



7-12-2020

[Entrega final]

[DAW]



Jorge Guerrero
2º DAW

Contenido

- Apache:.....	2
- 02 Apache virtual hosting:.....	2
- 03 Apache mapeo de URL.....	3
- 04 Apache control de acceso, autenticación	4
- Vsftpd.....	5
- Git (iniciación)	7
- Configuración global inicial	7
- Creación de repositorio local y remoto.....	7
- Comandos básicos para la gestión de archivos usando git.....	9

- Apache:

- Lo primero para realizarlo será contar con el servicio instalado de apache. (sudo apt -get install apache2)
- Procederemos a verificar que todo está funcionando correctamente. (sudo systemctl status apache2)
- Una vez que veamos que todo funciona correctamente comenzaremos con las tareas a realizar:

- 02 Apache virtual hosting:

1. Nos iremos a /etc/apache2/sites-available y copiaremos el archivo 000-default.conf con otro nombre por ejemplo 1daw.conf y 2daw.conf, sobre el que trabajaremos y realizaremos modificaciones.

```
usuario@hobbit:/etc/apache2/sites-available$ ls
000-default.conf  1daw.conf  2daw.conf  default-ssl.conf
```

2. Añadiremos en ambos archivos un ServerName 1daw.jorge.net y ServerName 2daw.jorge.net respectivamente en ambos archivos.
3. Modificaremos el DocumentRoot y pondremos /var/www/1daw y 2daw en el segundo archivo .conf

```
# However, you must set it for any further v
ServerName 1daw.jorge.net

ServerAdmin webmaster@localhost
DocumentRoot /var/www/1daw
```

4. Ponemos nombres distintos a los archivos logs para diferenciar los tipos de errores y que sea más fácil de encontrarlos y filtrarlos.

```
ErrorLog ${APACHE_LOG_DIR}/1daw-error.log
CustomLog ${APACHE_LOG_DIR}/1daw-access.log combined
```

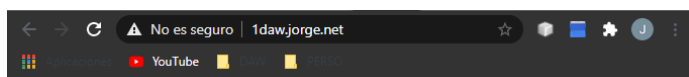
5. Nos vamos a /var/www y tendremos que crear un directorio 1daw y otro 2daw y le cambiamos los permisos para que el propietario de estas carpetas sean del usuario apache. (chown -R www-data:www-data 1daw/2daw)

```
usuario@hobbit:/etc/apache2/sites-available$ cd /var/www
usuario@hobbit:/var/www$ ls
1daw 2daw html
usuario@hobbit:/var/www$ ls -la
total 20
drwxr-xr-x  5 root    root    4096 Nov  6 18:32 .
drwxr-xr-x 14 root    root    4096 Nov  6 09:36 ..
drwxr-xr-x  3 www-data www-data 4096 Nov 20 18:26 1daw
drwxr-xr-x  2 www-data www-data 4096 Nov  6 18:52 2daw
drwxr-xr-x  2 root    root    4096 Nov 20 18:27 html
usuario@hobbit:/var/www$
```

6. Nos iremos a /etc/apache2/sites-enabled y usaremos a2ensite 1daw y a2ensite 2daw para crear los enlaces.

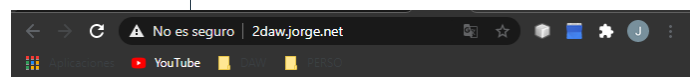
7. Dentro de las carpetas de www/1daw y 2daw tendremos que crear un index.html con el contenido que queramos y será el que se mostrará al acceder.
8. Deshabilitamos el sitio por defecto en sites-enabled con a2dissite 000-default.conf
9. Añadimos la url al archivo de C:\Windows\System32\drivers\etc\hosts en Windows y al acceder ya nos debería de salir el HTML que hemos creado anteriormente.

```
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
192.168.2.20    1daw.jorge.net
192.168.2.20    2daw.jorge.net
```



1Â° DAW

- BDD
- PROGRAMACION
- ED
- LDM
- FOL



2Â° DAW

- DAW
- DTW
- DWEC
- DWES
- EIE
- LC

- 03 Apache mapeo de URL

1. Usaremos la misma carpeta 1daw y 1daw.conf que en el 02. Por lo tanto nos ahorramos todos esos pasos.
2. En los archivos .conf añadiremos las siguientes líneas:

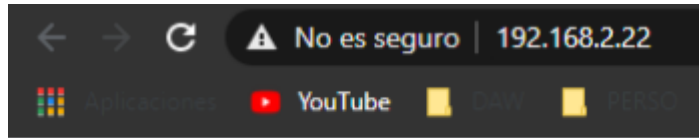
```
# However, you must set it for any further v
ServerName 1daw.jorge.net

ServerAdmin webmaster@localhost
DocumentRoot /var/www/1daw
<Directory /var/www/1daw/internacional>
    Options +Multiviews
</Directory>
```

3. En la carpeta /etc/www/1daw crearemos varios archivos .html que acaben en el idioma que queramos usar, en este caso .es y .en.

```
usuario@hobbit:/var/www/1daw$  
index.html.en index.html.es
```

4. Al cambiar el idioma del navegador deberá mostrarnos según el idioma que seleccionemos.



Bienvenido



- 04 Apache control de acceso, autenticación

1. Usaremos la carpeta 1daw y 1daw.conf de los ejemplos anteriores.
2. Accederemos a 1daw.conf y lo que haremos será denegar a todos menos a la red interna. Sustituiremos /servidor/interna por la carpeta que tengamos, en mi caso 1daw.

```
<Directory /var/www/servidor/interna>  
    Order Allow,Deny  
    Allow from 172.22.0  
</Directory>
```

3. Para el siguiente ejemplo usaremos autenticación:
4. Crearemos un archivo de contraseñas en /etc/apache2/claves con el comando "htpasswd -c passwd.txt usuario1" y os pedirá que introduzcáis la clave de dicho usuario. Para añadir otro usuario sería igual pero sin la opción -c y cambiando el nombre del usuario.
5. Nos iremos a 1daw.config y añadiremos la siguiente línea justo bajo DocumentRoot:

```
<Directory "/var/www/1daw">  
    AuthUserFile "/etc/apache2/claves/passwd.txt"  
    AuthName "Palabra de paso"  
    AuthType Basic  
    Require valid-user  
</Directory>
```

6. Cuando accedamos al sitio web nos pedirá usuario y contraseña.

- Vsftpd

1. Instalaremos el servicio

```
usuario@hobbit:/var/www/1daw$ sudo apt-get install vsftpd
```

2. Realizaremos una copia de seguridad del archivo de configuración.

```
usuario@hobbit:/var/www/1daw$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.original
usuario@hobbit:/var/www/1daw$
```

3. Vamos a habilitar el tráfico ftp, si vemos el estado(`sudo ufw status`) deberá salir inactivo por lo que vamos a ejecutar los siguientes comandos y deberá salir en el status así:

```
usuario@hobbit:/var/www/1daw$ sudo ufw allow 20/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
usuario@hobbit:/var/www/1daw$ sudo ufw allow 21/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
usuario@hobbit:/var/www/1daw$ sudo ufw allow 990/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
usuario@hobbit:/var/www/1daw$ sudo ufw allow 40000:50000/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
usuario@hobbit:/var/www/1daw$ sudo ufw status
Status: active
```

To	Action	From
--	-----	----
20/tcp	ALLOW	Anywhere
21/tcp	ALLOW	Anywhere
990/tcp	ALLOW	Anywhere
40000:50000/tcp	ALLOW	Anywhere
20/tcp (v6)	ALLOW	Anywhere (v6)
21/tcp (v6)	ALLOW	Anywhere (v6)
990/tcp (v6)	ALLOW	Anywhere (v6)
40000:50000/tcp (v6)	ALLOW	Anywhere (v6)

4. Creamos un usuario y en su home una carpeta que se llame ftp.

```
usuario@hobbit:/var/www/1daw$ sudo mkdir /home/jorge/ftp
usuario@hobbit:/var/www/1daw$ _
```

5. Modificamos los permisos de esta carpeta ejecutando los siguientes comandos:

```
usuario@hobbit:/var/www/1daw$ sudo chown nobody:nogroup /home/jorge/ftp/
usuario@hobbit:/var/www/1daw$ sudo chmod a-w /home/jorge/ftp/
usuario@hobbit:/var/www/1daw$ sudo ls -la /home/jorge/ftp/
total 8
dr-xr-xr-x 2 nobody nogroup 4096 Dec  7 16:33 .
drwxr-xr-x 3 jorge jorge 4096 Dec  7 16:33 ..
```

6. Creamos la carpeta de los archivos, modificamos los permisos y añadimos un archivo de pruebas:

```
usuario@hobbit:/var/www/1daw$ sudo mkdir /home/jorge/ftp/files
usuario@hobbit:/var/www/1daw$ sudo chown jorge:jorge /home/jorge/ftp/files/
usuario@hobbit:/var/www/1daw$ echo "vsftpd sample file" | sudo tee /home/jorge/ftp/files/vsftpd sample file
```

7. Accedemos al archivo `sudo nano /etc/vsftpd.conf` y des comentamos las líneas que dicen "**write_enable=YES**" y "**chroot_local_user=YES**".

8. Para que la conexión funcione con el usuario actual y todos los demás añadiremos al archivo:

```
user_sub_token=$USER
local_root=/home/$USER/ftp
```

9. Para garantizar que haya una cantidad considerable de conexiones disponibles:

```
pasv_min_port = 40000
pasv_max_port = 50000
```

10. Reiniciamos el servicio: **sudo systemctl restart vsftpd**

11. Procedemos a instalar el SSL para que sea seguro:

12. En primer lugar, debemos crear el certificado SSL y usarlo para proteger el servidor FTP de Ubuntu:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem
```

13. Abrimos el archivo de configuración nuevamente.

14. Debemos de comentar estas dos líneas para que queden tal que así:

```
# rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
# rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
```

15. Añadiremos estas para que redireccione hacia el nuevo certificado creado anteriormente:

```
rsa_cert_file=/etc/ssl/private/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
```

16. Habilitamos el SSL:

ssl_enable=YES

17. Prueba la conexión ftp a través del navegador o de un cliente y debes de acceder con el usuario creado anteriormente.

- Git (iniciación)

- Configuración global inicial

1. Instalaremos git (sudo apt install git)

- Creación de repositorio local y remoto

1. Creamos una carpeta, en mi caso la he creado en la carpeta raíz del usuario. Una vez estemos en ella ejecutamos el comando para inicializarlo con git.

```
usuario@hobbit:~/git$ git init
Initialized empty Git repository in /home/usuario/git/.git/
usuario@hobbit:~/git$ ls -la
total 12
drwxrwxr-x 3 usuario usuario 4096 Dec  7 17:10 .
drwxr-xr-x 6 usuario usuario 4096 Dec  7 17:10 ..
drwxrwxr-x 7 usuario usuario 4096 Dec  7 17:10 .git
```

2. Creamos un archivo de prueba dentro del repositorio.

```
usuario@hobbit:~/git$ echo "# Primera linea del archivo de prueba" > readme.md
usuario@hobbit:~/git$ _
```

3. Este archivo esta creado pero no subido, para subirlo usaremos el siguiente comando:

```
usuario@hobbit:~/git$ git add readme.md
usuario@hobbit:~/git$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   readme.md

usuario@hobbit:~/git$
```

4. Para modificar el archivo/aprobarlo deberemos de añadir un sistema de identificación que nos solicita git:

```
usuario@hobbit:~/git$ git config --global user.email "jgmruezgijon@gmail.com"
usuario@hobbit:~/git$ git config --global user.name "Jorge Guerrero"
```

5. Realizaremos la primera revisión/aprobación de git con el comando:

```
usuario@hobbit:~/git$ git commit -m "Mi primer commit"
[master (root-commit) 7409a90] Mi primer commit
1 file changed, 1 insertion(+)
 create mode 100644 readme.md
usuario@hobbit:~/git$ _
```


6. Creamos .gitignore que es para que todas las mascaras que se encuentren en dicho archivo no se incluyan en la revision.

```
GNU nano 5.2 .gitignore
*.tmp
```

7. Con git add . añade todos los archivos, que como vemos no ha añadido los .tmp incluidos en .gitignore.

```
usuario@hobbit:~/git$ ls
readme.md temp1.tmp temp2.tmp temp3.tmp
usuario@hobbit:~/git$ git add .
usuario@hobbit:~/git$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .gitignore
usuario@hobbit:~/git$
```

8. Para la creación de un git remoto deberemos seguir estos pasos (REALIZADO EN WINDOWS PARA SUBIR ESTE TRABAJO AL GIT Y REALIZAR LA ENTREGA):

Crearemos un repositorio en github por ejemplo y ejecutaremos una serie de comandos:

```
C:\Users\Jorge Guerrero\Desktop>cd "Entrega final"
C:\Users\Jorge Guerrero\Desktop\Entrega final>git init
Initialized empty Git repository in C:/Users/Jorge Guerrero/Desktop/Entrega final/.git/
C:\Users\Jorge Guerrero\Desktop\Entrega final>git add .
C:\Users\Jorge Guerrero\Desktop\Entrega final>git add .
C:\Users\Jorge Guerrero\Desktop\Entrega final>git commit -m "first commit"
[master (root-commit) a8294ba] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 prueba.txt
C:\Users\Jorge Guerrero\Desktop\Entrega final>git remote add origin https://github.com/JorgeGuerrero7/entrega_final_daw.git
C:\Users\Jorge Guerrero\Desktop\Entrega final>git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 207 bytes | 103.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/JorgeGuerrero7/entrega_final_daw.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
C:\Users\Jorge Guerrero\Desktop\Entrega final>
```

- Comandos básicos para la gestión de archivos usando git

ELIMINAR ARCHIVOS EN GIT

20) `git rm index.html` ==> elimina un archivo del Repository. Si hacemos un `git status` nos indicará que se ha borrado dicho archivo. Para eliminarlo del todo habría que hacer un commit: `git commit -m "eliminando index.html"`. Si se hace un "git status" el archivo "index.html" ya no está ni en el directorio de trabajo

MOVER ARCHIVOS EN GIT

21) Crear carpeta "style" y mover un archivo css a dicha carpeta desde la carpeta raíz del proyecto(`mkdir` y `mv`)

22) `git status` ==> nos indica que se ha borrado el archivo. Solución:

23) `git add .` (se añade de forma recursiva)

24) `git commit -m "nueva carpeta css"`

IGNORAR ARCHIVOS EN GIT

25) Crea un archivo `.log`. Queremos que ese tipo de archivos no se añada nunca

`nano .gitignore` ==> escribir `"*.log"`.

26) `git status` ==> aparece el archivo `.gitignore` para ser enviado a STAGE

27) Solución: `git add.` y `git commit -m ".ignore"`

GIT REMOTO

28) mkdir .ssh (en nuestra carpeta personal==> EN NUESTRO HOME)

29) cd .ssh

30) ssh-keygen -t rsa -C "transistor47@gmail.com"

31) ls -al ==> para ver los nuevos archivos creados. El archivo .pub es la clave pública que se tendrá que enviar a GitHub. Abriremos dicho archivo y copiaremos su contenido (en el editor vi se pulsará la tecla "V" para seleccionar el texto a copiar, "yy" para copiar, "p" para pegar)

32) Se inicia sesión en GitHub >> Settings >> SSH Keys >> crear una nueva clave. IMPORTANTE: Se tendrá que usar un entorno gráfico para realizar dichas acciones en GitHub

CREAR UN NUEVO REPOSITORIO

33) echo "#prueba" >> README.md

34) git add .

35) git commit -m "primer commit"

36) git remote add origin https://github.com/grafeno30/prueba.git

37) git push -u origin master

===MODIFICAR WORKING DIRECTORY Y ACTUALIZAR REPOSITORY===

37) Modifica cualquier archivo

38) git add .

39) git commit -m "Cambios realizados"

40) git status ==> comenta que hay cambios pendientes de sincronizar con el repositorio remoto

41) git pull origin master ==> IMPORTANTE: Solamente se hará esta operación si se trabaja en equipo,

pues lo que está haciendo es traerse al host local los últimos cambios

(Al hacer pull, el sistema recupera y trata de unir la rama remota con la local)

42) git push origin master ==> subimos los cambios