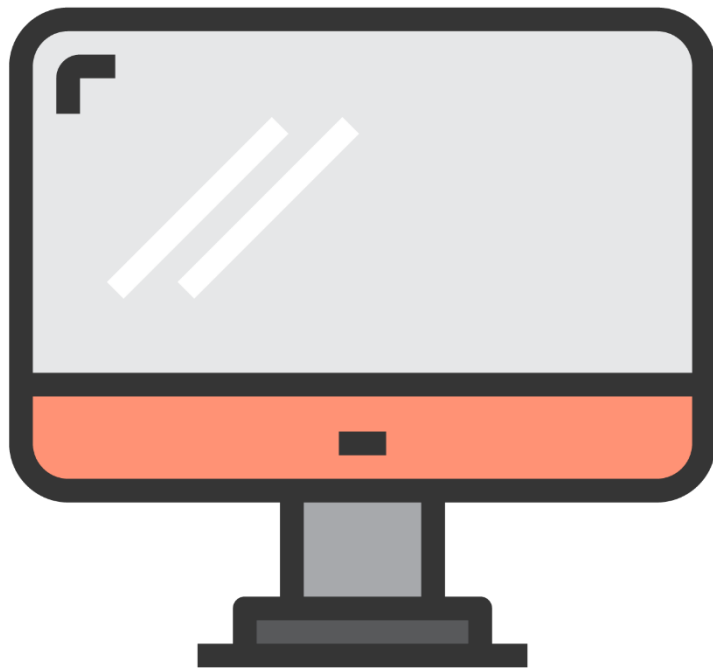


Documentación de instalación y configuración del entorno de desarrollo y explotación



INDICE

1. Ubuntu Server 20.04	4
1.1 Datos de configuración maquina Ubuntu Server 20.04	4
1.2. Cambiar nombre del host	4
1.3. Configurar dirección IP	6
1.4. Instalar servicio SSH	7
1.4.1. Instalación	7
1.4.2. Monitorización	7
1.5. Instalar Apache	8
1.5.1 Instalación	8
1.5.2 Configuración	8
1.5.3 Monitorización	16
1.6. Crear usuario operadorweb y asignar permisos.....	17
1.6.1 Configuración	17
1.7. Instalación y configuración del módulo de PHP 7.4.....	19
1.7.1 Instalación	19
1.7.2 Configuración	19
1.8. Instalación y configuración de MySQL.....	23
1.8.1. Instalación	23
1.8.2 Configuración	24
1.8.3 Ejemplo de uso	28
2. Windows 10	30
2.1. Datos de configuración maquina Windows 10.....	30
2.2. Cambiar nombre del host	31
2.3. Configurar dirección IP	32
2.4. Instalación de navegador Mozilla	33
2.5. Instalación de PuTTY.....	34
2.6. Instalación de Filezilla	35
2.7. Instalación de NetBeans	36
2.8. Conectar Proyecto de NetBeans con Ubuntu Server	37
2.8.1 Configuración	37
2.8.2 Ejemplo de uso	41
2.9. Administrar un repositorio con GitHub.....	43
2.9.1. Configuración	43

1. Ubuntu Server 20.04

1.1 Datos de configuración maquina Ubuntu Server 20.04

Nombre de la maquina	AFR-USED
Nombre del host	AFR-USED
Sistema Operativo	Ubuntu Server 20.04
Discos / Particiones	Disco de 500GB Particiones - 1M Boot System - 20GB Sistema en / - 480GB Datos en /var
RAM	2048MB
Usuarios y contraseñas	Nombre usuario: admin dwes Contraseña: admin Nombre de usuario: operadorweb Contraseña: operadorweb
Configuración de Red	Dirección de red: 192.168.3.10x Mascara: 255.255.255.0 Gateway: 192.168.3.1 DNS: 8.8.8.8

1.2. Cambiar nombre del host

Lo primero que haremos será cambiar el nombre del equipo, lo haremos con el siguiente comando:

```
admin dwes@d wes:~$ sudo hostnamectl set-hostname AFR-USED  
admin dwes@d wes:~$ reboot_
```

Para comprobar que se ha cambiado reiniciamos la maquina con el comando **reboot**.

Y una vez reiniciada introducimos el comando **cat /etc/hostname** para comprobar que se ha cambiado.

```
admin dwes@AFR-USED:~$ cat /etc/hostname  
AFR-USED
```

También lo podemos comprobar con el comando **hostname**.

```
admin dwes@AFR-USED:~$ hostname  
AFR-USED
```

Una vez cambiado en esta parte, también deberemos de cambiarlo a mano dentro del fichero **/etc/hosts**, para ello lo editaremos con el comando **sudo nano /etc/hosts**

```
admin dwes@AFR-USED:~$ sudo nano /etc/hosts
```

```
GNU nano 4.8 /etc/hosts Modified  
127.0.0.1 localhost  
127.0.1.1 AFR-USED_  
  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters
```

Reiniciamos de nuevo la máquina virtual y comprobamos que se ha cambiado correctamente con el comando **cat /etc/hosts**

```
admin@dwes@AFR-USED:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 AFR-USED

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0  ip6-localnet
ff00::0  ip6-mcastprefix
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
```

1.3. Configurar dirección IP

Lo primero que tenemos que hacer es ver el nombre de la tarjeta de red que tenemos que configurar, con el comando **ip link** nos lo mostrara, en mi caso la tarjeta a configurar tiene el nombre de **enp0s3**.

```
root@dwes:/etc/netplan# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:50:32:b7 brd ff:ff:ff:ff:ff:ff
root@dwes:/etc/netplan#
```

Una vez tenemos el nombre de la tarjeta de red que vamos a configurar, lo siguiente es entrar al archivo de configuración de red. Este archivo se encuentra en la carpeta **/etc/netplan** que debemos acceder con **cd**, para editar el archivo usaremos el comando **sudo nano** y le pondremos el nombre de **01-netcfg.yaml** (Todos los archivos de configuración de red deben tener la extensión **yaml** para que funcionen y puedan ser validados).

```
root@dwes:~# cd /etc/netplan
root@dwes:/etc/netplan# sudo nano 01-netcfg.yaml
```

Una vez dentro del archivo introduciremos la siguiente información con el nombre obtenido anteriormente de la tarjeta de red, nuestra ip y nuestra Gateway correspondiente. (IMPORTANTE no usar tabulador, todos los espacios tienen que ser puestos a mano para poder validar el archivo más adelante).

```
GNU nano 4.8                                01-netcfg.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.1.107/24
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
```

Una vez guardado el archivo de configuración, vamos a aplicarlo, para aplicarlo usaremos el comando **sudo netplan apply**.

```
root@dwes:/etc/netplan# sudo netplan apply
```

Por último, revisaremos de nuevo la tarjeta de red para ver si se ha aplicado la configuración correctamente. Lo haremos con el comando **ip addr show dev enp0s3** (en mi caso es **enp0s3** porque es el nombre de mi tarjeta de red, cada uno el que le corresponda).

```
root@dwes:/etc/netplan# ip addr show dev enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:50:32:b7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.107/24 brd 192.168.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe50:32b7/64 scope link
        valid_lft forever preferred_lft forever
root@dwes:/etc/netplan#
```

1.4. Instalar servicio SSH

1.4.1. Instalación

Antes de instalar el servicio SSH comprobaremos si está instalado en el sistema con el siguiente comando.

```
adminwes@AFR-USED:~$ ps -ef |grep ssh
adminw+  974      944  0 14:54 tty1      00:00:00 grep --color=auto ssh
```

Si en la pantalla nos muestra lo siguiente es que no está instalado.

Para instalarlo, antes de hacerlo, tendremos que hacer **sudo apt update** y **sudo apt upgrade** para actualizar el sistema.

```
adminwes@AFR-USED:~$ sudo apt update_
```

```
adminwes@AFR-USED:~$ sudo apt upgrade_
```

Una vez hecho lo anterior ya podemos instalar el servicio SSH con el siguiente comando:

```
adminwes@AFR-USED:~$ sudo apt install openssh-server
```

1.4.2. Monitorización

Una vez instalado, comprobaremos que está bien con el siguiente comando:

```
adminwes@AFR-USED:~$ ps -ef |grep ssh
root      40377      1  0 15:07 ?                00:00:00 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 s
tartups
adminw+   40611      944  0 15:08 tty1      00:00:00 grep --color=auto ssh
```

Si sale de esta manera y nos indica [listener] es que está instalado y funcionando.

Ahora ya podremos conectarnos a nuestro servidor usando una conexión remota con el programa PuTTY

1.5. Instalar Apache

1.5.1 Instalación

Antes de instalar el servidor Apache 2 comprobaremos si está instalado en el sistema con el siguiente comando:

```
adminwes@AFR-USED:~$ ps -ef |grep apache2
adminw+ 1539 1435 0 15:42 pts/0 00:00:00 grep --color=auto apache2
```

Si en la pantalla nos muestra lo siguiente es que no se encuentra instalado.

Para instalarlo, antes de hacerlo, tendremos que hacer **sudo apt update** y **sudo apt upgrade** para actualizar el sistema.

```
adminwes@AFR-USED:~$ sudo apt update
```

```
adminwes@AFR-USED:~$ sudo apt upgrade
```

Una vez hecho lo anterior ya podemos instalar el servidor Apache 2 con el siguiente comando:

```
adminwes@AFR-USED:~$ sudo apt install apache2
```

1.5.2 Configuración

En la carpeta apache2 podemos ver con **ls** todo lo que tiene dentro y entrar a ver el archivo ports el cual contiene los puertos que usa nuestro sitio web.

```
adminwes@AFR-USED:~$ cd /etc/apache2
adminwes@AFR-USED:/etc/apache2$ ls
apache2.conf  conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars      mods-available  ports.conf    sites-enabled
```

Entramos al archivo ports con el comando **sudo nano ports.conf** y podemos observar que está escuchando por el 80(http) y tiene disponible también el 443(https) mediante el módulo SSL que configuraremos más adelante.



```
adminwes@AFR-USED: /etc/apache2
GNU nano 4.8 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
```

Con este comando podemos ver los mods disponibles, filtrando con el parámetro SSL que es el mod del protocolo https que configuraremos más adelante.

```
adminwes@AFR-USED:/etc/apache2$ ls mods-available |grep ssl
ssl.conf
ssl.load
```


Vemos los mods que hay activados de apache2, lo cual no sale ninguno porque no hemos activado ninguno aún.

```
admindwes@AFR-USED:/etc/apache2$ ls mods-enabled |grep ssl
```

Lo siguiente que vamos a hacer es configurar el módulo SSL, el primer paso es activarlo con el siguiente comando:

```
admindwes@AFR-USED:/etc/apache2$ sudo 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 s
elf-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

Una vez activado reiniciaremos apache2 con el siguiente comando:

```
admindwes@AFR-USED:/etc/apache2$ systemctl restart apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to restart 'apache2.service'.
Authenticating as: admin (admindwes)
Password:
==== AUTHENTICATION COMPLETE ====
```

Cuando tengamos hecho lo anterior, nos vamos a ir a [sites-enabled](#) y vamos a hacer un **ls** para ver que archivos contiene.

```
admindwes@AFR-USED:/etc/apache2$ ls sites-enabled/
000-default.conf
```

También haremos un **ls** en [sites-available](#) para ver los archivos que contiene.

```
admindwes@AFR-USED:/etc/apache2$ ls sites-available/
000-default.conf  default-ssl.conf
```

Lo primero que haremos será echar un vistazo al archivo de configuración de sites-enabled, lo haremos con un **cat** y el nombre del archivo.

```
admin@dwes@AFR-USED:/etc/apache2$ cd sites-enabled/
admin@dwes@AFR-USED:/etc/apache2/sites-enabled$ cat 000-default.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
```

Ahora vamos a ver el status de apache, en este caso nos dará un error, el cual nos dice que www-browser: not found, significa que no tenemos instalado un navegador web en el sistema.

```
admin@dwes@AFR-USED:/etc/apache2/sites-enabled$ sudo apachectl status
/usr/sbin/apachectl: 113: www-browser: not found
'www-browser -dump http://localhost:80/server-status' failed.
Maybe you need to install a package providing www-browser or you
need to adjust the APACHE_LYNX variable in /etc/apache2/envvars
```

Ya que nos indica que no tenemos instalado un navegador, lo que vamos a hacer es instalarlo, en este caso he instalado el Lynx, con el siguiente comando:

```
adminwes@AFR-USED:/etc/apache2/sites-enabled$ sudo apt install lynx
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libidn11 lynx-common
Se instalarán los siguientes paquetes NUEVOS:
  libidn11 lynx lynx-common
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 1.586 kB de archivos.
Se utilizarán 5.731 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://es.archive.ubuntu.com/ubuntu focal/main amd64 libidn11 amd64 1.33-2.2ubuntu2 [46,2 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 lynx-common all 2.9.0dev.5-1 [914 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 lynx amd64 2.9.0dev.5-1 [626 kB]
Descargados 1.586 kB en 1s (1.324 kB/s)
Seleccionando el paquete libidn11:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 108827 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libidn11_1.33-2.2ubuntu2_amd64.deb ...
Desempaquetando libidn11:amd64 (1.33-2.2ubuntu2) ...
Seleccionando el paquete lynx-common previamente no seleccionado.
Preparando para desempaquetar .../lynx-common_2.9.0dev.5-1_all.deb ...
Desempaquetando lynx-common (2.9.0dev.5-1) ...
Seleccionando el paquete lynx previamente no seleccionado.
Preparando para desempaquetar .../lynx_2.9.0dev.5-1_amd64.deb ...
Desempaquetando lynx (2.9.0dev.5-1) ...
Configurando libidn11:amd64 (1.33-2.2ubuntu2) ...
Configurando lynx-common (2.9.0dev.5-1) ...
Configurando lynx (2.9.0dev.5-1) ...
update-alternatives: utilizando /usr/bin/lynx para proveer /usr/bin/www-browser (www-browser) en modo automático
Procesando disparadores para libc-bin (2.31-0ubuntu9.2) ...
Procesando disparadores para man-db (2.9.1-1) ...
Procesando disparadores para mime-support (3.64ubuntu1) ...
```

Una vez instalado, ya podremos acceder con él a nuestra página web, con poner el nombre del navegador y localhost nos abrirá la web, con el siguiente comando:

```
adminwes@AFR-USED:/etc/apache2/sites-enabled$ lynx http://localhost
```

Si tenemos activados los archivos de errores, etc... Ahora podemos ver que contiene nuestro archivo de errores de nuestra web. Para ello mostramos el contenido de la carpeta apache2 con el siguiente comando:

```
adminwes@AFR-USED:/etc/apache2$ ls /var/log/apache2/
access.log      access.log.2.gz  error.log.1      other_vhosts_access.log
access.log.1    error.log        error.log.2.gz
```

Una vez vistos los archivos, mostraremos el contenido de **error.log** con el siguiente comando (el archivo error log, archiva todos los errores que se producen):

```
admindwes@AFR-USED:/etc/apache2$ tail /var/log/apache2/error.log
[Fri Oct 08 10:25:05.841195 2021] [core:notice] [pid 785] AH00094: Command line:
'/usr/sbin/apache2'
[Fri Oct 08 10:43:33.046068 2021] [mpm_prefork:notice] [pid 785] AH00169: caught
SIGTERM, shutting down
[Fri Oct 08 10:43:33.154191 2021] [mpm_prefork:notice] [pid 1633] AH00163: Apache
2.4.41 (Ubuntu) configured -- resuming normal operations
[Fri Oct 08 10:43:33.156634 2021] [core:notice] [pid 1633] AH00094: Command line
: '/usr/sbin/apache2'
[Fri Oct 08 10:57:31.593205 2021] [mpm_prefork:notice] [pid 1633] AH00169: caught
SIGTERM, shutting down
[Fri Oct 08 10:57:31.709302 2021] [mpm_prefork:notice] [pid 1848] AH00163: Apache
2.4.41 (Ubuntu) configured -- resuming normal operations
[Fri Oct 08 10:57:31.709340 2021] [core:notice] [pid 1848] AH00094: Command line
: '/usr/sbin/apache2'
[Fri Oct 08 11:03:07.515379 2021] [mpm_prefork:notice] [pid 1848] AH00169: caught
SIGTERM, shutting down
[Fri Oct 08 11:03:07.630082 2021] [mpm_prefork:notice] [pid 1995] AH00163: Apache
2.4.41 (Ubuntu) OpenSSL/1.1.1f configured -- resuming normal operations
[Fri Oct 08 11:03:07.630115 2021] [core:notice] [pid 1995] AH00094: Command line
: '/usr/sbin/apache2'
```

Lo siguiente que vamos a hacer es cambiar la ruta de los archivos donde se guardan los errores y ponerlos en nuestra carpeta [/var/www/html](#)

Antes mostramos la información de este archivo con un cat para ver su contenido, pero ahora vamos a editarlo, antes de editarlo haremos una copia de dicho archivo por seguridad con el siguiente comando:

```
admindwes@AFR-USED:/etc/apache2$ sudo cp sites-enabled/000-default.conf sites-enabled/000-default.conf.backup
```

Una vez echa la copia de seguridad del archivo, ya entraremos a editarlo con el comando **sudo nano sites-enabled/000-default.conf**

Dentro de dicho archivo tenemos que añadir los nuevos directorios donde queremos que se guarden tanto el archivo error.log como el archivo Access.log.

ErrorLog /var/www/html/error.log

CustomLog /var/www/html/access.log combined

```
admin@dwes@AFR-USED: /etc/apache2
GNU nano 4.8      sites-enabled/000-default.conf      Modified
<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
ErrorLog /var/www/html/error.log
CustomLog /var/www/html/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
```

Una vez editado el archivo, lo guardaremos y lo cerramos. Ahora reiniciaremos el servicio apache2. Haremos un **ls** de la carpeta html y comprobaremos que estén los dos archivos que hemos metido con anterioridad al editar el archivo.

```
admin@dwes@AFR-USED:/etc/apache2$ sudo service apache2 restart
admin@dwes@AFR-USED:/etc/apache2$ ls /var/www/html/
access.log  error.log  index.html  proyectoDEAW  proyectoDWEK  proyectoTema3
doc         img       index.php  _proyectoDIW  proyectoDWES  webroot
```

Lo siguiente que haremos será configurar el sitio web con el módulo SSL, con el siguiente comando podemos ver que contiene el archivo de configuración del módulo.

```
admin@dwes@AFR-USED:/etc/apache2$ sudo cat sites-available/default-ssl.conf
```

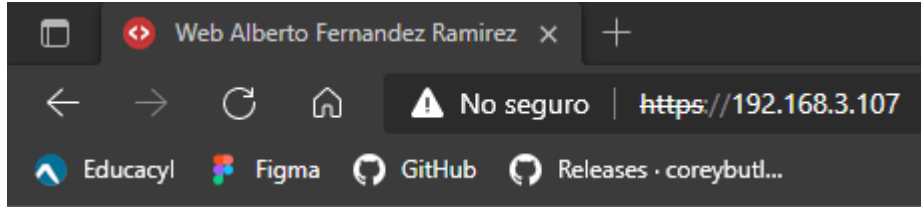
Lo primero que haremos es activar el módulo con el siguiente comando

```
admin@dwes@AFR-USED:/etc/apache2$ sudo a2ensite default-ssl.conf
Enabling site default-ssl.
To activate the new configuration, you need to run:
systemctl reload apache2
```

Ahora recargaremos el servicio apache con el siguiente comando:

```
|admindwes@AFR-USED:/etc/apache2$ sudo systemctl reload apache2
```

Una vez activado el módulo si accedemos a nuestro sitio web con https:// ya podremos ver que funciona, aunque nos muestra que no es seguro.



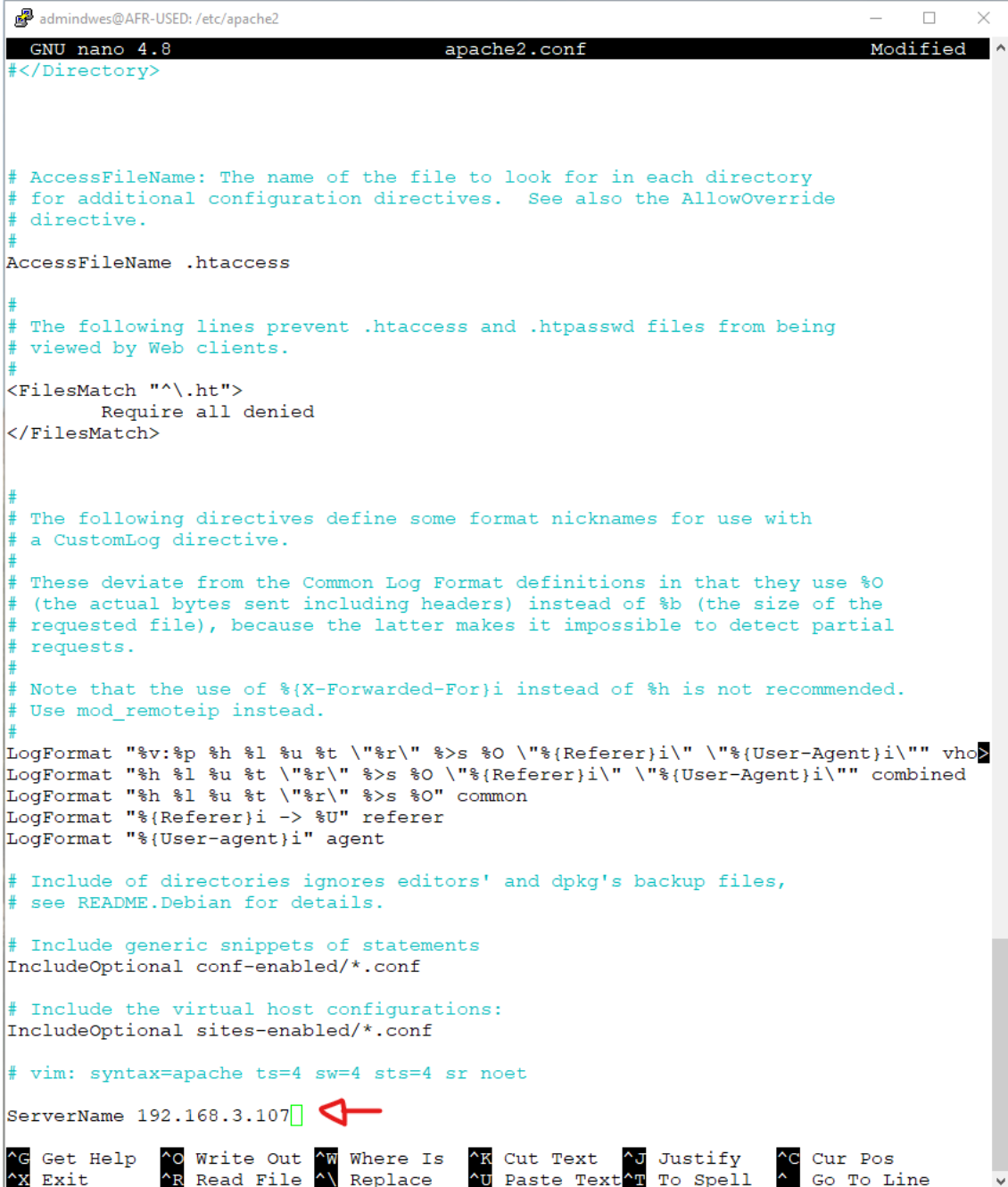
Si ejecutamos el comando `sudo apache2ctl -S` podremos ver el primer mensaje que nos muestra que no está configurado el `ServerName` en las directivas globales.

```
admindwes@AFR-USED:/etc/apache2$ sudo apache2ctl -S
AH00558: apache2: Could not reliably determine the server's fully qualified domain
name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this mes
sage
VirtualHost configuration:
*:80                  127.0.1.1 (/etc/apache2/sites-enabled/000-default.conf:1)
*:443                 127.0.1.1 (/etc/apache2/sites-enabled/default-ssl.conf:2)
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"
Mutex default: dir="/var/run/apache2/" mechanism=default
Mutex mpm-accept: using_defaults
Mutex watchdog-callback: using_defaults
Mutex ssl-stapling-refresh: using_defaults
Mutex ssl-stapling: using_defaults
Mutex ssl-cache: using_defaults
PidFile: "/var/run/apache2/apache2.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="www-data" id=33
Group: name="www-data" id=33
```

Para eliminar este error, editaremos el archivo `apache2.conf` con el siguiente comando.

```
|admindwes@AFR-USED:/etc/apache2$ sudo nano apache2.conf
```


Una vez dentro del archivo, indicaremos que el ServerName es la IP del servidor, en este caso la 192.168.3.107 una vez editado guardamos y ya desaparecerá dicho error.



```
admin@dwes@AFR-USED: /etc/apache2
GNU nano 4.8 apache2.conf Modified
#</Directory>

# AccessFileName: The name of the file to look for in each directory
# for additional configuration directives.  See also the AllowOverride
# directive.
#
AccessFileName .htaccess

#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<FilesMatch "\.ht">
    Require all denied
</FilesMatch>

#
# The following directives define some format nicknames for use with
# a CustomLog directive.
#
# These deviate from the Common Log Format definitions in that they use %O
# (the actual bytes sent including headers) instead of %b (the size of the
# requested file), because the latter makes it impossible to detect partial
# requests.
#
# Note that the use of %{X-Forwarded-For}i instead of %h is not recommended.
# Use mod_remoteip instead.
#
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" who
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %O" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf

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

ServerName 192.168.3.107
```

Para desactivar el módulo SSL de nuestro sitio web lo podremos hacer con el siguiente comando, que nos desactiva el archivo de configuración del módulo SSL:

```
admin@dwes@AFR-USED:/etc/apache2$ sudo a2dissite default-ssl.conf
Site default-ssl disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
```

Después será necesario también desactivar el módulo SSL con el siguiente comando:

```
adminwes@AFR-USED:/etc/apache2$ sudo a2dismod ssl
Module ssl disabled.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

Por último, reiniciamos el servicio apache2 para que se apliquen los cambios en el servidor con el siguiente comando:

```
adminwes@AFR-USED:/etc/apache2$ sudo service apache2 restart
```

1.5.3 Monitorización

Una vez instalado, comprobaremos que está bien con el siguiente comando:

```
adminwes@AFR-USED:~$ ps -ef |grep apache2
root          2706      1   0 15:43 ?           00:00:00 /usr/sbin/apache2 -k start
www-data      2708     2706   0 15:43 ?           00:00:00 /usr/sbin/apache2 -k start
www-data      2709     2706   0 15:43 ?           00:00:00 /usr/sbin/apache2 -k start
adminw+       3138     1435   0 15:43 pts/0      00:00:00 grep --color=auto apache2
```

Si nos muestra lo siguiente, es que apache 2 se ha instalado y funciona.

1.6. Crear usuario operadorweb y asignar permisos

1.6.1 Configuración

Necesitaremos un usuario operadorweb para poder subir ficheros con Filezilla y NetBeans en el directorio `/var/www/html`.

Creemos el usuario operador web en `/var/www/html` dentro del grupo `www-data`, indicando que no cree home y que este en el grupo `www-data` con el siguiente comando:

```
admin@AFR-USED:~$ sudo adduser --home /var/www/html --no-create-home --ingroup www-data operadorweb
```

Comprobamos que se ha creado correctamente y en el sitio que le hemos indicado con el comando `id operadorweb`

```
admin@AFR-USED:~$ id operadorweb
uid=1001(operadorweb) gid=33(www-data) groups=33(www-data)
```

Ahora le cambiamos el propietario a el directorio `html` poniéndole `operadorweb` e indicándole el grupo `www-data`. Nos meteremos en el directorio con `cd /var/www` y lo realizaremos con el siguiente comando:

```
admin@AFR-USED:/var/www$ sudo chown -R operadorweb:www-data html
```

Comprobamos que se ha cambiado correctamente con el comando `ll html`

```
admin@AFR-USED:/var/www$ ll html
total 36
drwxr-xr-x 8 operadorweb www-data 4096 oct 6 09:52 ./
drwxr-xr-x 3 root         root    4096 oct 5 10:32 ../
drwxrwxr-x 2 operadorweb www-data 4096 oct 6 07:57 doc/
drwxrwxr-x 2 operadorweb www-data 4096 oct 6 07:57 img/
-rw-rw-r-- 1 operadorweb www-data 3184 oct 6 08:10 index.php
drwxrwxr-x 2 operadorweb www-data 4096 oct 6 09:53 proyectoDEAW/
drwxrwxr-x 4 operadorweb www-data 4096 oct 6 08:01 proyectoDWES/
drwxrwxr-x 8 operadorweb www-data 4096 oct 6 08:01 proyectoTema3/
drwxrwxr-x 3 operadorweb www-data 4096 oct 6 07:57 webroot/
```

Lo siguiente es darle permisos 2775 sobre el directorio `html` con el siguiente comando:

```
admin@AFR-USED:/var/www$ sudo chmod -R 2775 html
```

Y comprobaremos que se ha hecho correctamente con el comando `ll html`

```
admin@AFR-USED:/var/www$ ll html
total 36
drwxrwsr-x 8 operadorweb www-data 4096 oct 6 09:52 ./
drwxr-xr-x 3 root         root    4096 oct 5 10:32 ../
drwxrwsr-x 2 operadorweb www-data 4096 oct 6 07:57 doc/
drwxrwsr-x 2 operadorweb www-data 4096 oct 6 07:57 img/
-rwxrwsr-x 1 operadorweb www-data 3184 oct 6 08:10 index.php*
drwxrwsr-x 2 operadorweb www-data 4096 oct 6 09:53 proyectoDEAW/
drwxrwsr-x 4 operadorweb www-data 4096 oct 6 08:01 proyectoDWES/
drwxrwsr-x 8 operadorweb www-data 4096 oct 6 08:01 proyectoTema3/
drwxrwsr-x 3 operadorweb www-data 4096 oct 6 07:57 webroot/
```


1.7. Instalación y configuración del módulo de PHP 7.4

1.7.1 Instalación

Para instalar el módulo de PHP en apache 2, lo primero que tenemos que hacer es un update y un upgrade del sistema con los siguientes comandos:

```
adminwes@AFR-USED:~$ sudo apt update
```

```
adminwes@AFR-USED:~$ sudo apt upgrade
```

Una vez realizado lo anterior, ya podemos instalar el módulo PHP con el siguiente comando (dicho comando nos instalara también la librería de PHP para apache2):

```
adminwes@AFR-USED:~$ sudo apt install php libapache2-mod-php
```

Una vez instalados los paquetes, reiniciaremos el servicio Apache:

```
adminwes@AFR-USED:~$ sudo systemctl restart apache2
```

1.7.2 Configuración

Para configurar PHP lo primero es buscar el archivo de configuración, nos meteremos en `/etc/php` y haremos un listado mediante el comando `ls`

```
adminwes@AFR-USED:~$ cd /etc/php
```

```
adminwes@AFR-USED:/etc/php$ ls
```

```
7.4
```

Una vez nos salga la carpeta la cual indica la versión de PHP nos meteremos en ella con `cd 7.4`

```
adminwes@AFR-USED:/etc/php/7.4$ ls
```

```
apache2  cli  mods-available
```

Cuando estemos dentro de la carpeta que indica la versión haremos un tree para ver el árbol de directorios y buscar el archivo de configuración de PHP en mi caso lo he hecho con **tree . /more** para ver todo.

```
adminwes@AFR-USED:/etc/php/7.4$ tree . /more
.
├── apache2
│   ├── conf.d
│   │   ├── 10-opcache.ini -> /etc/php/7.4/mods-available/opcache.ini
│   │   ├── 10-pdo.ini -> /etc/php/7.4/mods-available/pdo.ini
│   │   ├── 20-calendar.ini -> /etc/php/7.4/mods-available/calendar.ini
│   │   ├── 20-ctype.ini -> /etc/php/7.4/mods-available/ctype.ini
│   │   ├── 20-exif.ini -> /etc/php/7.4/mods-available/exif.ini
│   │   ├── 20-ffi.ini -> /etc/php/7.4/mods-available/ffi.ini
│   │   ├── 20-fileinfo.ini -> /etc/php/7.4/mods-available/fileinfo.ini
│   │   ├── 20-ftp.ini -> /etc/php/7.4/mods-available/ftp.ini
│   │   ├── 20-gettext.ini -> /etc/php/7.4/mods-available/gettext.ini
│   │   ├── 20-iconv.ini -> /etc/php/7.4/mods-available/iconv.ini
│   │   ├── 20-json.ini -> /etc/php/7.4/mods-available/json.ini
│   │   ├── 20-phar.ini -> /etc/php/7.4/mods-available/phar.ini
│   │   ├── 20-posix.ini -> /etc/php/7.4/mods-available/posix.ini
│   │   ├── 20-readline.ini -> /etc/php/7.4/mods-available/readline.ini
│   │   ├── 20-shmop.ini -> /etc/php/7.4/mods-available/shmop.ini
│   │   ├── 20-sockets.ini -> /etc/php/7.4/mods-available/sockets.ini
│   │   ├── 20-sysvmsg.ini -> /etc/php/7.4/mods-available/sysvmsg.ini
│   │   ├── 20-sysvsem.ini -> /etc/php/7.4/mods-available/sysvsem.ini
│   │   ├── 20-sysvshm.ini -> /etc/php/7.4/mods-available/sysvshm.ini
│   │   └── 20-tokenizer.ini -> /etc/php/7.4/mods-available/tokenizer.ini
│   └── php.ini
├── cli
│   ├── conf.d
│   │   ├── 10-opcache.ini -> /etc/php/7.4/mods-available/opcache.ini
│   │   ├── 10-pdo.ini -> /etc/php/7.4/mods-available/pdo.ini
│   │   ├── 20-calendar.ini -> /etc/php/7.4/mods-available/calendar.ini
│   │   ├── 20-ctype.ini -> /etc/php/7.4/mods-available/ctype.ini
│   │   ├── 20-exif.ini -> /etc/php/7.4/mods-available/exif.ini
│   │   ├── 20-ffi.ini -> /etc/php/7.4/mods-available/ffi.ini
│   │   ├── 20-fileinfo.ini -> /etc/php/7.4/mods-available/fileinfo.ini
│   │   ├── 20-ftp.ini -> /etc/php/7.4/mods-available/ftp.ini
│   │   ├── 20-gettext.ini -> /etc/php/7.4/mods-available/gettext.ini
│   │   ├── 20-iconv.ini -> /etc/php/7.4/mods-available/iconv.ini
│   │   ├── 20-json.ini -> /etc/php/7.4/mods-available/json.ini
│   │   ├── 20-phar.ini -> /etc/php/7.4/mods-available/phar.ini
│   │   ├── 20-posix.ini -> /etc/php/7.4/mods-available/posix.ini
│   │   ├── 20-readline.ini -> /etc/php/7.4/mods-available/readline.ini
│   │   ├── 20-shmop.ini -> /etc/php/7.4/mods-available/shmop.ini
│   │   ├── 20-sockets.ini -> /etc/php/7.4/mods-available/sockets.ini
│   │   ├── 20-sysvmsg.ini -> /etc/php/7.4/mods-available/sysvmsg.ini
│   │   ├── 20-sysvsem.ini -> /etc/php/7.4/mods-available/sysvsem.ini
│   │   ├── 20-sysvshm.ini -> /etc/php/7.4/mods-available/sysvshm.ini
│   │   └── 20-tokenizer.ini -> /etc/php/7.4/mods-available/tokenizer.ini
│   └── php.ini
└── mods-available
    ├── calendar.ini
    ├── ctype.ini
    ├── exif.ini
    ├── ffi.ini
    └── fileinfo.ini
```

Una vez encontrado el archivo llamado **php.ini**, antes de entrar a editar cualquier tipo de parámetro le tendremos que hacer una copia de seguridad al archivo con el siguiente comando, la cual en mi caso le he puesto el nombre del archivo **.backup**.

```
adminwes@AFR-USED:/etc/php/7.4$ sudo cp apache2/php.ini apache2/php.ini.backup
```

Una vez tenemos hecha la copia de seguridad del archivo. Dentro del archivo pasaremos a configurar los siguientes parámetros (Se busca mediante Ctrl+W y se edita según donde te encuentres, en este caso del lado de Desarrollo):

display_errors Determina si los errores deberían ser impresos en pantalla como parte de la salida o si deberían ocultarse al usuario. Lo pondremos en **On**.

```
; http://php.net/display-errors
display_errors = On
```

display_startup_errors Incluso cuando **display_errors** está activado, los errores que ocurren durante la secuencia de arranque de PHP no se muestran. Se recomienda encarecidamente mantener desactivado **display_startup_errors**, excepto para la depuración. Lo pondremos en **On**.

```
; http://php.net/display-startup-errors
display_startup_errors = On
```

html_errors Si se habilita, los mensajes de error incluirán etiquetas HTML. El formato para los errores HTML produce mensajes clicables que dirigen al usuario a una página que describe el error o función que causó el error. Lo pondremos en **On**.

```
; http://php.net/html-errors
html_errors = On
```

Una vez editemos los parámetros, guardamos el archivo y reiniciamos el servicio **apache2** con el siguiente comando para que se aplique la configuración.

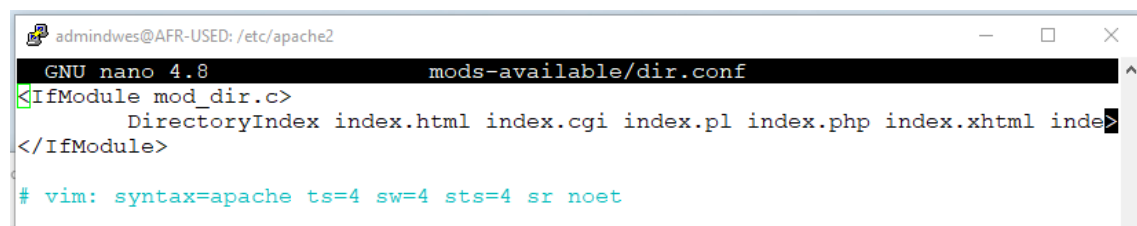
```
adminwes@AFR-USED:/etc/php/7.4$ sudo service apache2 restart
```

Lo siguiente que vamos a configurar es Otorgar prioridad a PHP, para ello nos situamos en el directorio que contiene los mods disponibles para PHP y hacemos un ls de los mods disponibles filtrando por dir con el siguiente comando:

```
adminwes@AFR-USED:/etc/apache2$ ls mods-available |grep dir
dir.conf
dir.load
userdir.conf
userdir.load
```

Antes de editar el archivo **dir.conf** lo abriremos para ver que contiene con el siguiente comando:

```
adminwes@AFR-USED:/etc/apache2$ sudo nano mods-available/dir.conf
```



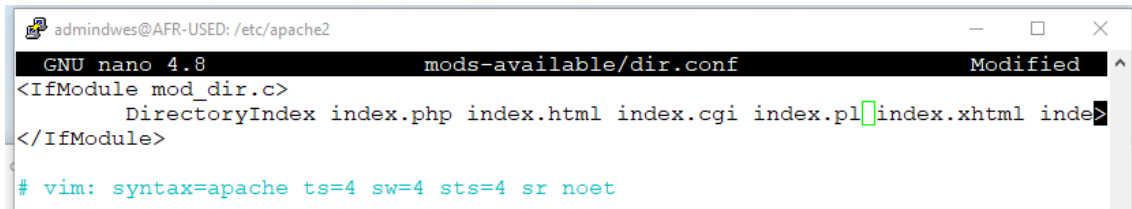
```
adminwes@AFR-USED: /etc/apache2
GNU nano 4.8 mods-available/dir.conf
<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml inde
</IfModule>

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

Antes de entrar a editar el archivo, crearemos un backup del archivo por seguridad con el siguiente comando:

```
admindwes@AFR-USED:/etc/apache2$ sudo cp mods-available/dir.conf mods-available/  
dir.conf.backup
```

Una vez tenemos hecha la copia de dicho archivo entramos a editar el archivo, debemos de priorizar el index de PHP y a continuación el de html de manera que nos ejecute el de PHP antes que el de html, una vez editado guardamos y salimos del archivo.



```
admindwes@AFR-USED: /etc/apache2
GNU nano 4.8 mods-available/dir.conf Modified
<IfModule mod_dir.c>
  DirectoryIndex index.php index.html index.cgi index.plindex.xhtml inde
</IfModule>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Reiniciamos el servicio apache2 para aplicar los cambios.

```
admindwes@AFR-USED:/etc/apache2$ sudo service apache2 restart
```

1.8. Instalación y configuración de MySQL

1.8.1. Instalación

Antes de empezar a instalar MySQL, haremos un update y un upgrade del sistema con los siguientes comandos:

```
adminwes@AFR-USED:~$ sudo apt update
```

```
|adminwes@AFR-USED:~$ sudo apt upgrade
```

Una vez hecho lo anterior, ya podemos instalar MySQL con el siguiente comando:

```
|adminwes@AFR-USED:~$ sudo apt install mysql-server
```

Una vez este instalado, entramos en la consola de MySQL con el comando **sudo mysql**, si nos sale lo siguiente y nos entra en la consola es que está instalado correctamente.

```
adminwes@AFR-USED:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.26-0ubuntu0.20.04.3 (Ubuntu)
```

```
Copyright (c) 2000, 2021, Oracle and/or its affiliates.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

1.8.2 Configuración

Podemos ver los puertos que usa MySQL con **ss -punta** y con **ss -puta** nos los muestra con el nombre.

```
adminwes@AFR-USED:~$ ss -punta
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
udp UNCONN 0 0 127.0.0.53%lo:53 0.0.0.0:*
tcp LISTEN 0 4096 127.0.0.53%lo:53 0.0.0.0:*
tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:*
tcp LISTEN 0 70 127.0.0.1:33060 0.0.0.0:*
tcp LISTEN 0 151 127.0.0.1:3306 0.0.0.0:*
tcp ESTAB 0 64 192.168.3.107:22 192.168.3.7:51018
tcp LISTEN 0 511 *:80 *:80
tcp LISTEN 0 128 [::]:22 [::]:*

adminwes@AFR-USED:~$ ss -puta
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
udp UNCONN 0 0 127.0.0.53%lo:domain 0.0.0.0:*
tcp LISTEN 0 4096 127.0.0.53%lo:domain 0.0.0.0:*
tcp LISTEN 0 128 0.0.0.0:ssh 0.0.0.0:*
tcp LISTEN 0 70 127.0.0.1:33060 0.0.0.0:*
tcp LISTEN 0 151 127.0.0.1:mysql 0.0.0.0:*
tcp ESTAB 0 64 192.168.3.107:ssh 192.168.3.7:51018
tcp LISTEN 0 511 *:http *:80
tcp LISTEN 0 128 [::]:ssh [::]:*
```

Ahora entraremos a configurar MySQL, para ello deberemos acceder al archivo `mysqld.cnf` y editarlo con el siguiente comando:

```
adminwes@AFR-USED:~$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

Una vez dentro del archivo, debemos comentar las líneas `bind-address` y `mysqlx-bind-address`, para comentarlas les pondremos una `#` delante a ambas líneas.

```
GNU nano 4.8 /etc/mysql/mysql.conf.d/mysqld.cnf Modified
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# Here is entries for some specific programs
# The following values assume you have at least 32M ram
#
[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file           = /var/run/mysqld/mysqld.pid
# socket             = /var/run/mysqld/mysqld.sock
# port               = 3306
# datadir            = /var/lib/mysql
#
# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.>
# tmpdir             = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address        = 127.0.0.1
#mysqlx-bind-address = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size     = 16M
# max_allowed_packet = 64M
# thread_stack       = 256K
#
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```


Una vez editado el archivo lo guardamos y lo cerramos, reiniciaremos el servicio de mysql para que se apliquen los cambios con el siguiente comando:

```
adminwes@AFR-USED:~$ sudo service mysql restart
```

Ahora entraremos a realizar la configuración de mysql con el siguiente comando:

```
adminwes@AFR-USED:~$ sudo mysql_secure_installation
```

Lo primero que nos pedirá es Validar el componente Password, le pondremos que sí.

```
Securing the MySQL server deployment.
```

```
Connecting to MySQL using a blank password.
```

```
VALIDATE PASSWORD COMPONENT can be used to test passwords  
and improve security. It checks the strength of password  
and allows the users to set only those passwords which are  
secure enough. Would you like to setup VALIDATE PASSWORD component?
```

```
Press y|Y for Yes, any other key for No: Y
```

Lo siguiente que nos dira es el nivel de la password, en mi caso voy a usar el LOW con lo que pondre un 0, seguidamente nos pedira introducir la password que queremos para acceder, en mi caso he usado P@ssw0rd y el ultimo paso he puesto si para continuar con la password usada.

```
There are three levels of password validation policy:
```

```
LOW      Length >= 8  
MEDIUM Length >= 8, numeric, mixed case, and special characters  
STRONG Length >= 8, numeric, mixed case, special characters and dictionary  
file
```

```
Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0  
Please set the password for root here.
```

```
New password:
```

```
Re-enter new password:
```

```
Estimated strength of the password: 100  
Do you wish to continue with the password provided?(Press y|Y for Yes, any other  
key for No) : Y
```

Lo siguiente que nos dirá es si queremos que nos cree un usuario anónimo por defecto, le diremos que no.

```
By default, a MySQL installation has an anonymous user,  
allowing anyone to log into MySQL without having to have  
a user account created for them. This is intended only for  
testing, and to make the installation go a bit smoother.  
You should remove them before moving into a production  
environment.
```

```
Remove anonymous users? (Press y|Y for Yes, any other key for No) : Y  
Success.
```

Lo siguiente que nos dirá es si queremos que root se conecte desde localhost, le diremos que sí.

```
Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.
```

```
Disallow root login remotely? (Press y|Y for Yes, any other key for No) : Y
Success.
```

Lo siguiente que nos dirá es que MySQL crea la base de datos test por defecto, y nos pregunta si queremos eliminarla, en este acaso le diremos que sí, ya que no la necesitamos.

```
By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.
```

```
Remove test database and access to it? (Press y|Y for Yes, any other key for No)
: Y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.
```

Por último, nos preguntara si queremos cargar los privilegios en las tablas ahora, le diremos que sí.

```
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.
```

```
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : Y
Success.
```

```
All done!
```

Lo siguiente que haremos será instalar el mod para MySQL de PHP con el siguiente comando:

```
|admindwes@AFR-USED:~$ sudo apt install libapache2-mod-php php-mysql
```

Una vez hecho, si ahora ponemos `sudo mysql` se nos debería de abrir la consola de `mysql`. Una vez se nos abra crearemos el usuario `admindb` y con la contraseña `P@ssw0rd` y le daremos todos los privilegios.

Para hacerlo usaremos dos comandos, uno para crearlo, **CREATE USER 'admindb'@'%' IDENTIFIED BY 'P@ssw0rd'**; y le daremos los permisos con el comando **GRANT ALL PRIVILEGES ON *.* TO 'admindb'@'%' WITH GRANT OPTION**;

Una vez lo creemos y le asignemos los permisos salimos con **exit**

```
adminwes@AFR-USED:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.26-0ubuntu0.20.04.3 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'admindb'@'%' IDENTIFIED BY 'P@ssw0rd';
Query OK, 0 rows affected (0,02 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'admindb'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0,01 sec)
```

Ahora ya podemos acceder a la consola de MySQL con el nuevo usuario `admindb` con el siguiente comando:

```
adminwes@AFR-USED:~$ mysql -u admindb -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.26-0ubuntu0.20.04.3 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Para ver los usuarios, dentro de la consola de MySQL pondremos el siguiente comando:

```
mysql> SELECT user, authentication_string, plugin, host FROM mysql.user;
+-----+-----+-----+-----+
| user                | authentication_string | plugin                | host                |
+-----+-----+-----+-----+
| admindb             | $A$005$b)uR\R'?k~F!7005/0d8/MEUWBikpBdjK/pi8F.hfAtEYvfL012KKo21 | caching_sha2_password | %                   |
| debian-sys-maint    | $A$005$m6,q)DPVo    %Y7"6x508EAacspyx/8TUg4fKYpMh1BJIiqZ/pQYpi/nXAC | caching_sha2_password | localhost           |
| mysql.infoschema    | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password | localhost           |
| mysql.session       | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password | localhost           |
| mysql.sys           | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password | localhost           |
| root                | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | auth_socket           | localhost           |
+-----+-----+-----+-----+
6 rows in set (0,00 sec)
```

1.8.3 Ejemplo de uso

Para crear una base de datos, un usuario y darle permisos usaremos los siguientes comandos:

```
mysql> CREATE DATABASE example_database;
Query OK, 1 row affected (0,01 sec)

mysql> CREATE USER 'example_user'@'%' IDENTIFIED WITH mysql_native_password BY 'P@ssw0rd';
Query OK, 0 rows affected (0,00 sec)

mysql> GRANT ALL ON example_database.* TO 'example_user'@'%';
Query OK, 0 rows affected (0,01 sec)

mysql> exit
Bye
```

Una vez esta creada la base de datos y el usuario, ahora ya podemos acceder a la consola de MySQL con dicho usuario.

```
admin@dwes:~$ mysql -u example_user -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.26-0ubuntu0.20.04.3 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

Dentro de la consola para asegurarnos que nos creó anteriormente la base de datos, usaremos el siguiente comando para ver las bases de datos:

```
mysql> show databases;
+-----+
| Database                |
+-----+
| example_database        |
| information_schema      |
+-----+
2 rows in set (0,00 sec)
```

Ahora pasaremos a crear una tabla con unos parámetros dentro de la tabla con el siguiente comando:

```
mysql> CREATE TABLE example_database.todo_list (
  -> item_id INT AUTO_INCREMENT,
  -> content VARCHAR(255),
  -> PRIMARY KEY(item_id)
  -> );
Query OK, 0 rows affected (0,03 sec)
```

Una vez tenemos creada la tabla, le insertamos valores con el siguiente comando:

```
mysql> INSERT INTO example_database.todo_list (content) VALUES ("Primer item");
Query OK, 1 row affected (0,01 sec)
```

```
mysql> INSERT INTO example_database.todo_list (content) VALUES ("Segundo item");
Query OK, 1 row affected (0,00 sec)

mysql> INSERT INTO example_database.todo_list (content) VALUES ("Tercer item");
Query OK, 1 row affected (0,01 sec)

mysql> INSERT INTO example_database.todo_list (content) VALUES ("Cuarto item");
Query OK, 1 row affected (0,01 sec)
```

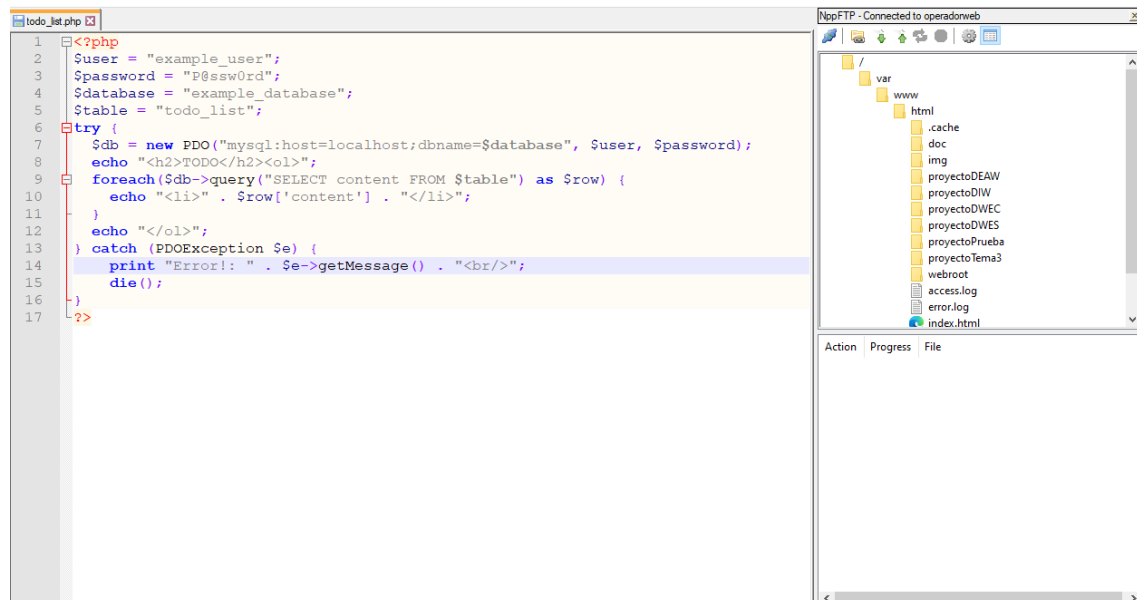
Una vez metidos todos los valores, comprobamos que se han introducido bien en la tabla con el siguiente comando:

```
mysql> SELECT * FROM example_database.todo_list;
+-----+-----+
| item_id | content |
+-----+-----+
|      1 | Primer item |
|      2 | Segundo item |
|      3 | Tercer item |
|      4 | Cuarto item |
+-----+-----+
4 rows in set (0,00 sec)
```

Ahora vamos a crear un archivo en PHP que muestre el contenido de la base de datos creada anteriormente, para ello nos situaremos en /var/www/html y hay crearemos el archivo todo_list.php con un nano, se puede realizar con el siguiente comando:

```
admin@dwes@AFR-USED:~$ nano /var/www/html/todo_list.php
```

Una vez creado, nos vamos a Notepad++ y insertamos este código, el cual permitirá leer la base de datos e imprimir su contenido por pantalla en nuestra página web.



2. Windows 10

2.1. Datos de configuración maquina Windows 10

Nombre de la maquina	AFR-WXED
Nombre del host	AFR-WXED
Sistema Operativo	Windows 10
Discos / Particiones	Disco de 500GB Particiones: - 50GB Sistema - 450GB Datos
RAM	4096MB
Usuarios y contraseñas	Nombre usuario: admin Contraseña: admin Nombre de usuario: operadorweb Contraseña: operadorweb
Configuración de Red	Dirección de red: 192.168.3.10x Mascara: 255.255.255.0 Gateway: 192.168.3.1 DNS: 8.8.8.8

2.2. Cambiar nombre del host

2.3. Configurar dirección IP

2.4. Instalación de navegador Mozilla

2.5. Instalación de PuTTY

2.6. Instalación de Filezilla

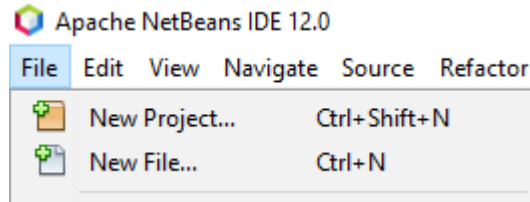
2.7. Instalación de NetBeans

2.8. Conectar Proyecto de NetBeans con Ubuntu Server

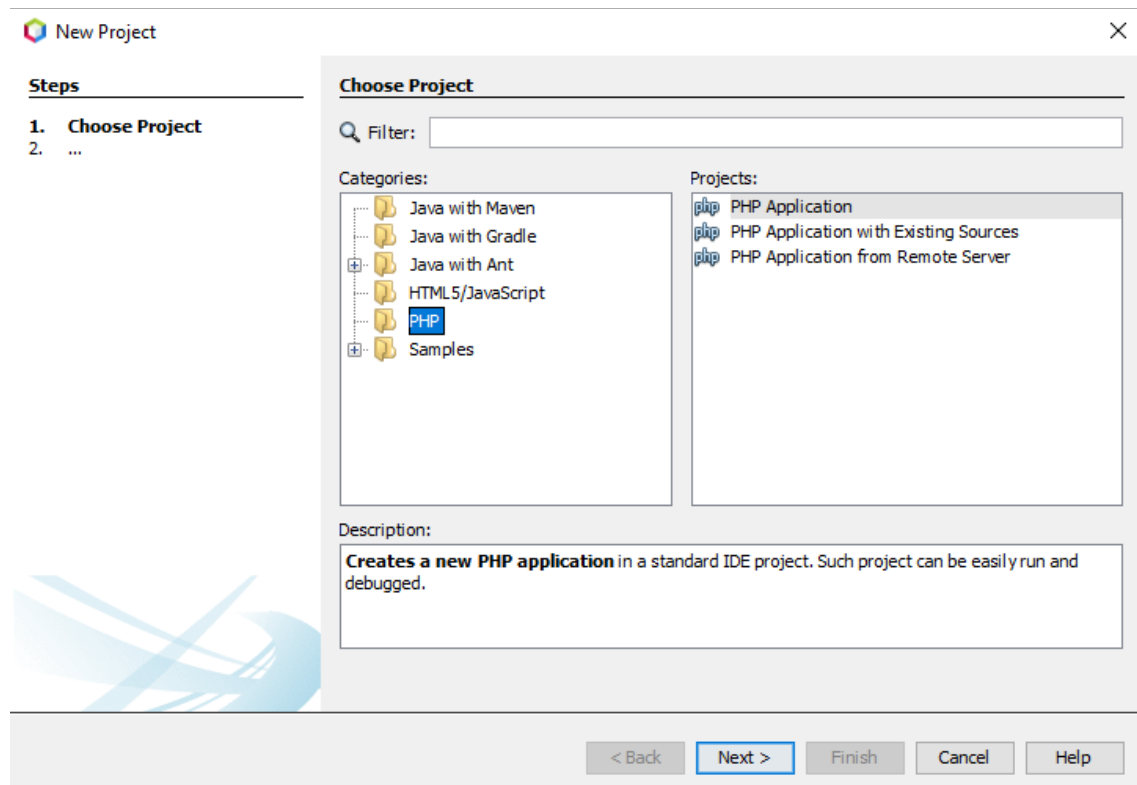
2.8.1 Configuración

Lo primero que tenemos que hacer es irnos a crear un proyecto nuevo en NetBeans.

IMPORTANTE: Tener NetBeans sin ningún proyecto abierto, tener vacío el directorio /var/www/html en el servidor de Ubuntu y tener arrancado Ubuntu Server.



Se nos abrirá un cuadro para configurar el proyecto nuevo, dentro de este seleccionamos la categoría PHP y el proyecto PHP Application, a continuación, le damos a siguiente.



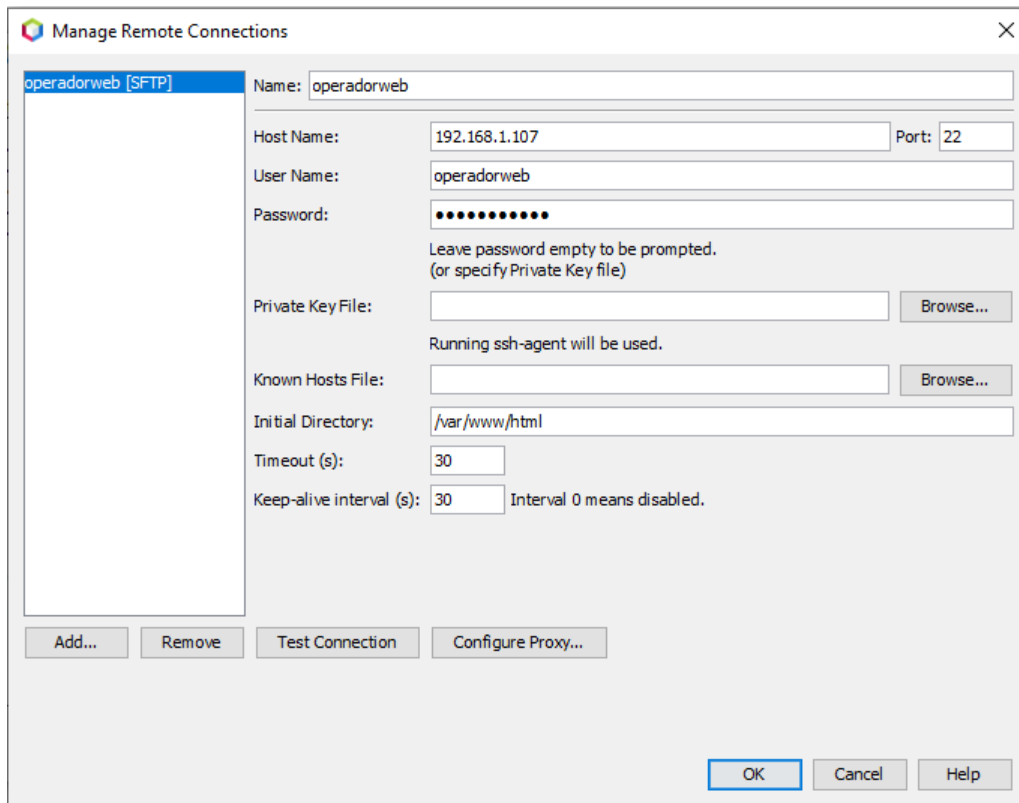
Ahora tenemos que darle un nombre al proyecto, también tendremos que seleccionar una carpeta vacía para meter el proyecto en dicha carpeta. Seleccionaremos la versión de PHP que usaremos en el proyecto y le daremos a siguiente.

The screenshot shows the 'New PHP Project' dialog box in NetBeans, specifically the 'Name and Location' step. On the left, a 'Steps' list shows: 1. Choose Project, 2. **Name and Location**, 3. Run Configuration, 4. PHP Frameworks, 5. Composer. The main area contains the following fields: 'Project Name' with the text 'ProyectoConectado'; 'Sources Folder' with a dropdown showing 'D:\ProyectoPrueba\ProyectoConectado' and a 'Browse...' button; 'PHP Version' with a dropdown showing 'PHP 7.4' and a note 'PHP version is used only for hints'; 'Default Encoding' with a dropdown showing 'UTF-8'; and an unchecked checkbox 'Put NetBeans metadata into a separate directory' with a 'Metadata Folder' field showing 'C:\Users\alber\Documents\NetBeansProjects\ProyectoConectado' and a 'Browse...' button. At the bottom are buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

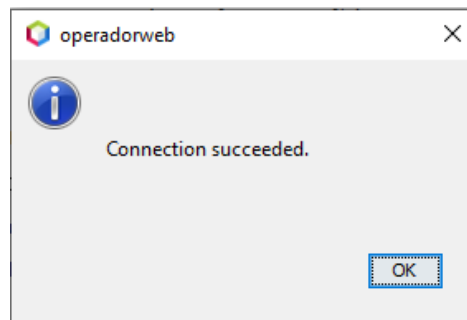
Lo siguiente es configurar como va a correr el proyecto, en este caso lo tendremos que poner en un sitio web remoto con FTP, SFTP. La URL será nuestra dirección IP de nuestro servidor de Ubuntu. Configuraremos que los archivos se suban al servidor al guardar. También tendremos que configurar la conexión remota, para ello le daremos a Manage.

The screenshot shows the 'New PHP Project' dialog box in NetBeans, specifically the 'Run Configuration' step. On the left, the 'Steps' list shows: 1. Choose Project, 2. Name and Location, 3. **Run Configuration**, 4. PHP Frameworks, 5. Composer. The main area contains the following fields: 'Run As' with a dropdown showing 'Remote Web Site (FTP, SFTP)'; 'Project URL' with the text 'http://192.168.1.107'; 'Remote Connection' with a dropdown showing 'operadorweb' and a 'Manage...' button; 'Upload Directory' with a text field containing '/' and a sub-field showing 'sftp://192.168.1.107/var/www/html'; and 'Upload Files' with a dropdown showing 'On Save' and a note 'Files will follow all local changes (create, edit, rename, delete)'. At the bottom are buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

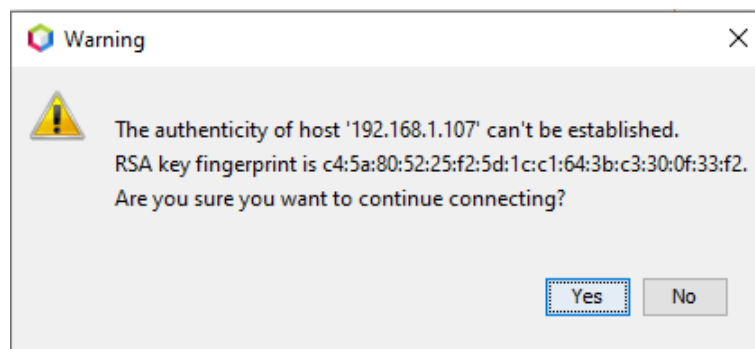
Una vez dentro de Manage, aquí pondremos la ip del host, el puerto 22, el usuario de conexión y su Password. Una vez introducidos los datos haremos un Test Connection dándole a dicho botón.



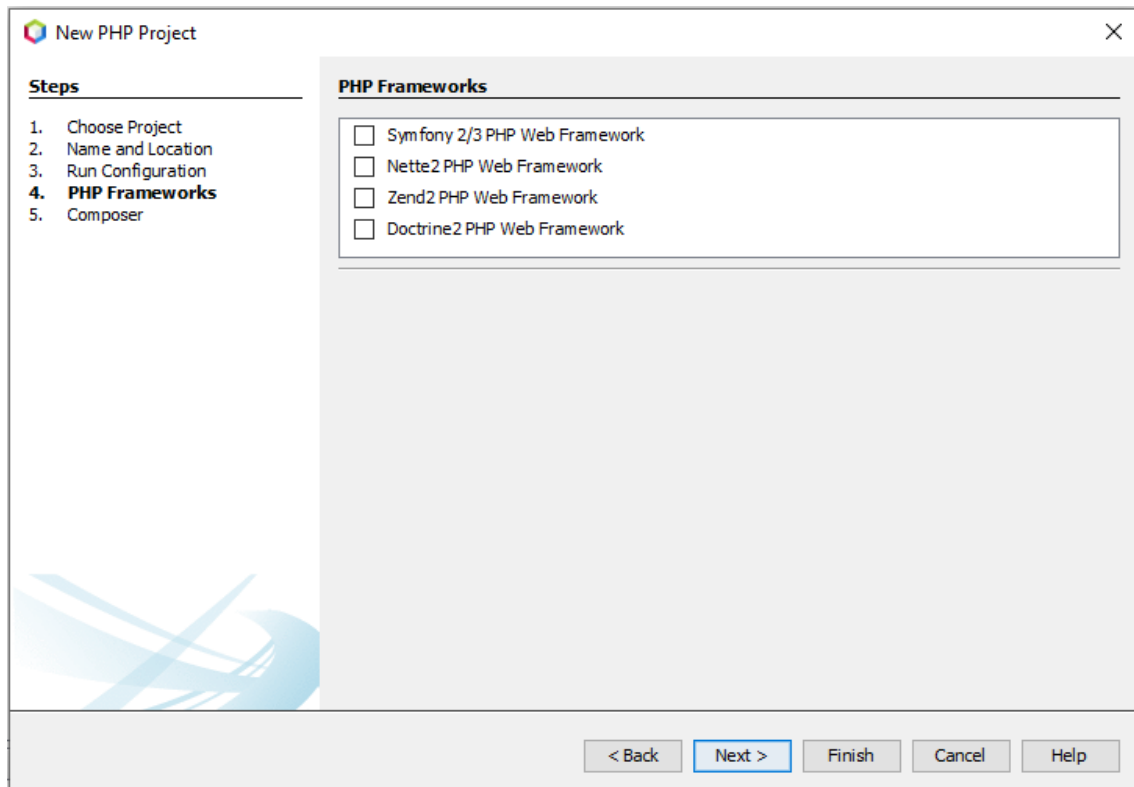
Una vez le damos al botón de Test Connection, probara que hay conexión con el Servidor de Ubuntu, si es correcta nos mostrara dicho mensaje.



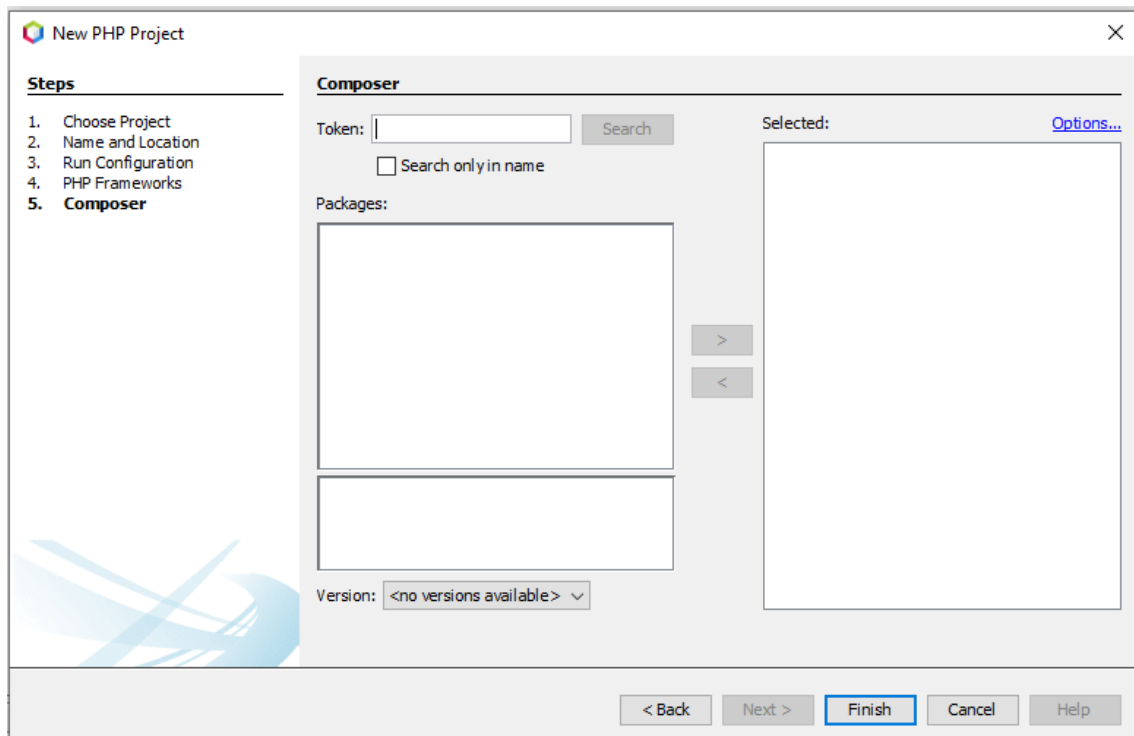
Una vez hecho este proceso le daremos a siguiente, y se nos mostrara dicho mensaje, el cual nos indica si estamos seguros de hacer la conexión, le daremos a que sí.



Ahora nos preguntara si vamos a usar algún framework de PHP, como no vamos a usar ninguno por ahora le daremos a siguiente.

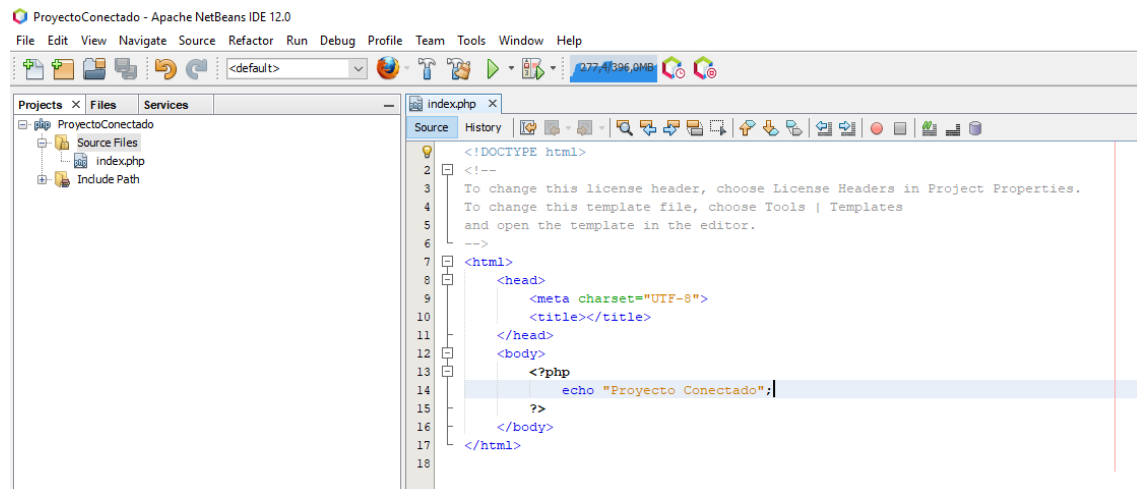


Por último, nos pide configurar un Composer, como no lo necesitamos tampoco por ahora le daremos a Finish.

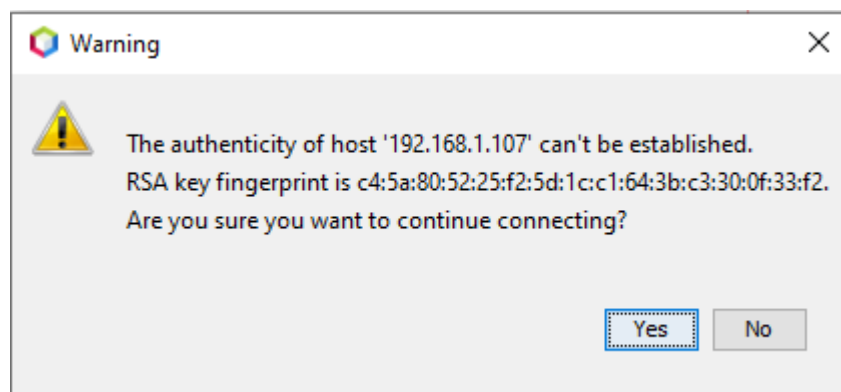


2.8.2 Ejemplo de uso

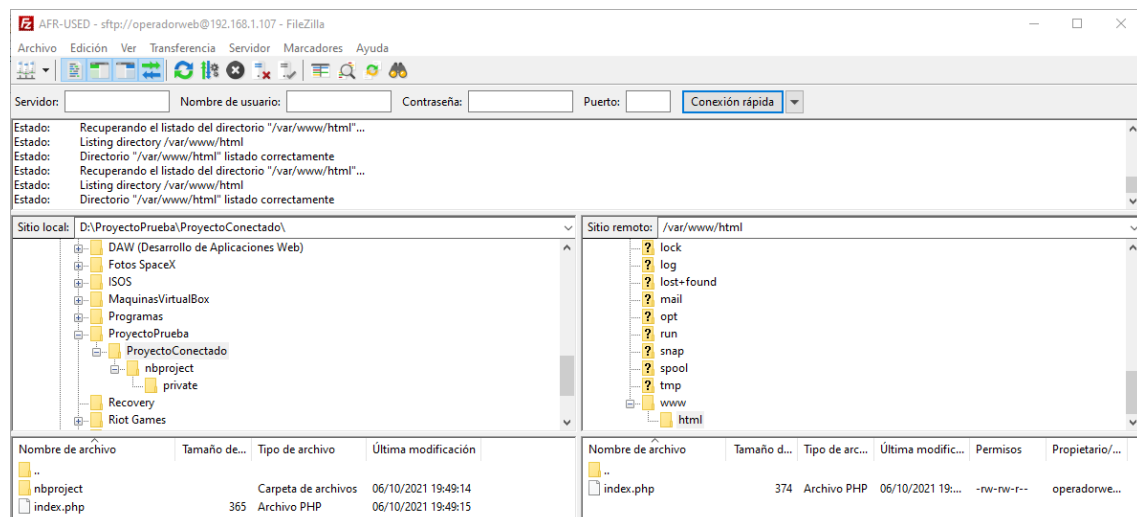
Una vez se acaba el menú de configuración el proyecto estará creado, dicho proyecto está conectado con el servidor de Ubuntu y cualquier cambio que se realice en él se cambiará en el servidor una vez guardemos.



Para que se suba el Index, le pondremos una frase en PHP y le daremos a guardar, se nos abrirá dicho cuadro que pregunta si estas seguro de continuar para hacer los cambios, le daremos que sí.



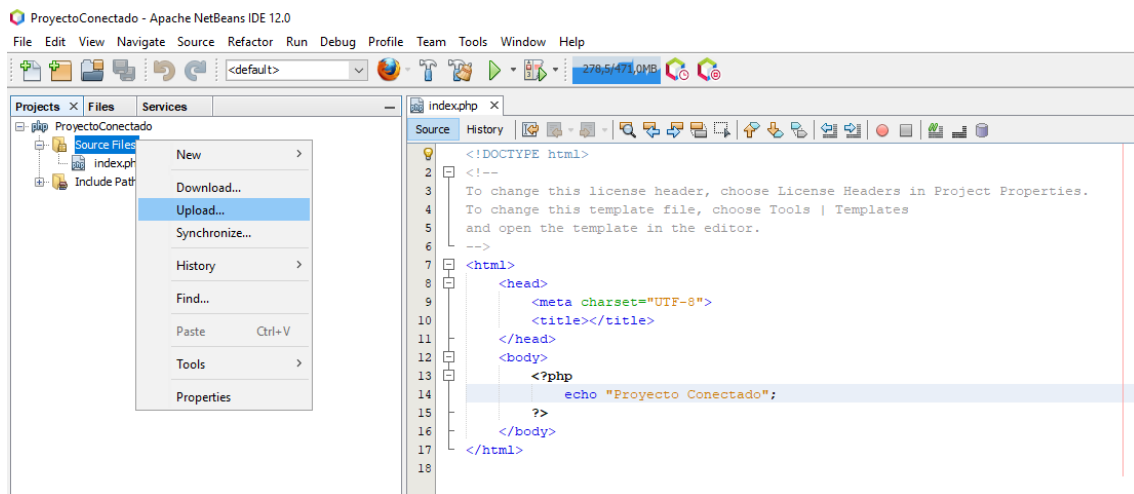
Una vez lo hemos hecho, podremos comprobar en Filezilla que se ha subido el archivo que hemos editado haciendo conexión con operadorweb.



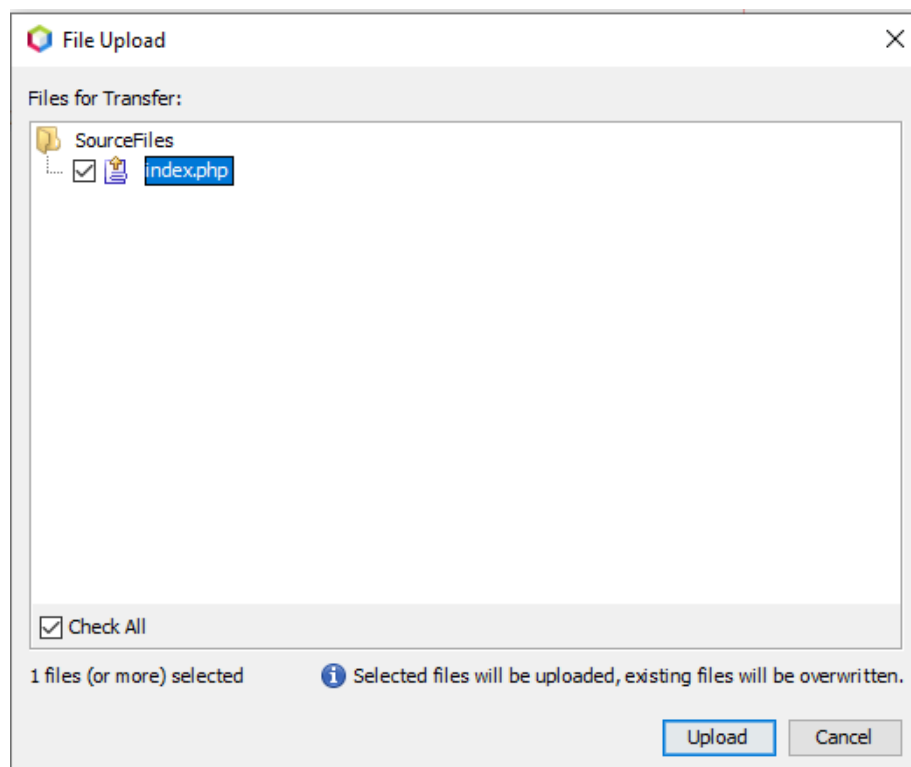
Para comprobar que funciona correctamente, abrimos cualquier navegador e introducimos la IP del servidor y veremos el mensaje escrito anteriormente en NetBeans, de haber hecho bien el proyecto y la conexión nos mostrara el mensaje de PHP.



Para subir otros archivos como imágenes, PDF u otras cosas, deberemos de hacer clic sobre la carpeta de Source Files, y nos iremos a Upload...



Se nos abrirá una ventana que nos dirá que archivos deseamos subir al servidor, seleccionamos los que necesitemos subir y de daremos a Upload.

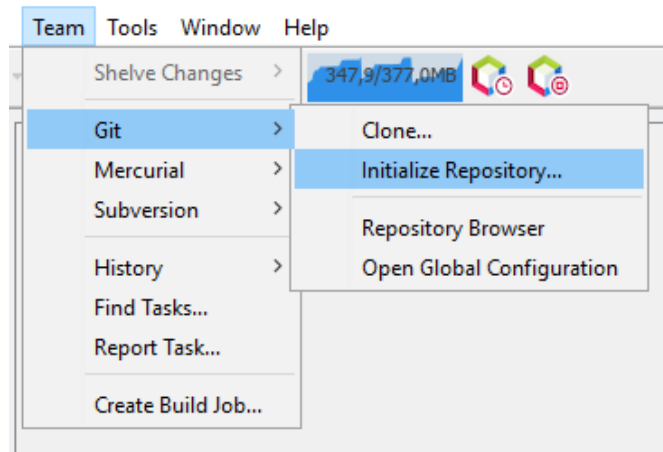


2.9. Administrar un repositorio con GitHub

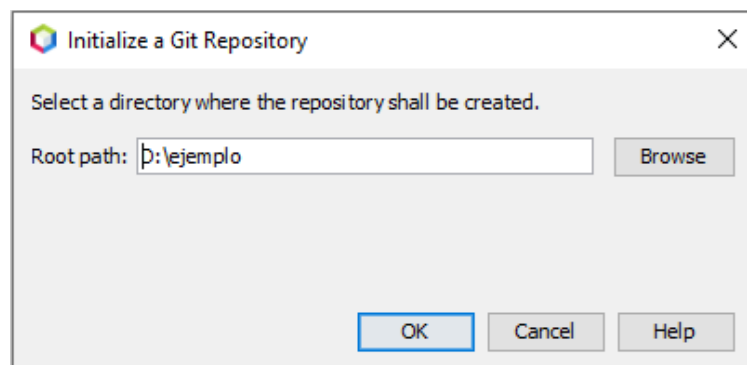
2.9.1. Configuración

2.9.1.1 Subir proyecto

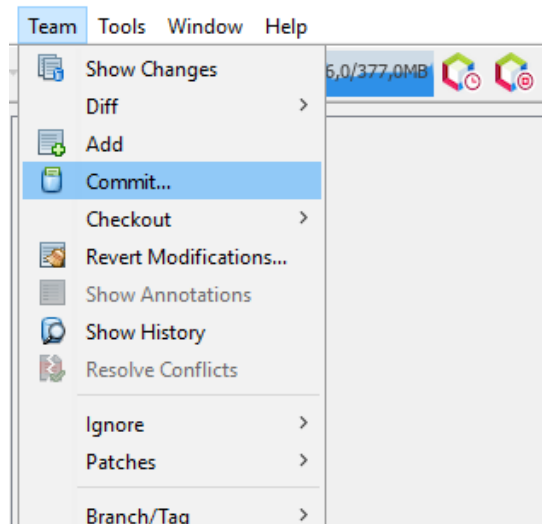
Para subir nuestro proyecto a GitHub, lo primero que tenemos que hacer es lo siguiente, con el proyecto abierto, nos vamos a el apartado Team, Git y Initialize Repository...



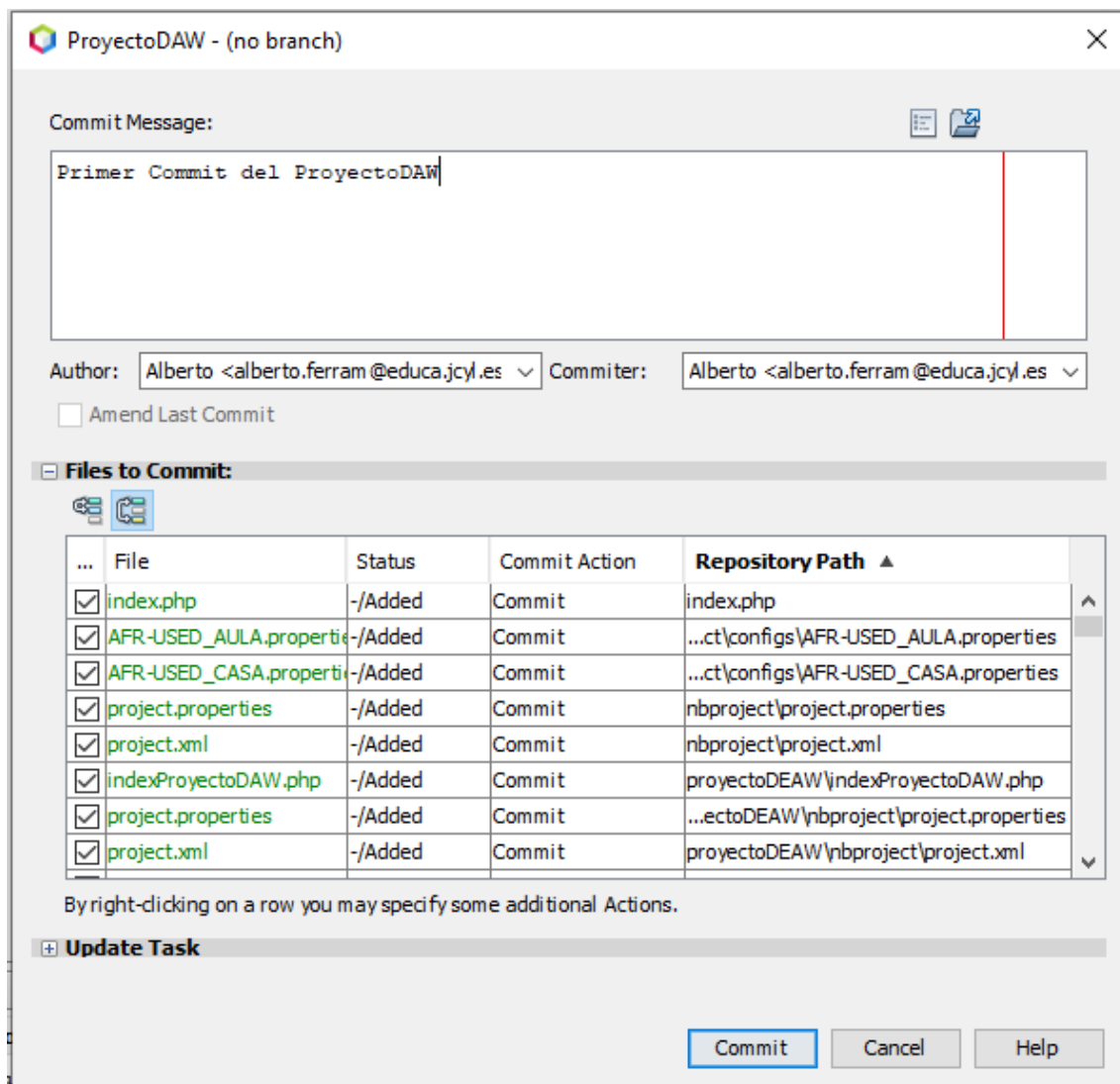
Se nos abrirá esta ventana, la cual nos pedirá que indiquemos el Root path, en mi caso deje la ruta que me dio por defecto ya que era la carpeta donde tengo mi proyecto.



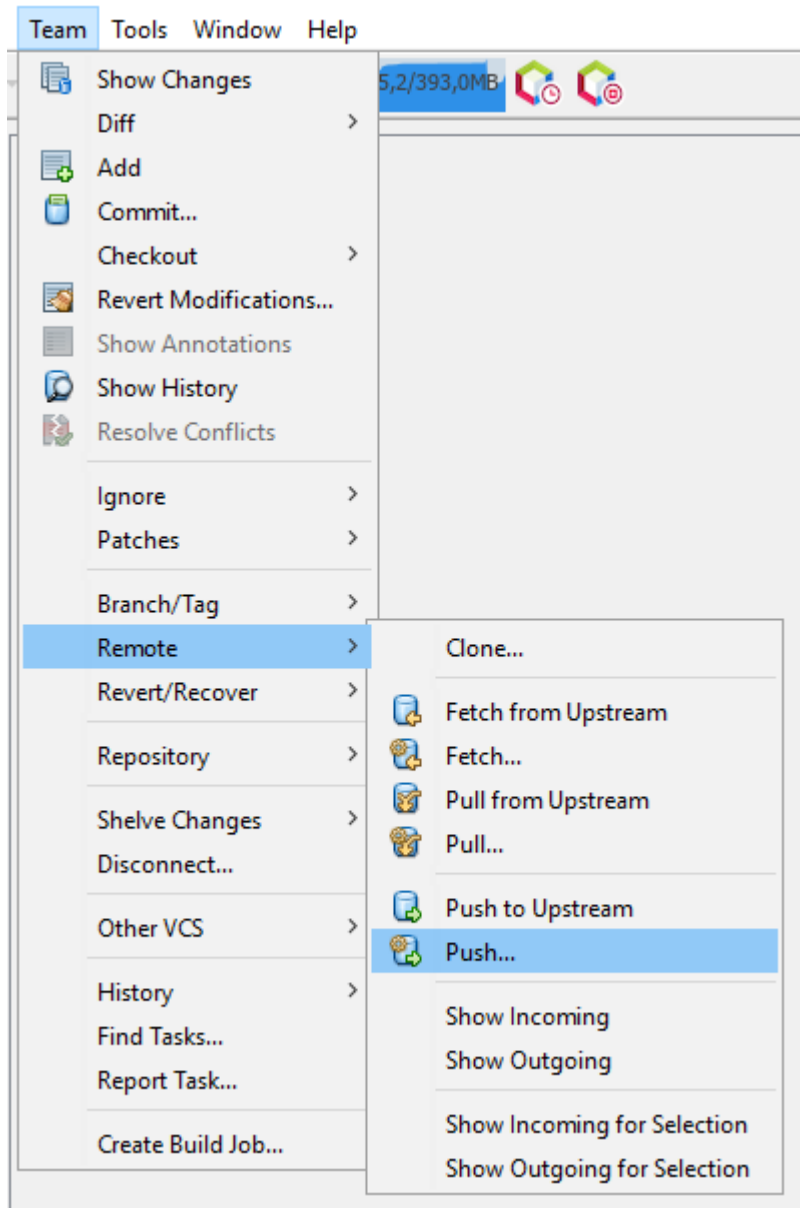
Una vez hecho lo anterior, ahora haremos un Commit de lo que vamos a subir, para ello nos iremos a Team, Commit...



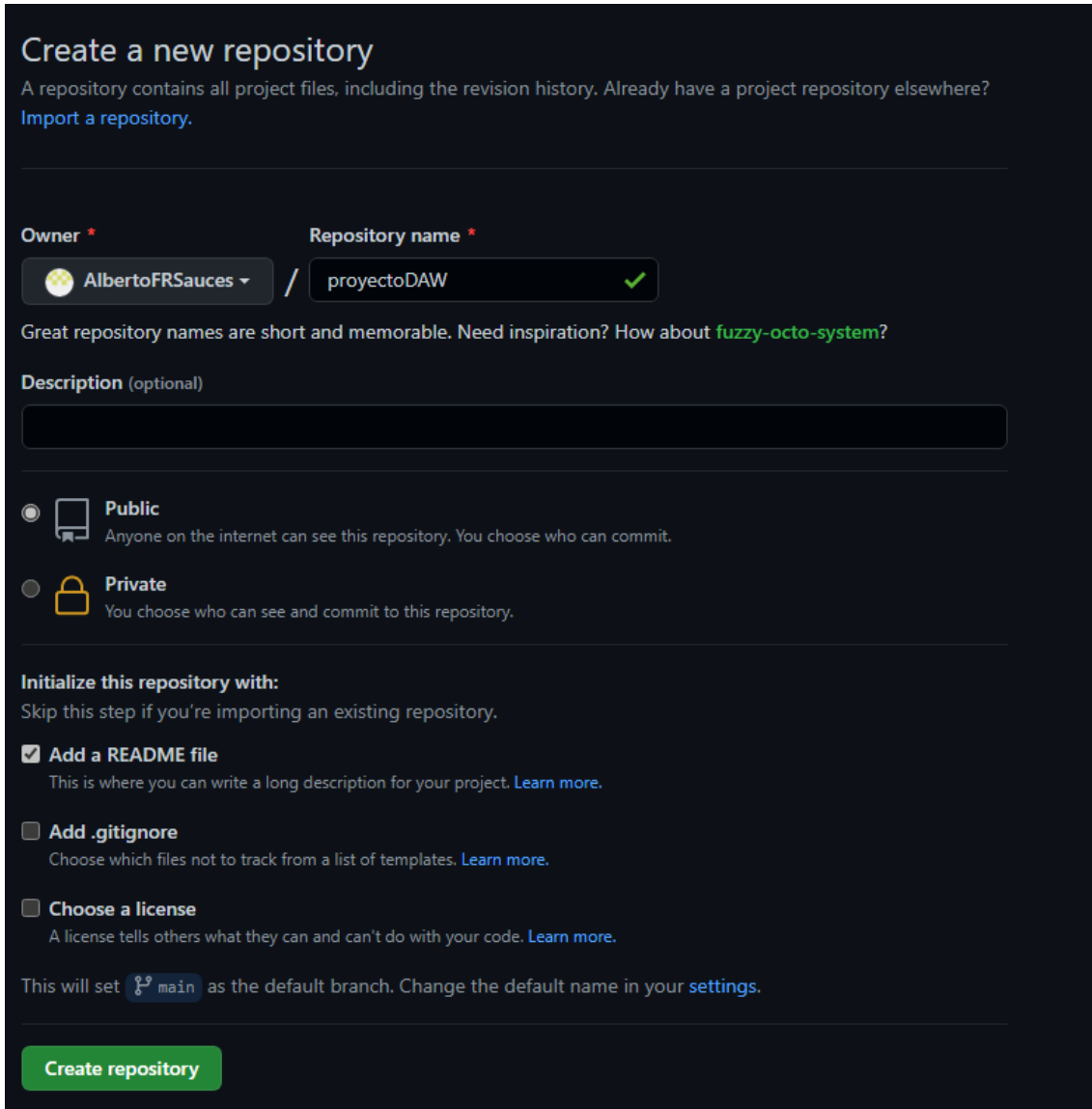
Se nos abrirá la siguiente ventana, en dicha ventana tendremos que indicar el mensaje del Commit, en mi caso he puesto Primer Commit del ProyectoDAW, indicaremos el autor que es nuestro correo, y también indicaremos lo que queremos hacer Commit, en mi caso todo ya que es la primera vez que lo hago.



Una vez hecho el Commit, lo siguiente es hacer el Push, para ello nos iremos a Team, Remote, Push...

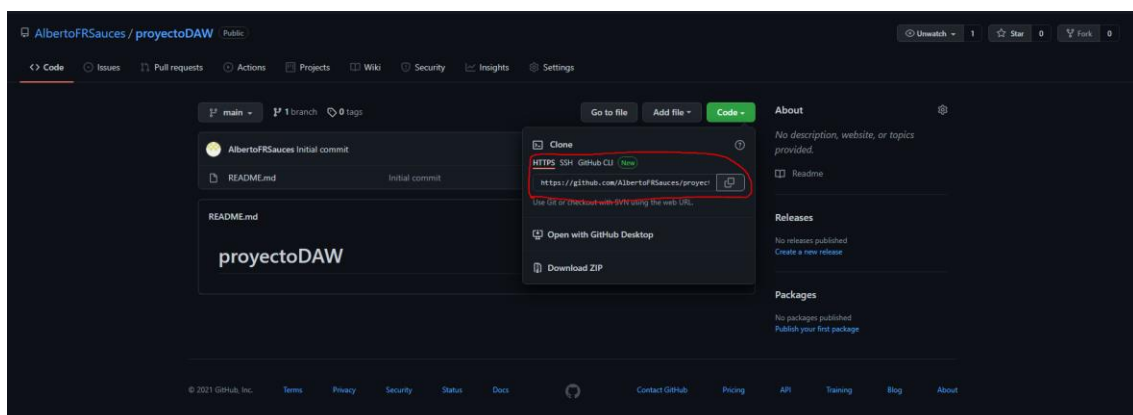


Antes de continuar, necesitamos crear un repositorio en nuestra cuenta de GitHub, yo lo he creado con el nombre proyectoDAW, que sea público y que tenga el archivo README. Con todo eso ya le podemos dar a Create Repository.



The screenshot shows the 'Create a new repository' form on GitHub. The 'Owner' is 'AlbertoFRSauces' and the 'Repository name' is 'proyectoDAW', which has a green checkmark. Below this, there's a suggestion for 'fuzzy-octo-system?'. The 'Description' field is empty. Under 'Visibility', 'Public' is selected with a radio button. The 'Initialize this repository with:' section has three checkboxes: 'Add a README file' (checked), 'Add .gitignore' (unchecked), and 'Choose a license' (unchecked). At the bottom, there's a green 'Create repository' button.

Una vez se nos crea, copiamos la URL de HTTPS ya que la vamos a necesitar en el siguiente paso.



Volvemos a el Push, cuando le damos se nos abrirá esta ventana, aquí debemos indicar el URL del repositorio, el cual es el que hemos copiado antes de GitHub y también tendremos que indicar nuestro usuario y contraseña. Una vez hecho esto le damos a siguiente.

(**IMPORTANTE:** Si al finalizar todos los pasos nos salta continuamente una ventana en NetBeans de conexión que pide usuario y contraseña de GitHub para subir el proyecto al repositorio, se trata de un ERROR.

Este error se debe a que no se puede usar la contraseña de la cuenta de GitHub para conectarnos con el repositorio. Nos obliga a usar un token, el cual hay que generar en la web de GitHub e iniciar sesión poniendo ese Token en el apartado de Password.

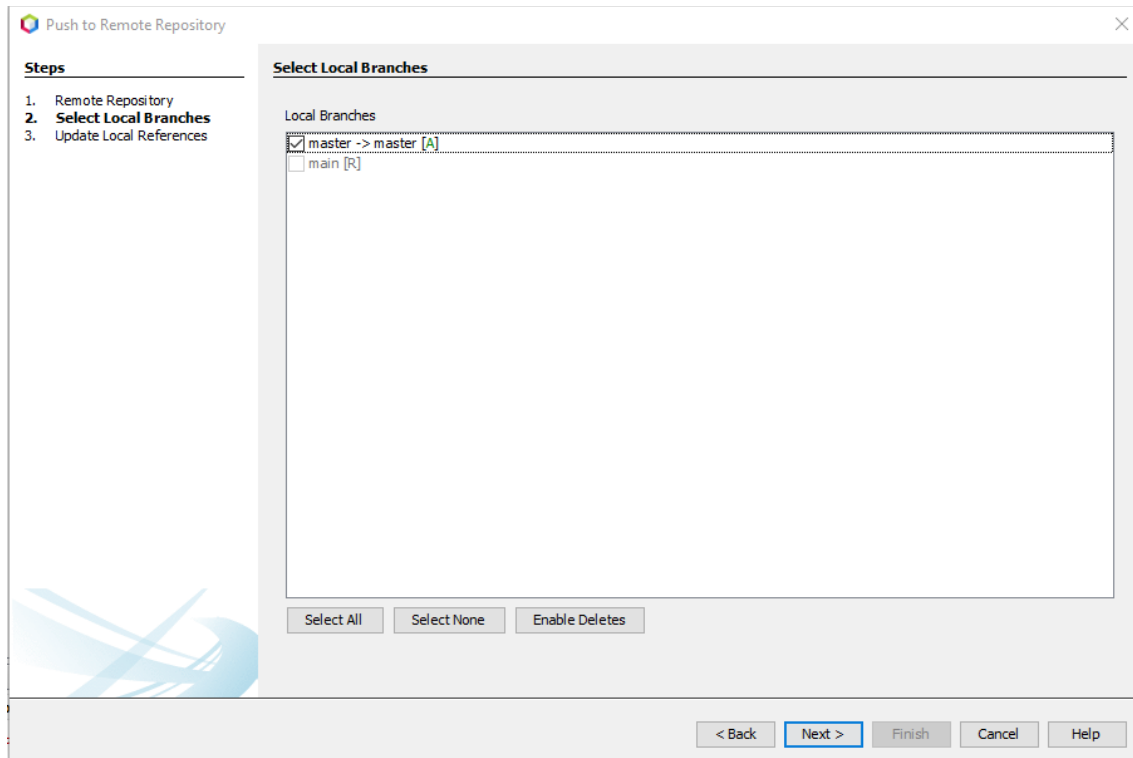
Para crear un Token: <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token>)

The screenshot shows the 'Push to Remote Repository' dialog box in NetBeans. The dialog has a title bar with a close button (X). On the left, there is a 'Steps' panel with three steps: 1. Remote Repository (selected), 2. Select Local Branches, and 3. Update Local References. The main area is titled 'Remote Repository' and contains the following options and fields:

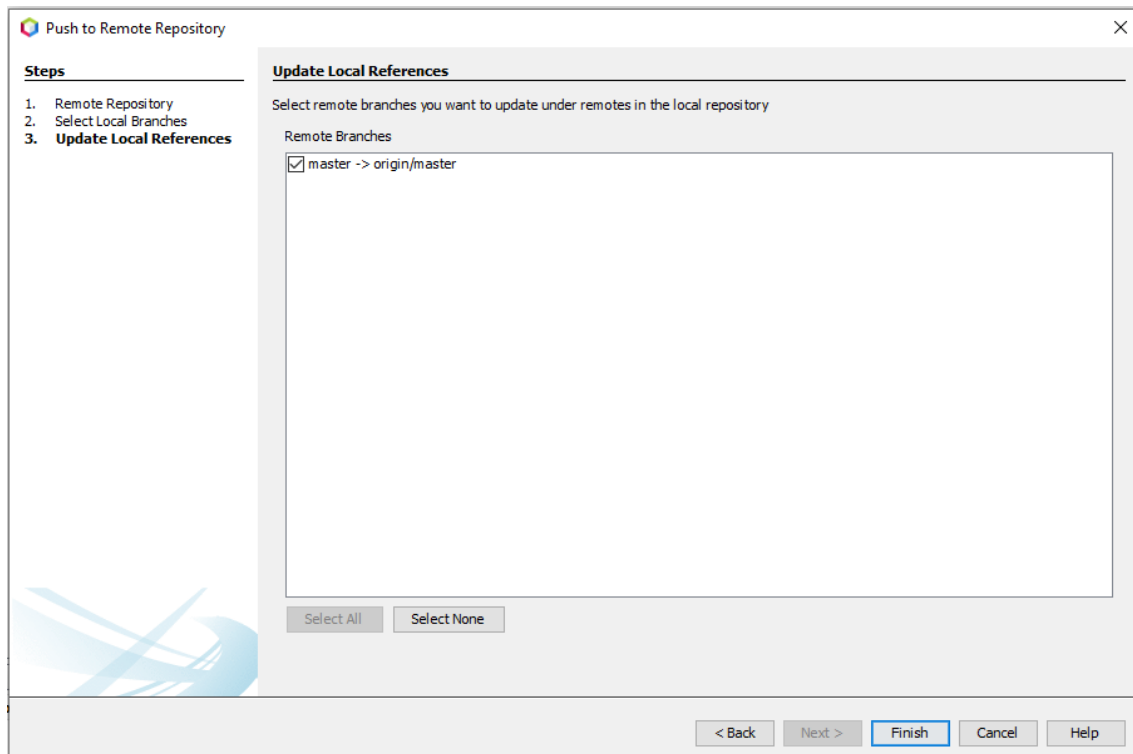
- ☐ Select Configured Git Repository Location: This option is disabled and has a dropdown menu below it.
- ☒ Specify Git Repository Location: This option is selected.
- Remote Name: A dropdown menu showing 'origin'. To its right is a checked checkbox labeled 'Persist Remote'.
- Repository URL: A text field containing 'https://github.com/AlbertoFRSauces/proyectoDAW.git'. Below it is a smaller text field with the placeholder 'http[s]://host.xz[:port]/path/to/repo.git/'.
- User: A text field containing 'ertoFRSauces'. To its right is a note '(leave blank for anonymous access)'.
- Password: A text field with masked characters (dots). To its right is a checked checkbox labeled 'Save Password'.
- Below the password field is a button labeled 'Proxy Configuration...'.

At the bottom of the dialog, there are five buttons: '< Back', 'Next >' (highlighted with a blue border), 'Finish', 'Cancel', and 'Help'.

Nos pedirá seleccionar la rama que queremos subir desde local. En este caso la master.



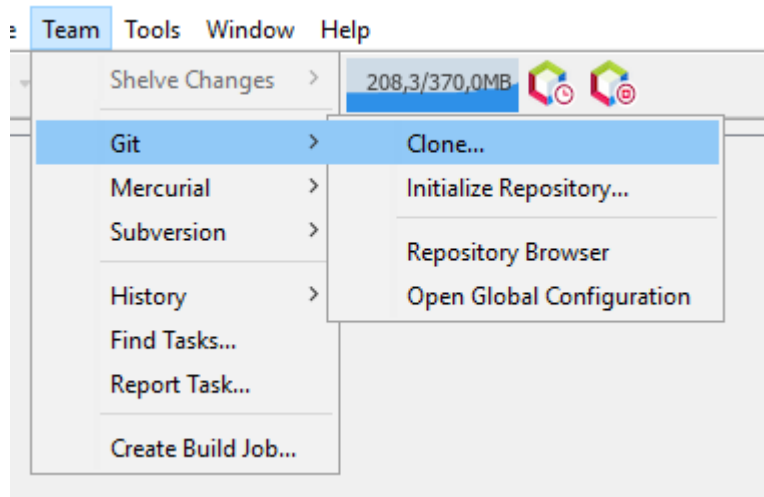
Y también nos pedirá la rama remota para actualizar el repositorio. En este caso la master. Una vez hecho esto le daremos a Finish y ya tendríamos el proyecto en nuestro repositorio de GitHub.



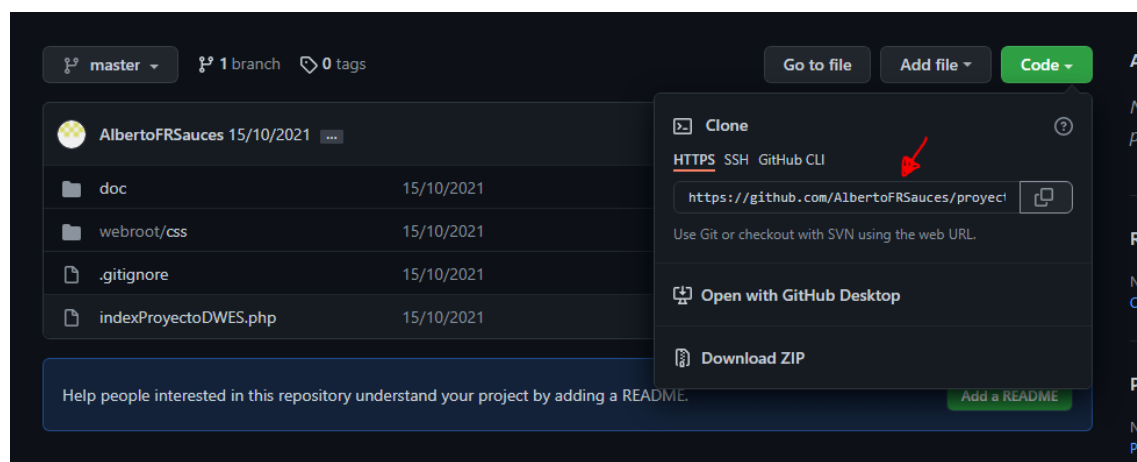
2.9.1.2 Bajar proyecto

Una vez subido el proyecto, tendremos que bajarlo en otro sitio si queremos hacer cambios sobre él, para ello haremos lo siguiente (**IMPORTANTE: para hacer lo siguiente necesitamos una carpeta vacía donde dejar nuestro proyecto y también necesitaremos un repositorio con contenido en GitHub**).

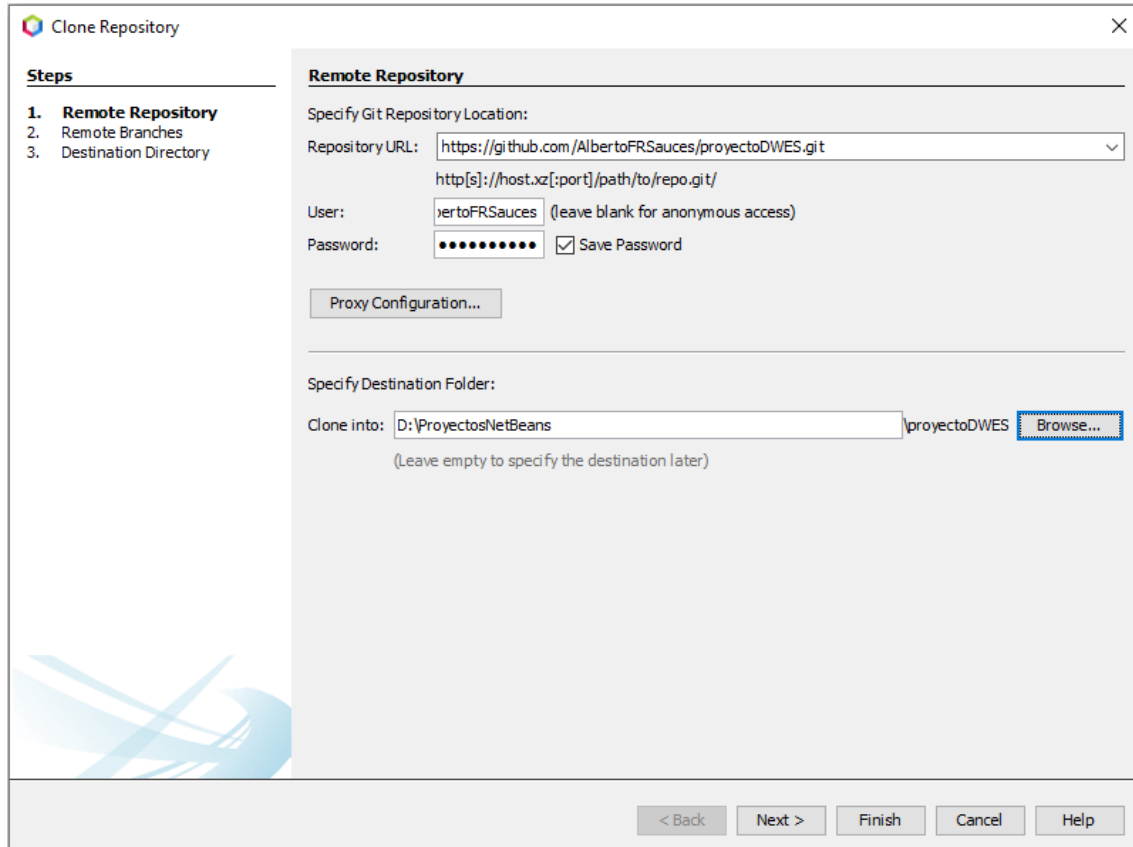
Lo primero será irnos en NetBeans a Team, después a Git y le daremos a Clone...



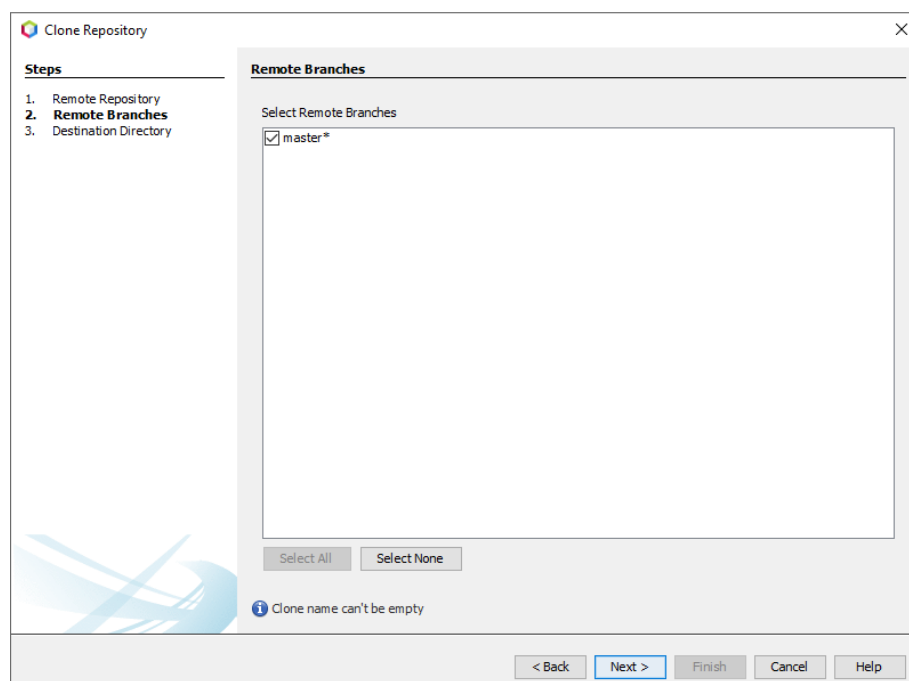
Antes de seguir, nos iremos a nuestro repositorio de GitHub, le daremos a Code y en el desplegable copiaremos la URL del apartado de HTTPS por que la vamos a necesitar a continuación.



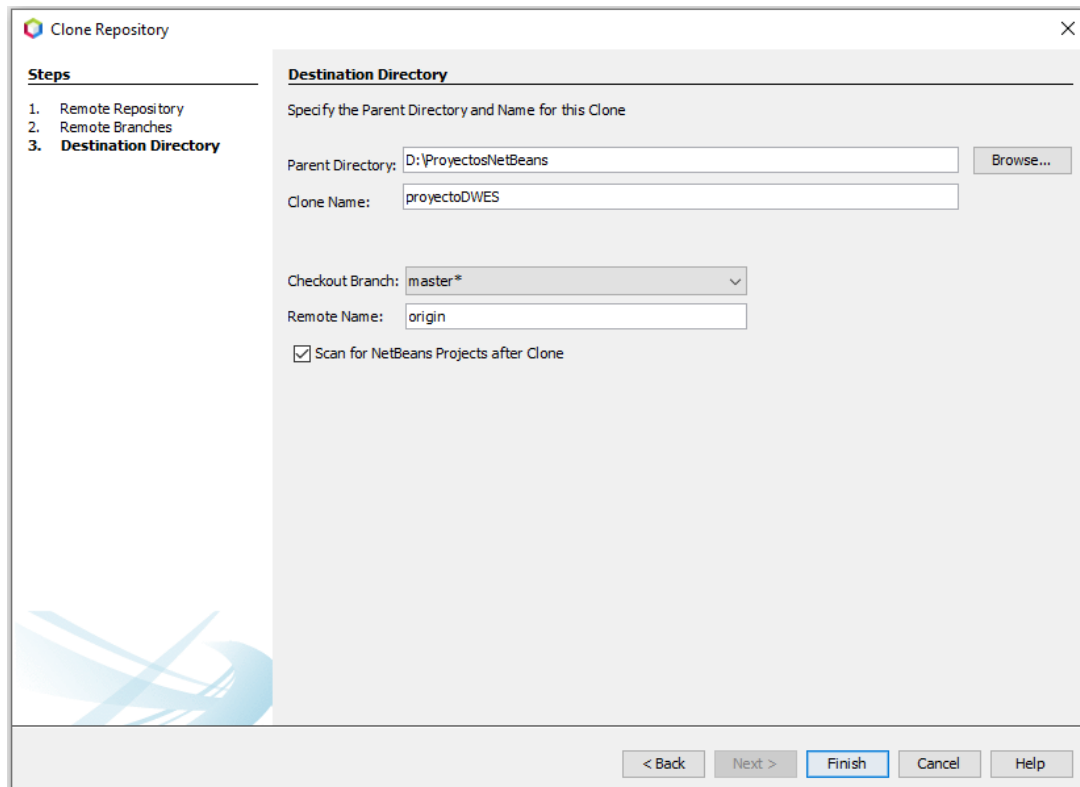
Una vez le hemos dado a la opción Clone, se nos abrirá esta ventana, aquí configuraremos el URL del repositorio que necesitamos (la URL la hemos copiado en el paso anterior). Introduciremos nuestros datos de nuestra cuenta de GitHub, y por último seleccionaremos donde se va a clonar el proyecto (en mi caso he creado una carpeta para ello). Una vez completo todo le damos a siguiente.



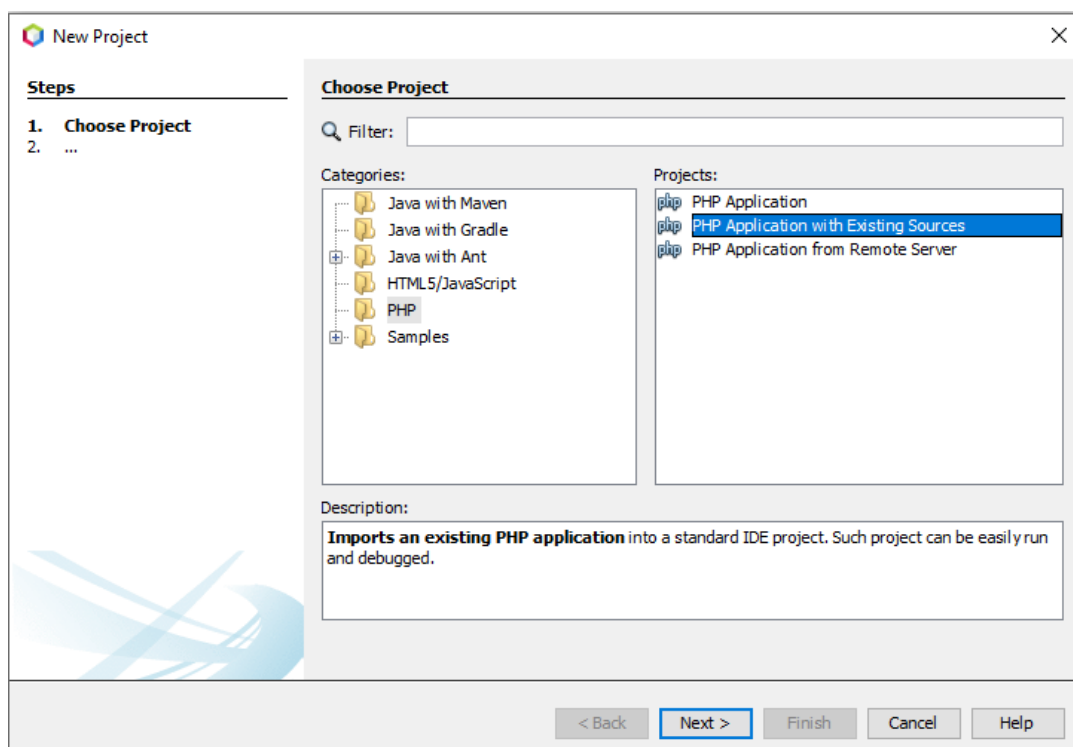
Lo siguiente será seleccionar la rama que vamos a bajar desde GitHub, en mi caso solo tengo la master, en caso de tener varias, seleccionamos la rama que queramos.



Por último, nos preguntara donde guardar los archivos y que nombre le vamos a dar a la carpeta, en mi caso elijo la carpeta creada anteriormente y le pongo el nombre del proyecto a el nombre de la clonación.



Cuando acabamos de configurar todos los pasos de clonar, nos preguntará en una ventana si queremos crear un proyecto, le daremos a que sí, y se nos abrirá ya directamente esta ventana. En esta ventana seleccionamos PHP y Aplicación PHP con archivos existentes, ya que los hemos bajado de GitHub anteriormente. Y siguiente.



Ahora nos preguntara donde están los archivos, seleccionamos la carpeta que nos ha bajado antes donde le indicamos, nos pedirá también un nombre para el proyecto. Una vez completado le daremos a siguiente.

New PHP Project with Existing Sources

Steps

1. Choose Project
- 2. Name and Location**
3. Run Configuration

Name and Location

Sources Folder: D:\ProyectosNetBeans\proyectoDWES Browse...

Folder with existing PHP scripts.

Project Name: proyectoDWES

PHP Version: PHP 7.3 ▼

PHP version is used only for hints

Default Encoding: UTF-8 ▼

☐ Put NetBeans metadata into a separate directory

Metadata Folder: D:\ProyectosNetBeans\proyectoDWES\proyectoDWES Browse...

< Back Next > Finish Cancel Help

Por último, nos pedirá configurar la manera de correr el proyecto, en mi caso seleccionare SFTP ya que tendré conectado el proyecto con mi servidor web.

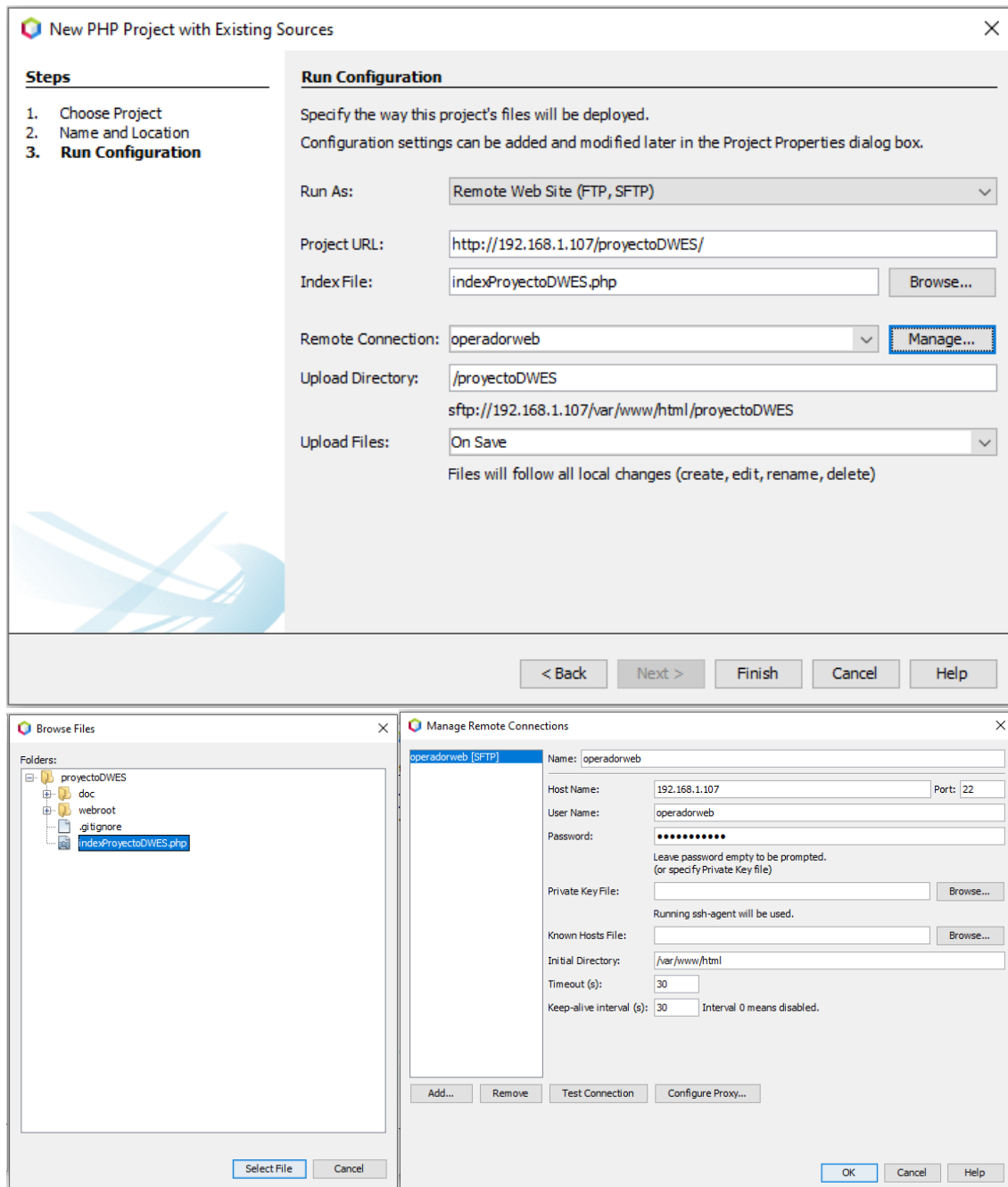
La URL del proyecto será nuestra IP mas el nombre de la carpeta del proyecto.

El index, le daremos a browse y seleccionaremos el archivo que es el index, en mi caso se llama `indexProyectoDWES.php`

La conexión remota la configuraremos como tengamos los datos de nuestro servidor.

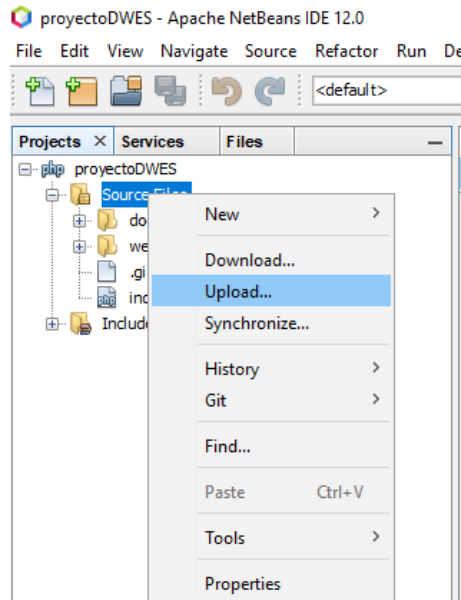
Configuramos el directorio de nuestro servidor donde se van a subir los archivos, en este caso la carpeta proyectoDWES.

Y configuramos, en mi caso, que se suba a el servidor cuando se guarde algo.

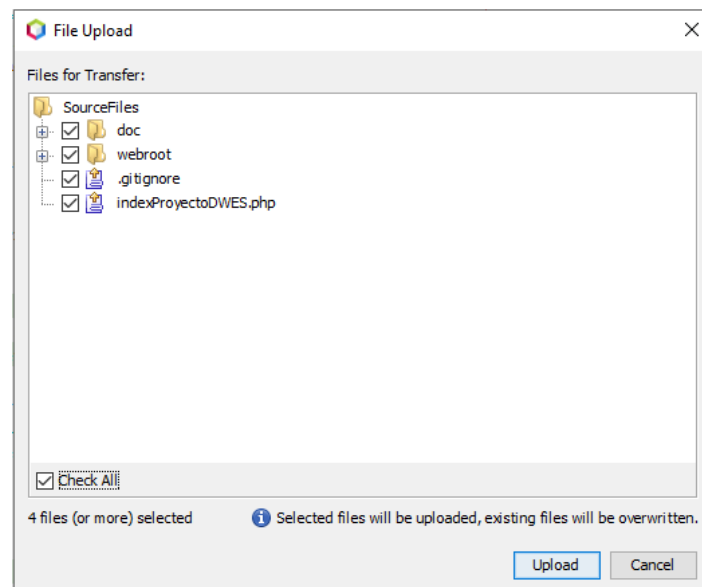


Una vez finalizado el proceso anterior, ya tenemos nuestro proyecto listo para trabajar en él, poder subirlo a nuestro servidor Ubuntu para ver los resultados y hacer Commit para seguir trabajando en el y subirlo a GitHub también.

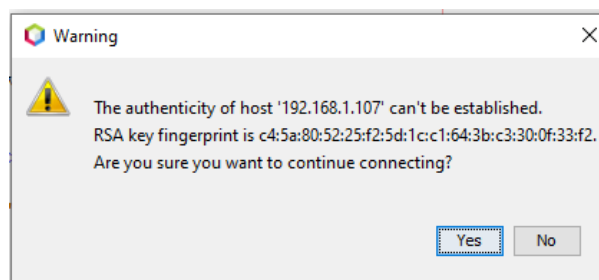
Ahora haremos un Upload de los archivos a nuestro servidor de Ubuntu ya conectado.



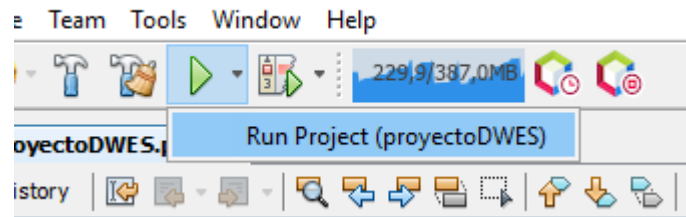
Nos pedirá que archivos subir, en este caso todos porque está vacío el servidor.



Nos pedirá que, si estamos seguros de seguir con la conexión con el servidor, le diremos que si para que se realice la subida de los archivos.



Una vez subidos los archivos a nuestro servidor, ya podemos darle a el botón Play desde NetBeans para abrir nuestro proyecto en el navegador.



Si todo se ha hecho correctamente, veremos nuestro proyecto corriendo correctamente en el navegador.

