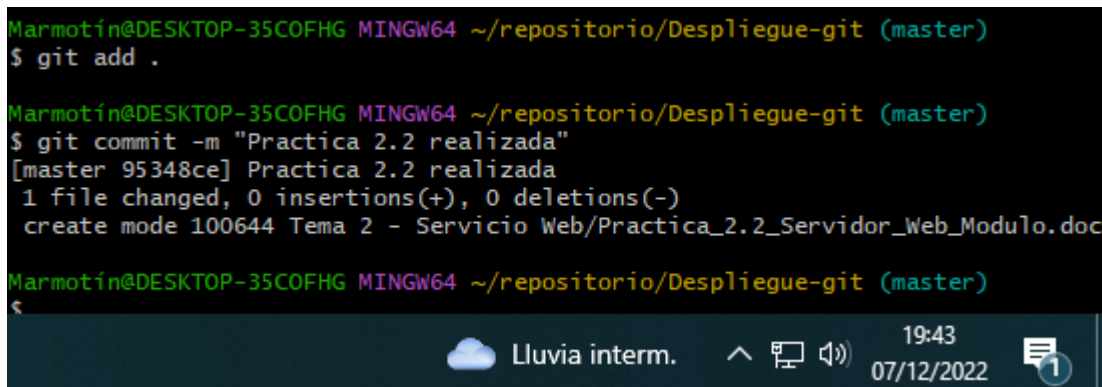


PASO 2) Una vez establecido la ruta, creamos una carpeta con nuestro archivo que se va a subir al repositorio. Hacemos “git add .” para añadir el documento a la zona de intercambio temporal, luego “git commit -m” para subir el documento a nuestro repositorio local y añadiéndole un mensaje. Y, por último, hacemos “git push” 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 2.2 realizada"
[master 95348ce] Practica 2.2 realizada
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Tema 2 - Servicio Web/Practica_2.2_Servidor_Web_Modulo.docx

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



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