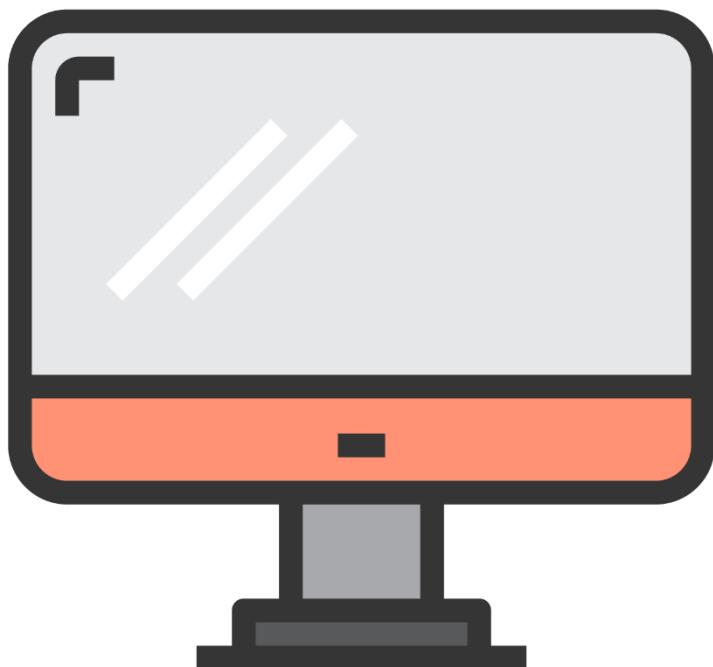


Documentación de instalación y configuración del entorno de desarrollo y explotación



INDICE

1. Ubuntu Server 20.04	5
1.1 Datos de configuración maquina Ubuntu Server 20.04	5
1.2 Cambiar nombre del host	6
1.3 Configurar dirección IP	7
1.4 Instalar servicio SSH	8
1.4.1 Instalación	8
1.4.2 Monitorización	8
1.5 Instalar Apache	9
1.5.1 Instalación	9
1.5.2 Configuración	10
1.5.3 Monitorización	21
1.6 Crear usuario operadorweb y asignar permisos.....	22
1.6.1 Configuración	22
1.7 Instalación y configuración del módulo de PHP 7.4.....	23
1.7.1 Instalación	23
1.7.2 Configuración	23
1.8 Instalación y configuración de MySQL	27
1.8.1 Instalación	27
1.8.2 Configuración	28
1.8.3 Ejemplo de uso	32
1.9 Instalación y Configuración de XDebug	34
1.9.1 Instalación	34
1.9.2 Configuración	35
1.10 Conexión mediante clave privada con SSH.....	40
1.10.1 Configuración	40
1.10.2 Generar clave	41
1.11 Enjaular usuarios con SSH	42
1.11.1 Configuración	42
1.12 Instalación y configuración de DNS.....	45
1.12.1 Instalación	45
1.12.2 Configuración	45
1.12.3 Configuración Arpa	48
1.13 Configuración de dos sitios Web en el mismo Servidor	51

1.13.1 Configuración	51
1.14 Configuración de un sitio web con HTTPS	56
1.14.1 Configuración	56
1.14.2 Redireccionar http a https	60
1.14.3 Configurar la negociación de contenidos mediante Multiview	61
1.15 Instalación de phpDocumentor	62
1.15.1 Instalación y configuración con phar	62
1.15.2 Ejemplo de uso phar	63
1.16 Autenticación y autorización en aplicación	64
1.16.1 Autenticación básica mediante usuarios	64
1.16.2 Autenticación Digest mediante usuarios	67
1.16.3 Autenticación Digest mediante grupos	69
1.17 Uso de comandos en Git	71
1.17.1 Copia de archivos en una rama de un repositorio de GitHub	71
1.18 Instalación y configuración del servicio LDAP	73
1.18.1 Configuración de las máquinas virtuales	73
1.18.2 Instalación y configuración	75
1.18.3 Prueba de uso de conexión desde NetBeans	89
1.18.4 Configuración modulo autenticación	94
1.18.5 Prueba de uso de inicio de sesión desde el navegador	98
1.18.6 Administración con el cliente “Apache Directory Studio”	99
2. Windows 10	113
2.1 Datos de configuración maquina Windows 10	113
2.2 Cambiar nombre del host	114
2.3 Configurar dirección IP	118
2.4 Instalación de navegador Mozilla	121
2.4.1 Instalación	121
2.5 Instalación de PuTTY	124
2.5.1 Instalación	124
2.5.2 Configuración	128
2.5.3 Ejemplo de uso	129
2.6 Instalación de FileZilla	131
2.6.1 Instalación	131
2.6.2 Configuración	135
2.6.3 Ejemplo de uso	137
2.7 Instalación de NetBeans	139

2.7.1 Instalación	139
2.8. Conectar Proyecto de NetBeans con Ubuntu Server	145
2.8.1 Configuración	145
2.8.2 Ejemplo de uso	149
2.9 Administrar un repositorio con GitHub	151
2.9.1 Configuración	151
2.10 XDebug.....	164
2.10.1 Ejemplo de uso	164
2.11 Uso de Git con Visual Studio Code	166
2.11.1 Configuración	166
2.11.2 Ejemplo de uso	167

1. Ubuntu Server 20.04

1.1 Datos de configuración maquina Ubuntu Server 20.04

Nombre de la maquina	AFR-US
Nombre del host	AFR-US
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: admindwes 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:

```
admindwes@dwes:~$ sudo hostnamectl set-hostname AFR-USED
admindwes@dwes:~$ 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.

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

También lo podemos comprobar con el comando *hostname*.

```
admindwes@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*

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

```
GNU nano 4.8
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*

```
admindwes@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: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: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.

```
admindwes@AFR-US ED:~$ ps -ef |grep ssh
admindw+ 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.

```
admindwes@AFR-US ED:~$ sudo apt update
```

```
admindwes@AFR-US ED:~$ sudo apt upgrade
```

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

```
admindwes@AFR-US ED:~$ sudo apt install openssh-server
```

1.4.2 Monitorización

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

```
admindwes@AFR-US ED:~$ 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
admindw+ 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:

```
admindwes@AFR-USDED:~$ ps -ef |grep apache2
admindw+ 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.

```
admindwes@AFR-USDED:~$ sudo apt update|
admindwes@AFR-USDED:~$ sudo apt upgrade
```

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

```
admindwes@AFR-USDED:~$ sudo apt install apache2|
```

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

```
admindwes@AFR-USDED:~$ 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
admindw+ 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.

1.5.2 Configuración

Rutas personalizadas de archivos de errores

Lo primero que haremos será entrar a sites-enabled y veremos el contenido del archivo 000-default.conf, lo haremos con un *cat* y el nombre del archivo.

Podremos ver donde tiene configuradas las rutas de los dos archivos de errores.

```
admindwes@AFR-USED:/etc/apache2$ cd sites-enabled/  
admindwes@AFR-USED:/etc/apache2/sites-enabled$ cat 000-default.conf  
<VirtualHost *:80>  
    # The ServerName directive sets the request scheme, hostname and port th  
at  
    # 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
```

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

```
admindwes@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'
```

Una vez visto el archivo de errores, 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**

Tendremos que editar el archivo de configuración llamado 000-default.conf. Antes de editarlo haremos una copia de dicho archivo por seguridad con el siguiente comando:

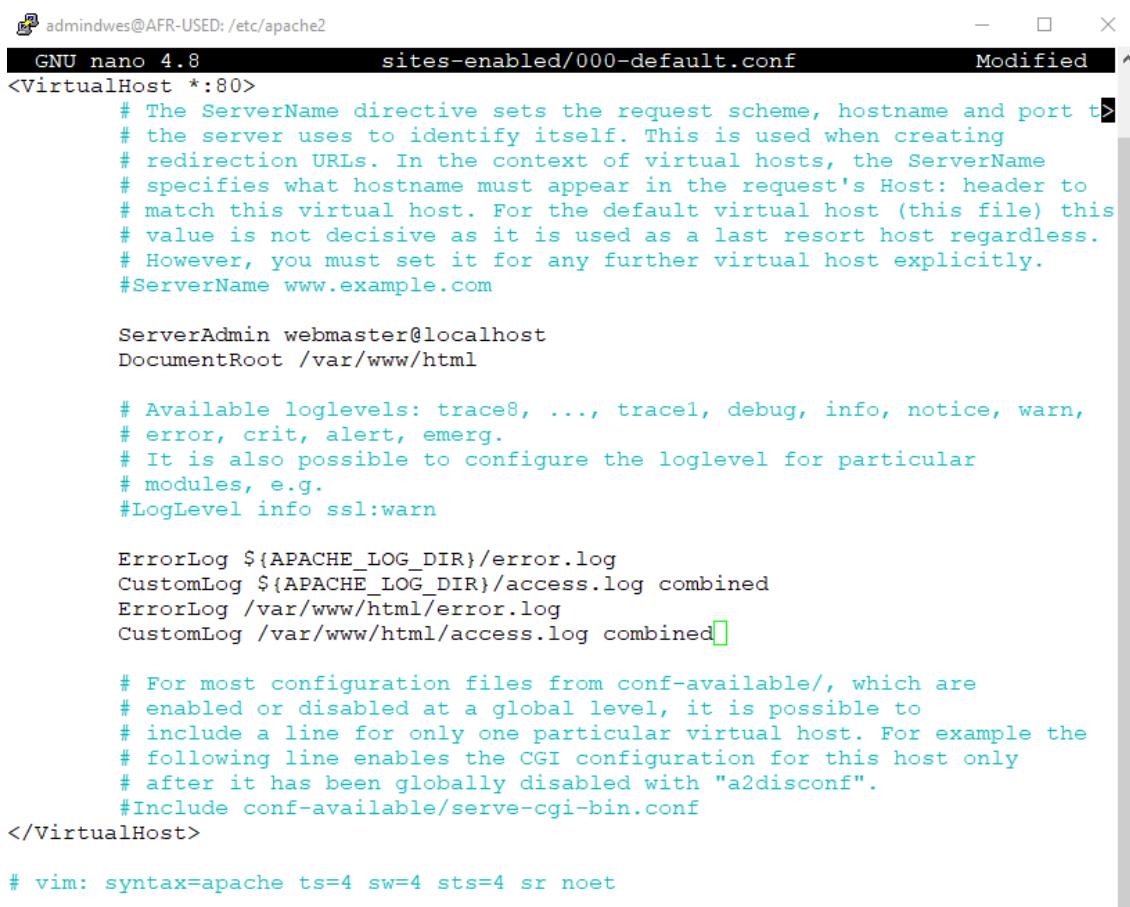
```
|adminwes@AFR-US...:/etc/apache2$ sudo cp sites-enabled/000-default.conf sites-enabled/000-default.conf.backup
```

Una vez echo la copia de seguridad del archivo, ya entraremos a editarla 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



```
GNU nano 4.8          sites-enabled/000-default.conf      Modified ^

<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port t>
    # 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 en dicho directorio.

```
adminwes@AFR-US...:/etc/apache2$ sudo service apache2 restart
adminwes@AFR-US...:/etc/apache2$ ls /var/www/html/
access.log  error.log  index.html  proyectoDEAW  proyectoDWEC  proyectoTema3
doc         img        index.php  _ proyectoDIW  proyectoDWES  webroot
```

Configurar el módulo SSL

En la carpeta apache2 haremos `ls` para ver lo que tiene dentro y entrar a ver el archivo ports el cual contiene los puertos que usa nuestro sitio web.

```
adminwes@AFR-US:~$ cd /etc/apache2
adminwes@AFR-US:/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.



```
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.

```
adminwes@AFR-US:/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.

```
adminwes@AFR-US:/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:

```
adminwes@AFR-US:/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 self-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-USDED:/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-USDED:/etc/apache2$ ls sites-enabled/
000-default.conf
```

También haremos un **ls** en **sites-available** para ver los archivos que contiene y comprobar que tenemos el archivo del módulo SSL llamado default-ssl.conf

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

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.

```
admindwes@AFR-USDED:/etc/apache2$ sudo cat sites-available/default-ssl.conf
```

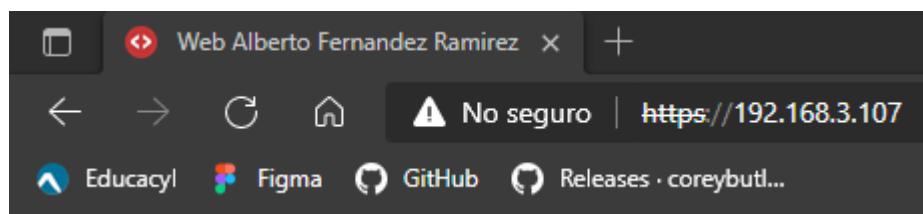
Lo primero que haremos es activar el módulo con el siguiente comando

```
admindwes@AFR-USDED:/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-USDED:/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.



Desactivar el módulo SSL

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:

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

Quitar el Warning ServerName

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.

```
adminwes@AFR-US: /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.

```
|adminwes@AFR-US: /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.

```
GNU nano 4.8                               apache2.conf                                Modified
# Use mod_remoteip instead.
#
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" >
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combin>
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
```

Ahora vamos a Activar / Desactivar Indexes (indexación)

La generación automática de los índices se controla con la opción indexes. Es una medida de seguridad, vamos a deshabilitar la indexación para todos los directorios a partir de /var/www, evitando así que ningún usuario pueda ver qué archivos hay en un directorio en el caso de que falte los archivos indicados en la directiva DirectoryIndex.

Nos vamos al directorio /etc/apache2 y editamos el archivo apache2.conf con el siguiente comando:

```
|adminwes@AFR-US...:/etc/apache2$ sudo nano apache2.conf|
```

Se busca (Ctrl+W) /var/www/ en el apache2.conf y, dentro de ese directorio se inserta delante de Indexes un '-' (quitas los permisos en los indexes) y delante de FollowSymLinks un '+' (otorgas permisos para el resto).

```
GNU nano 4.8                                     apache2.conf
<Directory /var/www/>
    Options -Indexes +FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

Una vez este editado el archivo ejecutamos el siguiente comando:

```
adminwes@AFR-US...:/etc/apache2$ sudo apachectl configtest
```

Por último, reiniciamos el servicio con el siguiente comando:

```
adminwes@AFR-US...:/etc/apache2$ sudo service apache2 restart|
```

Ahora configuraremos el Fichero .htaccess

El archivo htaccess (.htaccess) es un archivo de texto para configurar la forma en que tu servidor web responde a las peticiones de los usuarios. htaccess se convirtió en un estándar y es compatible con los servidores web más utilizados (Apache, Nginx, IIS).

Opera al nivel carpeta permitiendo anular o modificar ajustes de configuración global de tu servidor web, o asignar funciones o recursos a otras carpetas ubicadas en tu árbol de directorios en tu Hosting. Por ejemplo, si colocas una directiva en el htaccess de tu carpeta public, estas directivas serán heredadas hacia todos los directorios que están dentro, pero si ubicas una directiva dentro de una carpeta específica, solo de aplicará dentro de ella (y las subcarpetas que contenga dentro).

IMPORTANTE: debes saber que htaccess es muy poderoso, y cualquier directiva mal ingresada o tipeada podría terminar en un Error 500 (Internal Server Error) haciendo que tu sitio web deje de funcionar o volverlo inseguro.

Lo primero que haremos será permitir el uso del archivo .htaccess

Nos vamos al directorio /etc/apache2 y editamos el archivo apache2.conf con el siguiente comando:

```
|adminwes@AFR-US...:/etc/apache2$ sudo nano apache2.conf|
```

Dentro del directorio /var/www/ de apache.conf, se cambia el valor de AllowOverride, pasando de None a All.

```
GNU nano 4.8                                     apache2.conf
<Directory /var/www/>
    Options -Indexes +FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

Se comprueban que están bien realizados los cambios, se guarda y se cierra.

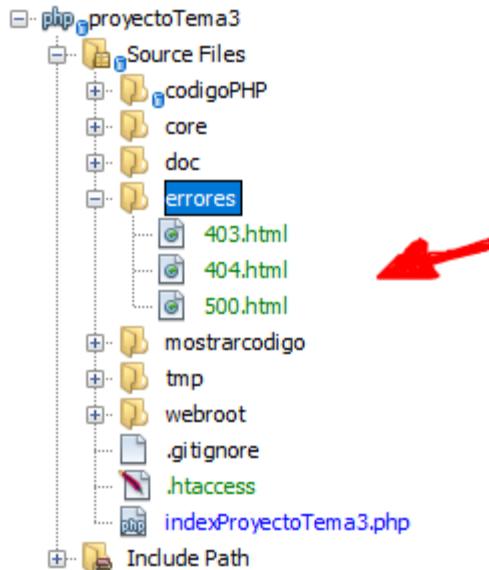
Se reinicia el servicio de apache2 con el siguiente comando:

```
|adminwes@AFR-US...:/etc/apache2$ sudo service apache2 restart|
```

Personalizar páginas de error

Los errores más comunes suelen ser 400 (petición incorrecta), 403 (acceso prohibido), 404 (archivo no encontrado), y los errores 500, tales como el error 500 (error interno del servidor), 502 (puerta de enlace no válida) o 503 (servicio no disponible).

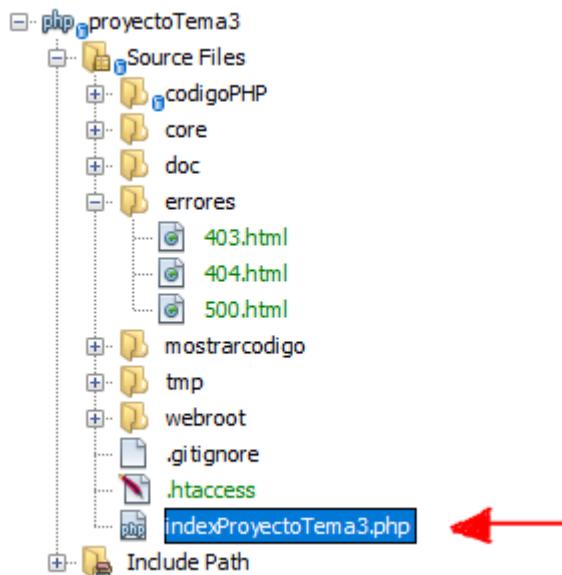
Para cada uno de los errores deberás crear primero un documento html y almacenarlo en tu Hosting o proyecto dentro de una carpeta, en mi caso se llama errores.



Si vas a personalizar una página de error para el 404, crea uno llamado 404.html como en la siguiente imagen y que este en el directorio errores:

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <style>
            div{
                color:red;
            }
        </style>
    </head>
    <body>
        <div>Acceso Prohibido</div>
    </body>
</html>
```

Una vez tengamos configurados todos los archivos .html para cada error, nos iremos al archivo .htaccess, este archivo tiene que estar en el directorio principal del proyecto.



El formato en que debes escribir las directivas es

ErrorDocument [espacio] número-de-error [espacio] ubicación.

Añadiremos estas directivas en el archivo .htaccess:

ErrorDocument 400 /errores/400.html

ErrorDocument 403 /errores/403.html

ErrorDocument 404 /errores/404.html

ErrorDocument 500 /errores/500.html

Nos tendrían que quedar de la siguiente manera:

The image shows a code editor window titled '.htaccess'. The tab 'Source' is selected. The code in the editor is:

```
1 ErrorDocument 400 /errores/400.html
2 ErrorDocument 403 /errores/403.html
3 ErrorDocument 404 /errores/404.html
4 ErrorDocument 500 /errores/500.html
```

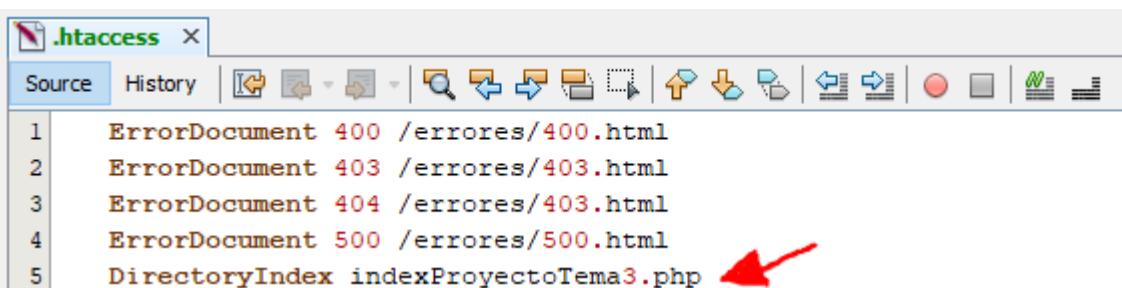
Una vez estén, ya funcionaría en nuestra página web cuando se produzca cualquier error de los configurados. **IMPORTANTE: Cualquier error en el archivo .htaccess hará que nuestro sitio web no funcione y de un error interno del servidor.**

Especificar archivo index

Cuando realizas una petición a un sitio web para una URL que no tiene un nombre de archivo específico, el servidor irá a buscar un archivo de índice definido por defecto. Esto ocurre cuando ingresas a tu dominio <https://tusitio.com> o cuando intentas ingresar a una carpeta (como <https://tusitio.com/carpeta>).

Los archivos de index más comunes son index.html o index.php, pero si lo deseas podrás cambiarlo por algo diferente. No es casual, pero tenemos muchos casos en los que utilizan main.html o home.html cuando se trata de sitios web un poco anticuados.

Forzaremos aquí el archivo de index a indexProyectoTema3.php agregando esta línea al archivo que creamos antes para las directivas llamado .htaccess:



```
1 ErrorDocument 400 /errores/400.html
2 ErrorDocument 403 /errores/403.html
3 ErrorDocument 404 /errores/403.html
4 ErrorDocument 500 /errores/500.html
5 DirectoryIndex indexProyectoTema3.php
```

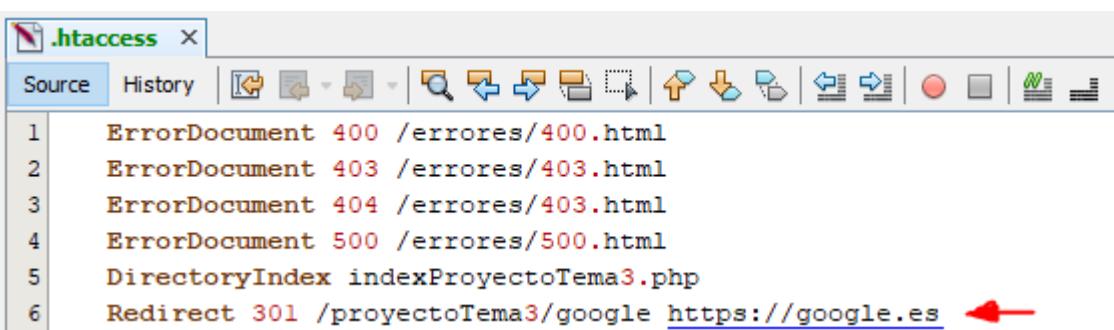
Redireccionar URL con htaccess

Esto es sumamente útil si estás actualizando o reordenando los contenidos de tu sitio web en el caso de que tus URL hayan cambiado. Utilizaremos aquí el método 301 (redirección permanente), la cual ayudará a mantener el peso de una URL ante cualquier cambio.

Si por ejemplo deseas redireccionar una página hacia otra utiliza:

Redirect 301 /url-relativa/archivo.html <https://example.com/full-url.html>

Haremos la redirección mediante el archivo .htaccess de nuevo. En mi caso hago que cuando se entre a la ruta /proyectoTema3/google y detecte que no existe, automáticamente te lleve a <https://google.es>



```
1 ErrorDocument 400 /errores/400.html
2 ErrorDocument 403 /errores/403.html
3 ErrorDocument 404 /errores/403.html
4 ErrorDocument 500 /errores/500.html
5 DirectoryIndex indexProyectoTema3.php
6 Redirect 301 /proyectoTema3/google https://google.es
```

Si deseas redireccionar un sitio web completo:

Redirect 301 / <https://nuevo-dominio.com>

1.5.3 Monitorización

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.

```
adminwes@AFR-US: /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-US: /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 2.9.0d
ev.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 actualmen
te.)
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 nuestra web, con el siguiente comando:

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

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**.

Creamos el usuario operador web, el home del usuario será /var/www/html para que entre directamente cuando se use, indicando que no cree home y dentro del grupo www-data con el siguiente comando:

```
admindwes@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**

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

Una vez tenemos el usuario creado, en el servidor, los archivos se encuentran en el directorio /var/www/html. Para que el usuario operadorweb pueda trabajar con este directorio, debe otorgársele la propiedad del mismo.

Le cambiamos el propietario a el directorio html poniéndole operadorweb e indicándole el grupo www-data, -R (recursive) supone aplicar el cambio a todos los directorios y archivos bajo el modificando. Por defecto todo lo que se encuentre en la carpeta html tiene como propietario el root, solo el superusuario puede modificarlo. Nos meteremos en el directorio con **cd /var/www** y lo realizaremos con el siguiente comando:

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

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

```
admindwes@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:

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

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

```
admindwes@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:

```
admindwes@AFR-US...:~$ sudo apt update|
```

```
admindwes@AFR-US...:~$ 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**):

```
admindwes@AFR-US...:~$ sudo apt install php libapache2-mod-php|
```

Una vez instalados los paquetes, reiniciaremos el servicio Apache:

```
admindwes@AFR-US...:~$ 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**

```
admindwes@AFR-US...:~$ cd /etc/php  
admindwes@AFR-US...:/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**

```
|admindwes@AFR-US...:/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-US ED:/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-US ED:/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.

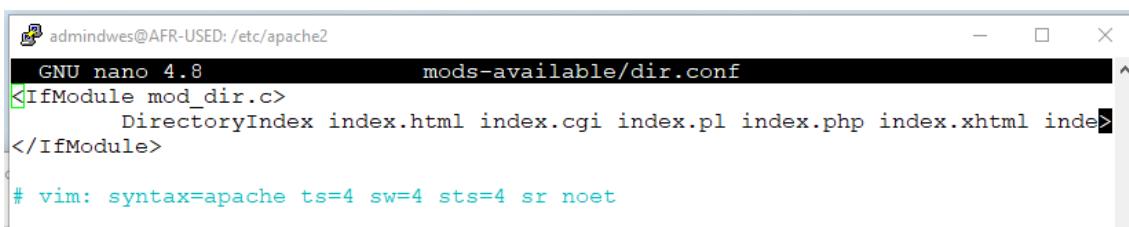
```
admindwes@AFR-USDED:/etc/php/7.4$ sudo service apache2 restart
```

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:

```
admindwes@AFR-USDED:/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:

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



The screenshot shows a terminal window titled "admindwes@AFR-USDED: /etc/apache2". The command "GNU nano 4.8" is displayed at the top. The file "mods-available/dir.conf" is open, showing the following content:

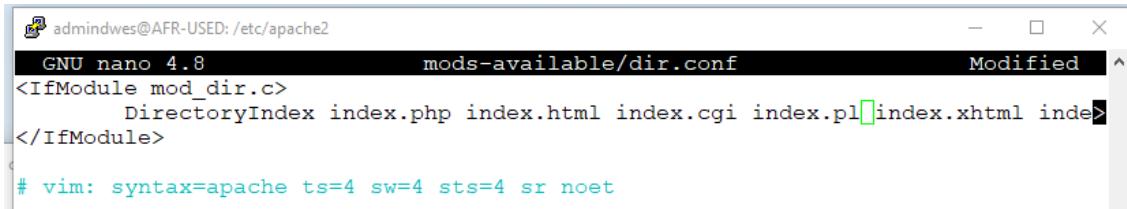
```
<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml inde>
</IfModule>

# 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:

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

Una vez tenemos echo 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.



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

Reiniciamos el servicio apache2 para aplicar los cambios.

```
adminwes@AFR-US: /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:

```
admindwes@AFR-US ED:~$ sudo apt update
```

```
|admindwes@AFR-US ED:~$ sudo apt upgrade
```

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

```
|admindwes@AFR-US ED:~$ 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.

```
admindwes@AFR-US ED:~$ 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-USDED:~$ 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                      *:*
tcp  LISTEN 0      128    [::]:22                  [::]:*
adminwes@AFR-USDED:~$ 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                   *:*
tcp  LISTEN 0      128    [::]:ssh                 [::]:*
```

Una vez instalado MySQL entraremos a configurar el archivo de MySQL que permite la conexión desde cualquier dispositivo, para ello deberemos acceder al archivo `mysqld.cnf` y editarlo con el siguiente comando:

```
adminwes@AFR-USDED:~$ 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 ^L Go To Line ^V
```

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:

```
admindwes@AFR-US ED:~$ sudo service mysql restart
```

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

```
admindwes@AFR-US ED:~$ 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 dirá es el nivel de la password, en mi caso voy a usar el LOW con lo que pondré un 0, seguidamente nos pedirá introducir la password que queremos para acceder, en mi caso he usado P@ssw0rd y el último 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 eliminar los usuarios anónimos por defecto, le diremos que sí.

```
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.
```

El siguiente paso nos preguntara si queremos deshabilitar los inicios de sesión raíz remotos, le daremos 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 las nuevas reglas para que MySQL respete inmediatamente los cambios realizados 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 queharemos será instalar el mod para MySQL de PHP con el siguiente comando:

```
|admin@AFR-US:~$ 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 creamos y le asignemos los permisos salimos con *exit*

```
admindwes@AFR-USUSED:~$ 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:

```
admindwes@AFR-USUSED:~$ 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 | SA5005$buR\!R?k-F17005/0d8/MfUWB1kpBdj/k/pi8F.hfAtEyvfLo12RKo21 | caching_sha2_password | % |
| debian-sys-maint | SA5005$bm6,qjDPVc %Y"6x508EaCspyx/8TUg4fKypMnlBJiqz/pQYpi/nXAC | caching_sha2_password | localhost |
| mysql.infoschema | SA5005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBEBUSED | caching_sha2_password | localhost |
| mysql.session | SA5005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBEBUSED | caching_sha2_password | localhost |
| mysql.sys | SA5005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBEBUSED | caching_sha2_password | localhost |
| root | | 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.

```
adminidwes@AFR-USED:~$ 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)
```

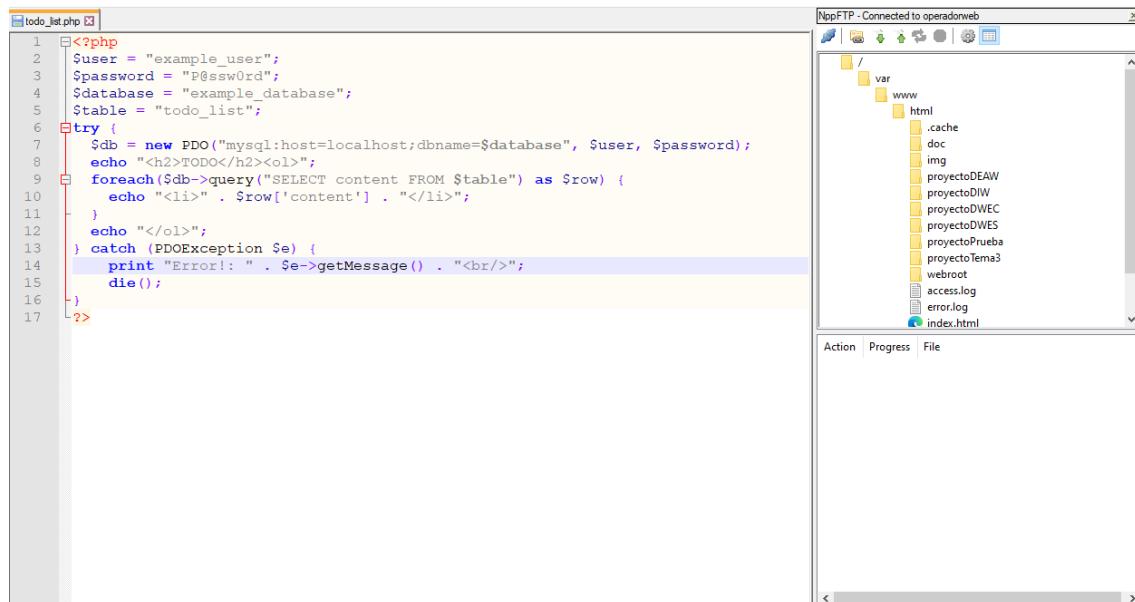
Una vez metidos todos los valores, comprobamos que se han introducido bien en la tabla con el siguiente comando:

```
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:

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

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



The screenshot shows two windows. On the left is a terminal window titled 'todo_list.php' containing the following PHP code:

```
1 <?php
2 $user = "example_user";
3 $password = "P@ssw0rd";
4 $database = "example_database";
5 $table = "todo_list";
6 try {
7     $db = new PDO("mysql:host=localhost;dbname=$database", $user, $password);
8     echo "<h2>TODO</h2><ol>";
9     foreach($db->query("SELECT content FROM $table") as $row) {
10         echo "<li>" . $row['content'] . "</li>";
11     }
12     echo "</ol>";
13 } catch (PDOException $e) {
14     print "Error! " . $e->getMessage() . "<br/>";
15     die();
16 }
?>
```

On the right is an NppFTP interface titled 'NppFTP - Connected to operadonweb'. It shows the directory structure of '/var/www/html' with files like .cache, doc, img, proyectoDEAW, proyectoDIW, proyectoDWEC, proyectoDWES, proyectoPrueba, proyectoTema3, webroot, access.log, error.log, and index.html.

1.9 Instalación y Configuración de XDebug

1.9.1 Instalación

Lo primero que tenemos que hacer antes de instalar el XDebug es un update y un upgrade del sistema.

```
admindwes@AFR-US...:~$ sudo apt update
```

```
|admindwes@AFR-US...:~$ sudo apt upgrade
```

Una vez realizado lo anterior, procederemos a instalar el XDebug con el comando *sudo apt install php-xdebug*

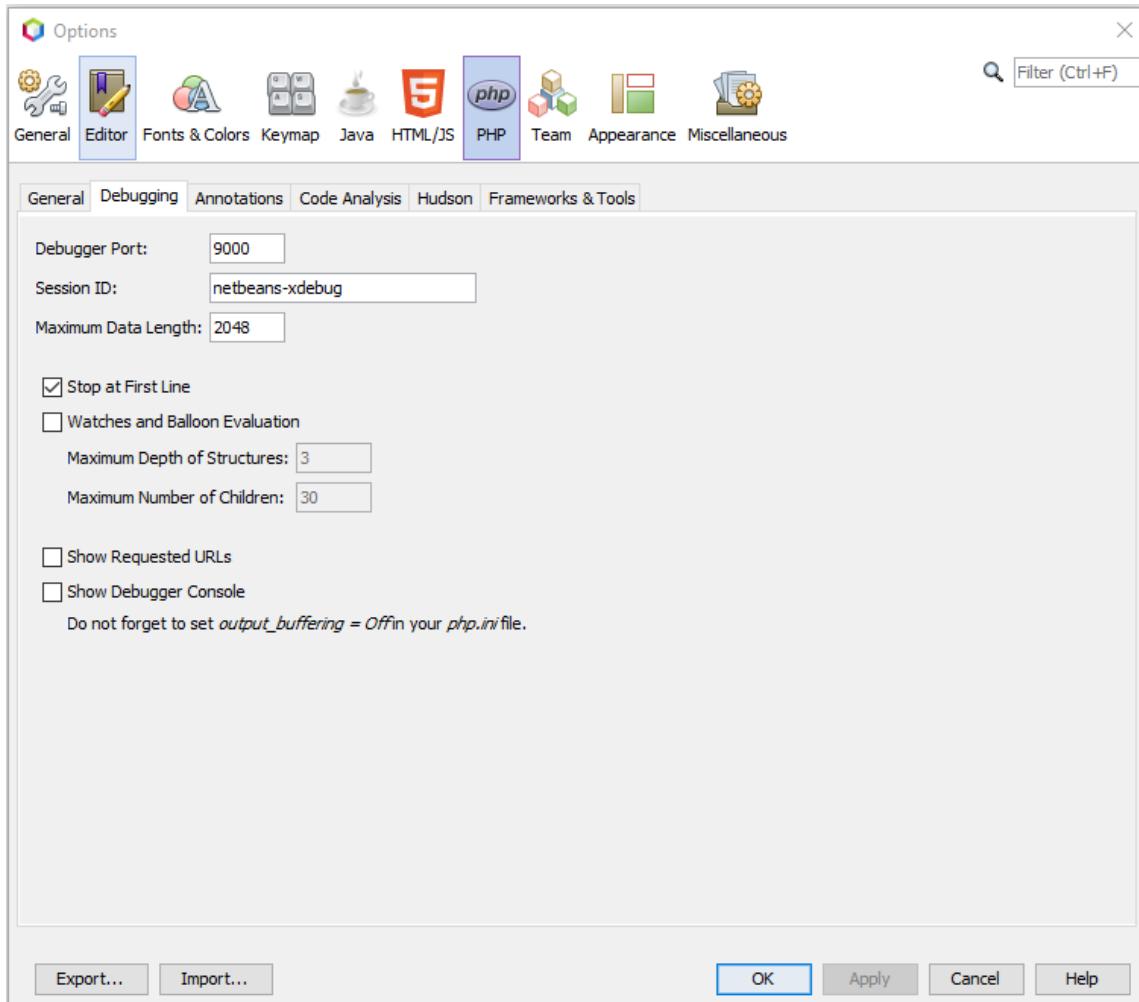
```
admindwes@AFR-US...:~$ sudo apt install php-xdebug
[sudo] password for admindwes:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  php-xdebug
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 473 kB de archivos.
Se utilizarán 2.174 kB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 php-xdebug amd64
2.9.2+2.8.1+2.5.5-1build1 [473 kB]
Descargados 473 kB en 0s (1.314 kB/s)
Seleccionando el paquete php-xdebug previamente no seleccionado.
(Leyendo la base de datos ... 109581 ficheros o directorios instalados actualmen
te.)
Preparando para desempaquetar .../php-xdebug_2.9.2+2.8.1+2.5.5-1build1_amd64.deb
...
Desempaquetando php-xdebug (2.9.2+2.8.1+2.5.5-1build1) ...
Configurando php-xdebug (2.9.2+2.8.1+2.5.5-1build1) ...
Procesando disparadores para libapache2-mod-php7.4 (7.4.3-4ubuntu2.6) ...
Procesando disparadores para php7.4-cli (7.4.3-4ubuntu2.6) ...
```

Una vez se instale, reiniciaremos el servicio de apache2 con el siguiente comando:

```
admindwes@AFR-US...:~$ sudo service apache2 restart
```

1.9.2 Configuración

Ahora iremos a la configuración de NetBeans para ver los parámetros del apartado PHP y después el de Debugging, estos parámetros será necesario que estén igual en el archivo de configuración xdebug.ini en el servidor para que funcione.



Ahora, una vez tenemos los datos para el archivo que hay que configurar, nos iremos a la carpeta de mods-available con el comando `cd /etc/php/7.4/mods-available` y haremos un `ls` para ver los ficheros que tiene, de todos estos editaremos el **xdebug.ini**

```
adminwes@AFR-USED:~$ cd /etc/php/7.4
adminwes@AFR-USED:/etc/php/7.4$ ls
apache2 cli mods-available
adminwes@AFR-USED:/etc/php/7.4$ cd mods-available/
adminwes@AFR-USED:/etc/php/7.4/mods-available$ ls
calendar.ini ftp.ini mysqlnd.ini posix.ini sysvsem.ini
ctype.ini gettext.ini opcache.ini readline.ini sysvshm.ini
exif.ini iconv.ini pdo.ini shmop.ini tokenizer.ini
ffi.ini json.ini pdo_mysql.ini sockets.ini xdebug.ini
fileinfo.ini mysqli.ini phar.ini sysvmsg.ini
```

Editaremos el archivo xdebug.ini con el siguiente comando:

```
adminwes@AFR-USED:/etc/php/7.4/mods-available$ sudo nano xdebug.ini
```

Una vez se nos abra el archivo introduciremos las siguientes líneas en dicho archivo, una vez introducidas, guardaremos y cerraremos el archivo. (En dicho archivo la primera línea **zend_extension=xdebug.so** ya viene introducida)

```
GNU nano 4.8                               xdebug.ini                                Modified
zend_extension=xdebug.so
xdebug.remote_enable=on
xdebug.idekey = netbeans-xdebug
xdebug.remote_host = "localhost"
xdebug.remote_port = 9000
```

Ahora reiniciaremos el servicio de apache2 con el siguiente comando para aplicar los cambios:

```
admindwes@AFR-USDED:/etc/php/7.4/mods-available$ service apache2 restart
===== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to restart 'apache2.service'.
Authenticating as: admin (admindwes)
Password:
===== AUTHENTICATION COMPLETE =====
```

Comprobaremos que este activo el servicio ufw (firewall) con el siguiente comando:

```
admindwes@AFR-USDED:/etc/php/7.4/mods-available$ sudo service ufw status
● ufw.service - Uncomplicated firewall
  Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset: enabled)
  Active: active (exited) since Tue 2021-10-19 08:21:11 UTC; 19min ago
    Docs: man:ufw(8)
   Process: 352 ExecStart=/lib/ufw/ufw-init start quiet (code=exited, status=0/SUCCESS)
 Main PID: 352 (code=exited, status=0/SUCCESS)

Warning: journal has been rotated since unit was started, output may be incomplete.
```

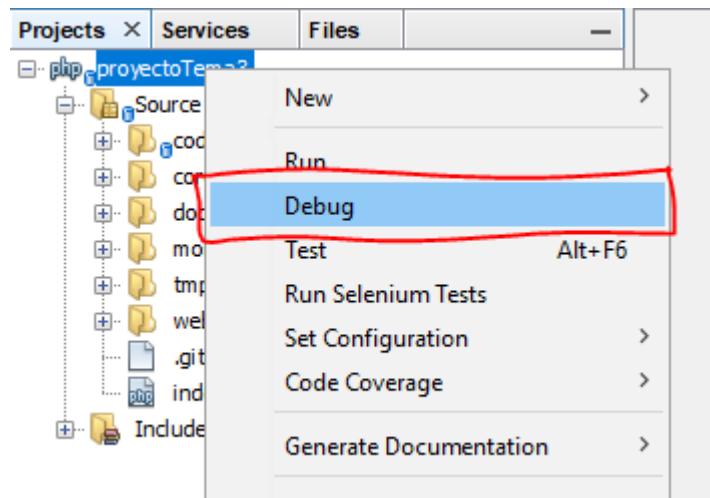
Daremos permiso por el puerto 9000 al firewall con el siguiente comando:

```
admindwes@AFR-USDED:/etc/php/7.4/mods-available$ sudo ufw allow 9000
Rules updated
Rules updated (v6)
```

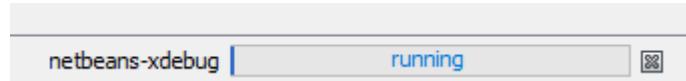
Y, por último, reiniciaremos el servicio ufw (firewall)

```
|admindwes@AFR-USDED:/etc/php/7.4/mods-available$ sudo service ufw restart
```

Ahora nos iremos a NetBeans, y haremos clic derecho sobre el proyecto y le daremos a Debug.



Si se ha configurado correctamente y funciona, en la parte inferior derecha, debe aparecer que XDebug está running y en la parte superior habrá un menú de opciones.



En caso de que no nos ponga running, nos iremos a el servidor y pondremos el siguiente comando, si nos pone inactivo, indica que no funciona.

```
admindwes@AFR-US: /etc/php/7.4/mods-available$ sudo ufw status numbered
Status: inactive
```

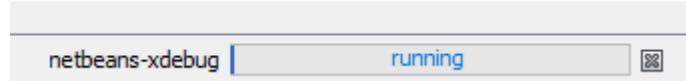
Para arreglar este problema, iremos de nuevo al archivo que configuramos anteriormente llamado xdebug.ini y tendremos que añadir más líneas de configuración, una vez estén, guardamos y cerramos el archivo.

```
GNU nano 4.8                                     xdebug.ini
zend_extension=xdebug.so
xdebug.show_error_trace=1
xdebug.remote_enable=on
xdebug.remote_handler=dbgp
xdebug.idekey = netbeans-xdebug
xdebug.remote_host = "localhost"
xdebug.remote_port = 9000
xdebug.remote_connect_back=1
```

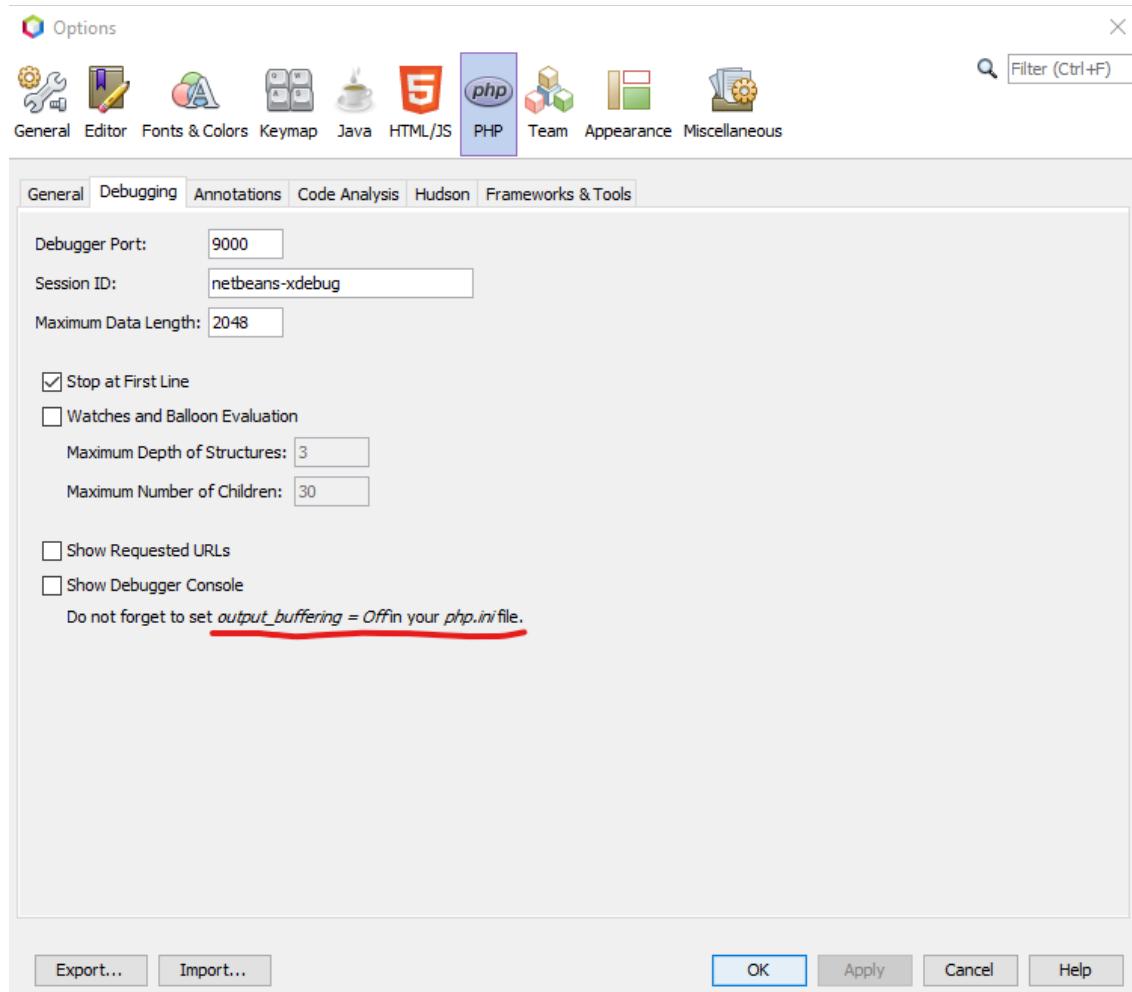
Reiniciaremos el servicio apache2 para aplicar los cambios realizados con el siguiente comando:

```
admindwes@AFR-US: /etc/php/7.4/mods-available$ service apache2 restart
===== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to restart 'apache2.service'.
Authenticating as: admin (admindwes)
Password:
===== AUTHENTICATION COMPLETE =====
```

Ahora, volveremos a NetBeans y podemos ya comprobar que se encuentra running el XDebug.



Para poder usar la consola Debugger, lo primero que habrá que hacer será activarlo en las opciones de PHP de NetBeans.



Una vez activado en NetBeans, lo siguiente será configurar lo necesario en el servidor, nos iremos a el directorio apache2 con el siguiente comando:

```
|adminwes@AFR-US:~$ cd /etc/php/7.4/apache2|
```

Una vez en el directorio, entraremos a editar el archivo php.ini con el siguiente comando:

```
|adminwes@AFR-US:~/etc/php/7.4/apache2$ sudo nano php.ini
```

Dentro de este archivo, buscaremos la frase output_buffering.

```
GNU nano 4.8          php.ini
; You may be able to send headers and cookies after you've already sent output
; through print or echo. You also may see performance benefits if your server is
; emitting less packets due to buffered output versus PHP streaming the output
; as it gets it. On production servers, 4096 bytes is a good setting for perfor>
; reasons.
; Note: Output buffering can also be controlled via Output Buffering Control
;       functions.
; Possible Values:
;   On = Enabled and buffer is unlimited. (Use with caution)
;   Off = Disabled
;   Integer = Enables the buffer and sets its maximum size in bytes.
; Note: This directive is hardcoded to Off for the CLI SAPI
; Default Value: Off
; Development Value: 4096
; Production Value: 4096
; http://php.net/output-buffering
output_buffering = 4096

; You can redirect all of the output of your scripts to a function. For
; example, if you set output_handler to "mb_output_handler", character

^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 encontrada, la comentaremos y añadiremos la frase output_buffering = off, una vez introducida, guardaremos el archivo y lo cerraremos.

```
GNU nano 4.8          php.ini          Modified
; You may be able to send headers and cookies after you've already sent output
; through print or echo. You also may see performance benefits if your server is
; emitting less packets due to buffered output versus PHP streaming the output
; as it gets it. On production servers, 4096 bytes is a good setting for perfor>
; reasons.
; Note: Output buffering can also be controlled via Output Buffering Control
;       functions.
; Possible Values:
;   On = Enabled and buffer is unlimited. (Use with caution)
;   Off = Disabled
;   Integer = Enables the buffer and sets its maximum size in bytes.
; Note: This directive is hardcoded to Off for the CLI SAPI
; Default Value: Off
; Development Value: 4096
; Production Value: 4096
; http://php.net/output-buffering
;output_buffering = 4096
output_buffering = off

; You can redirect all of the output of your scripts to a function. For

^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
```

Para que se apliquen los cambios, reiniciaremos el servicio de apache2 con el siguiente comando:

```
admindwes@AFR-US: /etc/php/7.4/apache2$ sudo service apache2 restart
```

1.10 Conexión mediante clave privada con SSH

1.10.1 Configuración

Tenemos un fichero de configuración de las claves publicas autorizadas con las que podemos conectarnos mediante SSH, para poder conectarnos con la clave hay que configurar un archivo llamado sshd_config, este archivo se encuentra en /etc/ssh.

Haremos primero un *ls /etc/ssh* para ver los dos archivos que tenemos de configuración, por un lado, ssh_config para la configuración de un cliente ssh y después sshd_config para la configuración del servidor ssh.

```
adminwes@AFR-USED:~$ ls /etc/ssh
moduli      ssh_host_dsa_key      ssh_host_ed25519_key.pub
ssh_config  ssh_host_dsa_key.pub  ssh_host_rsa_key
ssh_config.d ssh_host_ecdsa_key   ssh_host_rsa_key.pub
sshd_config  ssh_host_ecdsa_key.pub ssh_import_id
sshd_config.d ssh_host_ed25519_key
```

Una vez hemos visto los dos archivos de configuración, pasamos a editar el archivo sshd_config con el comando *sudo nano /etc/ssh/sshd_config*

Dentro del archivo, des comentamos la línea AuthorizedkeysFile, y añadimos los siguientes parámetros **.ssh/authorized_keys .ssh/authorized_keys2 %h/.ssh/authorized_keys**

Una vez introducidos, guardamos el archivo y lo cerramos.

```
GNU nano 4.8                                     sshd config                                         Modified
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2 %h/.ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

^G Get Help      ^Q Write Out     ^W Where Is      ^K Cut Text      ^J Justify      ^C Cur Pos      M-U Undo
^X Exit          ^R Read File    ^\ Replace       ^U Paste Text   ^T To Spell    ^  Go To Line   M-E Redo
```

Para que se apliquen los cambios será necesario reiniciar apache con el siguiente comando:

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

1.10.2 Generar clave

Para generar una clave ssh usaremos el siguiente comando, entre las comillas dobles deberemos introducir nuestro correo.

```
operadorweb@AFR-US...:~$ ssh-keygen -t rsa -b 4096 -C "alberto.ferram@educa.jcyl.es"
```

Una vez ejecutado el comando anterior, se nos generara una clave privada y una pública.

Se nos guardara en el directorio **/var/www/html/.ssh/id_rsa** si no le decimos nada.

Nos pedirá una Password para ver la clave, en mi caso no le voy a poner ninguna, pero es recomendable ponerle una.

```
Generating public/private rsa key pair.  
Enter file in which to save the key (/var/www/html/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /var/www/html/.ssh/id_rsa  
Your public key has been saved in /var/www/html/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:diG1oRn/nx/D1z7DmpfWwEC+rnKwTfiG2E25b8pKzWs alberto.ferram@educa.jcyl.es  
The key's randomart image is:  
+---[RSA 4096]---+  
| . o |  
| * o. |  
| + o |  
| . oo . |  
| S....+ o |  
| .o++ ..=oo |  
| o.Xoo oO+ |  
| ..=.E.o o==|  
| .*=..oo . |  
+---[SHA256]---
```

Agregaremos la clave publica al archivo authorized_keys desde el directorio /.ssh con el siguiente comando:

```
operadorweb@AFR-US...:~/.ssh$ cat id_rsa.pub >> authorized_keys|
```

Para ver si la clave privada y la clave publica se han guardado correctamente, nos iremos al directorio .ssh con el siguiente comando:

```
|operadorweb@AFR-US...:~$ cd /var/www/html/.ssh
```

Una vez dentro hacemos **ls** y deberíamos ver el archivo **id_rsa** que es la clave privada y por otro lado **id_rsa.pub** que es la clave pública.

También podremos comprobarlo con FileZilla en la carpeta .ssh si están las claves.

The screenshot shows the FileZilla interface with a 'Situs remoto' (Remote Site) set to '/var/www/html/.ssh'. The directory structure is displayed in the left pane, showing subfolders like var, www, html, .cache, .ssh, doc, errores, holaMundo, and img. The right pane displays a table of files:

Nombre de archivo	Tamaño d...	Tipo de arc...	Última modificación	Permisos	Propietario/...
..					
authorized_keys	1.508	Archivo	09/11/2021 10:56:46	-rw-r--r--	operadorwe...
id_rsa	3.401	Archivo	09/11/2021 10:53:42	-rw-----	operadorwe...
id_rsa.pub	754	Microsoft ...	09/11/2021 10:53:42	-rw-r--r--	operadorwe...

1.11 Enjaular usuarios con SSH

1.11.1 Configuración

Creación del usuario

Para enjaular a un usuario, lo primero que haremos será crear un grupo al que pertenecerán todos los usuarios creados para conexiones FTP. Crearemos el grupo con el siguiente comando:

```
admindwes@AFR-US ED:~$ sudo gropadd ftpusers
```

Una vez tenemos el grupo, pasaremos a crear el usuario, este usuario pertenecerá a el grupo www-data por defecto, también tendrá que pertenecer al grupo ftpusers y al directorio /var/www/DAW201(**Nombre de usuario**) y tendrá el nombre DAW201. Para realizar la creación del usuario usaremos el siguiente comando:

```
admindwes@AFR-US ED:~$ sudo useradd -g www-data -G ftpusers -m -d /var/www/DAW201 DAW201
```

Una vez creado el usuario le asignaremos una contraseña (yo he puesto paso de contraseña) con el siguiente comando:

```
admindwes@AFR-US ED:~$ sudo passwd DAW201  
New password:  
Retype new password:  
passwd: password updated successfully
```

Con el usuario creado correctamente, comprobamos si está en el grupo ftpusers con el siguiente comando:

```
admindwes@AFR-US ED:~$ cat /etc/group|grep ftpusers  
ftpusers:x:1001:DAW201
```

También comprobaremos si tiene el nombre bien y se encuentra en el directorio correcto con el siguiente comando:

```
admindwes@AFR-US ED:~$ cat /etc/passwd|grep DAW  
DAW201:x:1002:33::/var/www/DAW201:/bin/sh
```

La última comprobación que haremos será la siguiente, veremos los dos grupos y el uid con el nombre del usuario. Lo haremos con el siguiente comando:

```
admindwes@AFR-US ED:~$ id DAW201  
uid=1002 (DAW201) gid=33 (www-data) groups=33 (www-data),1001 (ftpusers)
```

Configuración de carpeta home del usuario

El propietario del directorio jaula y los directorios sobre este deben ser root, el home del usuario pertenece al root, entonces tendremos que cambiárselo con el siguiente comando:

```
admindwes@AFR-US ED:~$ sudo chown root:root /var/www/DAW201
```

Lo siguiente que haremos será eliminar el permiso de escritura con el siguiente comando:

```
admindwes@AFR-US ED:~$ sudo chmod -w /var/www/DAW201
```

Comprobamos que hemos realizado bien los cambios sobre el usuario DAW201 con el siguiente comando:

```
admindwes@AFR-USDED:~$ ls -l /var/www
total 8
dr-xr-xr-x  2 root      root      4096 nov 10 10:45 DAW201
drwxrwsr-x 17 operadorweb www-data 4096 nov  9 10:17 html
```

Crear carpeta public_html

Ahora crearemos la carpeta **public_html** en **/var/www/DAW201** con el siguiente comando:

```
admindwes@AFR-USDED:~$ sudo mkdir /var/www/DAW201/public_html
```

Una vez creada, le daremos permisos a la carpeta **public_html** con el siguiente comando:

```
admindwes@AFR-USDED:~$ sudo chmod 2775 -R /var/www/DAW201/public_html
```

Por último, le asignaremos los propietarios a la carpeta **public_html** con el siguiente comando:

```
|admindwes@AFR-USDED:~$ sudo chown DAW201:www-data -R /var/www/DAW201/public_html
```

Editar /etc/ssh/sshd_config

Ahora vamos a configurar el archivo **sshd_config**, en vez de editarla, usaremos uno de un repositorio para evitar meter errores en dicho archivo al hacer la edición, clonare el siguiente repositorio dentro de **/home/admindwes** con el siguiente comando y haremos un **ls** para ver si esta bajado correctamente:

```
admindwes@AFR-USDED:~$ git clone https://github.com/IESSAUCE/SERVERDAW.git
Cloning into 'SERVERDAW'...
remote: Enumerating objects: 189, done.
remote: Counting objects: 100% (45/45), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 189 (delta 31), reused 27 (delta 21), pack-reused 144
Receiving objects: 100% (189/189), 32.30 KiB | 972.00 KiB/s, done.
Resolving deltas: 100% (73/73), done.
admindwes@AFR-USDED:~$ ls
SERVERDAW
```

Una vez vemos la carpeta, entramos en ella con el siguiente comando:

```
admindwes@AFR-USDED:~$ cd SERVERDAW/
```

Hacemos un **ls** para ver que contiene, tendremos que ver una carpeta llamada **sftp**.

```
admindwes@AFR-USDED:~/SERVERDAW$ ls
AuthLDAPApache bind INSTALL_SERVERUS.md interfaces ldap LICENSE README.md sftp users
```

Entraremos en la carpeta **sftp** con **cd** y haremos un **ls** para ver que contiene.

```
admindwes@AFR-USDED:~/SERVERDAW$ cd sftp/
admindwes@AFR-USDED:~/SERVERDAW/sftp$ ls
INSTALL_SFTP.md sshd_config
```

Dicha carpeta contiene el archivo de configuración **sshd_config**, antes de copiarlo y remplazarlo por el nuestro, haremos una copia de seguridad de nuestro archivo con el siguiente comando:

```
admindwes@AFR-US ED:~/SERVERDAW/sftp$ sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.backup
```

Una vez hecha la copia, comprobamos con **ls** en **/etc/ssh** que se realizó la copia correctamente.

```
admindwes@AFR-US ED:~/SERVERDAW/sftp$ ls /etc/ssh
moduli      sshd_config.backup    ssh_host_ecdsa_key      ssh_host_rsa_key
ssh_config   sshd_config.d        ssh_host_ecdsa_key.pub  ssh_host_rsa_key.pub
ssh_config.d ssh_host_dsa_key    ssh_host_ed25519_key   ssh_import_id
sshd_config  ssh_host_dsa_key.pub ssh_host_ed25519_key.pub
```

Ahora que tenemos hecha la copia de seguridad de nuestro archivo, ya podemos copiar el archivo del repositorio y remplazarlo por el nuestro con el siguiente comando:

```
admindwes@AFR-US ED:~/SERVERDAW/sftp$ sudo cp
INSTALL_SFTP.md  sshd_config
admindwes@AFR-US ED:~/SERVERDAW/sftp$ sudo cp sshd_config /etc/ssh
```

Una vez tenemos el archivo, reiniciamos el servicio ssh con el siguiente comando:

```
admindwes@AFR-US ED:~/SERVERDAW/sftp$ sudo service ssh restart
```

(En caso de editar el archivo a mano y meterle los parámetros por teclado, los pasos serían solamente hacer una copia de seguridad del archivo por seguridad, editararlo, guardarlo y, por último, reiniciar el servicio.)

1.12 Instalación y configuración de DNS

1.12.1 Instalación

Para poder configurar el DNS lo primero que necesitamos es comprobar si tenemos instalado el bind9 con el siguiente comando:

```
admindwes@AFR-US ED:~$ dpkg -s bind9
      el paquete 'bind9' no está instalado y no hay ninguna información disponible.
      Use dpkg --info (= dpkg-deb --info) to examine archive files.
```

Si no está instalado, haremos antes de instalarlo un update del sistema con el siguiente comando:

```
admindwes@AFR-US ED:~$ sudo apt update
```

Una vez hecho el update, instalamos bind9 con el siguiente comando:

```
admindwes@AFR-US ED:~$ sudo apt install bind9
```

Una vez este instalado comprobamos que este activo con el siguiente comando:

```
admindwes@AFR-US ED:~$ sudo service bind9 status
● named.service - BIND Domain Name Server
  Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2021-11-24 08:34:05 UTC; 1min 3s ago
    Docs: man:named(8)
   Main PID: 2186 (named)
     Tasks: 5 (limit: 2279)
    Memory: 13.7M
       CGroup: /system.slice/named.service
               └─2186 /usr/sbin/named -f -u bind

nov 24 08:34:06 AFR-US ED named[2186]: network unreachable resolving './DNSKEY/IN': 2001:500:9f::42#53
nov 24 08:34:06 AFR-US ED named[2186]: network unreachable resolving './NS/IN': 2001:500:9f::42#53
nov 24 08:34:06 AFR-US ED named[2186]: network unreachable resolving './DNSKEY/IN': 2001:500:2::c#53
nov 24 08:34:06 AFR-US ED named[2186]: network unreachable resolving './NS/IN': 2001:500:2::c#53
nov 24 08:34:06 AFR-US ED named[2186]: network unreachable resolving './DNSKEY/IN': 2001:500:2d::d#53
nov 24 08:34:06 AFR-US ED named[2186]: network unreachable resolving './NS/IN': 2001:500:2d::d#53
nov 24 08:34:06 AFR-US ED named[2186]: network unreachable resolving './DNSKEY/IN': 2001:500:12::d0d#53
nov 24 08:34:06 AFR-US ED named[2186]: network unreachable resolving './NS/IN': 2001:500:12::d0d#53
nov 24 08:34:06 AFR-US ED named[2186]: managed-keys-zone: Initializing automatic trust anchor management f
nov 24 08:34:06 AFR-US ED named[2186]: resolver priming query complete
lines 1-20/20 (END)
```

1.12.2 Configuración

Para configurar el DNS accederemos al directorio bind con el comando `cd /etc/bind` y haremos un `ls` para ver los archivos que contiene.

```
admindwes@AFR-US ED:~$ cd /etc/bind
admindwes@AFR-US ED:/etc/bind$ ls
bind.keys  db.127  db.empty  named.conf          named.conf.local      rndc.key
db.0        db.255  db.local   named.conf.default-zones  named.conf.options  zones.rfc1918
```

Mostraremos el contenido del archivo `named.conf` con el siguiente comando:

```
admindwes@AFR-US ED:/etc/bind$ cat named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

Una vez visto, podremos entrar a configurar el archivo named.conf.local con el siguiente comando:

```
admindwes@AFR-US ED:/etc/bind$ sudo nano named.conf.local
```

Una vez dentro agregaremos la zona directa “nombre.local” y los demás parámetros.

Zona directa se encarga de registrar los registros necesarios para que resuelva nombres de dominio de un equipo o servicio y nos devuelva la IP.

```
GNU nano 4.8                                              named.conf.local
//                                                      
// Do any local configuration here
//                                                      
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "alberto.local" {
    type master;
    file "/etc/bind/db.alberto.local";
};
```

Una vez configurado el archivo, comprobaremos que los ficheros modificados no contienen ningún error sintáctico con el siguiente comando:

```
admindwes@AFR-US ED:/etc/bind$ sudo named-checkconf named.conf.local
admindwes@AFR-US ED:/etc/bind$
```

Para crear la zona directa copiamos el fichero db.local con el siguiente comando:

```
admindwes@AFR-US ED:/etc/bind$ sudo cp /etc/bind/db.local /etc/bind/db.alberto.local
```

Una vez esta copiado el archivo, entramos a configurarlo con el siguiente comando:

```
admindwes@AFR-US ED:/etc/bind$ sudo nano db.alberto.local
```

Editamos el fichero, los parámetros personalizables son **nombre.local** y **xxx-used.nombre.local**.

El "@" equivale al nombre de la zona. Una vez editado lo guardamos y lo cerramos.



The screenshot shows the terminal window with the command "sudo nano db.alberto.local" running. The nano editor interface is visible, displaying the zone configuration. The file contains the following content:

```
; BIND data file for alberto.local
; SOA AFR-US ED.alberto.local. root.localhost. (
    IN      SOA   AFR-US ED.alberto.local. root.localhost. (
        1           ; Serial
        604800       ; Refresh
        86400        ; Retry
        2419200      ; Expire
        604800 )     ; Negative Cache TTL
    ;
    @      IN      NS    AFR-US ED.alberto.local.
;Registros Host
    @      IN      A     192.168.3.107
    AFR-US   IN      A     192.168.3.107

;Registros Alias
    daw201    IN      CNAME AFR-US ED.alberto.local.
    daw202    IN      CNAME AFR-US ED.alberto.local.
    daw203    IN      CNAME AFR-US ED.alberto.local.
    daw204    IN      CNAME AFR-US ED.alberto.local.
    www.daw201 IN      CNAME AFR-US ED.alberto.local.
    www.daw202 IN      CNAME AFR-US ED.alberto.local.
    www.daw203 IN      CNAME AFR-US ED.alberto.local.
    www.daw204 IN      CNAME AFR-US ED.alberto.local.
```

Chequeamos el fichero que contiene la zona "nombre.local" con el siguiente comando:

```
admindwes@AFR-US ED:/etc/bind$ sudo named-checkzone alberto.local db.alberto.local
zone alberto.local/IN: loaded serial 1
OK
```

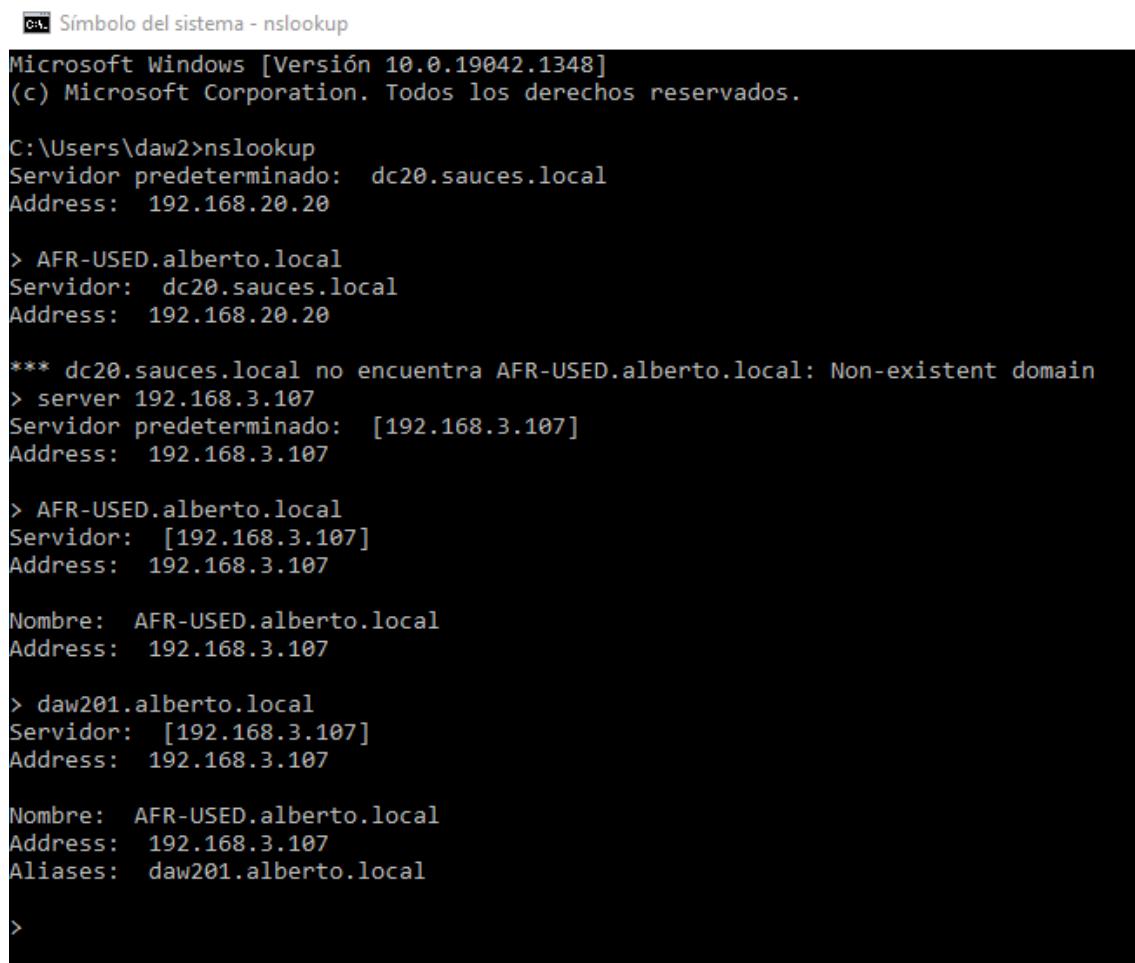
Por último, reiniciamos el servicio bind9 con el siguiente comando:

```
admindwes@AFR-US ED:/etc/bind$ sudo service bind9 restart
```

Para ver si funciona todo lo que hemos configurado, nos vamos a W10 y abrimos el cmd.

Pondremos el comando *nslookup* y una vez entremos tenemos que poner el comando *server 192.168.3.107* que es la IP de la máquina que contiene la configuración de la zona.

Ahora ya podremos poner alias para comprobar si funcionan como *xxx-used.nombre.local* o *daw201.nombre.local*



The screenshot shows a Windows Command Prompt window titled "Símbolo del sistema - nslookup". The output of the nslookup command is displayed, showing various lookups for hosts like AFR-US ED, daw201, and daw201.alberto.local, along with their addresses and aliases.

```
Símbolo del sistema - nslookup
Microsoft Windows [Versión 10.0.19042.1348]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\daw2>nslookup
Servidor predeterminado: dc20.sauces.local
Address: 192.168.20.20

> AFR-US ED.alberto.local
Servidor: dc20.sauces.local
Address: 192.168.20.20

*** dc20.sauces.local no encuentra AFR-US ED.alberto.local: Non-existent domain
> server 192.168.3.107
Servidor predeterminado: [192.168.3.107]
Address: 192.168.3.107

> AFR-US ED.alberto.local
Servidor: [192.168.3.107]
Address: 192.168.3.107

Nombre: AFR-US ED.alberto.local
Address: 192.168.3.107

> daw201.alberto.local
Servidor: [192.168.3.107]
Address: 192.168.3.107

Nombre: AFR-US ED.alberto.local
Address: 192.168.3.107
Aliases: daw201.alberto.local

>
```

1.12.3 Configuración Arpa

Para configurar la zona arpa, primero nos iremos al directorio bind con el siguiente comando y hacemos un `ls` para ver su contenido.

```
cd /etc/bind  
ls
```

Echaremos un vistazo al fichero named.conf con el siguiente comando:

```
cat named.conf
```

Ahora entramos a editar el fichero named.conf.local con el siguiente comando:

```
admindwes@AFR-US...:/etc/bind$ sudo nano named.conf.local
```

Una vez dentro del archivo añadiremos la zona arpa, lo haremos debajo de la zona directa que configuramos anteriormente. El contenido entrecomillado será la IP al revés sin el host más .in-addr.arpa.

Una vez editado guardamos y cerramos el archivo.

```
GNU nano 4.8                               named.conf.local                         Modified  
//  
// Do any local configuration here  
  
// Consider adding the 1918 zones here, if they are not used in your  
// organization  
//include "/etc/bind/zones.rfc1918";  
  
zone "alberto.local" {  
    type master;  
    file "/etc/bind/db.alberto.local";  
};  
zone "3.168.192.in-addr.arpa" {  
    type master;  
    file "/etc/bind/db.3.168.192.in-addr.arpa";  
};  
]  
  
^G Get Help      ^O Write Out     ^W Where Is      ^R Cut Text      ^J Justify      ^C Cur Pos  
^X Exit          ^R Read File     ^M Replace       ^U Paste Text    ^T To Spell     ^L Go To Line
```

Una vez editado el archivo chequeamos que está bien con el siguiente comando:

```
admindwes@AFR-US...:/etc/bind$ sudo named-checkconf
```

Haremos ahora `ls` para ver los archivos y vamos a copiar el archivo db.nombre.local para crear la zona arpa con el siguiente comando:

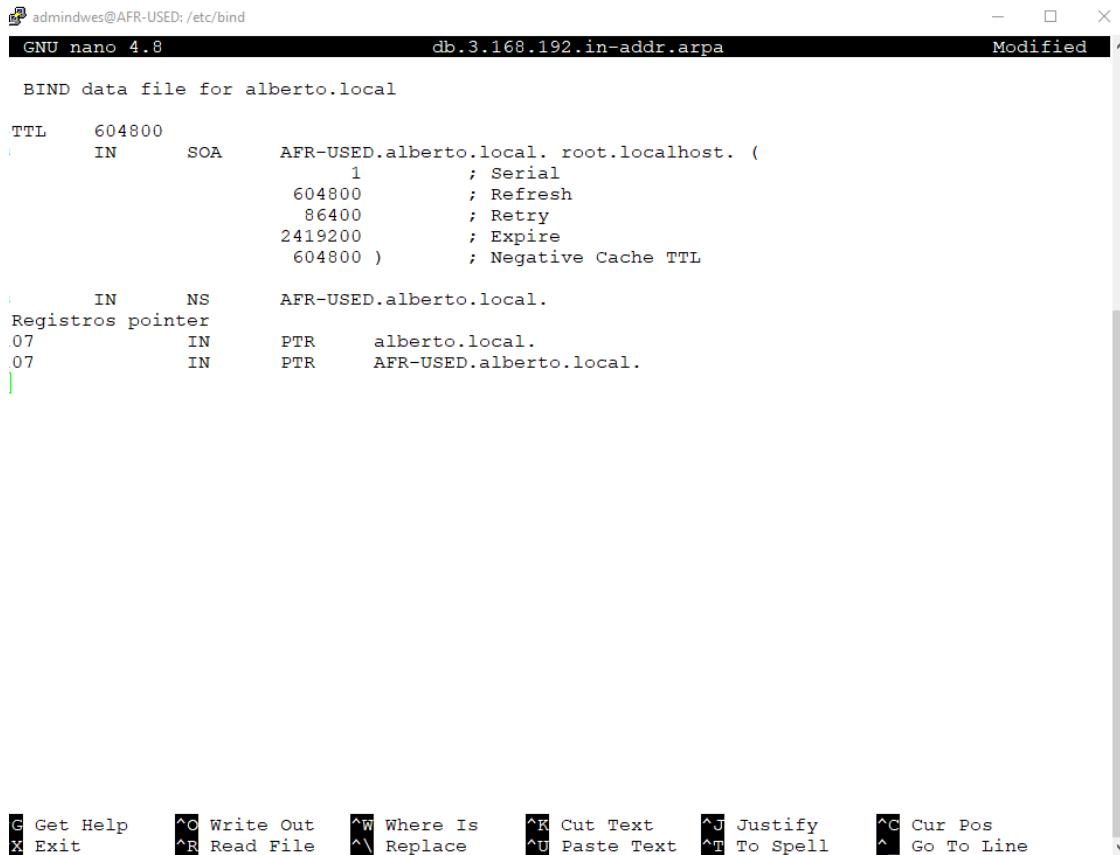
```
admindwes@AFR-US...:/etc/bind$ ls  
bind.keys  db.255           db.local           named.conf.local  zones.rfc1918  
db.0        db.alberto.local  named.conf       named.conf.options  
db.127      db.empty         named.conf.default-zones  rndc.key  
admindwes@AFR-US...:/etc/bind$ sudo cp db.alberto.local db.3.168.192.in-addr.arpa
```

Una vez copiado, comprobamos que está con `ls` y entramos a editarlo con el siguiente comando:

```
adminwes@AFR-US: /etc/bind$ ls
bind.keys    db.255      db.empty      named.conf.default-zones  rndc.key
db.0         db.3.168.192.in-addr.arpa db.local      named.conf.local      zones.rfc1918
db.127       db.alberto.local      named.conf      named.conf.options
adminwes@AFR-US: /etc/bind$ sudo nano db.3.168.192.in-addr.arpa
```

Editamos el fichero, los parámetros personalizables son **nombre.local** y **xxx-used.nombre.local**.

Una vez editado lo guardamos y lo cerramos.



```
adminwes@AFR-US: /etc/bind
GNU nano 4.8                               db.3.168.192.in-addr.arpa                         Modified ^

BIND data file for alberto.local

TTL     604800
IN      SOA      AFR-US.alberto.local. root.localhost. (
                1           ; Serial
                604800        ; Refresh
                86400         ; Retry
                2419200       ; Expire
                604800 )      ; Negative Cache TTL

IN      NS       AFR-US.alberto.local.
Registros pointer
07      IN      PTR      alberto.local.
07      IN      PTR      AFR-US.alberto.local.

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   ^L Go To Line
```

Chequeamos el fichero que contiene la zona arpa con el siguiente comando:

```
adminwes@AFR-US: /etc/bind$ sudo named-checkzone 3.168.192.in-addr.arpa db.3.168.192.in-addr.
arpa
zone 3.168.192.in-addr.arpa/IN: loaded serial 1
OK
adminwes@AFR-US: /etc/bind$
```

Por último, reiniciamos el servicio bind9 con el siguiente comando:

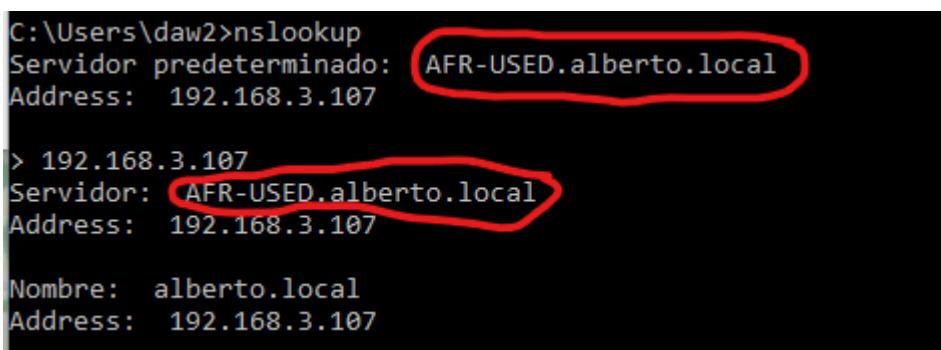
```
adminwes@AFR-US: /etc/bind$ sudo service bind9 restart
```

Mediante el comando *nslookup* podemos hacer consultas de nombres absolutos y relativos.

- En el propio servidor
- En el cliente Windows

Primero comprobaremos en el servidor, entramos a *nslookup* y ponemos la IP de la máquina *192.168.3.107*, seguidamente tendrá que salirnos la configuración del arp.
*admindwes@AFR-US*E: /etc/bind\$ nslookup
> 192.168.3.107
107.3.168.192.in-addr.arpa name = AFR-US.alberto.local.
107.3.168.192.in-addr.arpa name = alberto.local.

Seguidamente lo haremos en Windows 10, abrimos el cmd y ponemos el comando *nslookup*. Una vez dentro ponemos la IP del servidor *192.168.3.107* y nos saldrá el nombre del servidor que hemos configurado anteriormente, nos tiene que coincidir con el predeterminado.



```
C:\Users\daw2>nslookup
Servidor predeterminado: AFR-US.alberto.local
Address: 192.168.3.107

> 192.168.3.107
Servidor: AFR-US.alberto.local
Address: 192.168.3.107

Nombre: alberto.local
Address: 192.168.3.107
```

1.13 Configuración de dos sitios Web en el mismo Servidor

1.13.1 Configuración

Lo primero que haremos será situarnos en el directorio de sites-available con el siguiente comando, y haremos un `ls` para ver el contenido.

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

También nos situaremos en el directorio www con el siguiente comando y haremos un `ls -l` para ver su contenido y los permisos del usuario con los directorios.

```
admindwes@AFR-US...:~$ cd /var/www  
admindwes@AFR-US...:/var/www$ ls  
html  
admindwes@AFR-US...:/var/www$ ls -l  
total 4  
drwxrwsr-x 16 operadorweb www-data 4096 nov 5 12:50 html
```

Empezaremos añadiendo un nuevo grupo llamado ftpusers al cual pertenecerán los nuevos usuarios que necesitaremos, lo crearemos con el siguiente comando:

```
admindwes@AFR-US...:/var/www$ sudo addgroup ftpusers  
Adding group `ftpusers' (GID 1001) ...  
Done.
```

Para ver si se ha creado con éxito comprobaremos con el siguiente comando si se encuentra en group:

```
admindwes@AFR-US...:/var/www$ cat /etc/group |grep ftp  
ftpusers:x:1001:
```

Empezaremos añadiendo dos usuarios nuevos, se llamarán daw201 y daw202, tendrán el directorio home en /var/www/daw201 y /var/www/daw202, ambos usuarios pertenecerán al grupo creado anteriormente que es ftpusers, para crear los usuarios ejecutaremos los siguientes comandos:

```
admindwes@AFR-US...:/var/www$ sudo useradd -m --home /var/www/daw201 -g www-data -G ftpusers daw201  
admindwes@AFR-US...:/var/www$ sudo useradd -m --home /var/www/daw202 -g www-data -G ftpusers daw202
```

Le asignaremos las passwords a ambos usuarios con los siguientes comandos:

```
admindwes@AFR-US...:~$ sudo passwd daw201  
New password:  
Retype new password:  
passwd: password updated successfully  
admindwes@AFR-US...:~$ sudo passwd daw202  
New password:  
Retype new password:  
passwd: password updated successfully
```

Comprobamos que cuando creamos el usuario se crearon las carpetas daw201 y daw202 donde les indicamos, para ello nos colocaremos en www con `cd /var/www` y ejecutaremos el siguiente comando para ver si se han creado correctamente:

```
admindwes@AFR-US:~/var/www$ ls -l
total 12
drwxr-xr-x 2 daw201      www-data 4096 dic  1 11:03 daw201
drwxr-xr-x 2 daw202      www-data 4096 dic  1 11:03 daw202
drwxrwsr-x 16 operadorweb www-data 4096 nov  5 12:50 html
```

Ahora tendremos que crear a cada usuario en su directorio daw201 y daw202 su carpeta de publicación `public_html`, las crearemos con el siguiente comando:

```
admindwes@AFR-US:~/var/www$ sudo mkdir daw201/public_html
admindwes@AFR-US:~/var/www$ sudo mkdir daw202/public_html
```

Ahora cambiaremos el propietario de las nuevas carpetas de publicación(`public_html`) creadas anteriormente, asignaremos a cada una su usuario con los siguientes comandos:

```
admindwes@AFR-US:~$ sudo chown -R daw201:www-data /var/www/daw201/public_html
admindwes@AFR-US:~$ sudo chown -R daw202:www-data /var/www/daw202/public_html
```

Ahora le daremos los permisos al usuario sobre su carpeta de publicación(`public_html`) creadas anteriormente, les daremos los permisos con los siguientes comandos:

```
admindwes@AFR-US:~$ sudo chmod -R 2775 /var/www/daw201/public_html
admindwes@AFR-US:~$ sudo chmod -R 2775 /var/www/daw202/public_html
```

También haremos propietario a root sobre las carpetas de los usuarios con los siguientes comandos:

```
admindwes@AFR-US:~$ sudo chown root:root /var/www/daw201
admindwes@AFR-US:~$ sudo chown root:root /var/www/daw202
```

Le daremos permisos `-w` a ambas carpetas de los usuarios con los siguientes comandos:

```
admindwes@AFR-US:~$ sudo chmod -w /var/www/daw201
admindwes@AFR-US:~$ sudo chmod -w /var/www/daw202
```

Ahora vamos a ver el contenido de la carpeta apache2 con el siguiente comando:

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

Una vez visto el contenido, dentro de esa carpeta tendremos que ver que archivos contiene con un `ls`, para acceder a la carpeta usaremos `cd /etc/apache2/sites-available`

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

Vemos que contiene dos archivos de configuración, haremos una copia con el siguiente comando del archivo **000-default.conf** y le pondremos el nombre **daw201.conf** en este caso:

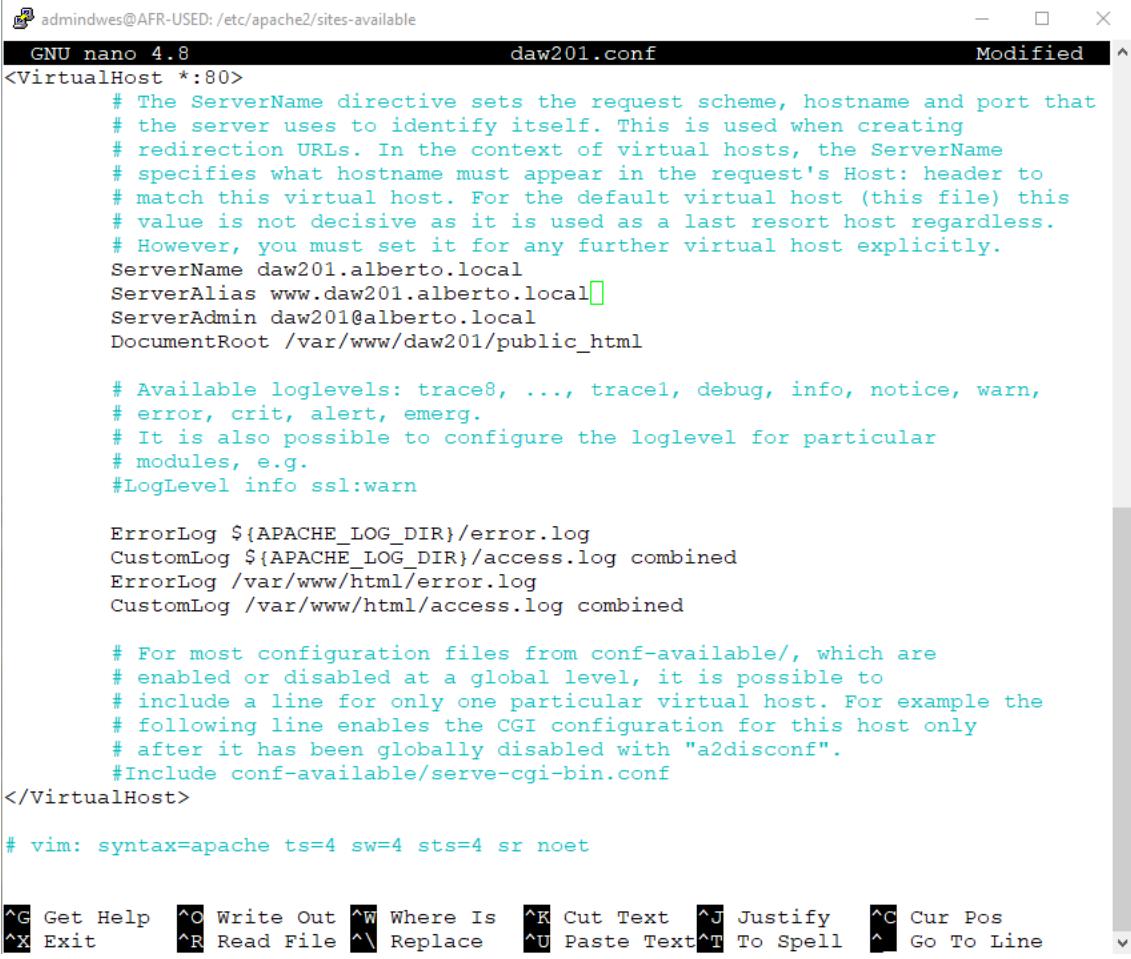
```
adminwes@AFR-US: /etc/apache2/sites-available$ sudo cp 000-default.conf daw201.co
nf
[sudo] password for adminwes:
adminwes@AFR-US: /etc/apache2/sites-available$ ls
000-default.conf  daw201.conf  default-ssl.conf
```

Una vez copiado el archivo pasaremos a editar lo con el siguiente comando:

```
adminwes@AFR-US: /etc/apache2/sites-available$ sudo nano daw201.conf
```

Dentro del archivo, editaremos el **ServerName(daw201.tunombre.local)**, **ServerAlias(www.daw201.tunombre.local)**, **ServerAdmin(daw201@tunombre.local)** y **DocumentRoot(directorio de daw201 public_html)**.

Una vez editado, guardamos y cerramos el archivo.



```
GNU nano 4.8          daw201.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 daw201.alberto.local
    ServerAlias www.daw201.alberto.local
    ServerAdmin daw201@alberto.local
    DocumentRoot /var/www/daw201/public_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

^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
```

Ahora configuraremos el archivo para daw202, para ello haremos una copia del fichero configurado para el 201 con el siguiente comando:

```
adminwes@AFR-US: /etc/apache2/sites-available$ sudo cp daw201.conf daw202.conf
```

Una vez copiado, vemos si se ha realizado correctamente haciendo un `ls` del directorio actual.

```
admindwes@AFR-US ED:/etc/apache2/sites-available$ ls  
000-default.conf daw201.conf daw202.conf default-ssl.conf
```

Entraremos a editar el archivo de configuración de daw202 con el siguiente comando:

```
admindwes@AFR-US ED:/etc/apache2/sites-available$ sudo nano daw202.conf
```

Dentro del archivo, editaremos el `ServerName(daw202.tunombre.local)`, `ServerAlias(www.daw202.tunombre.local)`, `ServerAdmin(daw202@tunombre.local)` y `DocumentRoot(directorio de daw202 public_html)`.

Una vez editado, guardamos y cerramos el archivo.



```
admindwes@AFR-US ED:/etc/apache2/sites-available  
GNU nano 4.8 daw202.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 daw202.alberto.local  
    ServerAlias www.daw202.alberto.local  
    ServerAdmin daw202@alberto.local  
    DocumentRoot /var/www/daw202/public_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  
  
^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 tengamos los dos archivos de configuración editados, tanto daw201 como daw202, ejecutaremos este comando para comprobar que están correctos, si nos dice Syntax OK es que todo ha ido bien:

```
admindwes@AFR-US ED:/etc/apache2/sites-available$ sudo apache2ctl configtest  
Syntax OK
```

Lo siguiente sera activar los sitios web, lo primero que haremos será un `ls` de sites-enabled con el siguiente comando:

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

Una vez hecho el ls, veremos que está activo 000-default.conf ya que sale en color azul claro, habrá que desactivarlo con el siguiente comando:

```
admindwes@AFR-US ED:/etc/apache2$ a2
a2disconf a2dismod a2dissite a2enconf a2enmod a2ensite a2query
admindwes@AFR-US ED:/etc/apache2$ sudo a2dissite 000-default.conf
```

Ahora que esta desactivado el sitio, ya podremos activar los nuevos sitios que hemos creado, tanto el daw201 como el daw202 con los siguientes comandos:

```
admindwes@AFR-US ED:/etc/apache2$ sudo a2ensite daw201
Enabling site daw201.
To activate the new configuration, you need to run:
    systemctl reload apache2
admindwes@AFR-US ED:/etc/apache2$ sudo a2ensite daw202
Enabling site daw202.
To activate the new configuration, you need to run:
    systemctl reload apache2
```

Por último, para que se apliquen los cambios y funcionen correctamente tendremos que reiniciar/recargar el servicio de apache2 con el siguiente comando:

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

1.14 Configuración de un sitio web con HTTPS

1.14.1 Configuración

Lo primero que tendremos que hacer es generar una clave privada RSA, la cual la llamare clave y será de tipo .key, para generarla usaremos el siguiente comando.

```
adminwes@AFR-US:~$ openssl genrsa 2048 > clave.key
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
```

Una vez tenemos la clave privada ahora generaremos la solicitud de la firma del certificado con la clave generada anteriormente, este archivo contendrá la información sobre la organización la cual rellenaremos más adelante.

```
adminwes@AFR-US:~$ openssl req -new -key clave.key > clave.csr
```

Una vez ejecutado el comando, tendremos que llenar los siguientes apartados, nombre de la ciudad, estado o provincia, nombre de la localidad, nombre de la organización nombre de la unidad, nombre de la página web el FQDN y email.

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Zamora
Locality Name (eg, city) []:Benavente
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Educacion
Organizational Unit Name (eg, section) []:Departamento de informatica
Common Name (e.g. server FQDN or YOUR name) []:www.alberto.local
Email Address []:alberto.ferram@educa.jcyl.es
```

También podremos ponerle una password y un nombre de compañía, estos dos apartados son opcionales.

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:paso
An optional company name []:
```

Una vez tenemos el archivo **clave.csr** configurado, generaremos el certificado digital mediante el archivo generado anteriormente con el siguiente comando:

```
adminwes@AFR-US:~$ openssl x509 -req -days 365 -in clave.csr -signkey clave.key > clave.crt
Signature ok
subject=C = ES, ST = Zamora, L = Benavente, O = Educacion, OU = Departamento de informatica, CN =
www.alberto.local, emailAddress = alberto.ferram@educa.jcyl.es
Getting Private key
```

Una vez hechos los pasos anteriores haremos un **ls** para comprobar que tenemos todos los archivos que hemos creado y configurado.

```
adminwes@AFR-US:~$ ls
clave.crt  clave.csr  clave.key
```

El siguiente paso será activar el módulo SSL el cual permite que la web funcione con https, lo activaremos con el siguiente comando:

```
admindwes@AFR-USDED:~$ 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:
Module socache_shmcb already enabled
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

Una vez activado, reiniciaremos el servicio de apache con el siguiente comando:

```
admindwes@AFR-USDED:~$ sudo service apache2 restart
```

Comprobamos con el comando ss -punta que el puerto 443 está escuchando para ver si funciona el módulo SSL.

```
admindwes@AFR-USDED:~$ ss -punta
Netid State Recv-Q Send-Q                               Local Address:Port      Peer Address:Port  Process
udp  UNCONN  0      0                           192.168.3.107:53  0.0.0.0:*
udp  UNCONN  0      0                           127.0.0.1:53   0.0.0.0:*
udp  UNCONN  0      0                           127.0.0.53%lo:53 0.0.0.0:*
udp  UNCONN  0      0                           [::]:53        [::]:*
udp  UNCONN  0      0   [fe80::a00:27ff:fe10:99b4]@enp0s3:53  [::]:*
tcp  LISTEN  0     10                          192.168.3.107:53  0.0.0.0:*
tcp  LISTEN  0     10                          127.0.0.1:53   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     4096                          127.0.0.1:953  0.0.0.0:*
tcp  ESTAB   0      0                           192.168.3.107:22 192.168.3.7:50704
tcp  LISTEN  0     10   [fe80::a00:27ff:fe10:99b4]@enp0s3:53  [::]:*
tcp  LISTEN  0     10                          [::]:53        [::]:*
tcp  LISTEN  0     128                          [::]:22        [::]:*
tcp  LISTEN  0     4096                         [::]:953       [::]:*
tcp  LISTEN  0     511                          *:443          *:*
tcp  LISTEN  0     70                           *:33060        *:*
tcp  LISTEN  0     151                          *:3306        *:*
tcp  LISTEN  0     511                          *:80          *:*
```

También podemos comprobar la configuración actual del módulo SSL con el siguiente comando:

```
admindwes@AFR-USDED:~$ apache2ctl -S
VirtualHost configuration:
*:80           is a NameVirtualHost
               default server daw201.alberto.local (/etc/apache2/sites-enabled/daw201.conf:1)
               port 80 namevhost daw201.alberto.local (/etc/apache2/sites-enabled/daw201.conf:1)
                     alias www.daw201.alberto.local
               port 80 namevhost daw202.alberto.local (/etc/apache2/sites-enabled/daw202.conf:1)
                     alias www.daw202.alberto.local
               port 80 namevhost login.alberto.local (/etc/apache2/sites-enabled/usuariologin.conf:1)
                     alias www.login.alberto.local
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 not_used
Group: name="www-data" id=33 not_used
```

Ahora moveremos la clave privada a la carpeta **/etc/ssl/private** con el siguiente comando:

```
admindwes@AFR-USDED:~$ sudo mv clave.key /etc/ssl/private
```

Una vez movida comprobamos que se encuentra en la carpeta con el siguiente comando:

```
admindwes@AFR-US ED:~$ sudo ls -l /etc/ssl/private
total 8
-rw-rw-r-- 1 admindwes admindwes 1679 dic 21 09:38 clave.key
-rw-r----- 1 root      ssl-cert   1704 oct  5 10:32 ssl-cert-snakeoil.key
```

Cambiaremos el propietario del archivo **clave.key** a root:ssl-cert con el siguiente comando:

```
admindwes@AFR-US ED:~$ sudo chown root:ssl-cert /etc/ssl/private/clave.key
```

Le cambiaremos también los permisos al archivo **clave.key** y le pondremos 640 con el siguiente comando:

```
admindwes@AFR-US ED:~$ sudo chmod 640 /etc/ssl/private/clave.key
```

Ahora moveremos el certificado a la carpeta **/etc/ssl/certs** con el siguiente comando:

```
admindwes@AFR-US ED:~$ sudo mv clave.crt /etc/ssl/certs
```

Cambiaremos el propietario del archivo clave.crt a root:root con el siguiente comando:

```
admindwes@AFR-US ED:~$ sudo chown root:root /etc/ssl/certs/clave.crt
```

Lo siguiente que haremos será configurar el sitio, para ello antes comprobaremos si tenemos el fichero **000-default.conf** en sites-available con el siguiente comando:

```
admindwes@AFR-US ED:~$ cd /etc/apache2/sites-available/
admindwes@AFR-US ED:/etc/apache2/sites-available$ ls
000-default.conf  daw201.conf  daw202.conf  default-ssl.conf  usuariologin.conf
```

Una vez comprobado que el fichero se encuentra en el directorio, haremos una copia y le pondremos de nombre el nombre del sitio seguido de **-ssl.conf** con el siguiente comando:

```
admindwes@AFR-US ED:/etc/apache2/sites-available$ sudo cp default-ssl.conf daw202-ssl.conf
admindwes@AFR-US ED:/etc/apache2/sites-available$ ls
000-default.conf  daw201.conf  daw202.conf  daw202-ssl.conf  default-ssl.conf  usuariologin.conf
```

Una vez copiado el archivo, entramos a configurarlo con el siguiente comando:

```
admindwes@AFR-US ED:/etc/apache2/sites-available$ sudo nano daw202-ssl.conf
```

Lo primero será cambiar la etiqueta de **VirtualHost** y ponerle *:443.

Después tendremos que editar el **ServerName**, **ServerAlias**, **ServerAdmin** y el **DocumentRoot**.

Ahora tendremos que cambiar el nombre del fichero tanto de **ErrorLog** como de **CustomLog**, a ambos les pondremos el nombre actual seguido de -daw202-ssl para poder identificarlos.

Por último, tendremos que configurar el nombre del fichero tanto del **SSLCertificateFile** como el del **SSLCertificateKeyFile** ya que están los de por defecto, pondremos el nombre de nuestros archivos, tanto el del certificado como el de la clave.

```
GNU nano 4.8                                         daw202-ssl.conf                                         Modified
<IfModule mod_ssl.c>
    <VirtualHost *:443>
        ServerName daw202.alberto.local
        ServerAlias www.daw202.alberto.local
        ServerAdmin daw202@alberto.local
        DocumentRoot /var/www/daw202/public_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-daw202-ssl.log
        CustomLog ${APACHE_LOG_DIR}/access-daw202-ssl.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

        # SSL Engine Switch:
        #   Enable/Disable SSL for this virtual host.
        SSLEngine on

        # A self-signed (snakeoil) certificate can be created by installing
        # the ssl-cert package. See
        # /usr/share/doc/apache2/README.Debian.gz for more info.
        # If both key and certificate are stored in the same file, only the
        # SSLCertificateFile directive is needed.
        SSLCertificateFile      /etc/ssl/certs/clave.crt
        SSLCertificateKeyFile  /etc/ssl/private/clave.key

        # Server Certificate Chain:
        # Point SSLCertificateChainFile at a file containing the
        # concatenation of PEM encoded CA certificates which form the
        # certificate chain for the server certificate. Alternatively
        # the referenced file can be the same as SSLCertificateFile
        # when the CA certificates are directly appended to the server

^G Get Help  ^O Write Out  ^W Where Is  ^R Cut Text  ^J Justify  ^C Cur Pos  M-U Undo
^X Exit      ^R Read File  ^L Replace   ^U Paste Text  ^T To Spell  ^S Go To Line M-E Redo
```

Una vez configurado el archivo, tendremos que activar el sitio https con el siguiente comando:

```
adminwes@AFR-US:~/etc/apache2/sites-available$ sudo a2ensite daw202-ssl.conf
Enabling site daw202-ssl.
To activate the new configuration, you need to run:
    systemctl reload apache2
```

Para que se apliquen los cambios tendremos que reiniciar apache con el siguiente comando:

```
adminwes@AFR-US:~/etc/apache2/sites-available$ systemctl reload apache2
===== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to reload 'apache2.service'.
Authenticating as: admin (adminwes)
Password:
===== AUTHENTICATION COMPLETE ===
```

Una vez hecho todo esto, ya estaría configurado el acceso mediante https y ya podremos acceder a nuestro sitio web con <https://daw202.alberto.local>

1.14.2 Redireccionar http a https

Ahora mandaremos todas las consultas hechas a **http** a **https**, para ello nos iremos a el directorio de publicación de nuestra web con el siguiente comando:

```
admindwes@AFR-US ED:/etc/apache2/sites-available$ cd /var/www/daw202/public_html/
```

Crearemos el fichero **.htaccess** si no lo tenemos con el siguiente comando:

```
admindwes@AFR-US ED:/var/www/daw202/public_html$ sudo touch .htaccess
```

Ahora entraremos a editarlo con el comando **sudo nano .htaccess**

Dentro del archivo agregaremos la siguiente información, el **RewriteEngine** activado, el **RewriteCond** del puerto 80 y el **RewriteRule** con el enlace al que hay que hacer la redirección y **/\$1**

```
GNU nano 4.8 .htaccess Modified
RewriteEngine On
RewriteCond %{SERVER_PORT} 80
RewriteRule ^(.*)$ https://www.daw202.alberto.local/$1 [R,L]
```

Para que nos funcione todo lo que hemos introducido en el archivo **.htaccess** necesitamos que este activado el módulo **rewrite**, el cual podremos activar con el siguiente comando:

```
admindwes@AFR-US ED:/etc/apache2$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

Una vez activado reiniciamos el servicio de apache con el siguiente comando:

```
admindwes@AFR-US ED:/var/www/daw202/public_html$ service apache2 restart
===== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to restart 'apache2.service'.
Authenticating as: admin (admindwes)
Password:
===== AUTHENTICATION COMPLETE ===
```

1.14.3 Configurar la negociación de contenidos mediante Multiview

Lo primero que debemos hacer es comprobar si el servidor dispone del mod **negotiation** para que pueda funcionar la configuración de la negociación de contenidos. Entraremos a ver los mods a la carpeta de todos los mods con el siguiente comando:

```
admindwes@AFR-US...:~$ cd /etc/apache2/mods-available/
```

Ahora haremos un **ls** filtrando por **negotiation** para comprobar si se encuentra con el siguiente comando:

```
admindwes@AFR-US...:/etc/apache2/mods-available$ ls -l|grep negotiation
-rw-r--r-- 1 root root 724 jul 5 2021 negotiation.conf
-rw-r--r-- 1 root root 74 jul 5 2021 negotiation.load
```

Una vez que sabemos que está el módulo, configuraremos el multiview, para ello nos iremos a el directorio de publicación de daw202 en mi caso con el siguiente comando:

```
admindwes@AFR-US...:~$ cd /var/www/daw202/public_html
```

Ahora entraremos a configurar el fichero **.htaccess** con el siguiente comando:

```
admindwes@AFR-US...:/var/www/daw202/public_html$ sudo nano .htaccess
```

Dentro del **.htaccess** agregaremos la opción siguiente “**Options +Multiviews**” y guardaremos y cerraremos el archivo.

GNU nano 4.8	.htaccess	Modified
RewriteEngine on		
RewriteCond %{SERVER_PORT} 80		
RewriteRule ^(.*)\$ https://www.daw202.alberto.local/\$1 [R,L]		
Options +MultiViews		

Por último, solo queda añadir los dos index correspondientes a cada idioma en la carpeta de publicación de daw202, en mi caso uno para en inglés “**index.php.en**” y otro para el español “**index.html.es**”.

1. htaccess	130	Archivo HT...	12/01/2022 12:11:17	-rw-r--r-- operadorwe...
2. index.php.en	253	Archivo EN	12/01/2022 12:11:14	-rw-r--r-- operadorwe...
3. index.php.es	255	Archivo ES	12/01/2022 12:11:11	-rw-r--r-- operadorwe...

Una vez hechos todos los pasos, nuestro servidor detectara el idioma del navegador y en base a ese idioma nos mostrara un index u otro.

1.15 Instalación de phpDocumentor

1.15.1 Instalación y configuración con phar

Lo primero que tendremos que hacer es estar situados en nuestro directorio home, y bajarnos el paquete de phpDocumentor de par con el siguiente comando:

```
admindwes@AFR-US:~$ ls
clave.csr
admindwes@AFR-US:~$ sudo wget https://phpdoc.org/phpDocumentor.phar|
```

Comprobaremos que el proceso se ha realizado correctamente haciendo un *ls* y comprobando que esta descargado.

```
|admindwes@AFR-US:~$ ls
clave.csr  phpDocumentor.phar
```

Ahora moveremos el **phar** de phpDocumentor que descargamos al directorio **/usr/local/bin/phpdoc** con el siguiente comando:

```
admindwes@AFR-US:~$ sudo mv phpDocumentor.phar /usr/local/bin/phpdoc
```

Ahora haremos que los archivos que están dentro de **/phpdoc** sean ejecutables con el siguiente comando:

```
|admindwes@AFR-US:~$ sudo chmod +x /usr/local/bin/phpdoc
```

Por último, podemos comprobar la versión de phpDocumentor con el siguiente comando y así garantizar que esta bien realizado el proceso de instalación.

```
admindwes@AFR-US:~$ phpdoc --version
phpDocumentor v3.3.0
```

1.15.2 Ejemplo de uso phar

Para generar la documentación con phpDocumentor tendremos que usar el siguiente comando (la primera ruta será la ruta donde se encuentra el proyecto a documentar, la segunda ruta, será la carpeta donde queremos almacenar la documentación generada, en mi caso en **/doc**):

```
adminwes@AFR-USED:~$ sudo phpdoc -d /var/www/html/207DWESLoginLogoutMulticapaPOO -t /var/www/html/207DWESLoginLogoutMulticapaPOO/doc
```

Si todo ha ido correctamente, se nos mostrara el mensaje **All done** después de generar la documentación.

phpDocumentor v3.3.0

```
Parsing files
35/35 [=====] 100%
Applying transformations (can take a while)
21/21 [=====] 100%
All done in 2 seconds!
```

Ahora ya podremos acceder al index de nuestra documentación generada con phpDocumentor. Dicha documentación se encuentra en la carpeta doc de nuestra aplicación.

The screenshot shows a web browser window displaying the generated documentation. The address bar shows the URL: `192.168.3.107/207DWESLoginLogoutMulticapaPOO/doc/index.html`. The page title is "Documentation". On the left, there is a sidebar with navigation links: "Packages" (Application, Validacion), "Reports" (Deprecated, Errors, Markers), and "Indices" (Files). The main content area has a search bar at the top right with the placeholder "Search (Press "/" to focus)". Below the search bar, the title "Documentation" is displayed. Under "Packages", there are two entries: "Application" and "Validacion", both marked with a green "P" icon. Below this, under "Interfaces, Classes, Traits and Enums", there is a list of classes: "DB", "UsuarioDB", "validacionFormularios", "AppError", "DBPDO", "Usuario", and "UsuarioPDO", each preceded by a green "C" icon. At the bottom of the main content area, there is a "Table of Contents" section. A small green arrow icon is located in the bottom right corner of the page.

1.16 Autenticación y autorización en aplicación

1.16.1 Autenticación básica mediante usuarios

Lo primero será comprobar que tenemos los módulos necesarios para que funcione la autenticación, los necesarios serán **authz** y **auth_basic**, lo comprobaremos con el siguiente comando:

```
admindwes@AFR-USDED:~$ cd /etc/apache2/
admindwes@AFR-USDED:/etc/apache2$ ls mods-enabled/*
mods-enabled/access_compat.load    mods-enabled/mime.load
mods-enabled/alias.conf            mods-enabled/mpm_prefork.conf
mods-enabled/alias.load           mods-enabled/mpm_prefork.load
mods-enabled/auth_basic.load      mods-enabled/negotiation.conf
mods-enabled/authn_core.load     mods-enabled/negotiation.load
mods-enabled/authn_file.load     mods-enabled/php7.4.conf
mods-enabled/authz_core.load     mods-enabled/php7.4.load
mods-enabled/authz_host.load     mods-enabled/reqtimeout.conf
mods-enabled/authz_user.load     mods-enabled/reqtimeout.load
mods-enabled/autoindex.conf      mods-enabled/rewrite.load
mods-enabled/autoindex.load      mods-enabled/setenvif.conf
mods-enabled/deflate.conf        mods-enabled/setenvif.load
mods-enabled/deflate.load        mods-enabled/socache_shmcb.load
mods-enabled/dir.conf            mods-enabled/ssl.conf
mods-enabled/dir.load           mods-enabled/ssl.load
mods-enabled/env.load            mods-enabled/status.conf
mods-enabled/filter.load         mods-enabled/status.load
mods-enabled/mime.conf
```

Lo primero que haremos será crear dos usuarios, **profesor1** y **profesor2** con los siguientes comandos:

```
admindwes@AFR-USDED:/etc/apache2$ sudo htpasswd -c /etc/apache2/passwd profesor1
New password:
Re-type new password:
Adding password for user profesor1
admindwes@AFR-USDED:/etc/apache2$ sudo htpasswd /etc/apache2/passwd profesor2
New password:
Re-type new password:
Adding password for user profesor2
```

Una vez tenemos los usuarios creados, nos situaremos en el directorio de publicación de daw202 y crearemos una carpeta llamada profesor con los siguientes comandos:

```
admindwes@AFR-USDED:/etc/apache2$ cd /var/www/daw202/public_html/
admindwes@AFR-USDED:/var/www/daw202/public_html$ sudo mkdir profesor
```

Una vez tenemos la carpeta, entramos en ella con **cd profesor** creamos dos archivos de texto, file1 y file2 con los siguientes comandos:

```
admindwes@AFR-USDED:/var/www/daw202/public_html/profesor$ sudo touch file1.txt
admindwes@AFR-USDED:/var/www/daw202/public_html/profesor$ sudo touch file2.txt
```

Una vez hecho lo anterior, nos iremos a el apartado de apache en sitios activos con **cd /etc/apache2/sites-enabled** y mostraremos los que están con el siguiente comando:

```
admindwes@AFR-USDED:/etc/apache2/sites-enabled$ ls
000-default.conf.backup  daw201.conf  daw202.conf
```

Entraremos a editar daw202 con el siguiente comando:

```
admindwes@AFR-USDED:/etc/apache2/sites-enabled$ sudo nano daw202.conf
```

Una vez dentro, tendremos que crear la etiqueta <Directory directorioPublicacion/profesor>

Dentro de la etiqueta directory tendremos que introducir la configuración del sitio que será la siguiente:

Options Indexes FollowSymLinks

AuthType Basic

AuthName "Acceso restringido"

AuthUserFile /etc/apache2/passwd

Require user profesor1 profesor2 <= **USUARIOS PARA INICIAR SESIÓN**

No olvidar cerrar la etiqueta </Directory>

```
GNU nano 4.8                               daw202.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 daw202.alberto.local
    ServerAlias www.daw202.alberto.local
    ServerAdmin daw202@alberto.local
    DocumentRoot /var/www/daw202/public_html
    <Directory /var/www/daw202/public_html/profesor>
        Options Indexes FollowSymLinks
        AuthType Basic
        AuthName "Acceso restringido"
        AuthUserFile /etc/apache2/passwd
        Require user profesor1 profesor2
    </Directory>

    # 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
```

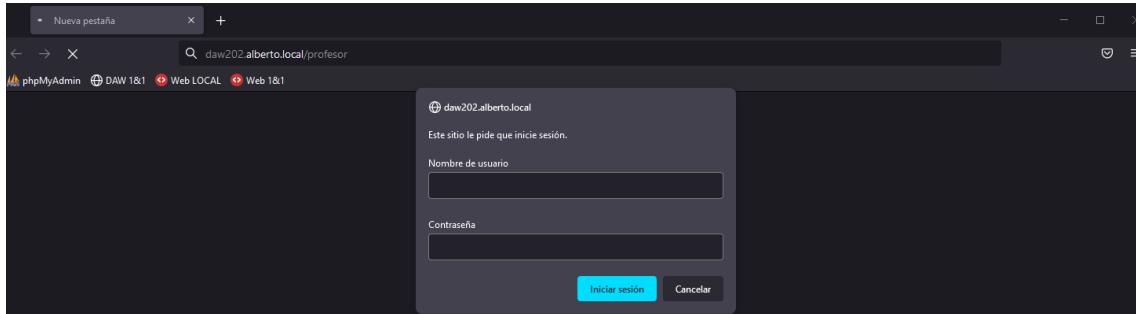
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^ Go To Line

Guardamos una vez este todo configurado y cerramos.

Para que se apliquen los cambios habrá que reiniciar el servicio de apache2

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

Una vez reiniciado ya podremos comprobar si nos funciona la autenticación en el sitio que está configurada.



1.16.2 Autenticación Digest mediante usuarios

La autenticación Digest es uno de los métodos mas usados en servidores para autenticarse.

Lo primero que tendremos que hacer será activar el módulo **digest** con el siguiente comando:

```
adminwes@AFR-USDED:/etc/apache2$ sudo a2enmod auth_digest
```

Una vez este activado, reiniciaremos apache con el siguiente comando:

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

Ahora crearemos dos usuarios, **ninja1** y **ninja2**, los tendremos que crear con el siguiente comando(cada usuario estará en el dominio **dominio-ninja**):

```
adminwes@AFR-USDED:/etc/apache2$ sudo htdigest -c /etc/apache2/.digest dominio-ninja
ninja1
Adding password for ninja1 in realm dominio-ninja.
New password:
Re-type new password:
adminwes@AFR-USDED:/etc/apache2$ sudo htdigest /etc/apache2/.digest dominio-ninja nin
ja2
Adding user ninja2 in realm dominio-ninja
New password:
Re-type new password:
```

Una vez tenemos los usuarios creados, nos situaremos en el directorio de publicación de daw202 y crearemos una carpeta llamada **ninja** con el siguiente comando:

```
adminwes@AFR-USDED:/var/www/daw202/public_html$ sudo mkdir ninja
```

Una vez hecho lo anterior, nos iremos a el apartado de apache en sitios activos con **cd /etc/apache2/sites-enabled** y mostraremos los que están con el siguiente comando:

```
adminwes@AFR-USDED:~$ cd /etc/apache2/sites-enabled/
adminwes@AFR-USDED:/etc/apache2/sites-enabled$ ls
000-default.conf.backup  daw201.conf  daw202.conf
```

Entraremos a editar daw202 con el siguiente comando:

```
adminwes@AFR-USDED:/etc/apache2/sites-enabled$ sudo nano daw202.conf
```

Una vez dentro, tendremos que crear otra etiqueta **<Directory directorioPublicacion/ninja>**

Dentro de la etiqueta **directory** tendremos que introducir la configuración del sitio que será la siguiente:

Options Indexes FollowSymLinks

AuthType Digest

AuthName "dominio-ninja"

AuthDigestProvider file

AuthUserFile /etc/apache2/.digest

Require valid-user

No olvidar cerrar la etiqueta </Directory>

```
GNU nano 4.8                               daw202.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 daw202.alberto.local
    ServerAlias www.daw202.alberto.local
    ServerAdmin daw202@alberto.local
    DocumentRoot /var/www/daw202/public_html
    <Directory /var/www/daw202/public_html/profesor>
        Options Indexes FollowSymLinks
        AuthType Basic
        AuthName "Acceso restringido"
        AuthUserFile /etc/apache2/passwd
        Require user profesor1 profesor2
    </Directory>
    <Directory /var/www/daw202/public_html/ninja>
        Options Indexes FollowSymLinks
        AuthType Digest
        AuthName "dominio-ninja"
        AuthDigestProvider file
        AuthUserFile /etc/apache2/.digest
        Require valid-user
    </Directory>

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
                                                [ Read 48 lines ]
^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
```

Chequearemos que el fichero esta bien configurado con el siguiente comando:

```
admindwes@AFR-US:~/etc/apache2/sites-enabled$ sudo apachectl configtest
```

Por último, reiniciaremos el servicio de apache2 con el siguiente comando y ya podremos probar la autenticación:

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

1.16.3 Autenticación Digest mediante grupos

Para realizar la autenticación con Digest mediante grupos necesitaremos instalar el mod **authz_groupfile** con el siguiente comando:

```
admindwes@AFR-USDED:/etc/apache2/sites-enabled$ sudo a2enmod authz_groupfile
```

Una vez instalado reiniciaremos apache2 para aplicar los cambios con el siguiente comando:

```
admindwes@AFR-USDED:/etc/apache2/sites-enabled$ systemctl restart apache2
```

Tendremos que crear un archivo .grupos dentro de apache2 con el siguiente comando para poder editarla a la vez que lo creamos:

```
admindwes@AFR-USDED:/etc/apache2$ sudo nano .grupos
```

Dentro introduciremos la siguiente información, en este caso un grupo1 con el usuario ninja1 y el grupo2 con el usuario ninja2.

```
GNU nano 4.8 .grupos Modified
grupo1: ninja1
grupo2: ninja2
```

Entraremos a editar daw202 con el siguiente comando:

```
admindwes@AFR-USDED:/etc/apache2/sites-enabled$ sudo nano daw202.conf
```

Una vez dentro, tendremos que crear otra etiqueta <Directory directorioPublicacion/ninja>

Dentro de la etiqueta directory tendremos que introducir la configuración del sitio que será la siguiente:

Options Indexes FollowSymLinks

AuthType Digest

AuthName "dominio-ninja"

AuthDigestProvider file

AuthUserFile /etc/apache2/.digest

AuthGroupFile /etc/apache2/.grupos

Require group grupo2 <= AUTORIZAR SOLO A USUARIOS DEL GRUPO2

No olvidar cerrar la etiqueta </Directory>

```
GNU nano 4.8                               daw202.conf                         Modified ^

# 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 daw202.alberto.local
ServerAlias www.daw202.alberto.local
ServerAdmin daw202@alberto.local
DocumentRoot /var/www/daw202/public_html
<Directory /var/www/daw202/public_html/profesor>
    Options Indexes FollowSymLinks
    AuthType Basic
    AuthName "Acceso restringido"
    AuthUserFile /etc/apache2/passwd
    Require user profesor1 profesor2
</Directory>
<Directory /var/www/daw202/public_html/ninja>
    Options Indexes FollowSymLinks
    AuthType Digest
    AuthName "dominio-ninja"
    AuthDigestProvider file
    AuthUserFile /etc/apache2/.digest
    AuthGroupFile /etc/apache2/.grupos
    Require group grupo2
</Directory>

# 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

`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   ^ Up Go To Line  v
```

Chequearemos que el fichero está bien configurado con el siguiente comando:

```
admindwes@AFR-US:~/etc/apache2/sites-enabled$ sudo apachectl configtest
Syntax OK
```

Por último, reiniciaremos el servicio de apache2 con el siguiente comando y ya podremos probar la autenticación:

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

1.17 Uso de comandos en Git

1.17.1 Copia de archivos en una rama de un repositorio de GitHub

Lo primero que haremos será clonar el repositorio que vamos a usar, con el comando *git clone enlace/repositorio*.

Una vez tenemos clonado el repositorio, haremos un *cd nombre* para colocarnos dentro de él.

```
admindwes@AFR-USDED:~$ cd DAWDespliegueAplicacionesWeb/
```

Revisaremos en que rama nos encontramos con *git Branch*, una vez comprobada crearemos nuestra propia rama con *git branch nombrerama*. Nos colocaremos dentro de la rama con *git checkout nombrerama*. Volvemos a hacer *git branch* para comprobar que estamos dentro de nuestra rama creada.

```
admindwes@AFR-USDED:~/DAWDespliegueAplicacionesWeb$ git branch
* master
admindwes@AFR-USDED:~/DAWDespliegueAplicacionesWeb$ git branch ramaAlberto
admindwes@AFR-USDED:~/DAWDespliegueAplicacionesWeb$ git branch
* master
    ramaAlberto
admindwes@AFR-USDED:~/DAWDespliegueAplicacionesWeb$ git checkout ramaAlberto
Switched to branch 'ramaAlberto'
admindwes@AFR-USDED:~/DAWDespliegueAplicacionesWeb$ git branch
    master
* ramaAlberto
```

Una vez tenemos la rama creada, entramos en bind con *cd bind* para borrar el contenido que tiene y colocar el nuestro.

```
admindwes@AFR-USDED:~/DAWDespliegueAplicacionesWeb$ cd bind
admindwes@AFR-USDED:~/DAWDespliegueAplicacionesWeb/bind$ ls
db.0.168.192.in-addr.arpa db.serverus.local named.conf.local
```

Una vez estamos dentro de bind, borraremos todo el contenido de la carpeta antes de colocar el nuestro con el siguiente comando:

```
admindwes@AFR-USDED:~/DAWDespliegueAplicacionesWeb/bind$ sudo rm -r *
[sudo] password for admindwes:
admindwes@AFR-USDED:~/DAWDespliegueAplicacionesWeb/bind$ ls
```

Haremos un *ls* para comprobar que la carpeta bind esta vacía.

```
admindwes@AFR-USDED:~/DAWDespliegueAplicacionesWeb/bind$ ls
```

Una vez vemos que esta vacía, copiaremos dentro de la carpeta de nuestra rama nueva los archivos que queremos guardar con los siguientes comandos:

```
admindwes@AFR-USDED:~/DAWDespliegueAplicacionesWeb/bind$ sudo cp /etc/bind/db.alberto.lo
cal .
admindwes@AFR-USDED:~/DAWDespliegueAplicacionesWeb/bind$ sudo cp /etc/bind/db.3.168.192.
in-addr.arpa .
admindwes@AFR-USDED:~/DAWDespliegueAplicacionesWeb/bind$ sudo cp /etc/bind/named.conf.lo
cal .
admindwes@AFR-USDED:~/DAWDespliegueAplicacionesWeb/bind$ ls
db.3.168.192.in-addr.arpa db.alberto.local named.conf.local
```

Comprobamos con *ls* que se han copiado correctamente.

```
admindwes@AFR-USDED:~/DAWDespliegueAplicacionesWeb/bind$ ls
db.3.168.192.in-addr.arpa db.alberto.local named.conf.local
```

Una vez están los archivos nuevos dentro de bind, haremos un *git add* para preparar los archivos para el Commit, vemos con *git status* los cambios a hacer Commit.

```
admindwes@AFR-US ED:~/DAWDespliegueAplicacionesWeb/bind$ git add .
admindwes@AFR-US ED:~/DAWDespliegueAplicacionesWeb/bind$ git status
On branch ramaAlberto
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    db.0.168.192.in-addr.arpa
    new file:   db.3.168.192.in-addr.arpa
    new file:   db.alberto.local
    deleted:   db.serverus.local
    modified:   named.conf.local
```

Lo siguiente será hacer el Commit con *git commit -m “mensajedelcommit”*

```
admindwes@AFR-US ED:~/DAWDespliegueAplicacionesWeb/bind$ git commit -m "03/02/2022 Actualización ficheros bind"
[ramaAlberto 78214ce] 03/02/2022 Actualización ficheros bind
 5 files changed, 53 insertions(+), 42 deletions(-)
 delete mode 100644 bind/db.0.168.192.in-addr.arpa
 create mode 100644 bind/db.3.168.192.in-addr.arpa
 create mode 100644 bind/db.alberto.local
 delete mode 100644 bind/db.serverus.local
```

Por último, haremos el push con *git push origin nombrerama* para subir los cambios al repositorio.

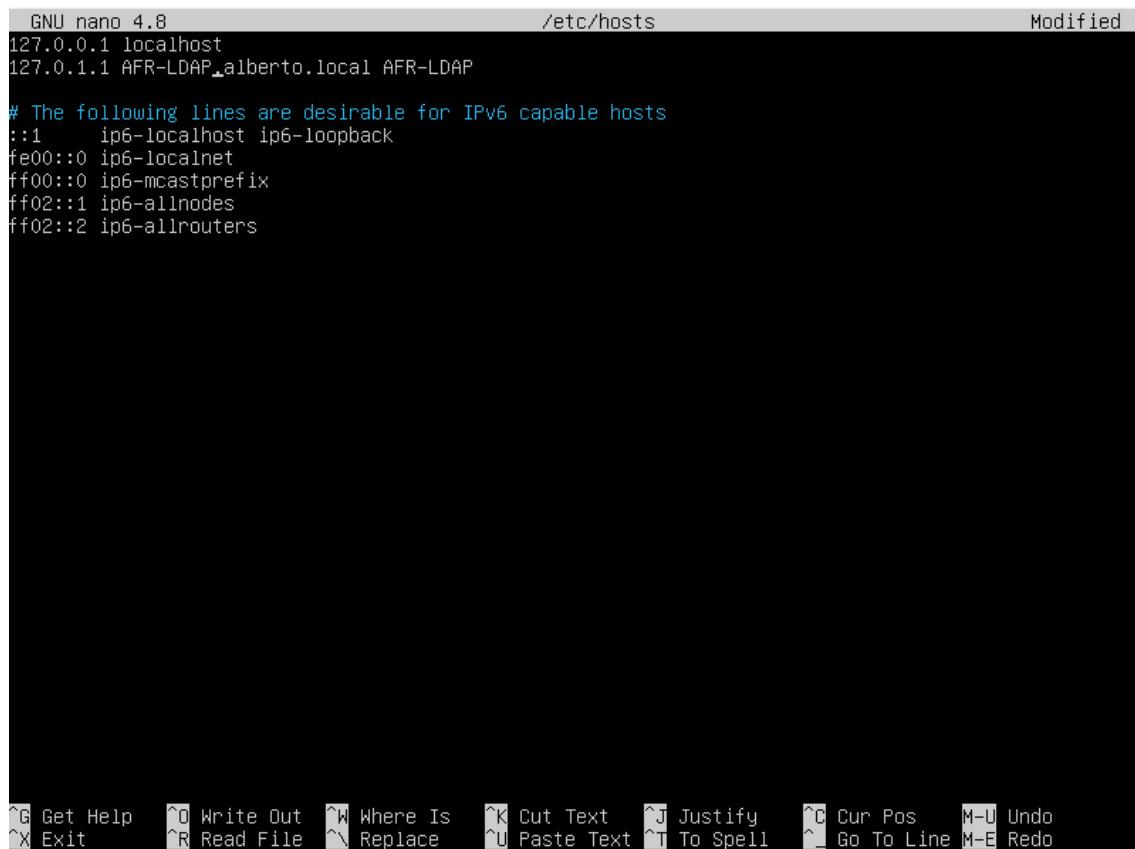
```
admindwes@AFR-US ED:~/DAWDespliegueAplicacionesWeb/bind$ git push origin ramaAlberto
Username for 'https://github.com': AlbertoFRSauces
Password for 'https://AlbertoFRSauces@github.com':
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 865 bytes | 865.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
remote:
remote: Create a pull request for 'ramaAlberto' on GitHub by visiting:
remote:   https://github.com/IESSAUCES/DAWDespliegueAplicacionesWeb/pull/new/ramaAlberto
remote:
To https://github.com/IESSAUCES/DAWDespliegueAplicacionesWeb.git
 * [new branch]      ramaAlberto -> ramaAlberto
```

1.18 Instalación y configuración del servicio LDAP

1.18.1 Configuración de las máquinas virtuales

Para configurar LDAP necesitaremos dos máquinas virtuales, la que ya teníamos que se llamaba **XXX-US**ED y la nueva que será **XXX-LDAP**, clonaremos una maquina limpia para hacer la de LDAP.

Una vez la tenemos la maquina **XXX-LDAP** clonada, editaremos el fichero **hosts** con el comando *sudo nano /etc/host*, dentro introduciremos **XXX-LDAP.nombre.local** **XXX-LDAP**, guardamos y cerramos el archivo.



```
GNU nano 4.8          /etc/hosts          Modified
127.0.0.1 localhost
127.0.1.1 AFR-LDAP alberto.local AFR-LDAP

# 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

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  M-U Undo
^X Exit     ^R Read File  ^M Replace  ^U Paste Text  ^T To Spell  ^L Go To Line M-E Redo
```

Ahora le cambiaremos el nombre a la maquina con el siguiente comando:

```
admindwes@AFR-US:~$ sudo hostnamectl set-hostname AFR-LDAP
```

Ahora accedemos con *cd /etc/netplan* a netplan y configuraremos la red de la maquina editando el fichero de configuración **01-netcfg.yaml** con el siguiente comando:

```
admindwes@AFR-LDAP:/etc/netplan$ sudo nano 01-netcfg.yaml
```

Le cambiaremos la ip a 192.168.3.20X y le pondremos el search: [nombre.local]

```
GNU nano 4.8                               01-netcfg.yaml                                Modified
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.3.207/24
      gateway4: 192.168.3.1
      nameservers:
        addresses: []
        - 8.8.8.8
      search: [alberto.local]
```

Una vez guardada la configuración de red, la aplicaremos con el siguiente comando:

```
admindwes@AFR-LDAP:/etc/netplan$ sudo netplan apply
```

Ahora reiniciaremos la máquina virtual con *reboot* para aplicar todos los cambios.

Una vez reiniciamos, comprobaremos que hay conectividad entre las dos máquinas y a internet con los siguientes comandos:

```
admindwes@AFR-USDED:~$ ping 192.168.3.207
PING 192.168.3.207 (192.168.3.207) 56(84) bytes of data.
64 bytes from 192.168.3.207: icmp_seq=1 ttl=64 time=0.444 ms
64 bytes from 192.168.3.207: icmp_seq=2 ttl=64 time=0.921 ms
64 bytes from 192.168.3.207: icmp_seq=3 ttl=64 time=1.11 ms
^Z
[1]+  Stopped                  ping 192.168.3.207
admindwes@AFR-USDED:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=112 time=10.6 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=112 time=10.4 ms
^Z
[2]+  Stopped                  ping 8.8.8.8
admindwes@AFR-LDAP:~$ ping 192.168.3.107
PING 192.168.3.107 (192.168.3.107) 56(84) bytes of data.
64 bytes from 192.168.3.107: icmp_seq=1 ttl=64 time=0.618 ms
64 bytes from 192.168.3.107: icmp_seq=2 ttl=64 time=0.889 ms
64 bytes from 192.168.3.107: icmp_seq=3 ttl=64 time=0.900 ms
^Z
[1]+  Stopped                  ping 192.168.3.107
admindwes@AFR-LDAP:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=112 time=10.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=112 time=11.0 ms
^Z
[2]+  Stopped                  ping 8.8.8.8
```

1.18.2 Instalación y configuración

1.18.2.1 bind9

Lo primero que necesitaremos sera configurar bind9, para ahorrarnos trabajo, clonare mi rama de GitHub en la cual ya tengo una copia de la configuracion de esos archivos y poder agilizar el proceso.

```
adminwes@AFR-LDAP:~$ git clone --branch ramaAlberto https://github.com/IESSAUCE/DWDespliegueAplicacionesWeb.git
Cloning into 'DAWDespliegueAplicacionesWeb'...
remote: Enumerating objects: 336, done.
remote: Counting objects: 100% (192/192), done.
remote: Compressing objects: 100% (117/117), done.
remote: Total 336 (delta 99), reused 139 (delta 63), pack-reused 144
Receiving objects: 100% (336/336), 51.91 KiB | 738.00 KiB/s, done.
Resolving deltas: 100% (141/141), done.
```

Antes de empezar, realizaremos los siguientes comandos, *sudo apt remove bind9*, *sudo apt clean bind9*y *sudo apt purge bind9* para asegurarnos de realizar una instalacion correcta.

```
adminwes@AFR-LDAP:~$ sudo apt remove bind9
[sudo] password for adminwes:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
El paquete «bind9» no está instalado, no se eliminará
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
adminwes@AFR-LDAP:~$ sudo apt clean bind9
adminwes@AFR-LDAP:~$ sudo apt purge bind9
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
El paquete «bind9» no está instalado, no se eliminará
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

Ahora procederemos a instalar bind9 con el siguiente comando:

```
adminwes@AFR-LDAP:~$ sudo apt install bind9
```

Una vez instalado accederemos a la carpeta de bind con el siguiente comando:

```
adminwes@AFR-LDAP:~$ cd /etc/bind
```

Como anteriormente baje los archivos preconfigurados de mi repositorio de GitHub, hare una copia de ellos a la carpeta de bind9. En caso de no hacerlo con archivos existentes, habra que configurarlos por completo los que ya se encuentran en el directorio bind9.

```
adminwes@AFR-LDAP:/etc/bind$ sudo cp /home/adminwes/DAWDespliegueAplicacionesWeb/bind/* .
```

Mostraremos que se han copiado con éxito los archivos con el siguiente comando (db.tunombre.local, db.3.168.192.in-addr.arpa):

```
adminwes@AFR-LDAP:/etc/bind$ ls
bind.keys    db.3.168.192.in-addr.arpa    named.conf          rndc.key
db.0          db.alberto.local           named.conf.default-zones zones.rfc1918
db.127        db.empty                 named.conf.local
db.255        db.local                named.conf.options
```

Editaremos el archivo db.tunombre.local con el siguiente comando:

```
adminwes@AFR-LDAP:/etc/bind$ sudo nano db.alberto.local
```

Dentro del archivo haremos los cambios de todo lo que esta subrayado en rojo, debera estar en XXX-LDAP y la ip 20X

```
GNU nano 4.8                               db.alberto.local                         Modified
;
; BIND data file for alberto.local
;
$TTL    604800
@       IN      SOA     AFR-LDAP.alberto.local. root.localhost. (
                          1           ; Serial
                          604800      ; Refresh
                          86400       ; Retry
                          2419200     ; Expire
                          604800 )     ; Negative Cache TTL
;
@       IN      NS      AFR-LDAP.alberto.local.
;Registros Host
@       IN      A       192.168.3.207
AFR-USED   IN      A       192.168.3.107
AFR-LDAP   IN      A       192.168.3.207

;Registros Alias
daw201    IN      CNAME   AFR-USED.alberto.local.
daw202    IN      CNAME   AFR-USED.alberto.local.
daw203    IN      CNAME   AFR-USED.alberto.local.
daw204    IN      CNAME   AFR-USED.alberto.local.
www.daw201 IN      CNAME   AFR-USED.alberto.local.
www.daw202 IN      CNAME   AFR-USED.alberto.local.
www.daw203 IN      CNAME   AFR-USED.alberto.local.
www.daw204 IN      CNAME   AFR-USED.alberto.local.
login     IN      CNAME   AFR-USED.alberto.local.
```

Una vez hechos los cambios, guardamos el archivo y lo cerramos.

Para comprobar que el archivo esta correcto, usaremos el siguiente comando, si nos dice OK es que esta todo bien:

```
admindwes@AFR-LDAP:/etc/bind$ sudo named-checkzone alberto.local db.alberto.local
zone alberto.local/IN: loaded serial 1
OK
```

Ahora pasaremos a configurar el archivo db.3.168.192.in-addr.arpa con el siguiente comando:

```
admindwes@AFR-LDAP:/etc/bind$ sudo nano db.3.168.192.in-addr.arpa
```

Dentro del archivo haremos los cambios de todo lo que esta subrayado en rojo, debera estar en XXX-LDAP y la ip 20X

```
GNU nano 4.8                               db.3.168.192.in-addr.arpa                         Modified
;
; BIND data file for alberto.local
;
$TTL    604800
@       IN      SOA     AFR-LDAP.alberto.local. root.localhost. (
                          1           ; Serial
                          604800      ; Refresh
                          86400       ; Retry
                          2419200     ; Expire
                          604800 )     ; Negative Cache TTL
;
@       IN      NS      AFR-LDAP.alberto.local.
;Registros pointer
107     IN      PTR     alberto.local.
107     IN      PTR     AFR-USED.alberto.local.
207     IN      PTR     AFR-LDAP.alberto.local.
```

Para comprobar que el archivo esta correcto, usaremos el siguiente comando, si nos dice OK es que esta todo bien:

```
adminwes@AFR-LDAP:/etc/bind$ sudo named-checkzone 3.168.192.in-addr.arpa db.3.168.192.i
n-addr.arpa
zone 3.168.192.in-addr.arpa/IN: loaded serial 1
OK
```

Por ultimo reiniciaremos el servicio bind9 para que se apliquen los cambios con el siguiente comando:

```
|adminwes@AFR-LDAP:/etc/bind$ sudo service bind9 restart
```

Para comprobar que la configuracion realizada del servicio bind9 esta funcionando correctamente, tendremos que usar en comando *nslookup* hacer peticiones a los siguientes nombres o IPs:

```
adminwes@AFR-LDAP:/etc/bind$ nslookup
> server 192.168.3.207
Default server: 192.168.3.207
Address: 192.168.3.207#53
> AFR-LDAP
Server:          192.168.3.207
Address:         192.168.3.207#53

Name:   AFR-LDAP.alberto.local
Address: 192.168.3.207
> DAW201
Server:          192.168.3.207
Address:         192.168.3.207#53

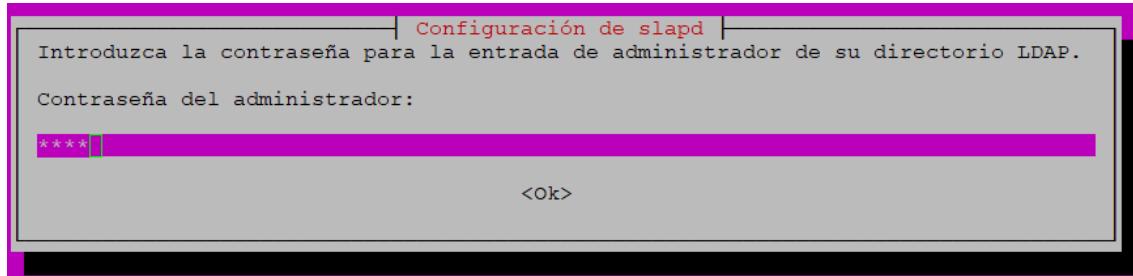
daw201.alberto.local      canonical name = AFR-USED.alberto.local.
Name:   AFR-USED.alberto.local
Address: 192.168.3.107
> 192.168.3.207
207.3.168.192.in-addr.arpa      name = AFR-LDAP.alberto.local.
```

1.18.2.2 LDAP

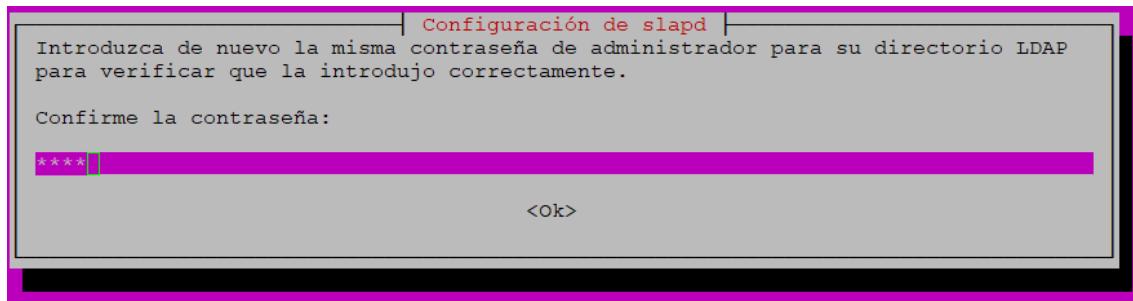
Una vez tenemos bind9 instalado y configurado, pasamos a configurar y instalar el LDAP, con el siguiente comando iniciaremos el primer proceso:

```
admindwes@AFR-LDAP:~$ sudo apt install slapd ldap-utils -y
```

Se nos iniciara el proceso de configuracion de slapd, indicaremos la contraseña para el administrador.



Confirmaremos la contraseña de nuevo.



Comprobaremos con el siguiente comando la version de ldap:

```
admindwes@AFR-LDAP:~$ slapd -V
@(#) $OpenLDAP: slapd (Ubuntu) (Apr  8 2021 04:22:01) $
      Debian OpenLDAP Maintainers <pkg-openldap-devel@lists.alioth.debian.org>
```

Tambien podemos comprobar el **puerto de Idap** con el comando ***ss -puta***, primero vemos el id de Idap:

```
admindwes@AFR-LDAP:~$ ss -puta
Netid State  Recv-Q Send-Q          Local Address:Port      Peer Address:Port
Process
udp   UNCONN 0      0              192.168.3.207:domain  0.0.0.0:*
udp   UNCONN 0      0              127.0.0.1:domain     0.0.0.0:*
udp   UNCONN 0      0              127.0.0.53%lo:domain 0.0.0.0:*
udp   UNCONN 0      0              [::1]:domain        [::]:*
udp   UNCONN 0      0      [fe80::a00:27ff:fe45:c1d6]@enp0s3:domain  [::]:*
tcp   LISTEN 0     1024            0.0.0.0:ldap        0.0.0.0:*
tcp   LISTEN 0     10              192.168.3.207:domain 0.0.0.0:*
tcp   LISTEN 0     10              127.0.0.1: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     4096            127.0.0.1:953       0.0.0.0:*
tcp   ESTAB  0      0              192.168.3.207:ssh    192.168.3.7:52248
tcp   LISTEN 0     1024            [::]:ldap          [::]:*
tcp   LISTEN 0     511             *:http             *:*
tcp   LISTEN 0     10      [fe80::a00:27ff:fe45:c1d6]@enp0s3:domain  [::]:*
tcp   LISTEN 0     10              [::1]:domain        [::]:*
tcp   LISTEN 0     128             [::]:ssh           [::]:*
tcp   LISTEN 0     4096            [::1]:953          [::]:*
```

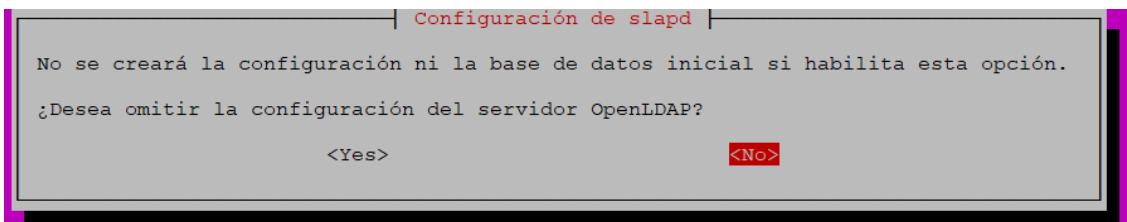
Una vez tenemos el id, para poder comprobar **el puerto** ahora con el comando ***ss -punta*** podemos comprobarlo con el numero anterior:

```
admindwes@AFR-LDAP:~$ ss -punta
Netid  State  Recv-Q Send-Q          Local Address:Port      Peer Address:Port  Process
udp   UNCONN 0      0              192.168.3.207:53    0.0.0.0:*
udp   UNCONN 0      0              127.0.0.1:53      0.0.0.0:*
udp   UNCONN 0      0              127.0.0.53%lo:53  0.0.0.0:*
udp   UNCONN 0      0              [::1]:53          [::]:*
udp   UNCONN 0      0      [fe80::a00:27ff:fe45:c1d6]@enp0s3:53  [::]:*
tcp   LISTEN 0     1024            0.0.0.0:389        0.0.0.0:*
tcp   LISTEN 0     10              192.168.3.207:53  0.0.0.0:*
tcp   LISTEN 0     10              127.0.0.1:53      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     4096            127.0.0.1:953     0.0.0.0:*
tcp   ESTAB  0      64              192.168.3.207:22  192.168.3.7:52248
tcp   LISTEN 0     1024            [::]:389          [::]:*
tcp   LISTEN 0     511             *:80              *:*
tcp   LISTEN 0     10      [fe80::a00:27ff:fe45:c1d6]@enp0s3:53  [::]:*
tcp   LISTEN 0     10              [::1]:53          [::]:*
tcp   LISTEN 0     128             [::]:22          [::]:*
tcp   LISTEN 0     4096            [::1]:953          [::]:*
```

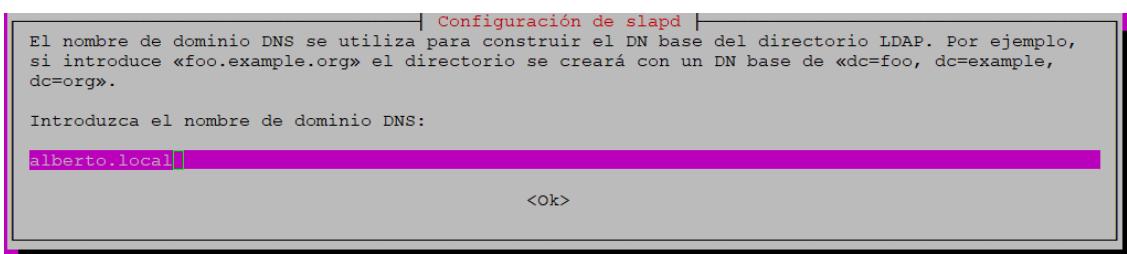
Para seguir con la configuración de ldap, usaremos el siguiente comando:

```
adminwes@AFR-LDAP:~$ sudo dpkg-reconfigure slapd
```

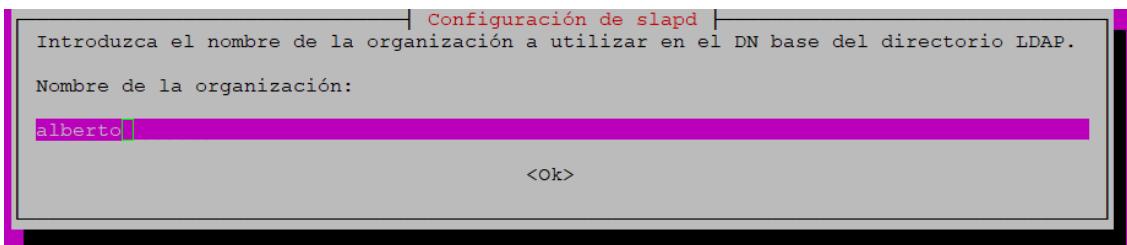
Indicaremos que **no** queremos omitir la configuración del servidor OpenLdap.



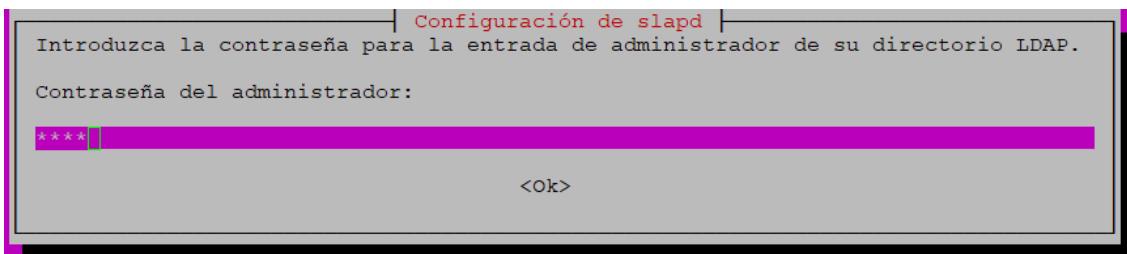
Introduciremos el nombre del dominio DNS que hemos configurado anteriormente, si todo lo anterior estaba bien configurado, debería de salir ya el nombre correcto, **tunombre.local**



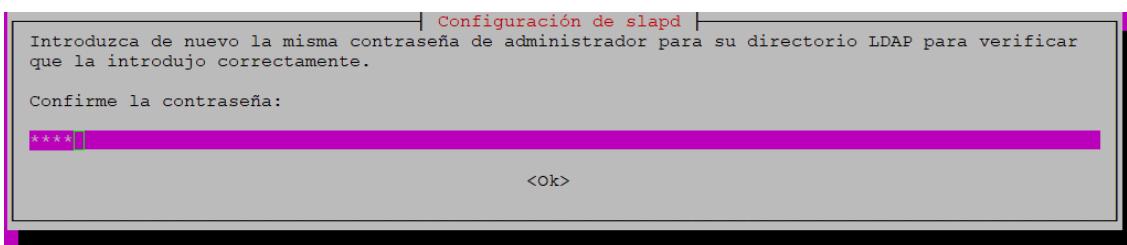
Introduciremos el nombre de la organización, **tunombre**



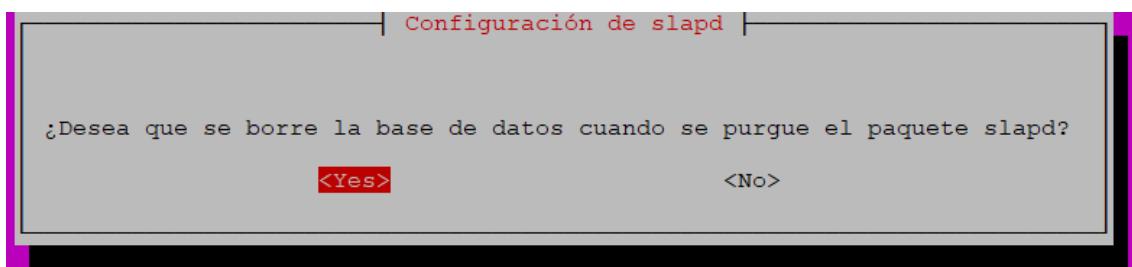
Asignaremos la **contraseña** del administrador.



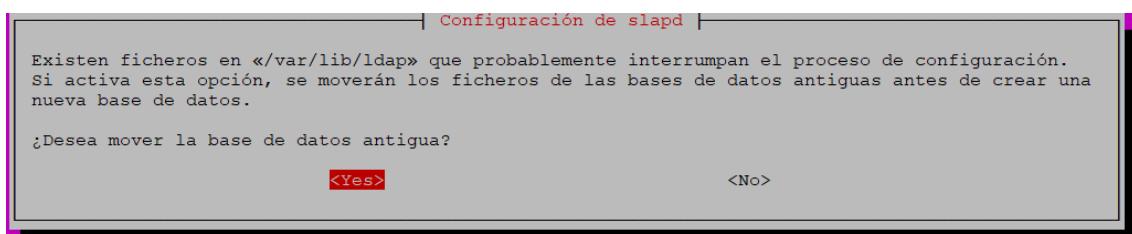
Introduciremos de nuevo la **contraseña** del administrador.



Indicaremos que se borre la **base de datos** cuando se purge slapd.



Moveremos la **base de datos** antigua.



Una vez terminada la configuracion, comprobaremos que la configuracion has sido correcta con el comando *sudo slapcat*

```
admindwes@AFR-LDAP:~$ sudo slapcat
dn: dc=alberto,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: alberto
dc: alberto
structuralObjectClass: organization
entryUUID: e905a06e-19fc-103c-9d80-271837653262
creatorsName: cn=admin,dc=alberto,dc=local
createTimestamp: 20220204115410Z
entryCSN: 20220204115410.955200Z#000000#000#000000
modifiersName: cn=admin,dc=alberto,dc=local
modifyTimestamp: 20220204115410Z

dn: cn=admin,dc=alberto,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9czhnNVhkY3lmRjNveFZyZHBZdFRQZ2xaa3FwTnZURzE=
structuralObjectClass: organizationalRole
entryUUID: e905f0aa-19fc-103c-9d81-271837653262
creatorsName: cn=admin,dc=alberto,dc=local
createTimestamp: 20220204115410Z
entryCSN: 20220204115410.957293Z#000000#000#000000
modifiersName: cn=admin,dc=alberto,dc=local
modifyTimestamp: 20220204115410Z
```

Para evitar futuros problemas en la configuracion, editaremos el fichero hosts con el comando `sudo nano /etc/hosts`, dentro introduciremos otra linea mas con la ip **192.168.3.20X XXX-LDAP.nombre.local XXX-LDAP**, guardamos y cerramos el archivo.

```
GNU nano 4.8                               /etc/hosts                         Modified
127.0.0.1 localhost
127.0.1.1 AFR-LDAP.alberto.local AFR-LDAP
192.168.3.207 AFR-LDAP.alberto.local AFR-LDAP

# 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 la maquina con el siguiente comando:

```
admindwes@AFR-LDAP:~$ reboot
```

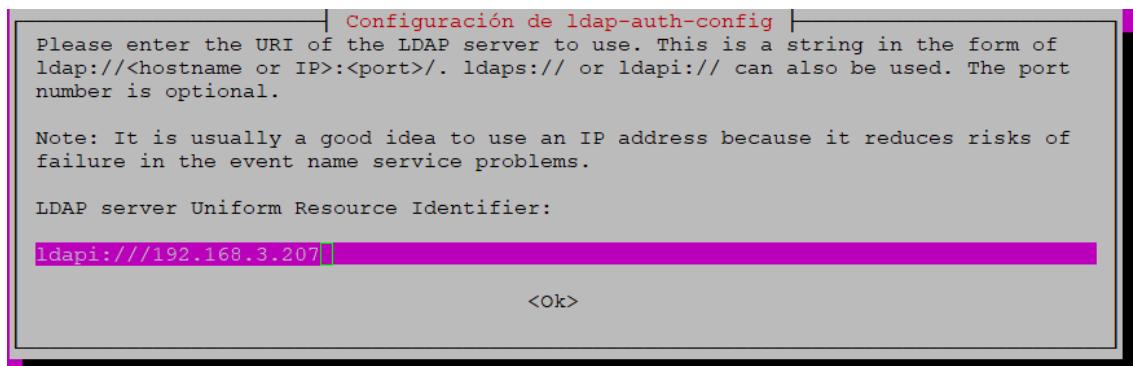
Comprobamos con el siguiente comando que los cambios se han realizado, si nos muestra **XXX-LDAP.tunombre.local** es que esta correcto:

```
admindwes@AFR-LDAP:~$ hostname -f
AFR-LDAP.alberto.local
```

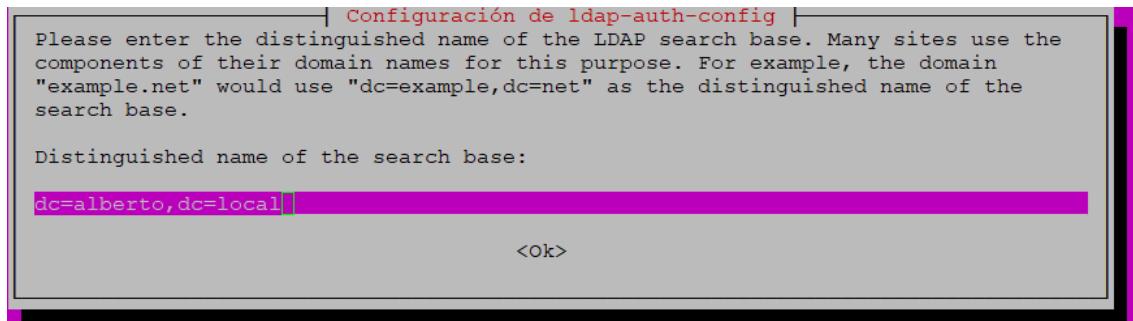
Lo siguiente sera instalar **libnss-ldap** con el siguiente comando:

```
admindwes@AFR-LDAP:~$ sudo apt install libnss-ldap -y
```

Introduciremos el identificador que sera la **IP de la maquina LDAP**, en nuestro caso **192.168.3.20X**



Introduciremos el **dc=tunombre** seguido de una coma y el segundo **dc=local**



Usaremos la versión 3 de LDAP.

```
| Configuración de ldap-auth-config |
Please enter which version of the LDAP protocol should be used by ldapns. It is
usually a good idea to set this to the highest available version.
```

LDAP version to use:

3
2

<Ok>

Indicaremos que el root sera el admin de la base de datos.

```
| Configuración de ldap-auth-config |
This option will allow you to make password utilities that use pam to behave like
you would be changing local passwords.
```

The password will be stored in a separate file which will be made readable to root
only.

If you are using NFS mounted /etc or any other custom setup, you should disable
this.

Make local root Database admin:

<Yes> <No>

No requeriremos que haya login en la database.

```
| Configuración de ldap-auth-config |
Choose this option if you are required to login to the database to retrieve
entries.
```

Note: Under a normal setup, this is not needed.

Does the LDAP database require login?

<Yes> <No>

Indicaremos la cuenta para el root, en nuestro caso sera
cn=admin,dc=tunombre,dc=local

```
| Configuración de ldap-auth-config |
This account will be used when root changes a password.
```

Note: This account has to be a privileged account.

LDAP account for root:

cn=admin,dc=alberto,dc=local

<Ok>

Introduciremos la **contraseña** del root a asignar.

```
| Configuración de ldap-auth-config |
Please enter the password to use when ldap-auth-config tries to login to the LDAP
directory using the LDAP account for root.

The password will be stored in a separate file /etc/ldap.secret which will be made
readable to root only.

Entering an empty password will re-use the old password.

LDAP root account password:
*****                                     <ok>
```

Comprobaremos con el siguiente comando que toda la información se ha configurado de manera correcta:

```
adminwes@AFR-LDAP:~$ sudo ldapsearch -xLLL -b dc=alberto,dc=local
dn: dc=alberto,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: alberto
dc: alberto

dn: cn=admin,dc=alberto,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
```

Ahora usaremos archivos que ya tenía preconfigurados en la rama Alberto de mi repositorio de GitHub para crear usuarios, etc...

```
adminwes@AFR-LDAP:~/DAWDespliegueAplicacionesWeb/ldap$ ls
add_GroupOfUniqueName.ldif  add_user.ldif  ejemplo.ldif      ldapusers.ldif
add_ou.ldif                 basedn.ldif    ldapgroups.ldif
```

Empezaremos creando dos unidades organizativas, editaremos el archivo con el siguiente comando(si no tenemos el archivo lo creamos con *sudo nano nombre.ldif*):

```
adminwes@AFR-LDAP:~/DAWDespliegueAplicacionesWeb/ldap$ sudo nano add_ou.ldif
```

Configuramos el archivo con nuestro dc=tunombre y guardamos.

```
GNU nano 4.8                                add_ou.ldif                               Modified
#####
### Alta de Unidad Organizativas
#####
dn: ou=usuarios,dc=alberto,dc=local
objectClass: organizationalUnit
ou: usuarios

dn: ou=grupos,dc=alberto,dc=local
objectClass: organizationalUnit
ou: grupos
```

Ejecutaremos el archivo para anadir las nuevas entradas con el siguiente comando(**dc=tunombre**):

```
adminwes@AFR-LDAP:~/DAWDespliegueAplicacionesWeb/ldap$ sudo ldapadd -x -D cn=admin,dc=alberto,dc=local -W -f add_ou.ldif
Enter LDAP Password:
adding new entry "ou=usuarios,dc=alberto,dc=local"
adding new entry "ou=grupos,dc=alberto,dc=local"
```

Comprobamos que se han creado correctamente con el siguiente comando(**dc=tunombre**):

```
adminwes@AFR-LDAP:~/DAWDespliegueAplicacionesWeb/ldap$ sudo ldapsearch -xLLL -b dc=alberto,dc=local
dn: dc=alberto,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: alberto
dc: alberto

dn: cn=admin,dc=alberto,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

dn: ou=usuarios,dc=alberto,dc=local
objectClass: organizationalUnit
ou: usuarios

dn: ou=grupos,dc=alberto,dc=local
objectClass: organizationalUnit
ou: grupos
```

Ahora crearemos los usuarios, editaremos el archivo con el siguiente comando(si no tenemos el archivo lo creamos con *sudo nano nombre.ldif*):

```
adminwes@AFR-LDAP:~/DAWDespliegueAplicacionesWeb/ldap$ sudo nano add_user.ldif
```

Configuramos el archivo con nuestro **dc=tunombre** y guardamos.

GNU nano 4.8	add_user.ldif	Modified
##### ## Alta de usuarios ## ##### # Usuarios zipi de la unidad organizativa usuarios dn: uid=zipi,ou=usuarios,dc=alberto,dc=local objectClass: inetOrgPerson objectClass: organizationalPerson objectClass: person objectClass: top uid: zipi sn: zapatilla cn: zipi mail: zipi@alberto.local userPassword: paso # Usuarios zape de la unidad organizativa usuarios dn: uid=zape,ou=usuarios,dc=alberto,dc=local objectClass: inetOrgPerson objectClass: organizationalPerson objectClass: person objectClass: top uid: zape sn: zapatilla cn: zape mail: zape@alberto.local userPassword: paso		

Ejecutaremos el archivo para anadir las nuevas entradas con el siguiente comando(**dc=tunombre**):

```
adminwes@AFR-LDAP:~/DAWDespliegueAplicacionesWeb/ldap$ sudo ldapadd -x -D cn=admin,dc=alberto,dc=local -W -f add_user.ldif
Enter LDAP Password:
adding new entry "uid=zipi,ou=usuarios,dc=alberto,dc=local"
adding new entry "uid=zape,ou=usuarios,dc=alberto,dc=local"
```

Comprobamos que se han creado correctamente con el siguiente comando(**dc=tunombre**):

```
adminwes@AFR-LDAP:~/DAWDespliegueAplicacionesWeb/ldap$ sudo ldapsearch -LLL -b dc=alberto,dc=local
dn: dc=alberto,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: alberto
dc: alberto

dn: cn=admin,dc=alberto,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

dn: ou=usuarios,dc=alberto,dc=local
objectClass: organizationalUnit
ou: usuarios

dn: ou=grupos,dc=alberto,dc=local
objectClass: organizationalUnit
ou: grupos

dn: uid=zipi,ou=usuarios,dc=alberto,dc=local
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
uid: zipi
sn: zapatilla
cn: zipi
mail: zipi@alberto.local

dn: uid=zape,ou=usuarios,dc=alberto,dc=local
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
uid: zape
sn: zapatilla
cn: zape
mail: zape@alberto.local
```

Ahora crearemos los grupos de nombres unicos, editaremos el archivo con el siguiente comando(si no tenemos el archivo lo creamos con *sudo nano nombre.ldif*):

```
adminwes@AFR-LDAP:~/DAWDespliegueAplicacionesWeb/ldap$ sudo nano add_GroupOfUniqueName.ldif
```

Configuramos el archivo con nuestro dc=tunombre y guardamos.

```
GNU nano 4.8                                add_GroupOfUniqueName.ldif                                Modified
#####
##### ALTA GRUPO DE NOMBRES UNICOS
#####
dn: cn=gemelos,ou=grupos,dc=alberto,dc=local
objectClass: groupOfUniqueNames
objectClass: top
cn:jefes
uniqueMember: uid=zipi,ou=usuarios,dc=alberto,dc=local
uniqueMember: uid=zape,ou=usuarios,dc=alberto,dc=local
```

Ejecutaremos el archivo para anadir las nuevas entradas con el siguiente comando(**dc=tunombre**):

```
adminwes@AFR-LDAP:~/DAWDespliegueAplicacionesWeb/ldap$ sudo ldapadd -x -D cn=admin,dc=alberto,dc=local  
-W -f add_GroupOfUniqueName.ldif  
Enter LDAP Password:  
adding new entry "cn=gemelos,ou=grupos,dc=alberto,dc=local"
```

Comprobamos que se han creado correctamente con el siguiente comando(**dc=tunombre**):

```
adminwes@AFR-LDAP:~/DAWDespliegueAplicacionesWeb/ldap$ sudo ldapsearch -xLLL -b dc=alberto,dc=local  
dn: dc=alberto,dc=local  
objectClass: top  
objectClass: dcObject  
objectClass: organization  
o: alberto  
dc: alberto  
  
dn: cn=admin,dc=alberto,dc=local  
objectClass: simpleSecurityObject  
objectClass: organizationalRole  
cn: admin  
description: LDAP administrator  
  
dn: ou=usuarios,dc=alberto,dc=local  
objectClass: organizationalUnit  
ou: usuarios  
  
dn: ou=grupos,dc=alberto,dc=local  
objectClass: organizationalUnit  
ou: grupos  
  
dn: uid=zipi,ou=usuarios,dc=alberto,dc=local  
objectClass: inetOrgPerson  
objectClass: organizationalPerson  
objectClass: person  
objectClass: top  
uid: zipi  
sn: zapatilla  
cn: zipi  
mail: zipi@alberto.local  
  
dn: uid=zape,ou=usuarios,dc=alberto,dc=local  
objectClass: inetOrgPerson  
objectClass: organizationalPerson  
objectClass: person  
objectClass: top  
uid: zape  
sn: zapatilla  
cn: zape  
mail: zape@alberto.local  
  
dn: cn=gemelos,ou=grupos,dc=alberto,dc=local  
objectClass: groupOfUniqueNames  
objectClass: top  
cn: jefes  
cn: gemelos  
uniqueMember: uid=zipi,ou=usuarios,dc=alberto,dc=local  
uniqueMember: uid=zape,ou=usuarios,dc=alberto,dc=local
```

Ahora editaremos informacion de los usuarios, creamos con *sudo nano nombre. ldif* un archivo para realizar los cambios necesarios:

```
GNU nano 4.8                                     edit user.ldif                                         Modified  
#anyadir el movil al usuario zipi  
dn: uid=zipi,ou=usuarios,dc=alberto,dc=local  
add: mobile  
mobile: 677888999  
  
dn: uid=zipi,ou=usuarios,dc=alberto,dc=local  
delete: mail  
  
dn: uid=zape,ou=usuarios,dc=alberto,dc=local  
changetype: modify  
replace: mail  
mail: zapatero@alberto.local
```

Ejecutaremos el archivo para editar las entradas que queremos cambiar con el siguiente comando(**dc=tunombre**):

```
adminwes@AFR-LDAP:~/DAWDespliegueAplicacionesWeb/ldap$ sudo ldapmodify -x -D cn=admin,dc=alberto,dc=local -W -f edit_user.ldif
Enter LDAP Password:
modifying entry "uid=zipi,ou=usuarios,dc=alberto,dc=local"
modifying entry "uid=zipi,ou=usuarios,dc=alberto,dc=local"
modifying entry "uid=zape,ou=usuarios,dc=alberto,dc=local"
```

Comprobamos que se han modificado correctamente con el siguiente comando(**dc=tunombre**):

```
adminwes@AFR-LDAP:~/DAWDespliegueAplicacionesWeb/ldap$ sudo ldapsearch -xLLL -b dc=alberto,dc=local
dn: dc=alberto,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: alberto
dc: alberto

dn: cn=admin,dc=alberto,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

dn: ou=usuarios,dc=alberto,dc=local
objectClass: organizationalUnit
ou: usuarios

dn: ou=grupos,dc=alberto,dc=local
objectClass: organizationalUnit
ou: grupos

dn: uid=zipi,ou=usuarios,dc=alberto,dc=local
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
uid: zipi
sn: zapatilla
cn: zipi
mobile: 677888999

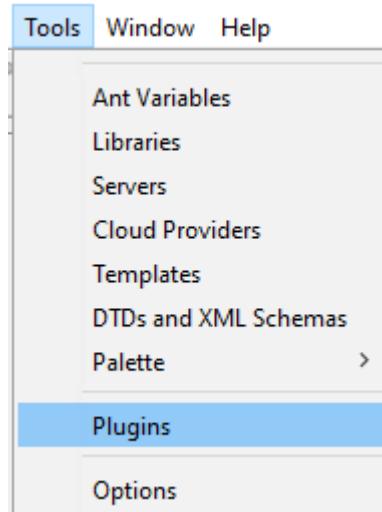
dn: uid=zape,ou=usuarios,dc=alberto,dc=local
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
uid: zape
sn: zapatilla
cn: zape
mail: zapatero@alberto.local

dn: cn=gemelos,ou=grupos,dc=alberto,dc=local
objectClass: groupOfUniqueNames
objectClass: top
cn: jefes
cn: gemelos
uniqueMember: uid=zipi,ou=usuarios,dc=alberto,dc=local
uniqueMember: uid=zape,ou=usuarios,dc=alberto,dc=local
```

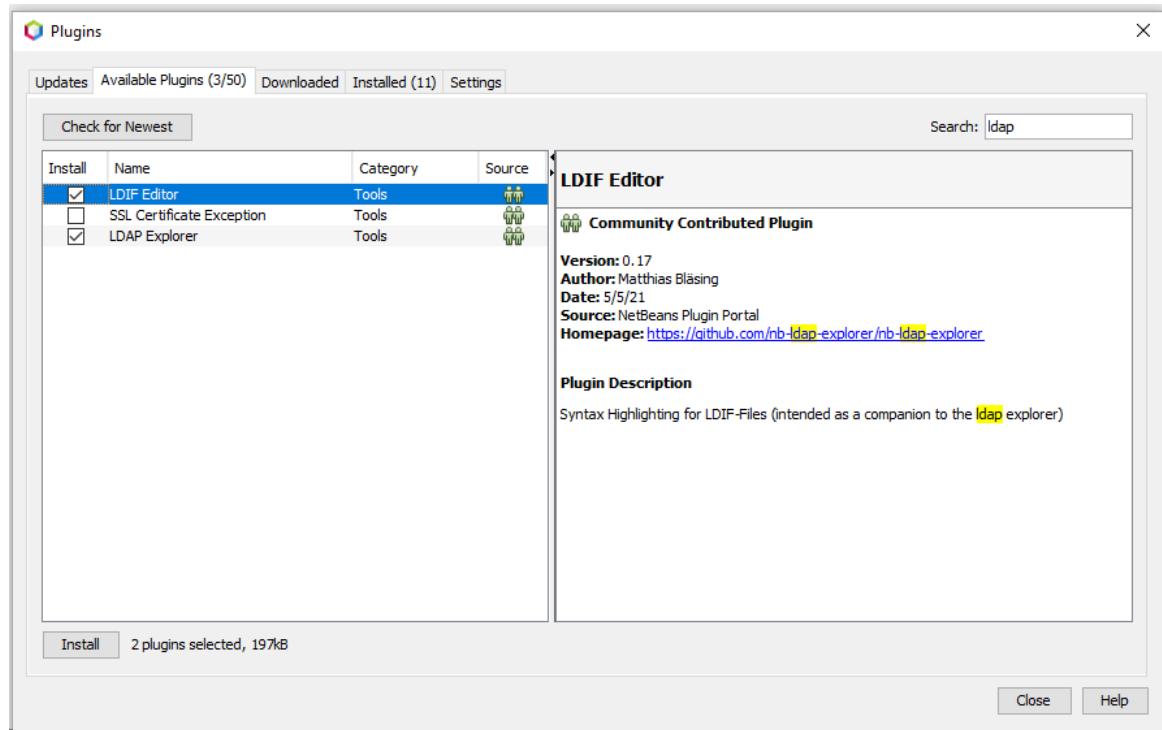
1.18.3 Prueba de uso de conexión desde NetBeans

Para comprobar que todo ha sido creado correctamente y esta funcionando, usaremos NetBeans para conectarnos al servidor LDAP y ver todo lo que hemos creado.

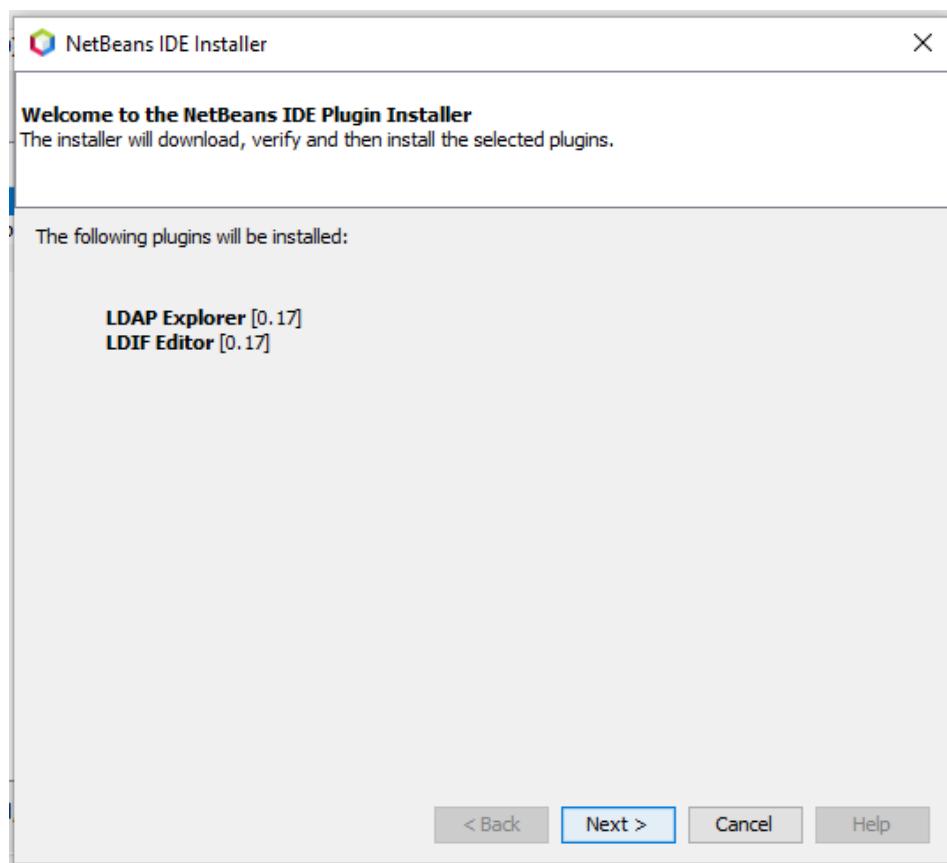
Lo primero sera instalar algunos plugins, nos iremos a **Tools** y a **Plugins**.



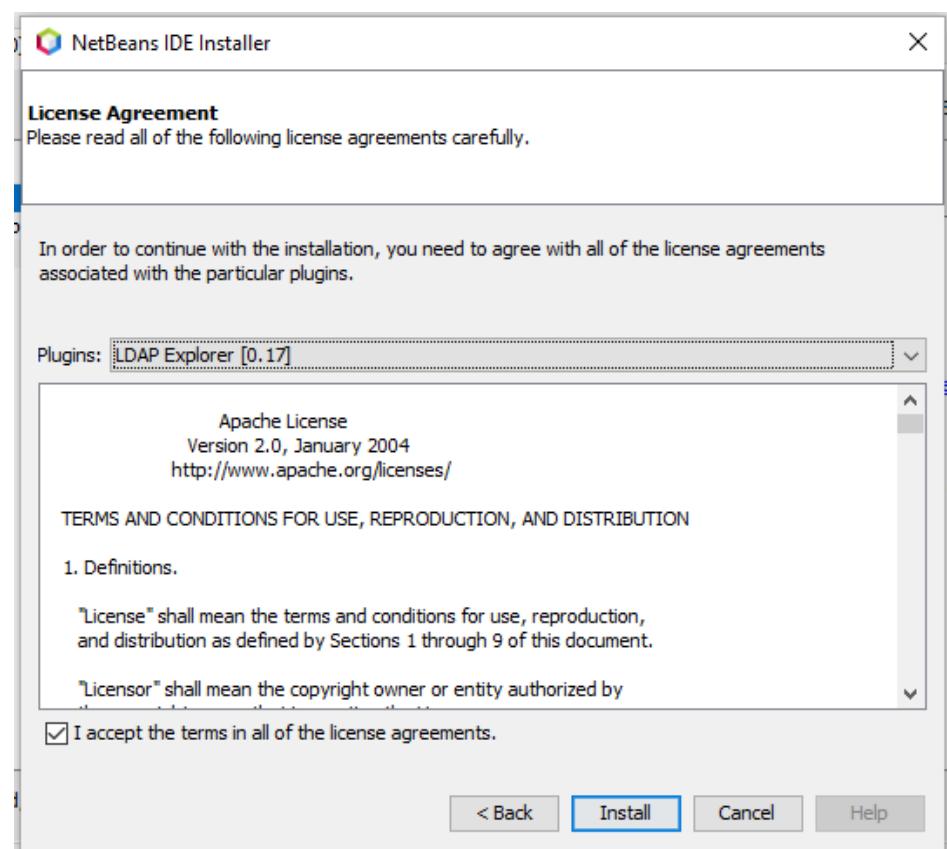
Una vez dentro de plugins, nos iremos a **Available Plugins**, y en el buscador buscaremos **ldap**, seleccionaremos **LDIF Editor** y **LDAP Explorer**. Una vez seleccionados le daremos a **Install**.



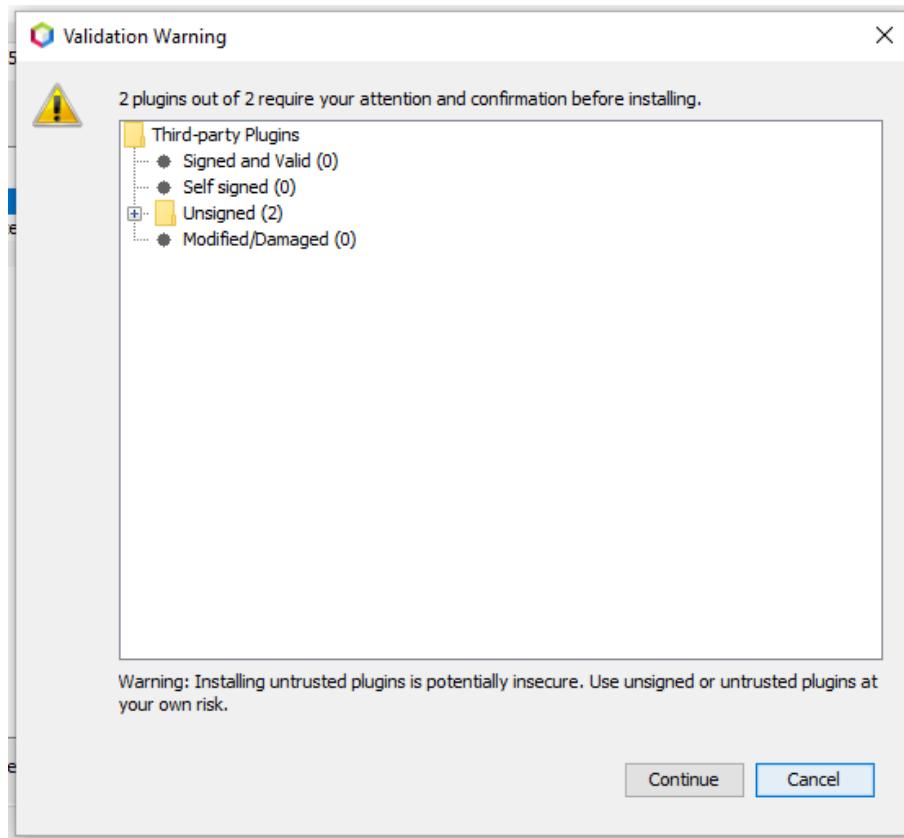
Nos mostrara los plugins que vamos a instalar, le daremos a **Next**.



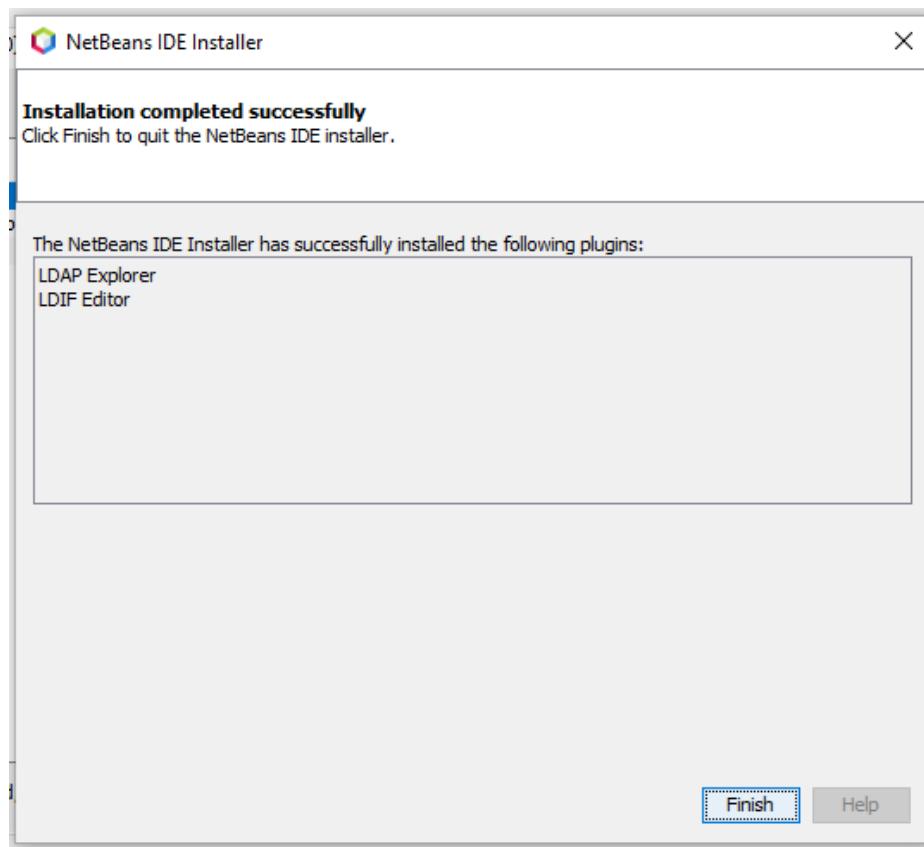
Aceptamos los terminos y le daremos a Next.



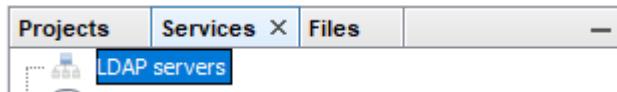
Nos saltara un problema de validacion, pero lo ignoraremos, le damos a **Continue**.



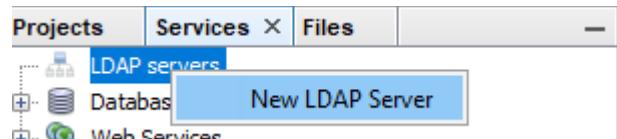
Si se han instalado correctamente se nos mostrara esta ventana, le daremos a **Finish**.



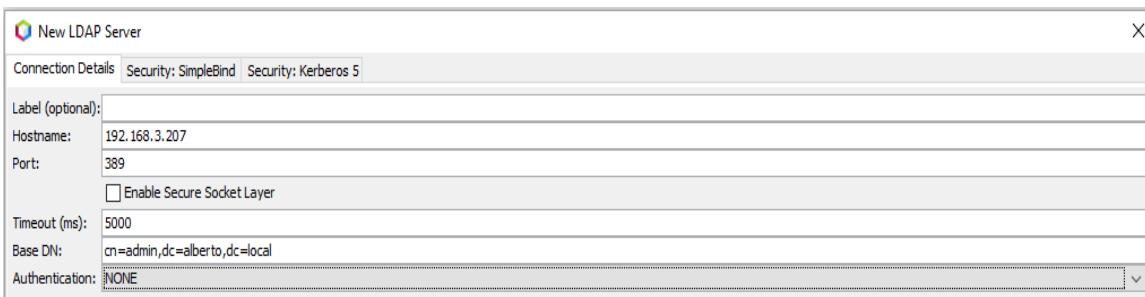
Una vez estan intalados los plugins, nos iremos a **Services**, y aquí dentro nos saldra **LDAP Servers**.



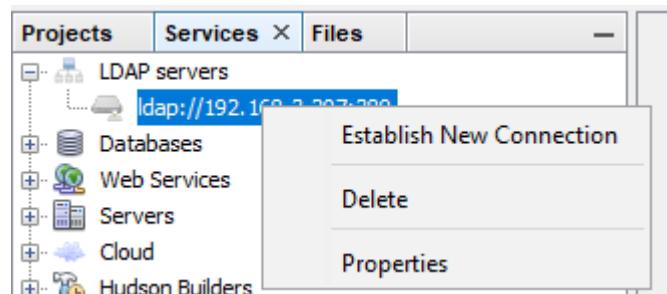
Haremos click derecho sobre **LDAP Servers** y le daremos a **New LDAP Server** para crear la conexión con el servidor LDAP.



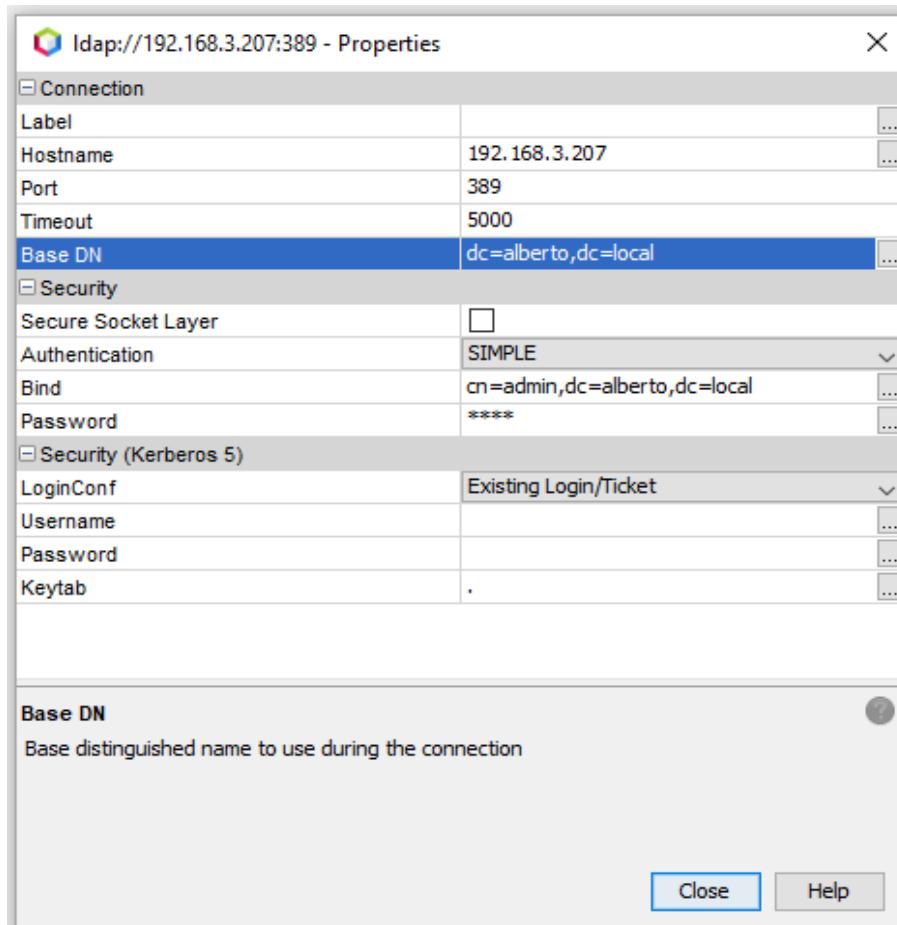
Configuraremos los datos, el hostname sera la IP de la maquina virtual del servidor LDAP, el puerto lo observamos anteriormente en la instalacion y era el 389, el Base DN sera cn=admin,dc=tunombre, cd=local y la authentication en NONE.



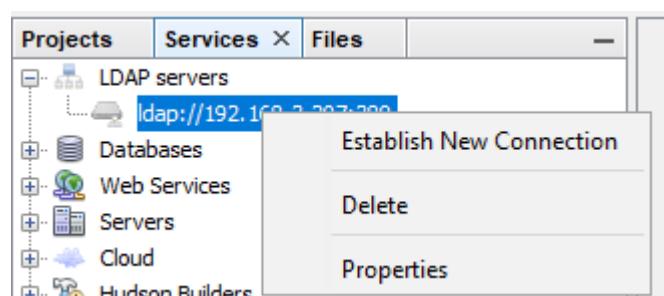
En caso de que la conexión no se realice, nos iremos a las propiedades de la conexión que hemos configurado, click derecho **Properties** .



Aquí dentro tendremos que configurar la parte de Security, pondremos la Authentication en SIMPLE, en el apartado de Bind podremos cn=admin,cd=tunombre, cd=local y en el apartado de Password introduciremos la password que se asigno en la instalacion de ldap al cn=admin. Una vez hecho esto ya podremos realizar la conexión.



Le daremos de nuevo click derecho a la conexión configurada, y le daremos a **Establish New Connection**.



Si todo ha funcionado correctamente, se realizara la conexión con el servidor LDAP y podremos ver usuarios, grupos, etc...

1.18.4 Configuración modulo autenticación

Antes de comenzar a configurar el módulo de autenticación, comprobaremos que tenemos bien configurada la red de las máquinas virtuales, tanto de XXX-USUSED como de XXX-LDAP.

Entraremos en la carpeta netplan de la maquina XXX-LDAP con `cd /etc/netplan` editaremos el archivo con el siguiente comando:

```
admindwes@AFR-LDAP:~$ cd /etc/netplan
admindwes@AFR-LDAP:/etc/netplan$ sudo nano 01-netcfg.yaml
```

Comprobaremos que en el apartado de **nameservers**: en concreto en el de **addresses**: tenemos la ip del servidor que está configurado con el DNS que es el de la maquina XXX-LDAP (192.168.3.20X)

```
GNU nano 4.8                                     01-netcfg.yaml                                         Modified
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.3.207/24
      gateway4: 192.168.3.1
      nameservers:
        search: [alberto.local]
        addresses: [192.168.3.207]
```

Una vez tenemos configurado el archivo, lo cerramos y lo guardamos. Aplicaremos los cambios con el siguiente comando:

```
admindwes@AFR-LDAP:~$ sudo netplan apply
```

Haremos lo mismo con el archivo de configuración de la maquina XXX-USUSED, entraremos a la carpeta netplan con `cd /etc/netplan` editaremos el archivo con el siguiente comando:

```
admindwes@AFR-USUSED:~$ cd /etc/netplan
admindwes@AFR-USUSED:/etc/netplan$ sudo nano 01-netcfg.yaml
```

Comprobaremos que en el apartado de **nameservers**: en concreto en el de **addresses**: tenemos la ip del servidor que está configurado con el DNS que es el de la maquina XXX-LDAP (192.168.3.20X)

```
GNU nano 4.8                                     01-netcfg.yaml                                         Modified
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.3.107/24
      gateway4: 192.168.3.1
      nameservers:
        search: [alberto.local]
        addresses: [192.168.3.207]
```

Una vez tenemos configurado el archivo, lo cerramos y lo guardamos. Aplicaremos los cambios con el siguiente comando:

```
admindwes@AFR-USUSED:~$ sudo netplan apply
```

Ahora ya podremos activar el módulo de autentificación llamado **authnz_ldap** con el siguiente comando:

```
adminwes@AFR-US: /etc/netplan$ sudo a2enmod authnz_ldap
Considering dependency ldap for authnz_ldap:
Enabling module ldap.
Enabling module authnz_ldap.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

Una vez activado, reiniciaremos el servicio de apache2 para aplicar los cambios con el siguiente comando:

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

IMPORTANTE: Antes de hacer los siguientes pasos, es importante tener activado **.htaccess** para todos los sitios virtuales o bien en el directorio de un sitio concreto.

Si no lo tenemos instalado, tendremos que editar el fichero de **apache2.conf** con el comando *sudo nano /etc/apache2/apache2.conf* y meter lo siguiente:

```
<Directory /var/www>
```

```
....
```

```
AllowOverrideAll
```

```
</Directory>
```

1.18.4.1 Configuración de la autenticación para un usuario

Lo primero que tendremos que hacer es crear una carpeta llamada **ldap** en el directorio de publicación (**public_html**) de la maquina XXX-USED, en mi caso lo he hecho en daw201.nombre.local



Nombre de archivo	Tamaño de a...	Tipo de arc...	Última modificación	Permisos	Propietario/...
..					
ldap		Carpeta de...	08/02/2022 10:47:51	drwxr-sr-x	daw201 ww...

Una vez tenemos la carpeta creada, crearemos dentro de ella el fichero **.htaccess** que configuraremos después.



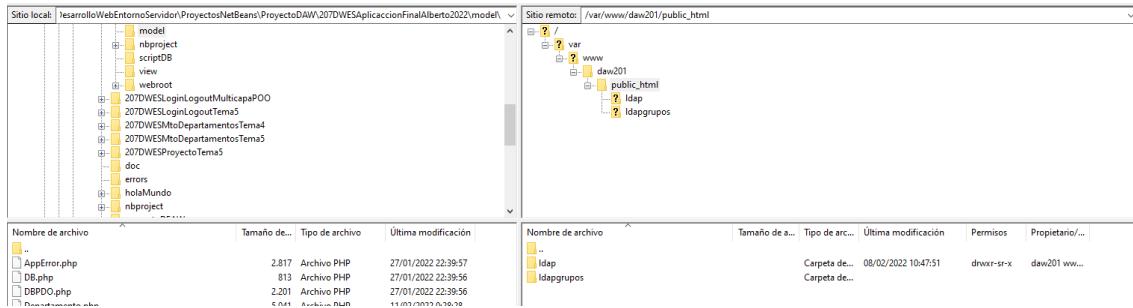
Nombre de archivo	Tamaño de a...	Tipo de arc...	Última modificación	Permisos	Propietario/...
..					
.htaccess	0	Archivo HT...	08/02/2022 10:47:51	-rw-r--r--	daw201 ww...

Ahora que tenemos el fichero, lo editaremos con el Notepad realizando una conexión a la maquina con el usuario daw201 y su Password. Una vez conectados y con el archivo abierto, configuraremos la **IP de la maquina LDAP y el dc=nombre** en **AuthLDAPURL**, indicaremos también el **cn=admin** y el **dc=nombre** en el **AuthLDAPBindDN**, cambiaremos el **AuthLDAPBindPassword** a la password a la que corresponda al admin y por último cambiaremos el **Require ldap-user** al nombre del usuario que iniciara sesión. Los demás parámetros del archivo pueden dejarse como están. Guardamos y cerramos.

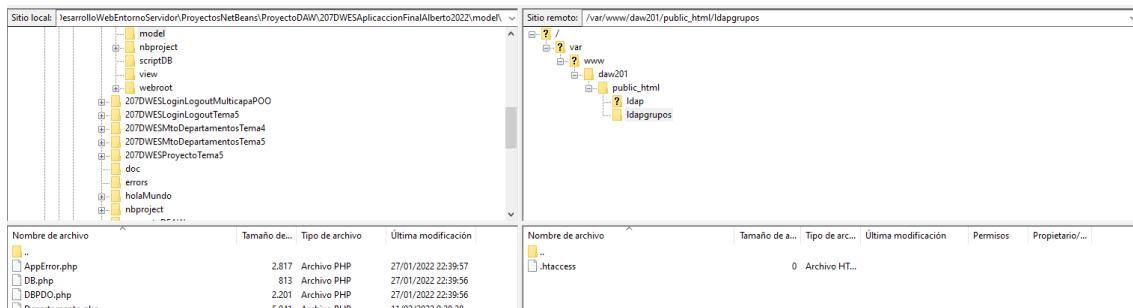
```
1 AuthType Basic
2 AuthBasicProvider ldap
3 AuthName "login..."
4 AuthLDAPURL "ldap://192.168.3.207/dc=alberto,dc=local?uid?sub?(objectClass=*)"
5 AuthLDAPBindDN "cn=admin,dc=alberto,dc=local"
6 AuthLDAPBindPassword pasow
7 Require ldap-user zipi
```

1.18.4.2 Configuración de la autenticación para un grupo

Lo primero que tendremos que hacer es crear una carpeta llamada **Idapgrupos** en el directorio de publicación (**public_html**) de la maquina XXX-USDED, en mi caso lo he hecho en daw201.nombre.local



Una vez tenemos la carpeta creada, crearemos dentro de ella el fichero **.htaccess** que configuraremos después.



Ahora que tenemos el fichero, lo editaremos con el Notepad realizando una conexión a la maquina con el usuario daw201 y su Password. Una vez conectados y con el archivo abierto, configuraremos la **IP de la maquina LDAP y el dc=nombre** en **AuthLDAPURL**, indicaremos también el **cn=admin** y el **dc=nombre en el AuthLDAPBindDN**, cambiaremos el **AuthLDAPBindPassword** a la password a la que corresponda al admin y por último cambiaremos el **Require ldap-group** al nombre del grupo que contiene los usuarios, en mi caso es **cn=gemelos,ou=grupos,dc=nombre,dc=local**. Los demás parámetros del archivo pueden dejarse como están. Guardamos y cerramos.

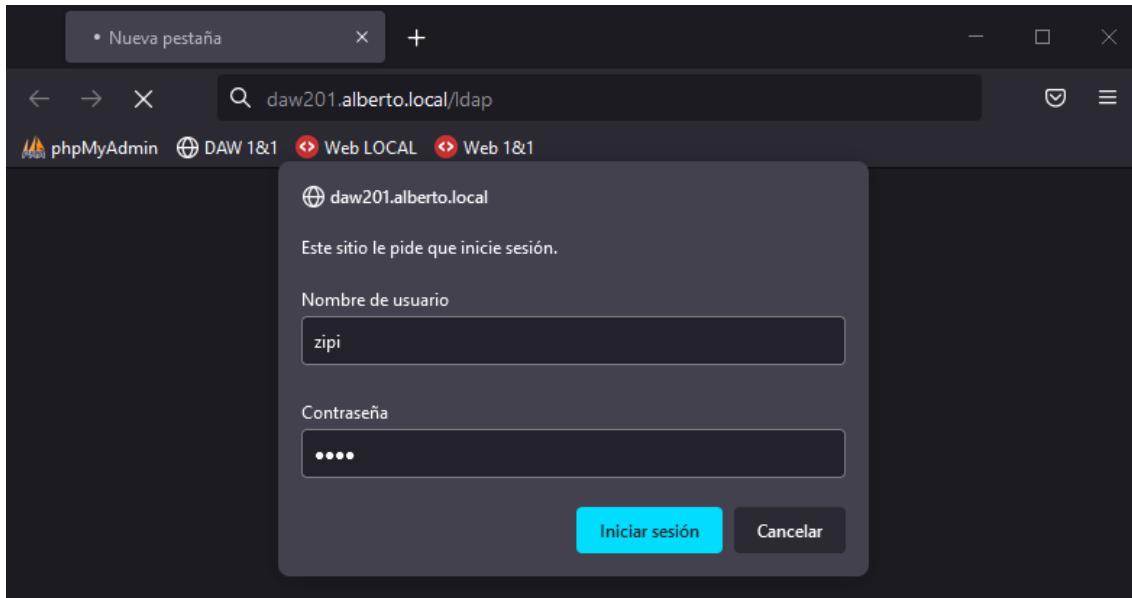
```

1 AuthType Basic
2 AuthBasicProvider ldap
3 AuthName "login...."
4 AuthLDAPURL "ldap://192.168.0.207/dc=alberto,dc=local?uid?sub?(objectClass=*)"
5 AuthLDAPBindDN "cn=admin,dc=alberto,dc=local"
6 AuthLDAPBindPassword paso
7 Require ldap-group cn=gemelos,ou=grupos,dc=alberto,dc=local

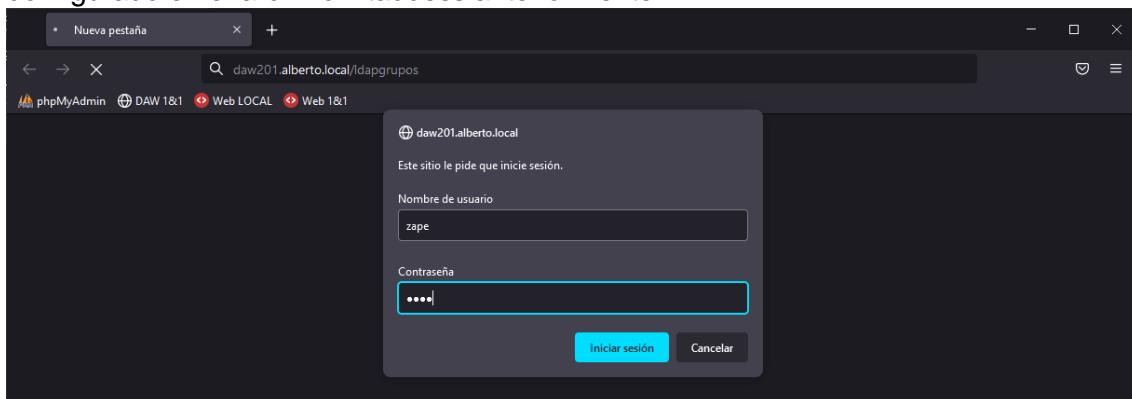
```

1.18.5 Prueba de uso de inicio de sesión desde el navegador

Una vez realizada la configuración por completo de todo lo anterior, solo nos quedara realizar la conexión entrando a la url **daw201.nombre.local/ldap** e iniciar sesión con el usuario que hemos configurado en el archivo .htaccess anteriormente.

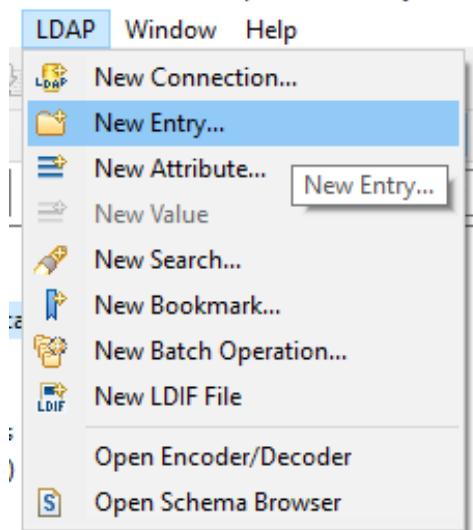


Realizaremos la conexión entrando a la url **daw201.nombre.local/ldapgrupos** e iniciar sesión con algún usuario de los que se encuentran en el grupo que hemos configurado en el archivo .htaccess anteriormente.

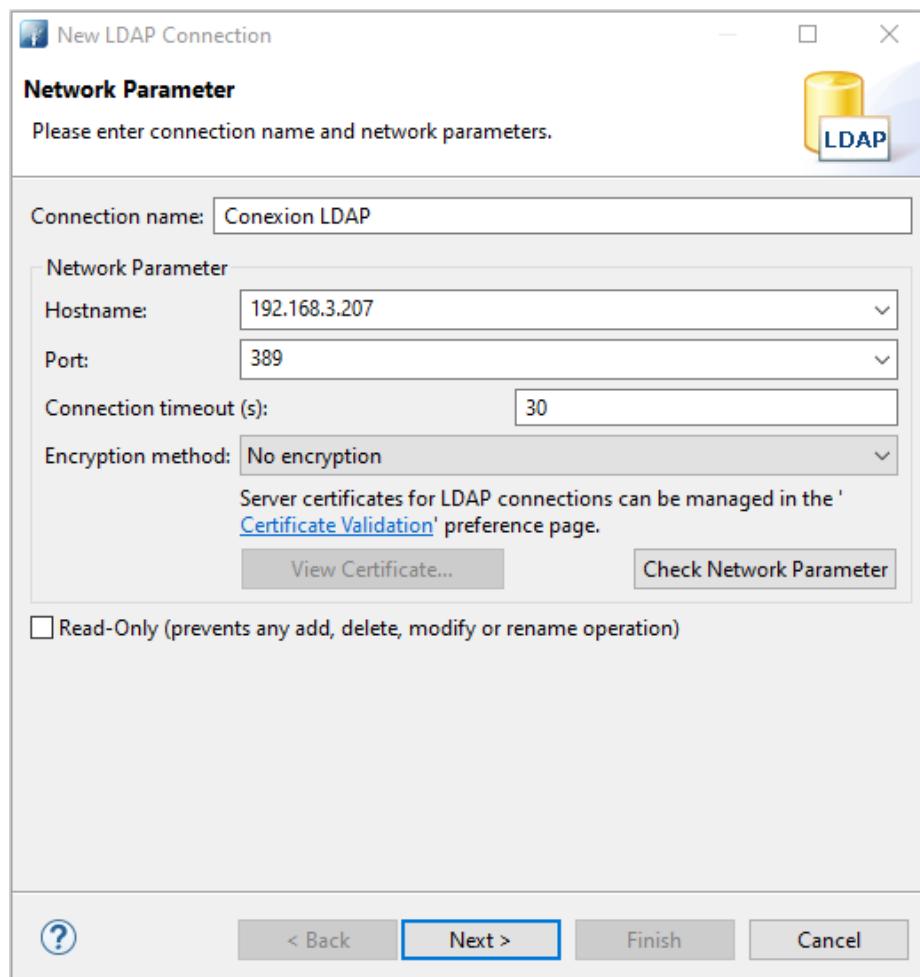


1.18.6 Administración con el cliente “Apache Directory Studio”

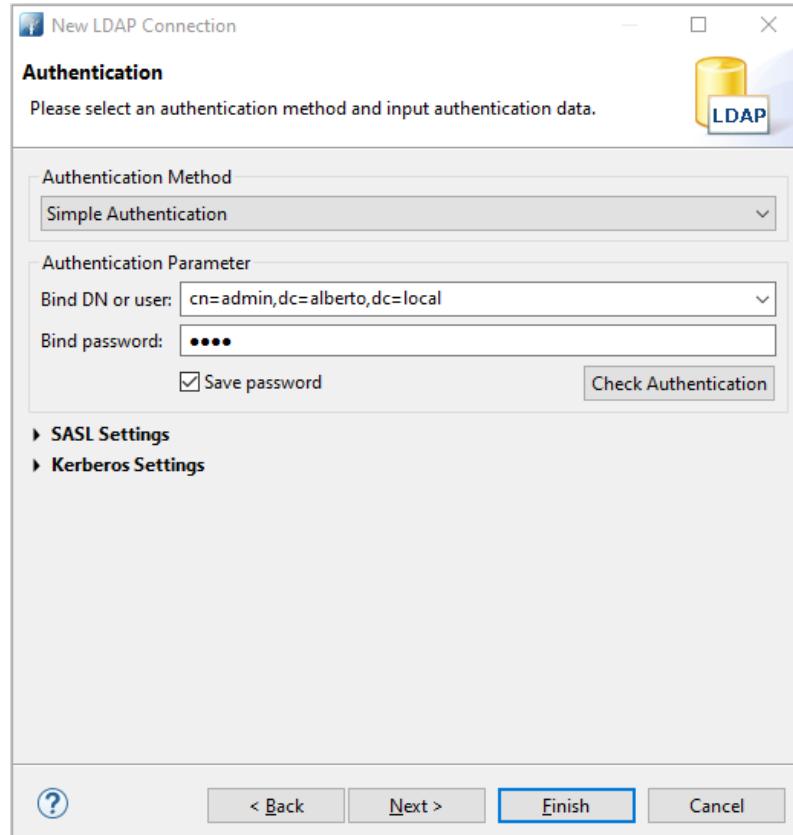
Una vez tenemos el programa instalado, lo abrimos y nos vamos a **LDAP** y a **New Connection** para conectarnos a nuestro servidor LDAP.



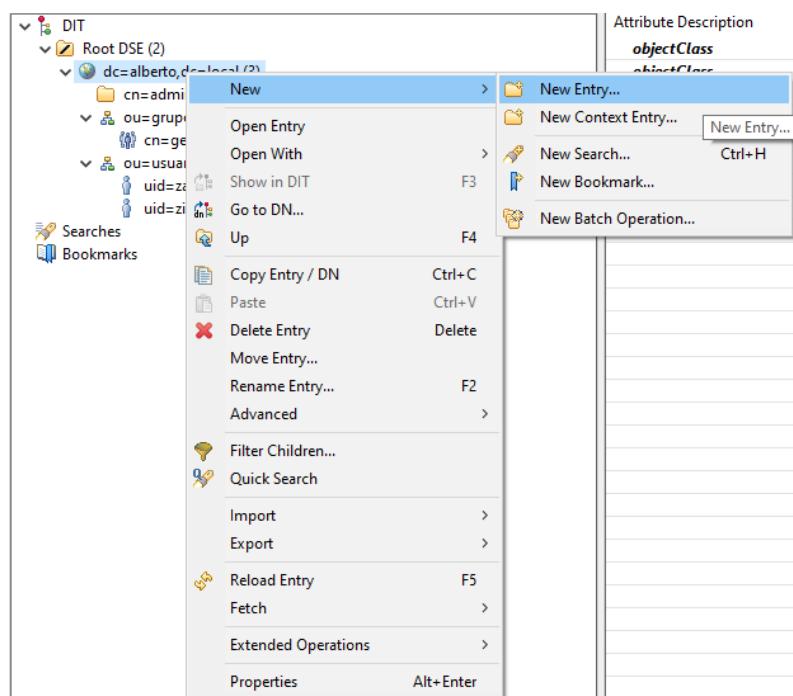
Configuramos los parámetros, le indicaremos el nombre de la conexión en **Connection name**, el **Hostname** que será la ip de la maquina LDAP y el **Port** esta por defecto que es el 389. Una vez configurado le damos a **Next**.



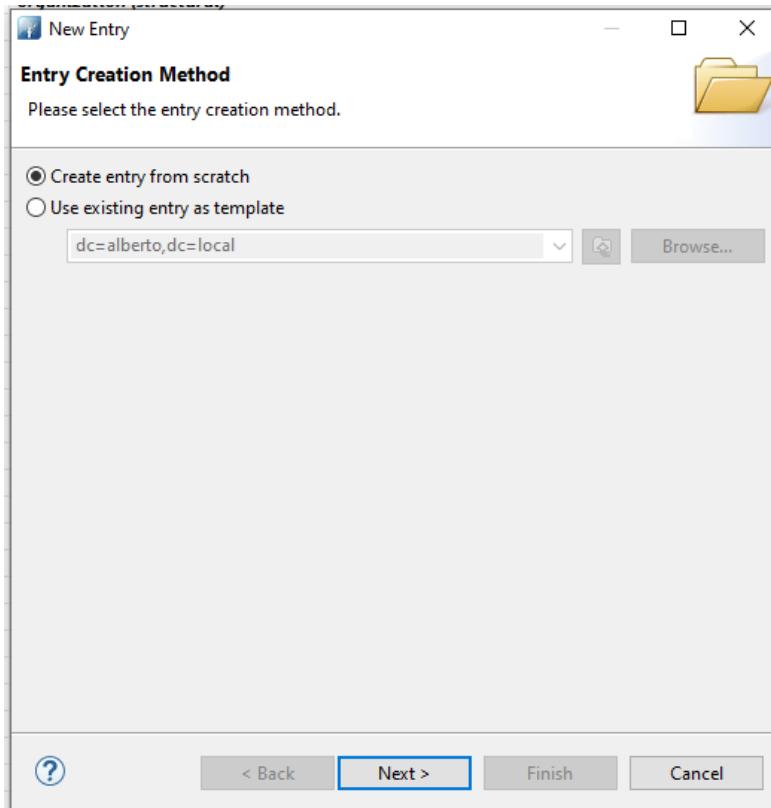
El siguiente paso será configurar el **Bind DN or user**, en este caso tendremos que poner el **cn=admin,dc=nombre,dc=local** y también configurar el **Bind Password** que será la contraseña que tenga el dc en nuestra configuración de LDAP. Una vez configurado le damos a **Finish**.



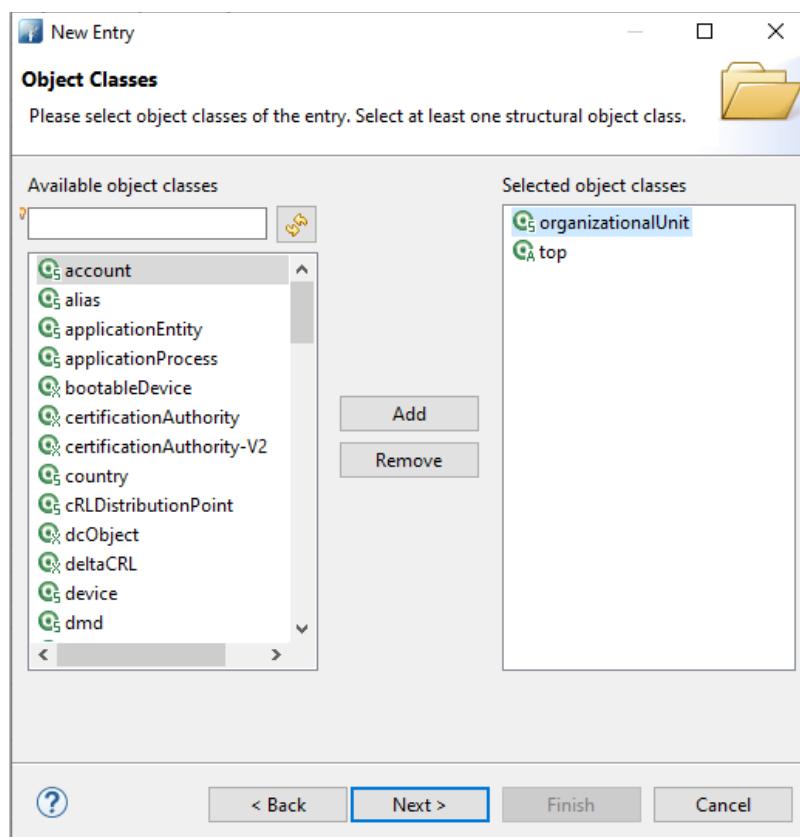
Una vez hemos realizado la conexión con el servidor LDAP, lo siguiente será crear una nueva **unidad organizativa** dentro de nuestro **dc=nombre,dc=local**, le daremos clic derecho a **dc=nombre,dc=local**, **New** y a **New Entry...**



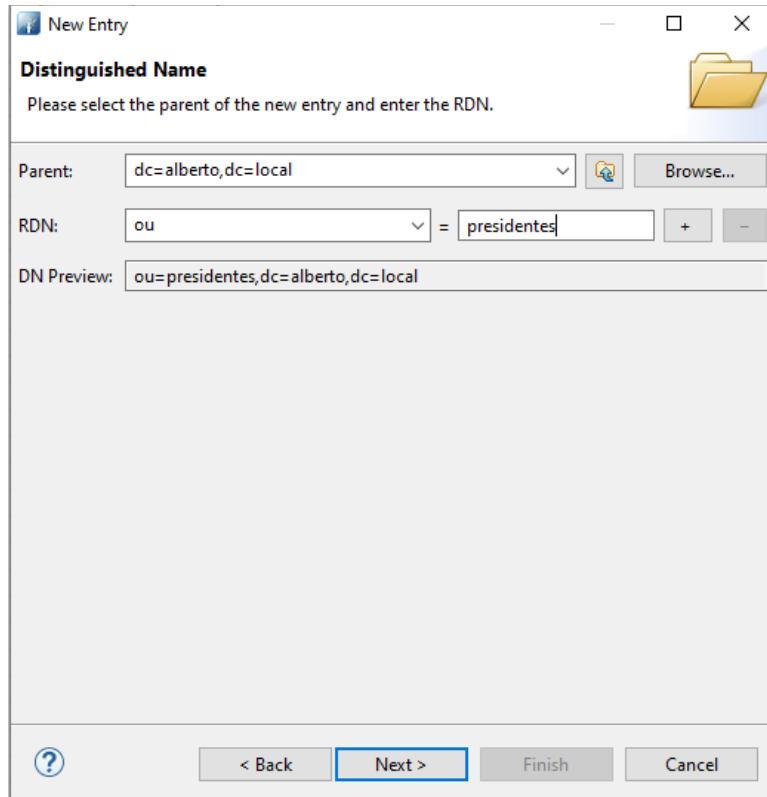
Dejaremos la primera opción marcada ya que no queremos usar una configuración ya creada, le daremos a **Next**.



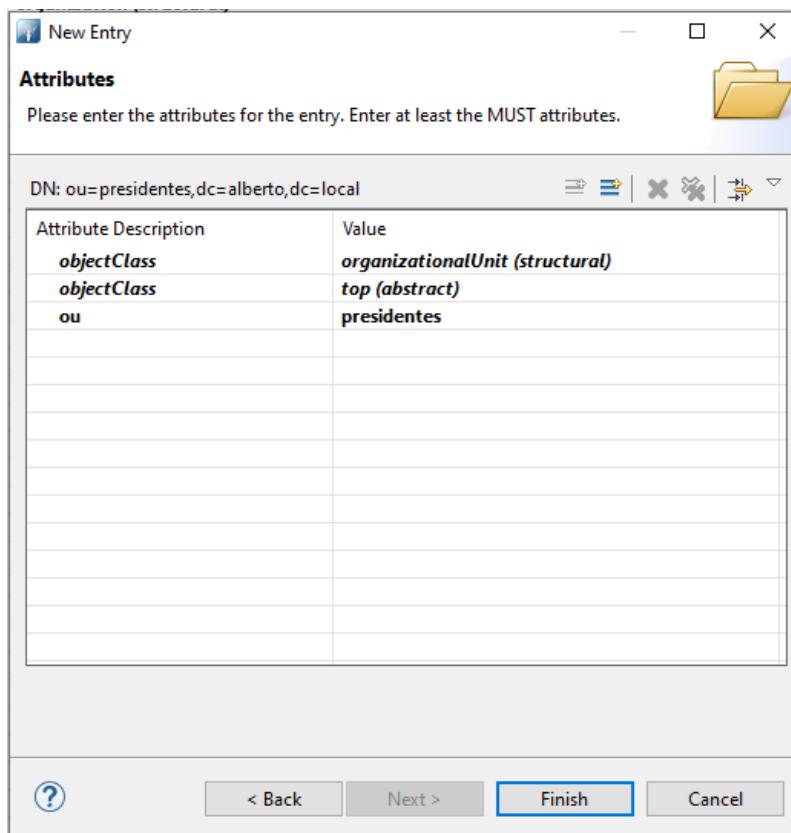
Agregaremos el objeto de clases llamado **organizationalUnit** buscándola en la lista y dándole a **Add**, le daremos a **Next**.



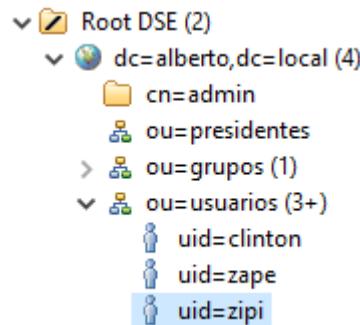
Lo siguiente será configurar el **RDN ou** que en mi caso el nuevo a crear será **presidentes** ya que es el nombre que he elegido. Le daremos a **Next**.



En el último paso se nos mostrara la información que se va a guardar, chequeamos que esta todo correcto y le daremos a **Finish**.

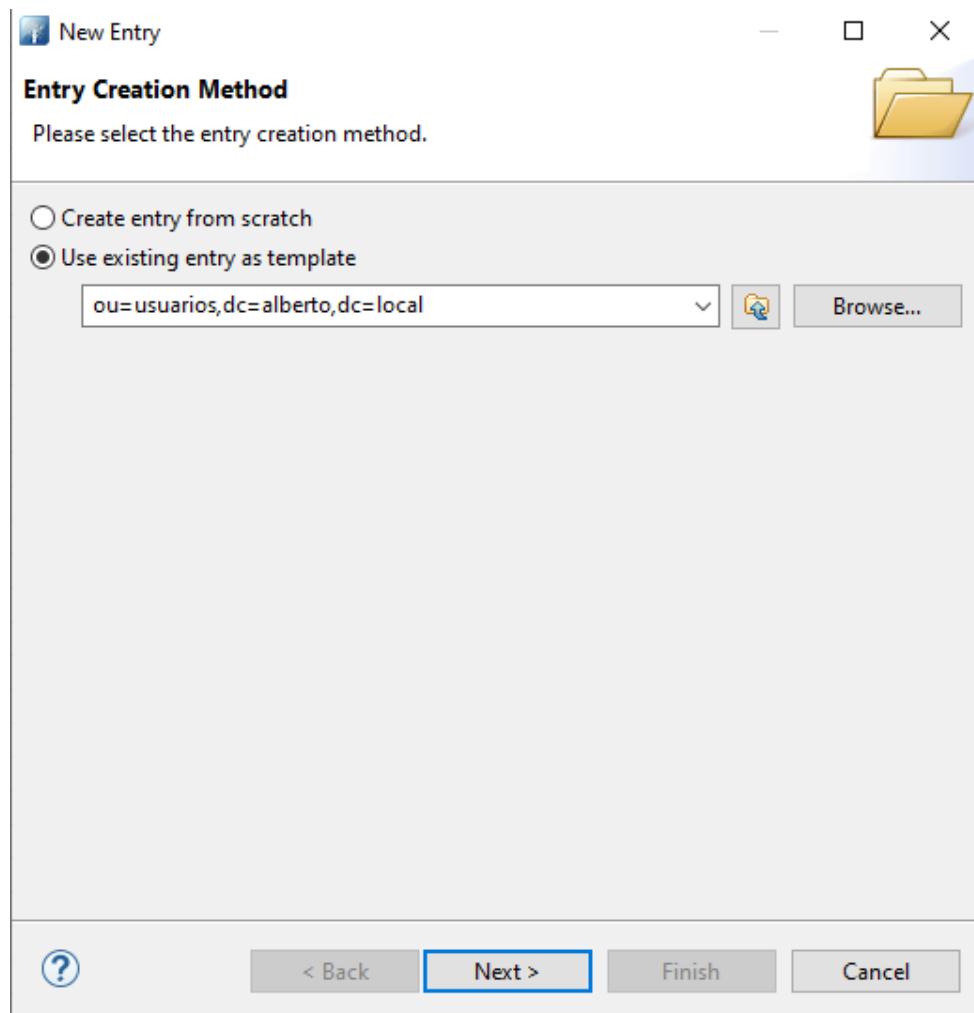


Una vez creada la nueva organización presidentes, pasaremos a crear los usuarios nuevos que irán dentro de la organización usuarios

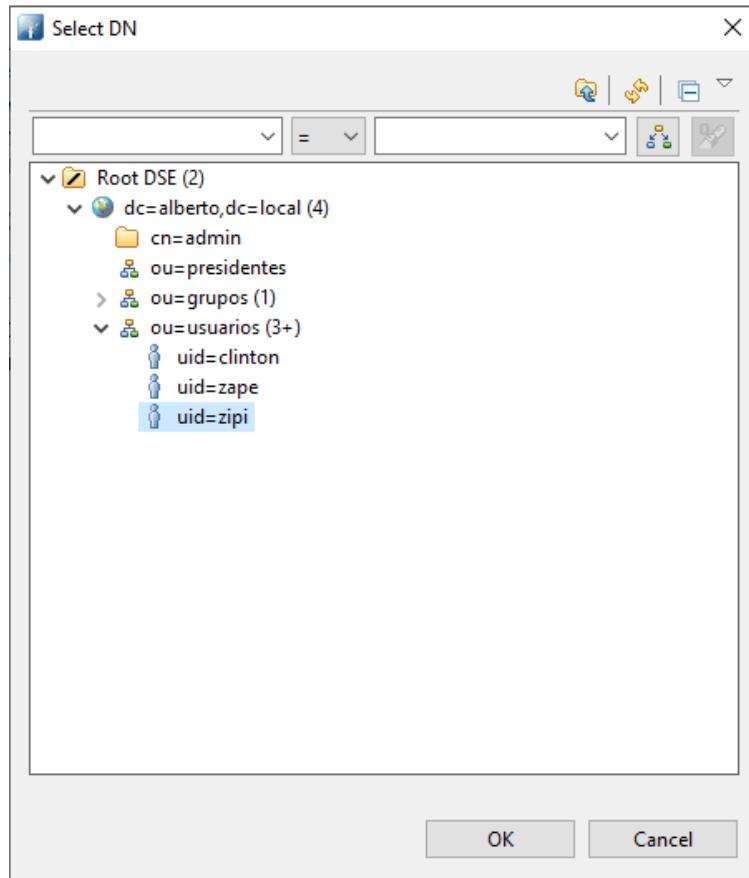


Haremos clic derecho en **ou=usuarios** y le daremos a **New** y después a **New Entry**.

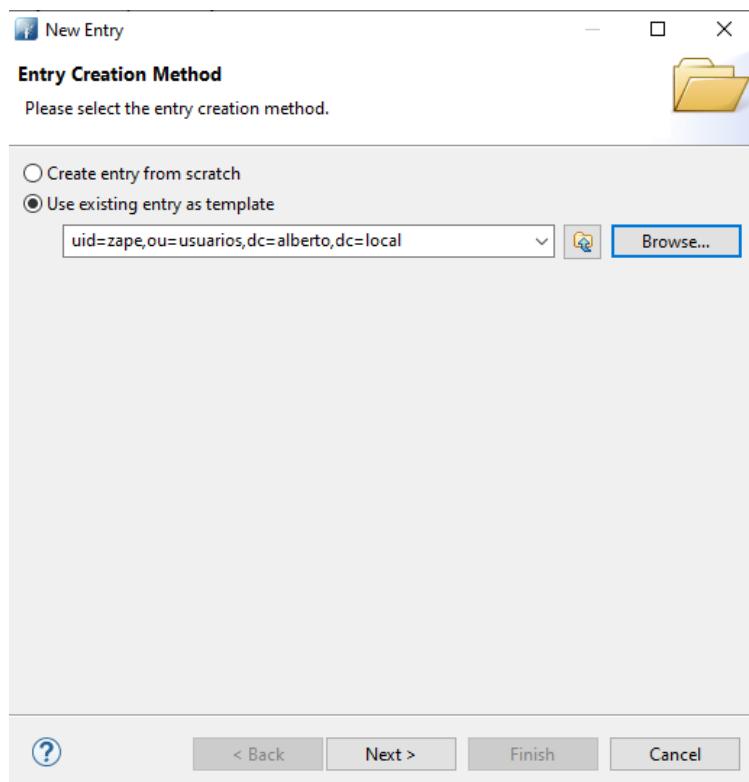
Una vez dentro de la ventana de configuración seleccionaremos **usar una plantilla existente** y le daremos a **Browse...** para seleccionarla.



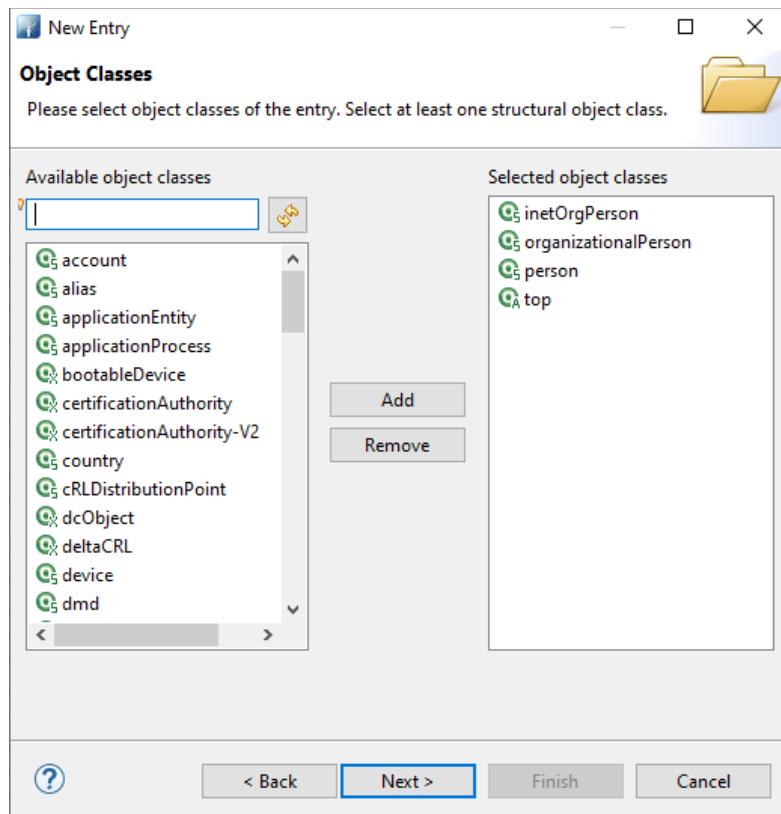
Seleccionaremos en este caso zipi o zape ya que ambas están creadas de la misma manera en nuestro servidor LDAP y nos sirven. Le daremos a **OK**.



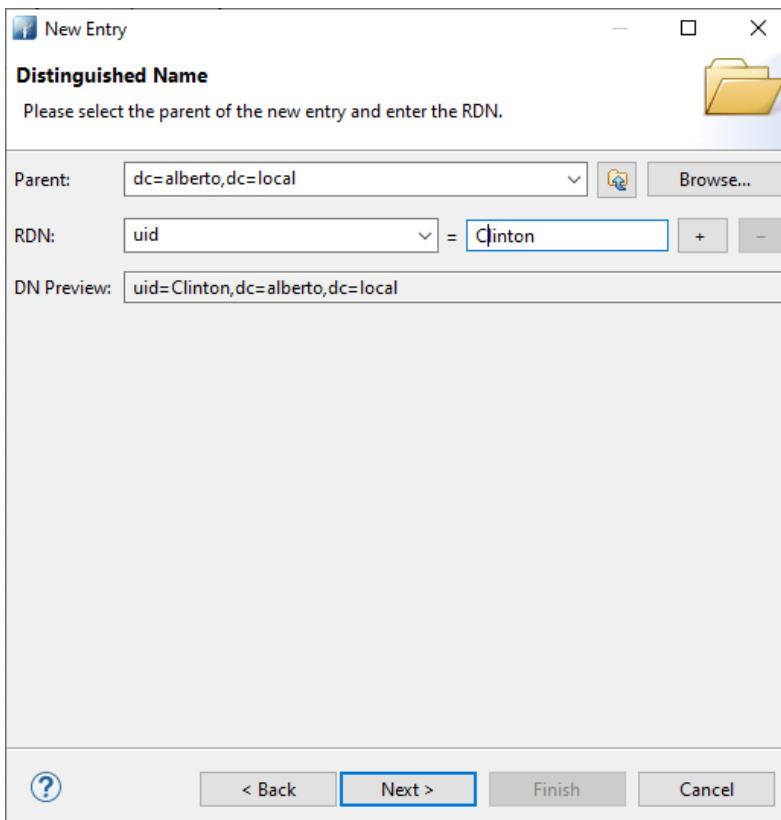
Una vez seleccionada ya podremos darle a **Next**.



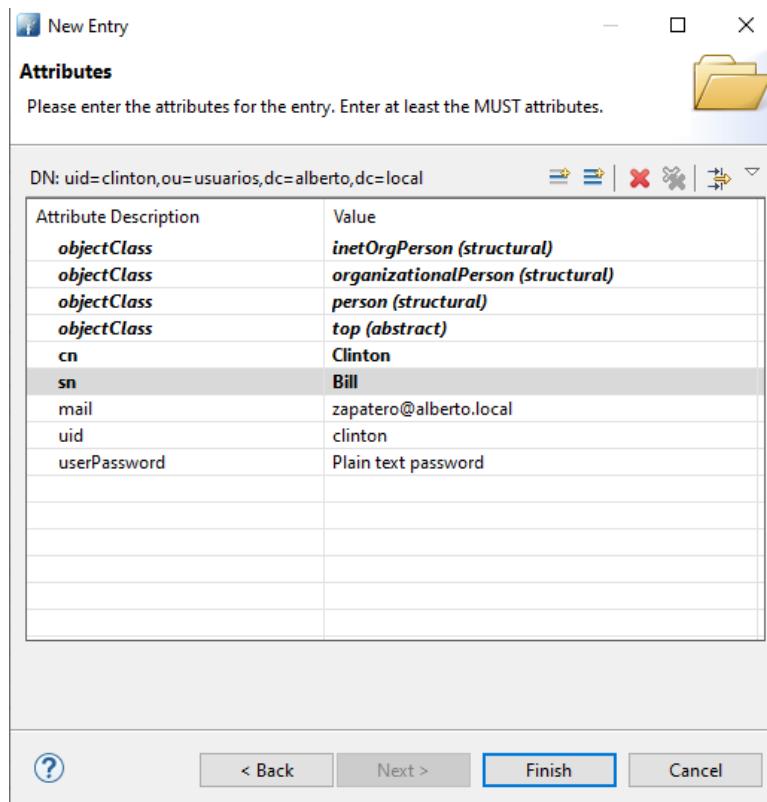
Las clases al usar una plantilla de usuario se seleccionarán automáticamente, le daremos a **Next**.



Ahora configuraremos el RDN: uid que en este caso será el nombre “Clinton” para el nuevo usuario. Le daremos a **Next**.

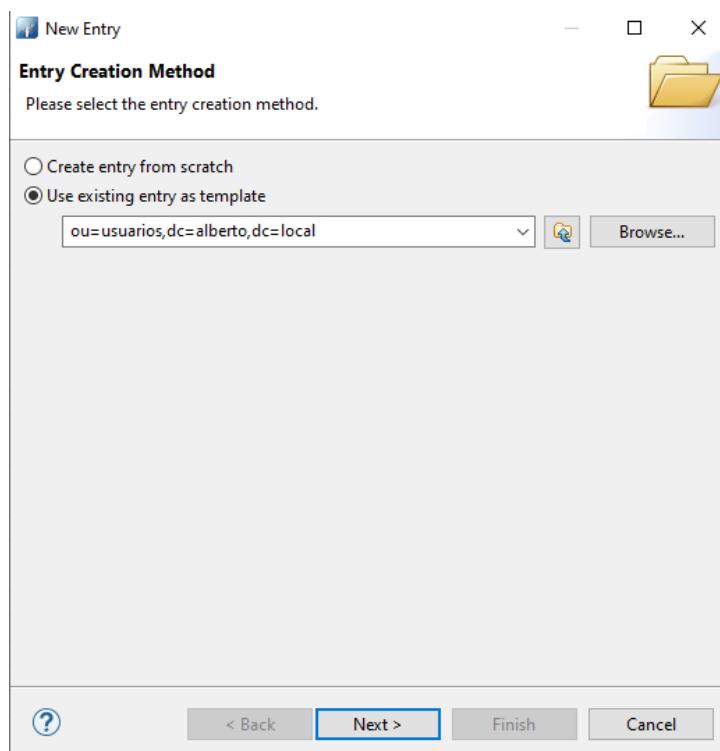


Por último, cambiaremos el **cn** y el **sn**, el **cn** será el **nombre del usuario** y el **sn** el **apellido** en mi caso. Una vez configurado le daremos a **Finish**.

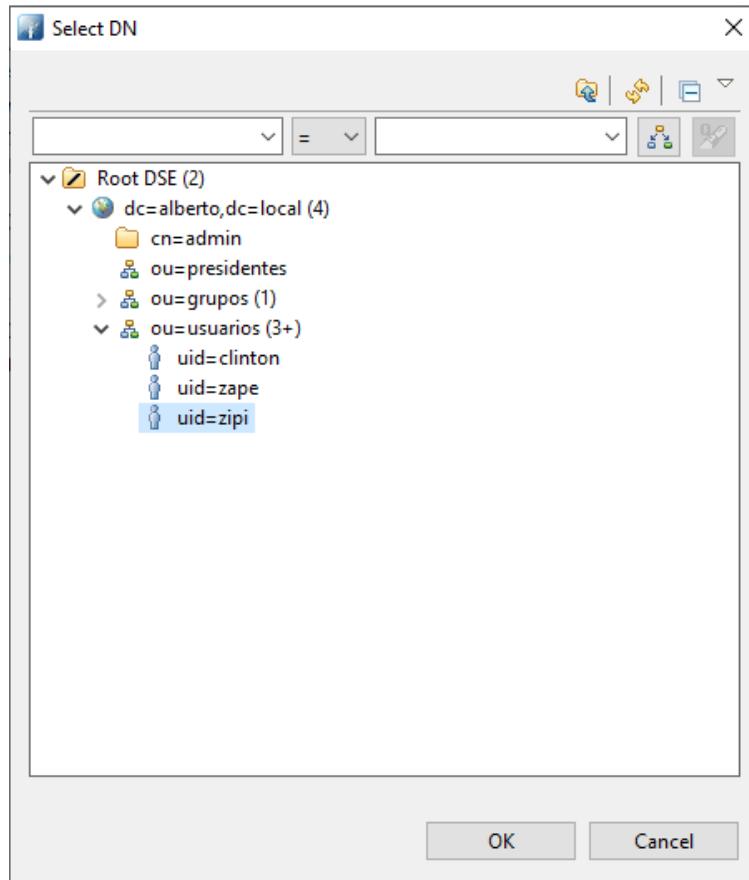


Para crear el otro usuario, seguiremos el mismo proceso. Haremos clic derecho en **ou=usuarios** y le daremos a **New** y después a **New Entry**.

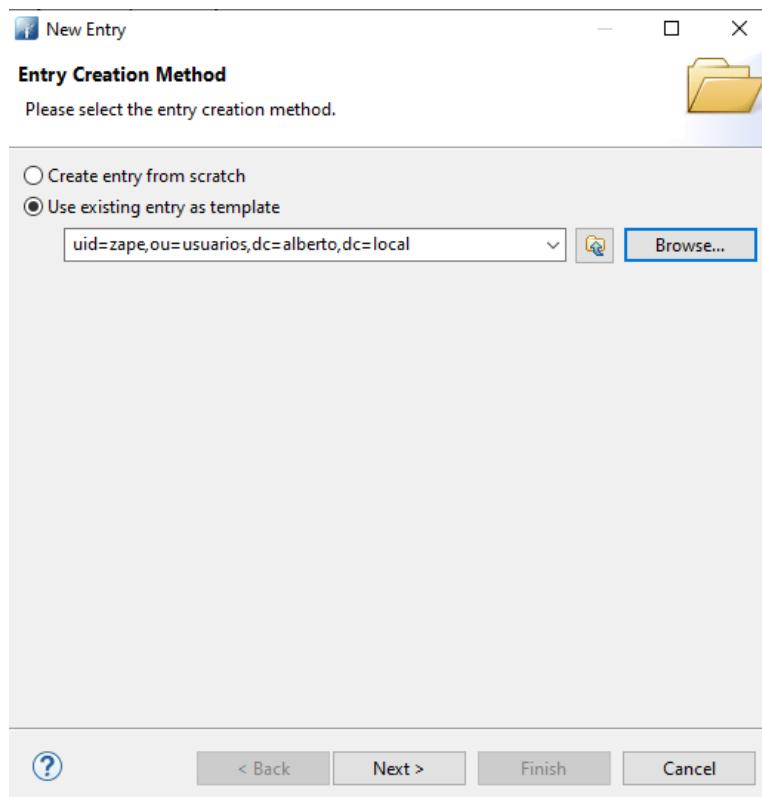
Una vez dentro de la ventana de configuración seleccionaremos **usar una plantilla existente** y le daremos a **Browse...** para seleccionarla.



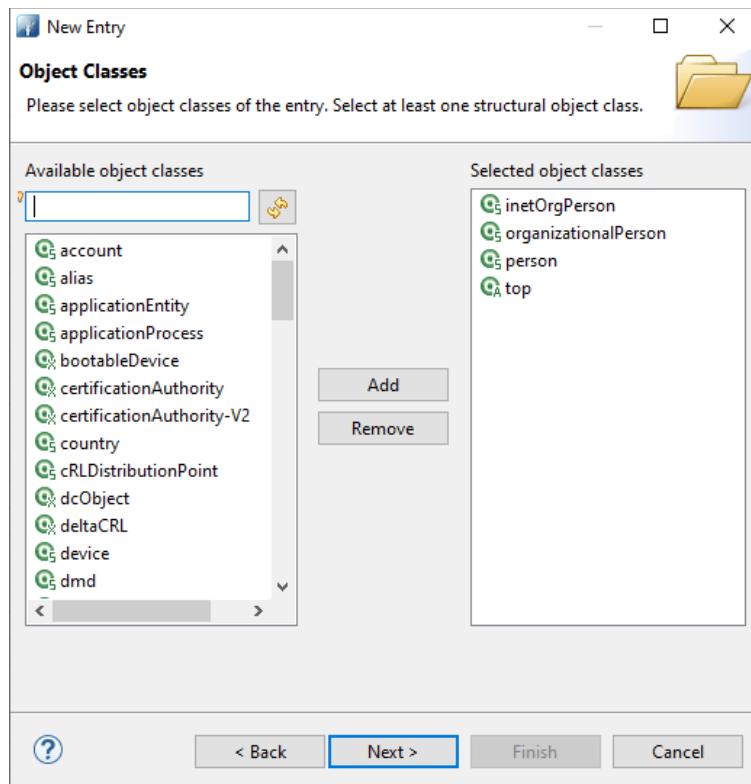
Seleccionaremos en este caso zipi o zape ya que ambas están creadas de la misma manera en nuestro servidor LDAP y nos sirven. Le daremos a **OK**.



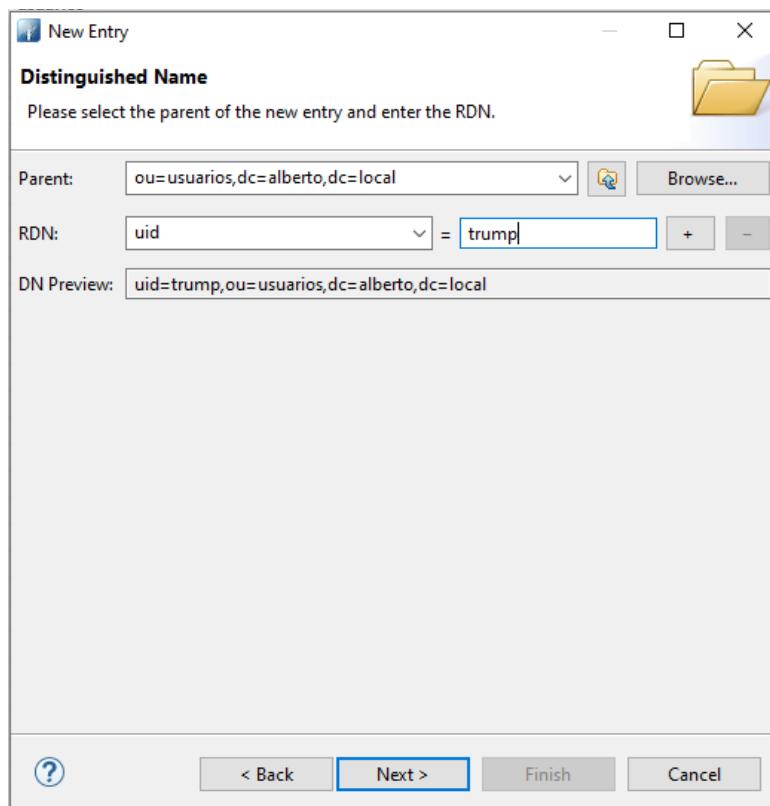
Una vez seleccionada ya podremos darle a **Next**.



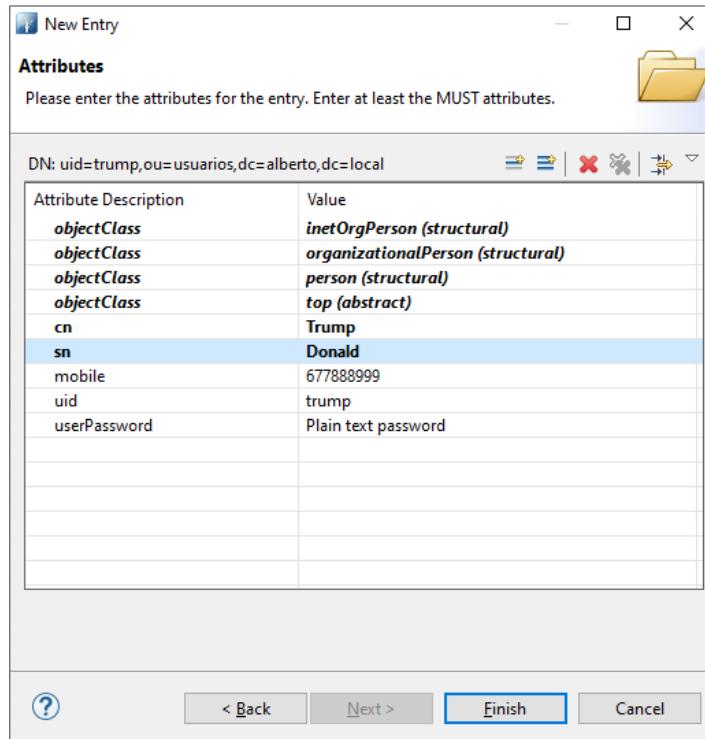
Las clases al usar una plantilla de usuario se seleccionarán automáticamente, le daremos a **Next**.



Ahora configuraremos el **RDN**: uid que en este caso será el nombre “Trump” para el nuevo usuario. Le daremos a **Next**.



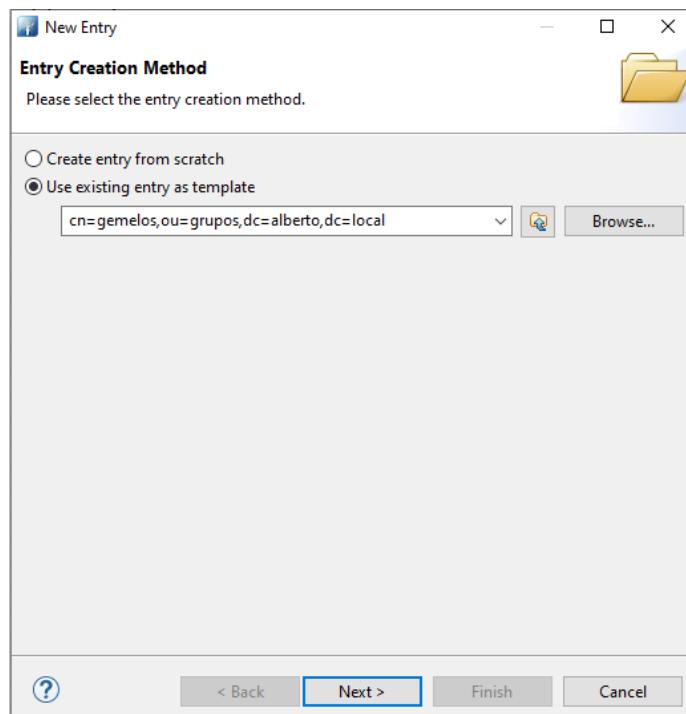
Por último, cambiaremos el **cn** y el **sn**, el **cn** será el nombre del usuario y el **sn** el apellido en mi caso. Una vez configurado le daremos a **Finish**.



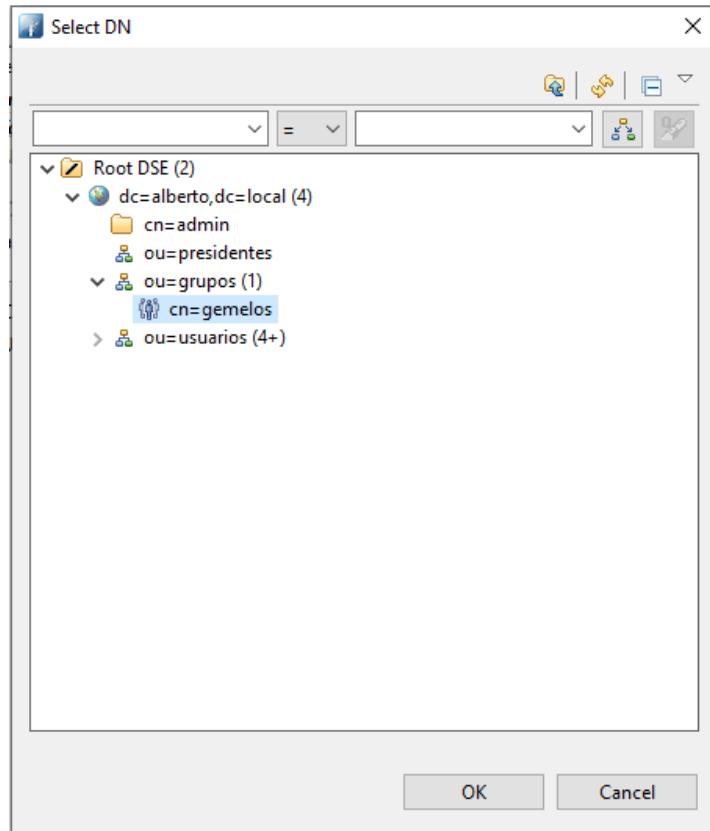
Lo último que haremos será crear el grupo presidentes y añadir los dos usuarios que hemos creado con anterioridad a este grupo.

Para ello haremos clic derecho en **ou=grupos** y le daremos a **New** y después a **New Entry**.

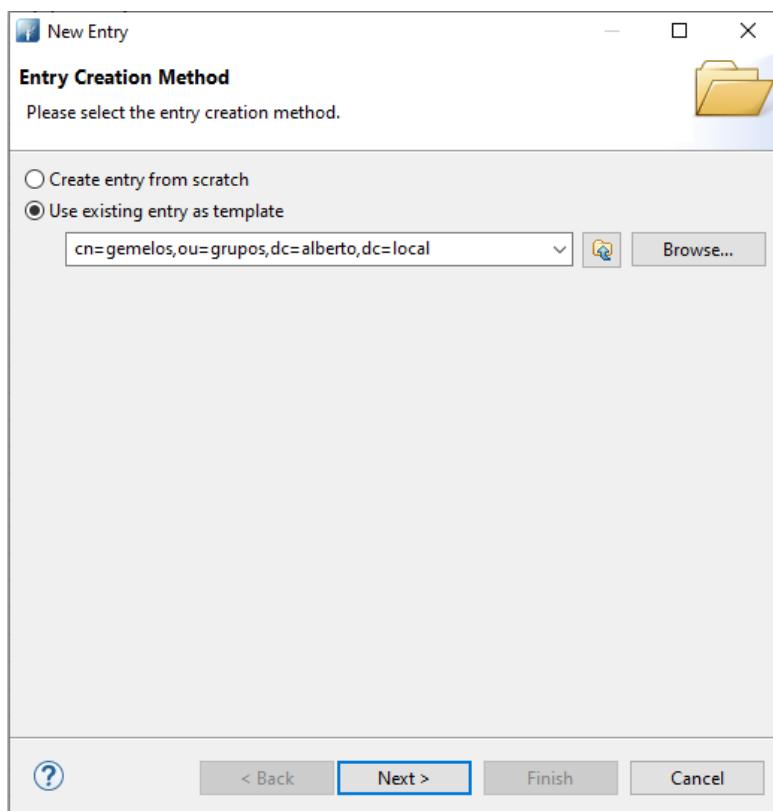
Una vez dentro de la ventana de configuración seleccionaremos **usar una plantilla existente** y le daremos a **Browse...** para seleccionarla.



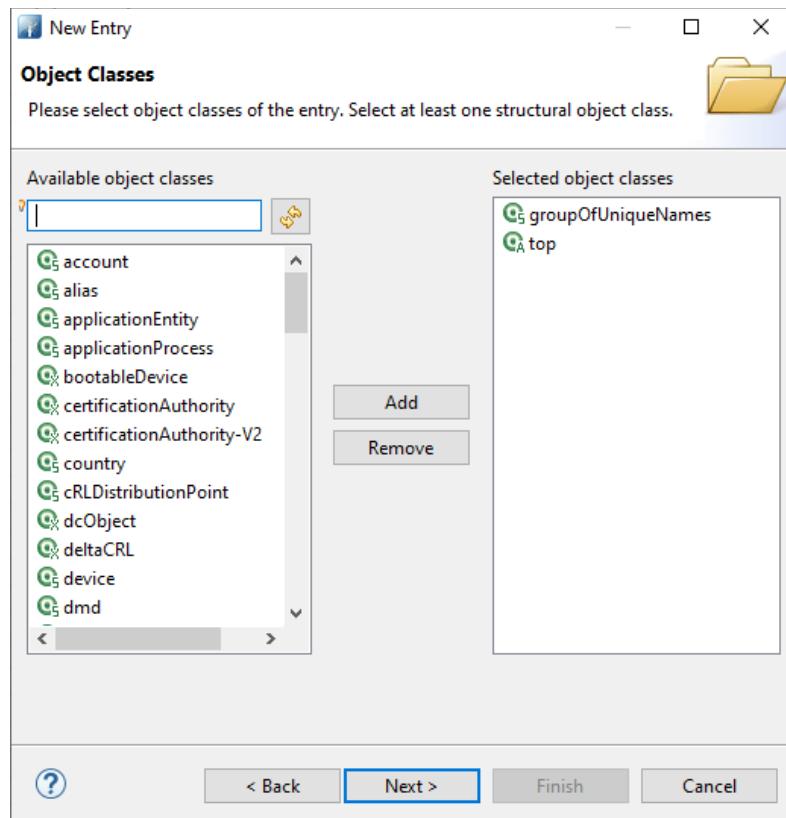
Seleccionaremos en este caso gemelos que está en nuestro servidor LDAP y nos sirve. Le daremos a **OK**.



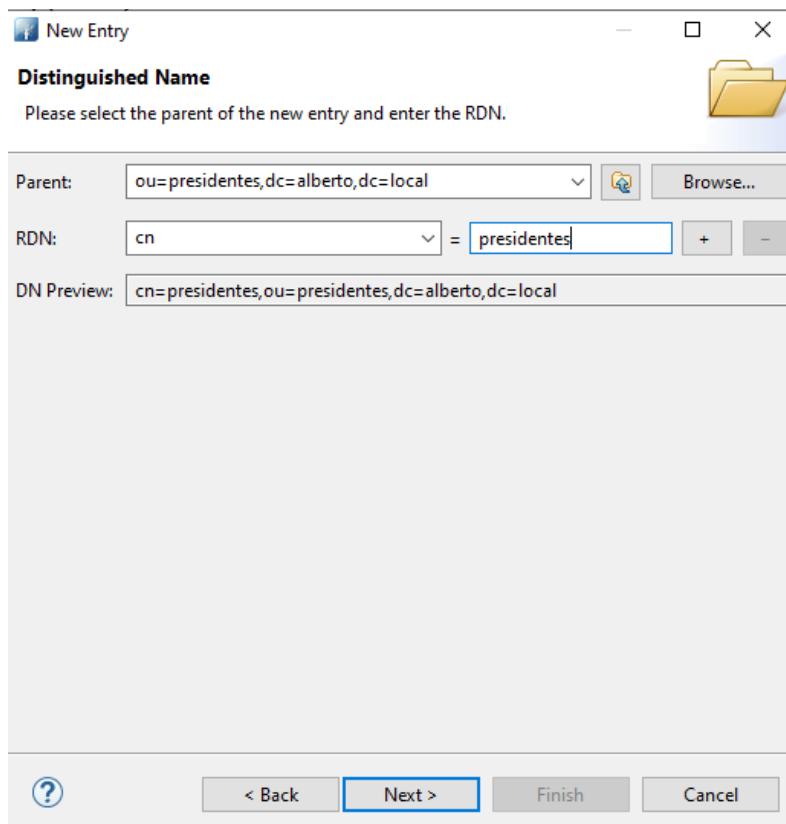
Una vez seleccionada ya podremos darle a **Next**.



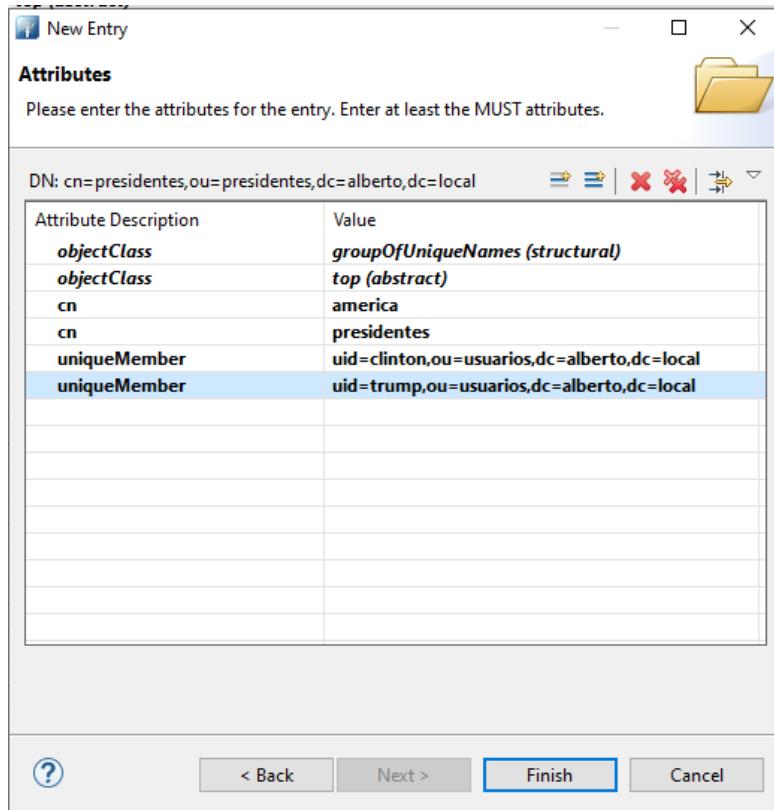
Las clases al usar una plantilla de usuario se seleccionarán automáticamente, le daremos a **Next**.



Ahora tendremos que asignar el **RDN**: **cn** que en este caso será el nombre del grupo nuevo. Le daremos a **Next**.



Por último, configuraremos el primer **cn a america** que será el tipo del grupo, después configuraremos los dos **uniqueMember** que serán los dos usuarios que hemos creado anteriormente, **tendrán uid=nombre,ou=usuarios,dc=tunombre,dc=local** y le daremos a **Finish**.



Con todo esto realizado, tendríamos configurado un nuevo grupo con dos nuevos usuarios dentro de una nueva unidad organizativa.

2. Windows 10

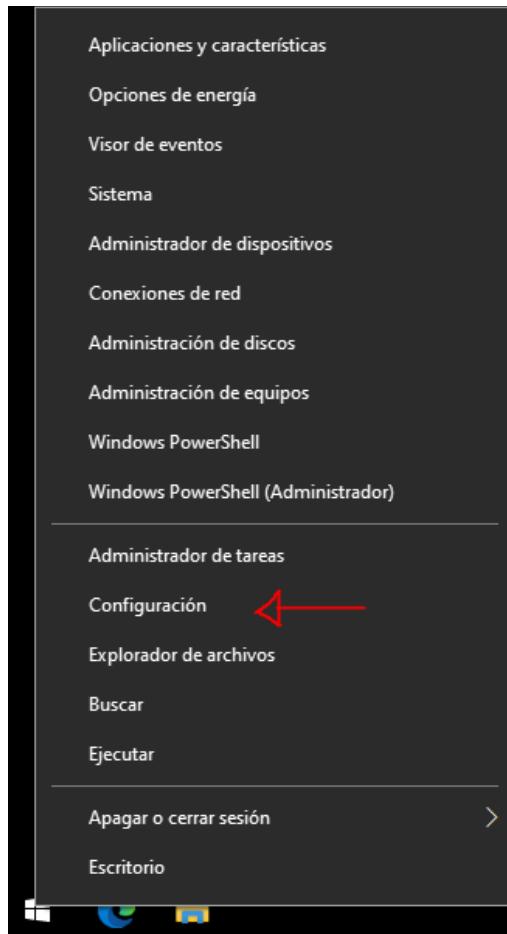
2.1 Datos de configuración maquina Windows 10

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: 192.168.3.20x Mascara: 255.255.255.0 Gateway: 192.168.3.1 DNS: 8.8.8.8

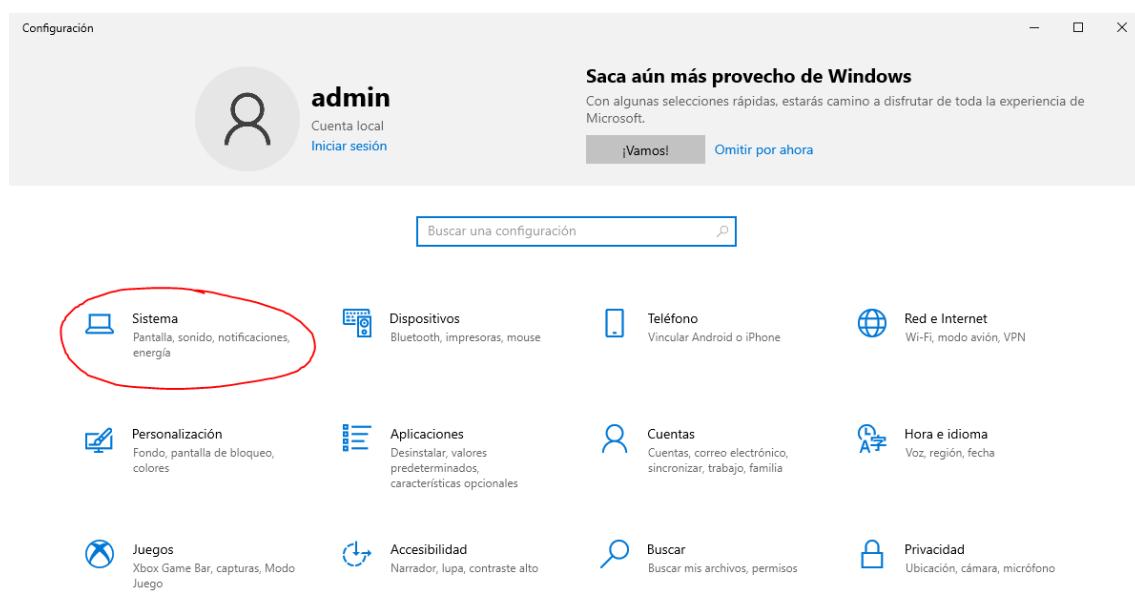
2.2 Cambiar nombre del host

Para cambiar el nombre del host en W10 lo primero que tienes que hacer es abrir el menú rápido haciendo clic derecho en el símbolo de Windows o con el atajo de teclado [Windows] + [X].

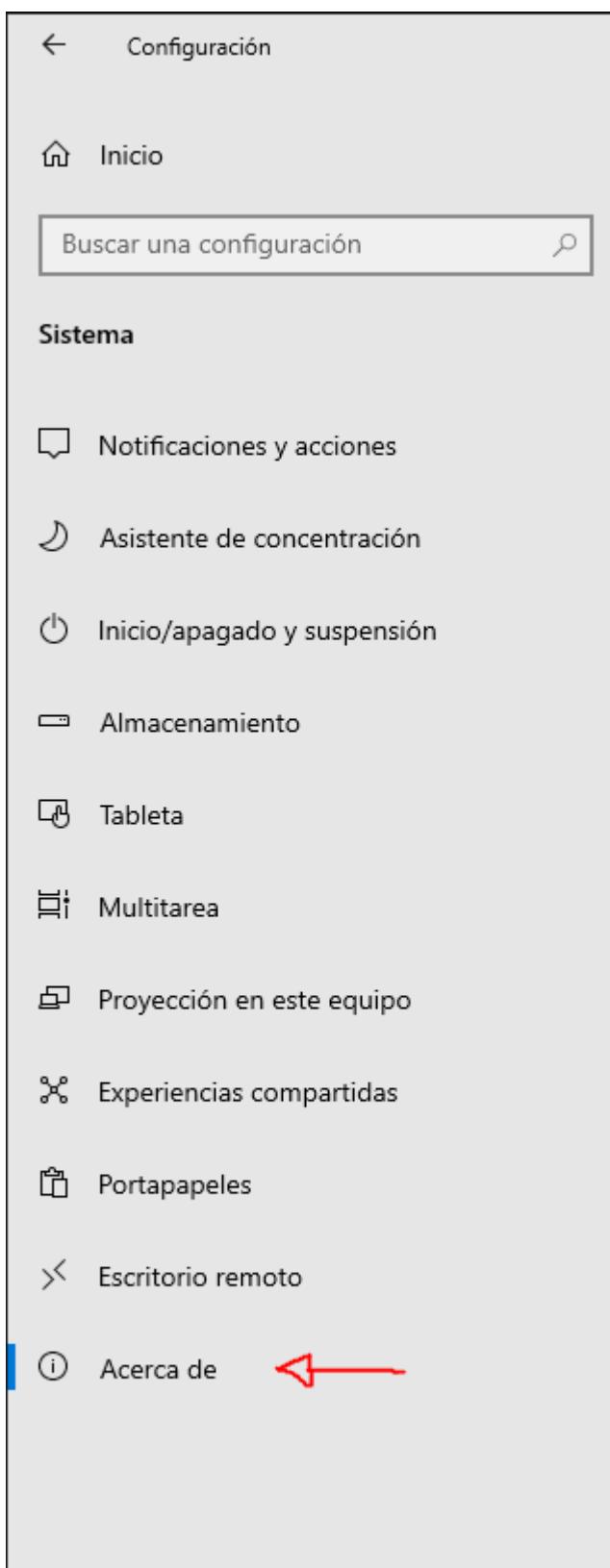
Selecciona “Configuración”.



En el cuadro de diálogo, selecciona “Sistema”.



En la barra de la izquierda, haz clic en “Acerca de”.



Haz clic en “Cambiar el nombre de este equipo”.

Acerca de

Tu equipo está supervisado y protegido.

[Ver detalles en Seguridad de Windows](#)

Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-V1CE0G5	Opciones de configuración relacionadas
Procesador	Intel(R) Core(TM) i5-9600K CPU @ 3.70GHz 3.70 GHz	Configuración de BitLocker
RAM instalada	8,00 GB	Administrador de dispositivos
Identificador de dispositivo	943C93B0-E12F-4FB2-A479-9EDDB6F98101	Escritorio remoto
Id. del producto	00330-80000-00000-AA563	Protección del sistema
Tipo de sistema	Sistema operativo de 64 bits, procesador basado en x64	Configuración avanzada del sistema
Lápiz y entrada táctil	La entrada táctil o manuscrita no está disponible para esta pantalla	Cambiar el nombre de este equipo (avanzado)

[Copiar](#)

[Cambiar el nombre de este equipo](#)

Especificaciones de Windows

Edición	Windows 10 Pro
Versión	21H1
Instalado el	22/09/2021
Compilación del sistema operativo	19043.928
Experiencia	Windows Feature Experience Pack

Ahora puedes cambiar el nombre del host, una vez cambiado le daremos a siguiente.

Cambiar el nombre de tu PC

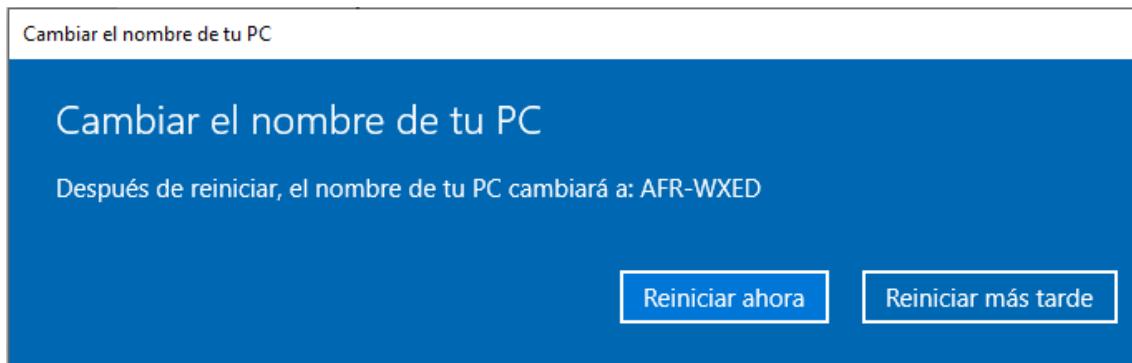
Cambiar el nombre de tu PC

Puedes usar una combinación de letras, guiones y números.

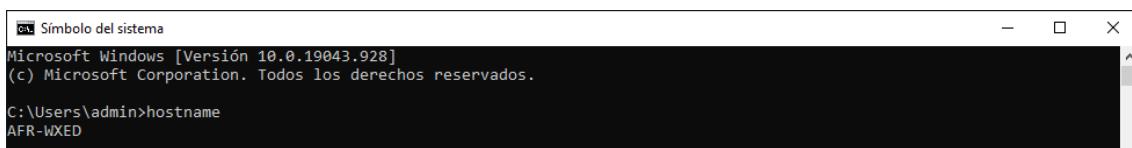
Nombre actual del equipo: AFR-USED

AFR-WXED

Ahora reiniciaremos el sistema para que se apliquen los cambios.

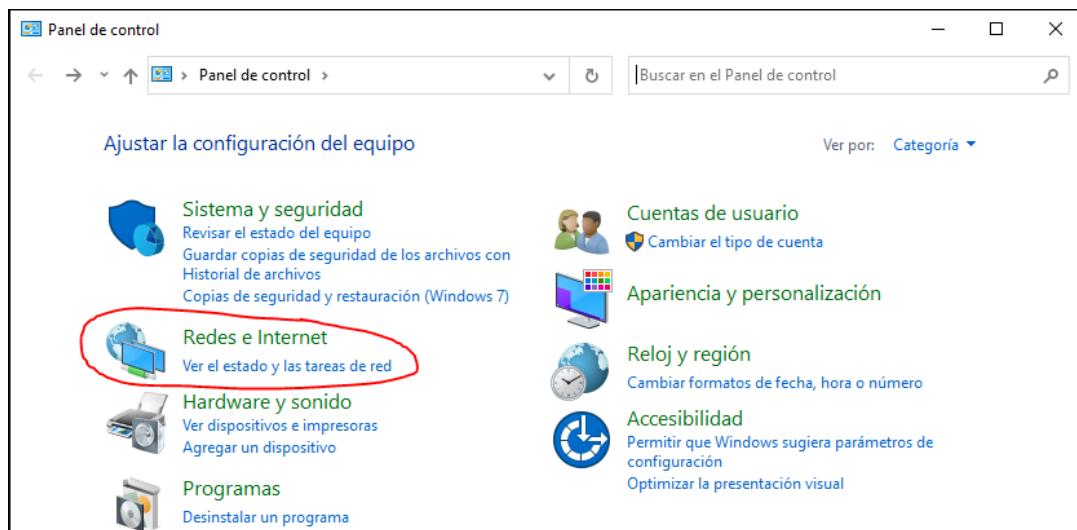


Una vez reiniciado el sistema, abriremos el CMD y ejecutaremos el comando `hostname` para ver que el nombre del host se ha cambiado correctamente.

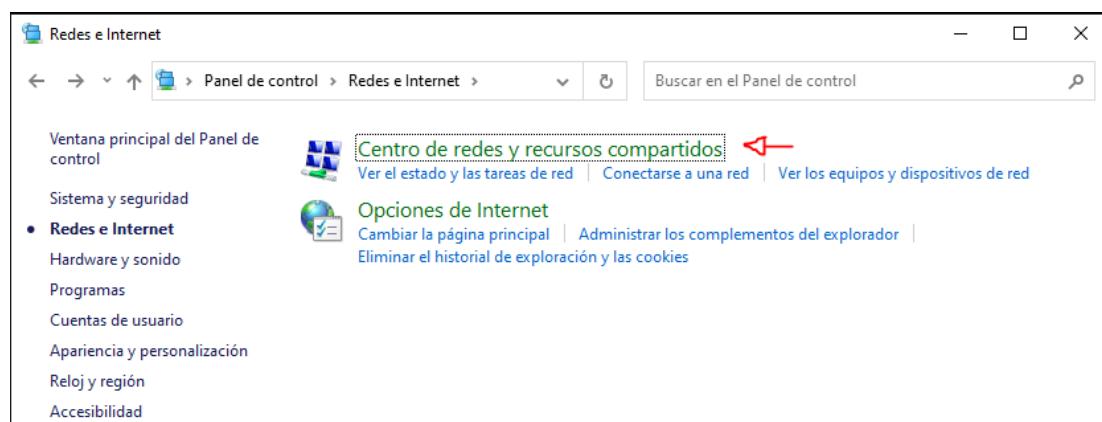


2.3 Configurar dirección IP

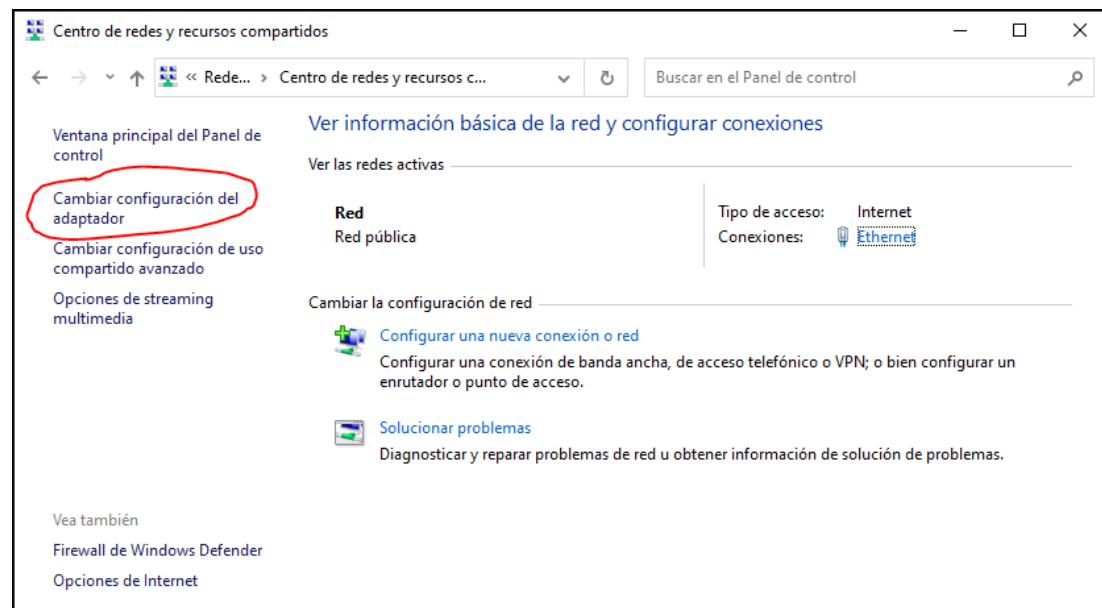
Lo primero que haremos será irnos al panel de control del equipo, una vez dentro entraremos en “Redes e Internet”.



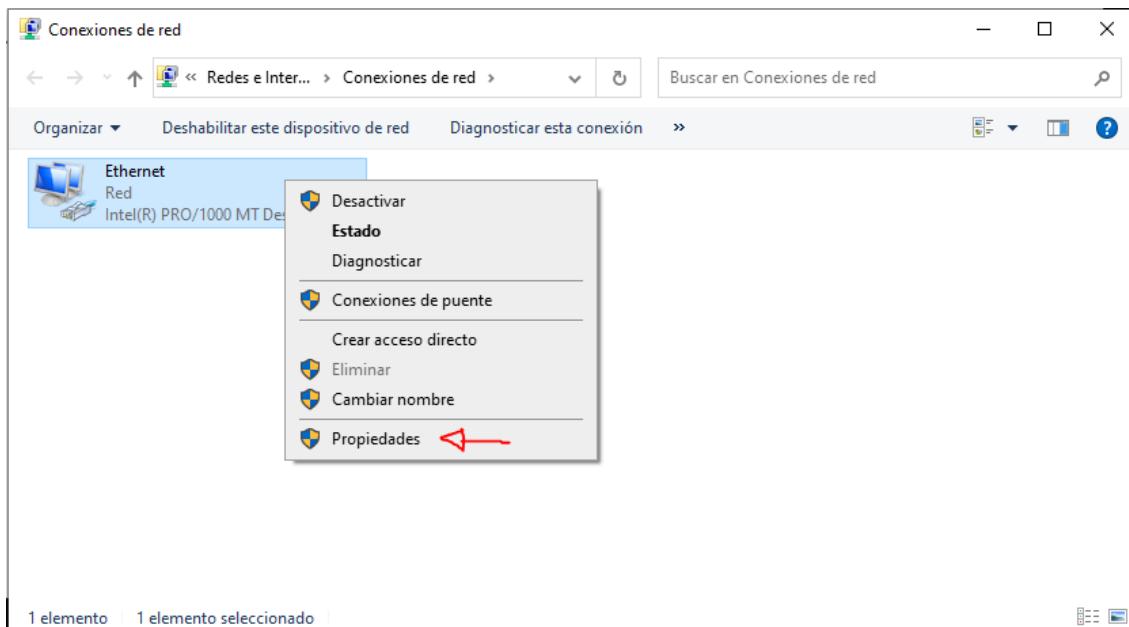
Dentro de redes e internet, entraremos en “Centro de redes y recursos compartidos”.



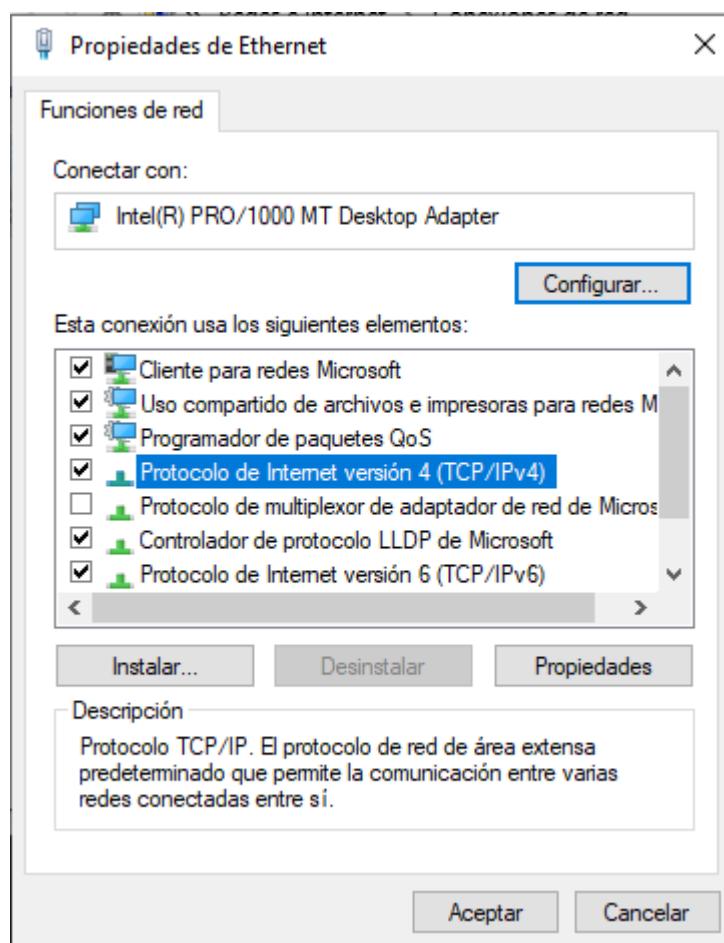
Una vez dentro, entraremos a “Cambiar configuración del adaptador”.



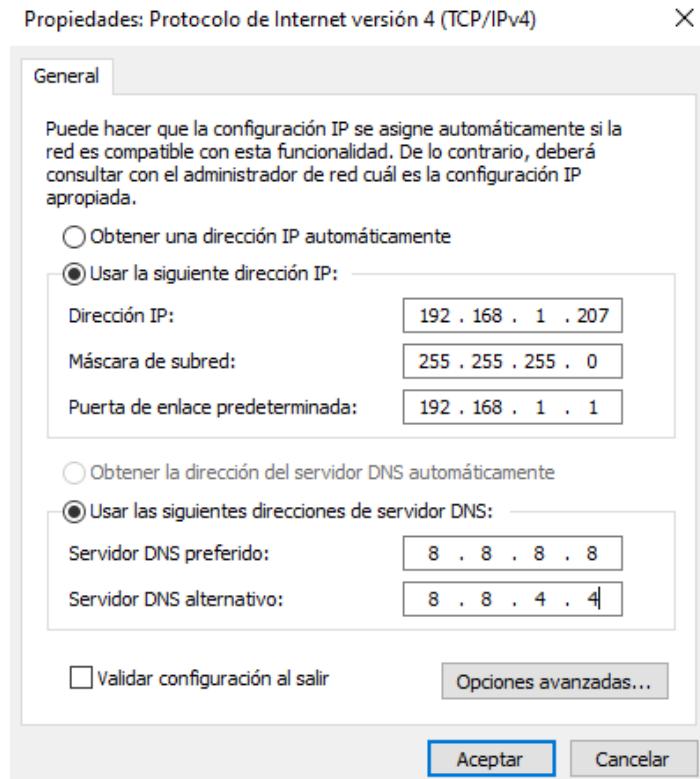
Dentro de las conexiones de red, en la conexión que nos sale, daremos clic derecho y en el menú le daremos a **Propiedades**.



Se nos abrirán las propiedades del adaptador, para configurar la IP nos iremos a el “Protocolo de Internet versión 4” y le daremos a **Propiedades**.



Una vez dentro, procederemos a configurar la IP, la máscara de subred, la puerta de enlace y los servidores DNS. Una vez configurados le damos a **Aceptar**.



Para comprobar que la configuración se ha guardado correctamente, nos iremos al CMD y ejecutaremos el comando *ipconfig* para ver la configuración del adaptador.

También podemos probar si tenemos conexión a internet haciendo una prueba de conexión con el siguiente comando `ping 8.8.8.8`

```
C:\Users\admin>ping 8.8.8.8

Haciendo ping a 8.8.8.8 con 32 bytes de datos:
Respuesta desde 8.8.8.8: bytes=32 tiempo=12ms TTL=116
Respuesta desde 8.8.8.8: bytes=32 tiempo=13ms TTL=116
Respuesta desde 8.8.8.8: bytes=32 tiempo=13ms TTL=116
Respuesta desde 8.8.8.8: bytes=32 tiempo=12ms TTL=116

Estadísticas de ping para 8.8.8.8:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
                (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 12ms, Máximo = 13ms, Media = 12ms
```

2.4 Instalación de navegador Mozilla

2.4.1 Instalación

Para instalar el navegador Mozilla, lo primero será buscar en el buscador “Mozilla”. Una vez se encuentra la búsqueda, entramos en la página web de mozilla.org.

También podemos ir directamente a la [página web de descarga](https://mozilla.org).

Aproximadamente 2.360.000.000 resultados (0,48 segundos)

<https://www.mozilla.org> > es-ES > firefox > new ▾

Descarga Navegador Firefox — Rápido, privado y gratis

Descarga **Firefox**, un navegador web gratuito proporcionado por **Mozilla**, una organización sin ánimo de lucro dedicada a la salud de internet y a la privacidad ...

[Todos los idiomas](#) · [Firefox para Android](#) · [Características de Firefox](#)

<https://www.mozilla.org> > es-ES > firefox ▾

Firefox - Protege tu vida en línea con productos que ... - Mozilla

Firefox es mucho más que un navegador. Descubre más detalles sobre los productos **Firefox** que gestionan tus datos con respeto y que están diseñados para ...

Una vez dentro de la web oficial de Mozilla, le daremos al botón **Descargar Firefox** para iniciar la descarga del instalador.

moz://a Navegadores Firefox Productos Quiénes somos Innovación Descargar Firefox

Firefox para escritorio Características Ayuda Complementos Todos los idiomas

Firefox Browser

Descarga el navegador que protege lo que es importante

No hay políticas de privacidad sospechosas o puertas traseras para los anunciantes. Solo un navegador súper rápido que no te vende.

Descargar Firefox

Aviso de privacidad de Firefox Opciones de descarga y otros idiomas Ayuda de Navegador Firefox

Una vez se descargue, nos vamos a descargas y ejecutamos el instalador.

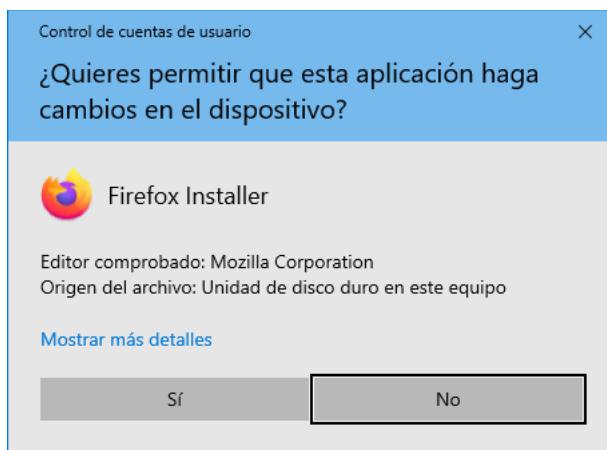
↓ | ↻ | ↺ | Descargas Archivo Inicio Compartir Vista

Este equipo > Descargas

Nombre	Fecha de modificación	Tipo	Tamaño
Firefox Installer	07/12/2021 0:26	Aplicación	327 KB

Acceso rápido Escritorio Descargas Documentos Imágenes Música proyectoPrueba Videos Este equipo Red

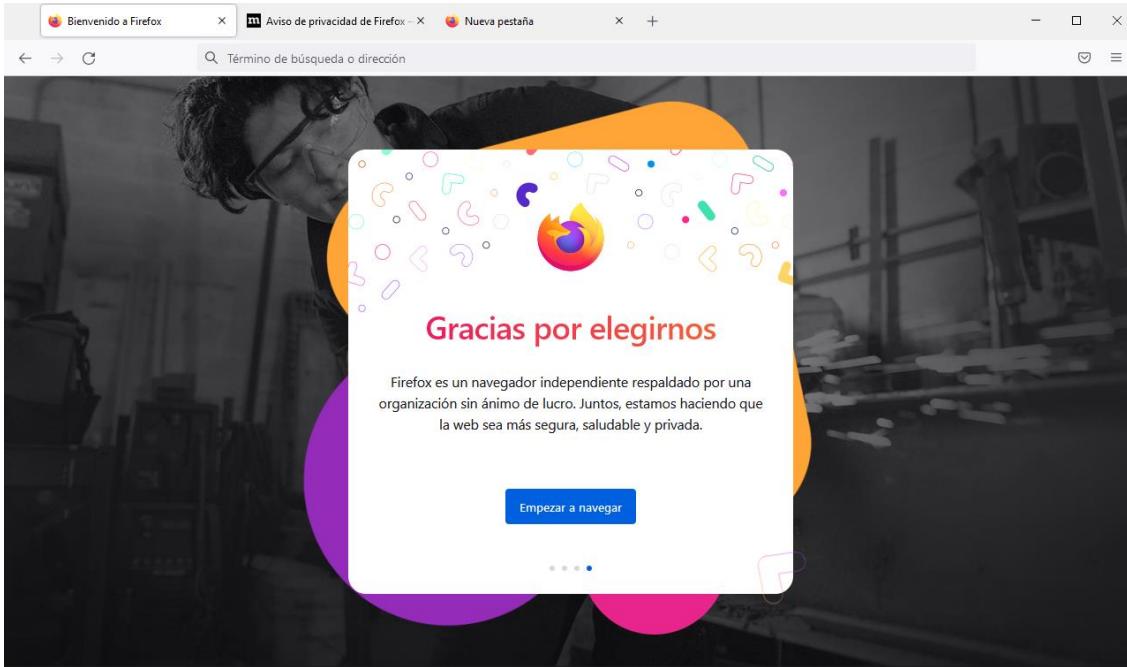
Nos pedirá permisos para realizar la instalación, le daremos a que sí.



El instalador arrancara y se instalará automáticamente en nuestro equipo sin necesidad de dar ninguna indicación más.



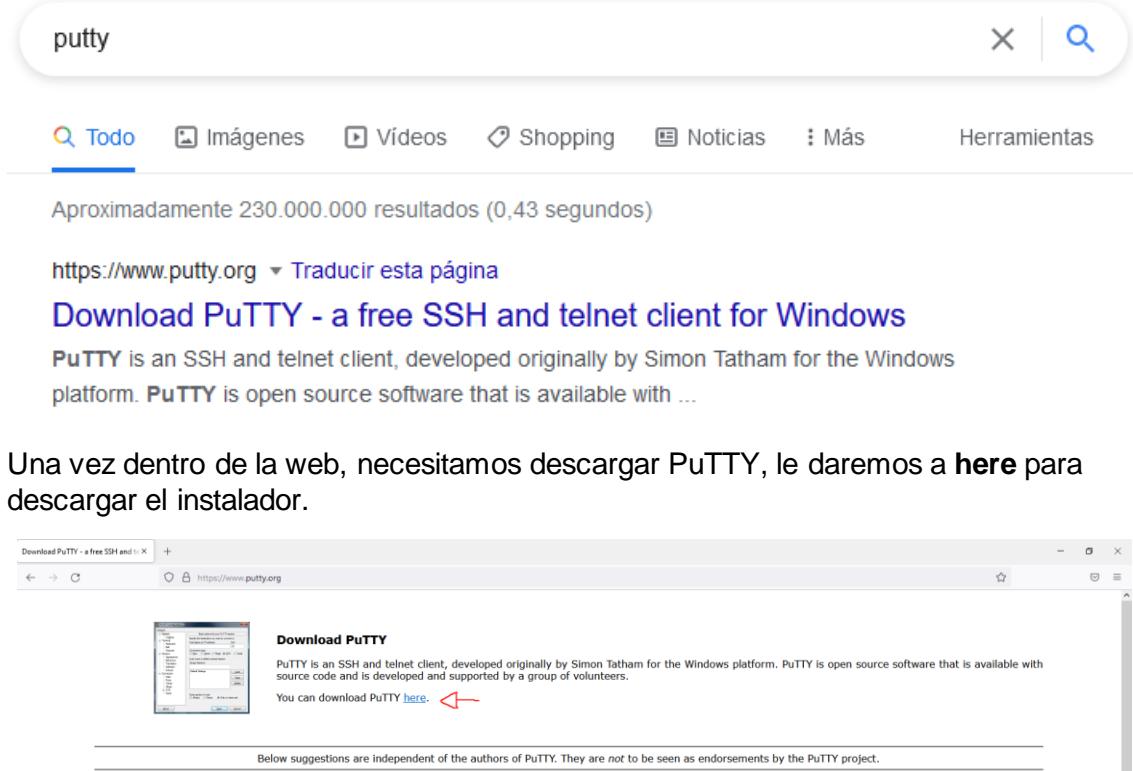
Una vez finalice la instalación, el navegador se abrirá automáticamente y ya estará disponible para su uso.



2.5 Instalación de PuTTY

2.5.1 Instalación

Lo primero será buscar en el navegador “PuTTY” y entrar a la página oficial [putty.org](https://www.putty.org). También podemos ir directamente a la [página web de descarga](#).



Aproximadamente 230.000.000 resultados (0,43 segundos)

<https://www.putty.org> ▾ Traducir esta página

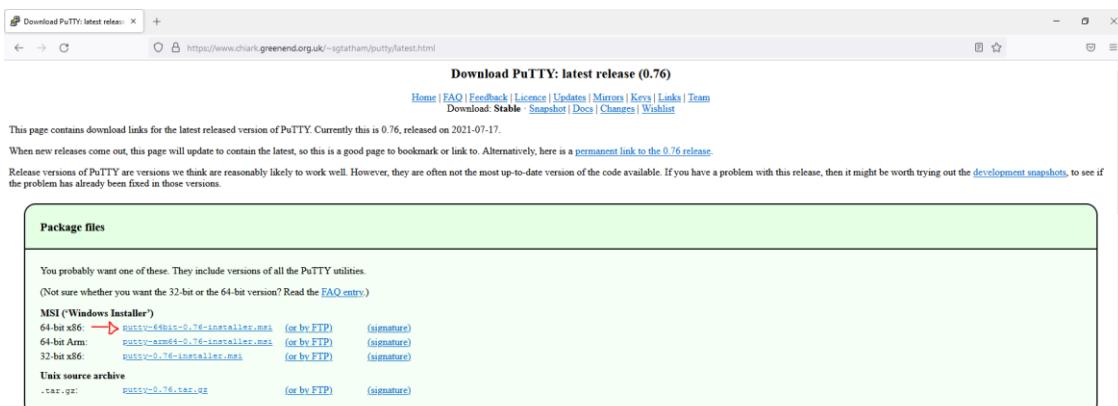
Download PuTTY - a free SSH and telnet client for Windows

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with ...

Una vez dentro de la web, necesitamos descargar PuTTY, le daremos a **here** para descargar el instalador.

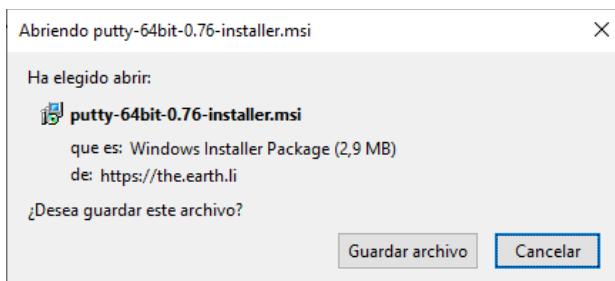


Nos llevará a la página en la que nos dice qué paquete de archivos descargar, en nuestro caso el Instalador de 64-bits x86.

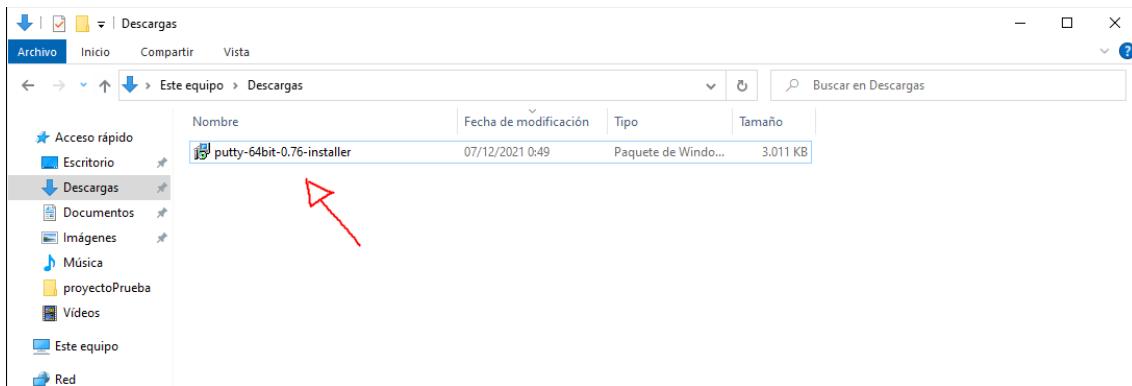


MSI ("Windows Installer")
64-bit x86: [putty-64bit-0.76-installer.msi](#) (or by FTP) (signature)
64-bit Arm: [putty-arm64-0.76-installer.msi](#) (or by FTP) (signature)
32-bit x86: [putty-0.76-installer.msi](#) (or by FTP) (signature)

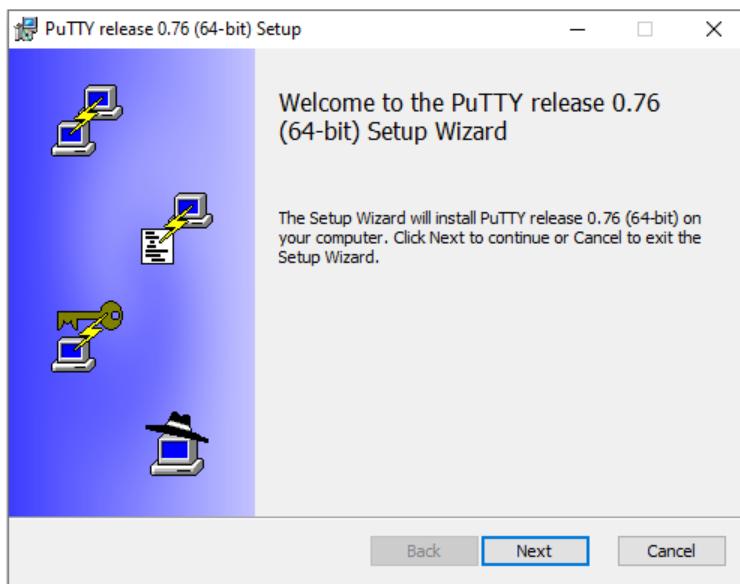
Nos preguntará si queremos guardar el archivo, le damos a **Guardar archivo**.



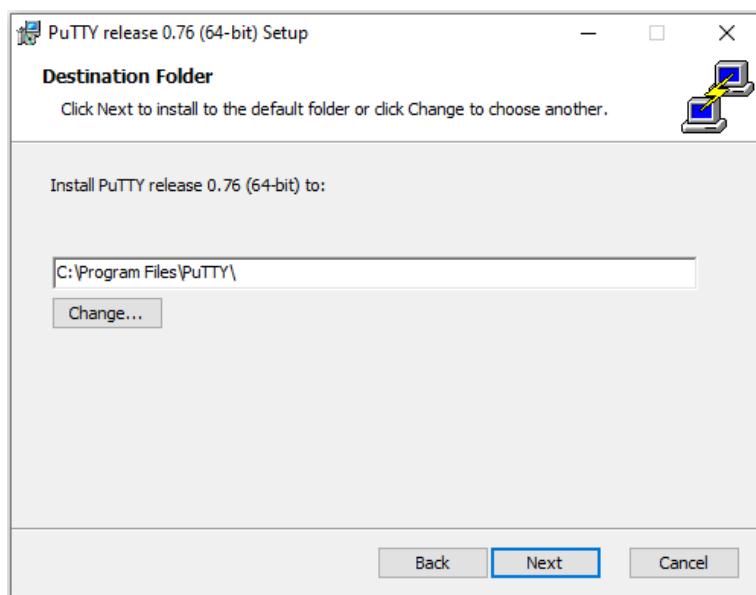
Una vez se descargue, nos vamos a descargas y ejecutamos el instalador.



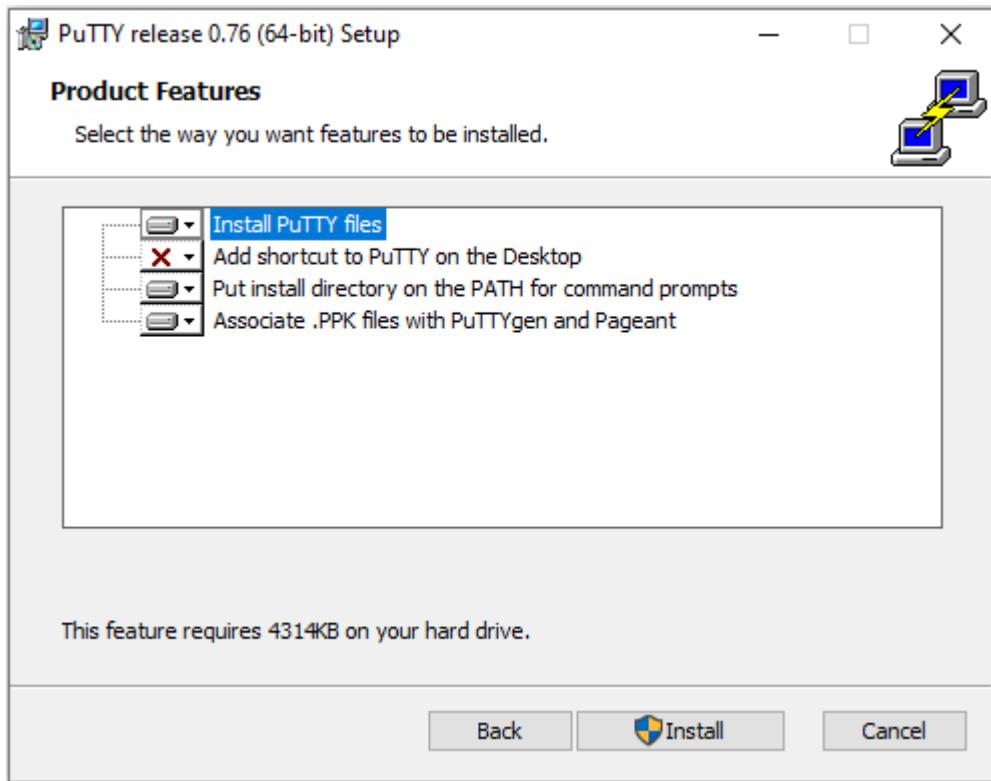
Una vez iniciado el instalador, se abrirá la primera ventana que da la bienvenida al instalador, le daremos a **next**.



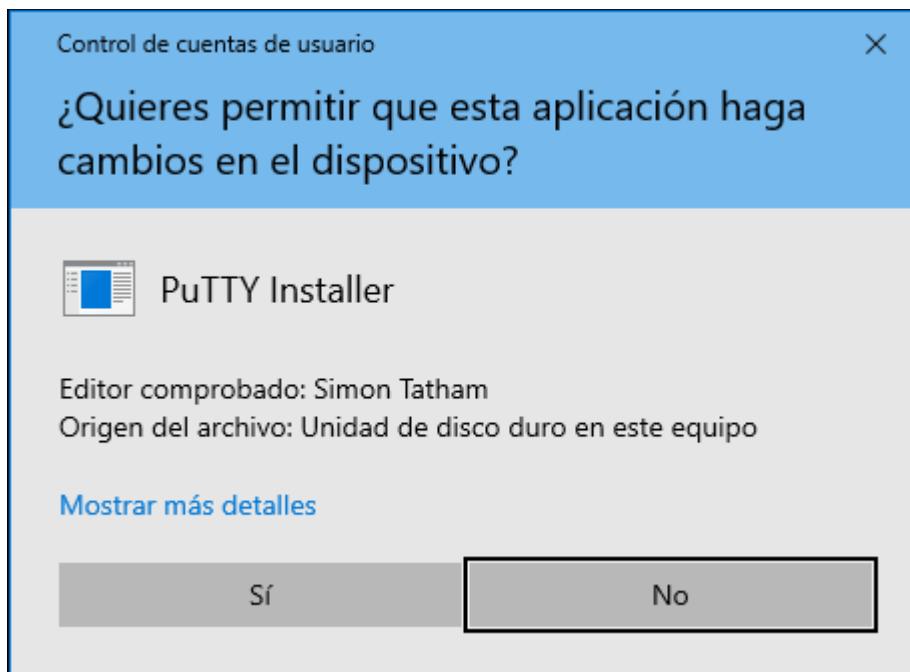
Nos preguntara en que directorio queremos que se instale, en mi caso lo he dejado por defecto. Una vez seleccionado el directorio, le daremos a **next**.



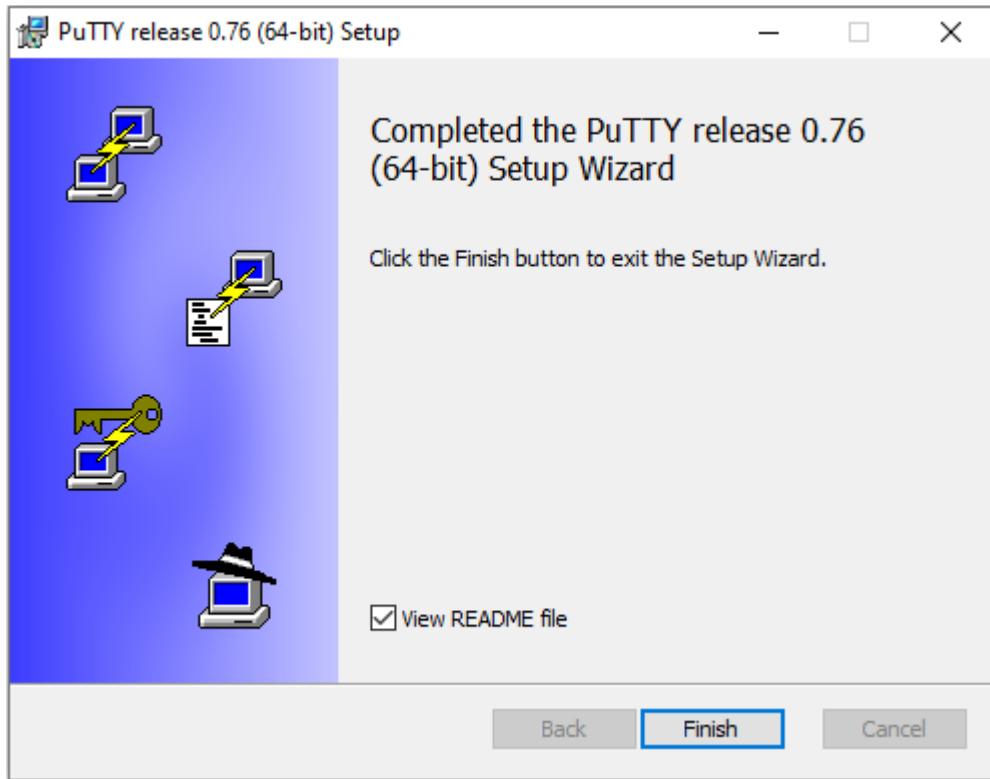
Ahora nos preguntara que características queremos instalar, en mi caso lo he dejado por defecto. Una vez seleccionadas las necesarias, le daremos a **Install**.



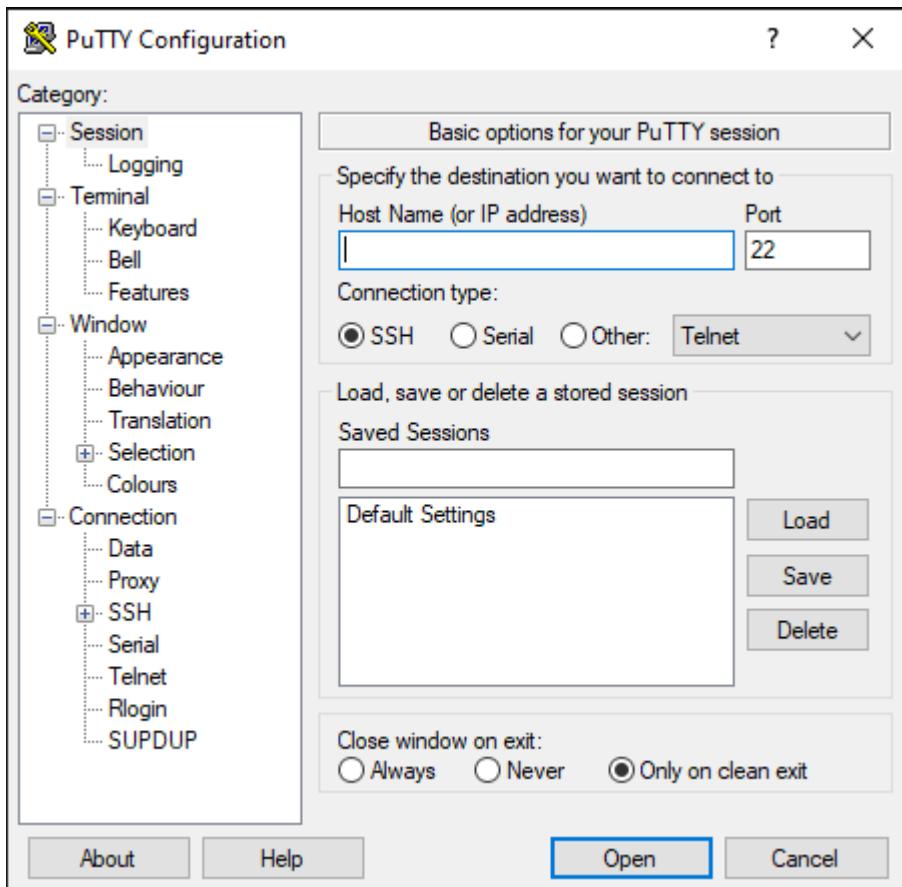
Nos pedirá permisos para realizar la instalación, le daremos a que **sí**.



Una vez se realice la instalación con éxito nos mostrara que ha sido completada, le daremos a **Finish**.



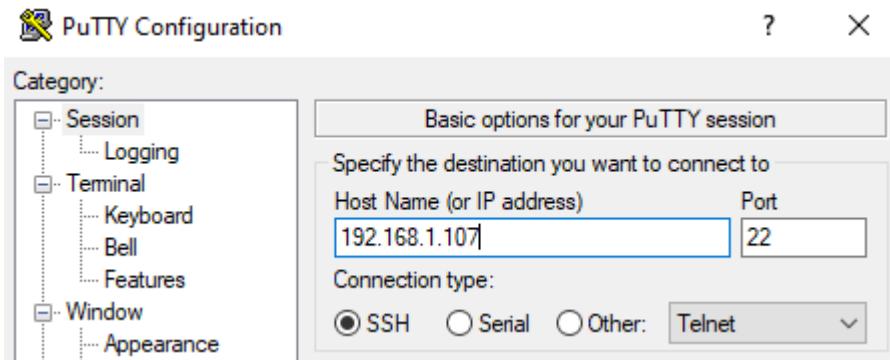
Una vez instalado, podremos ya abrir el PuTTY.



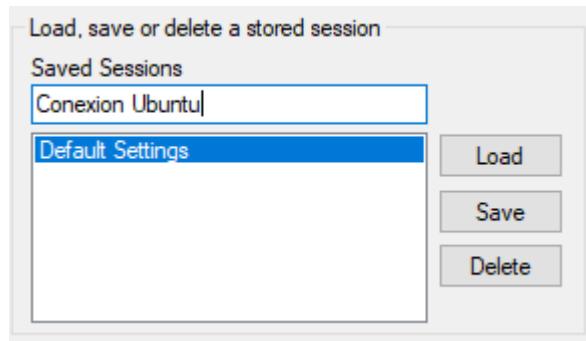
2.5.2 Configuración

Ahora que tenemos el PuTTY instalado, vamos a configurar la conexión a nuestro servidor de Ubuntu por SSH.

Con el PuTTY abierto, dentro de **Session** vamos a introducir la IP del servidor, el puerto será el 22 y el tipo de conexión será SSH.

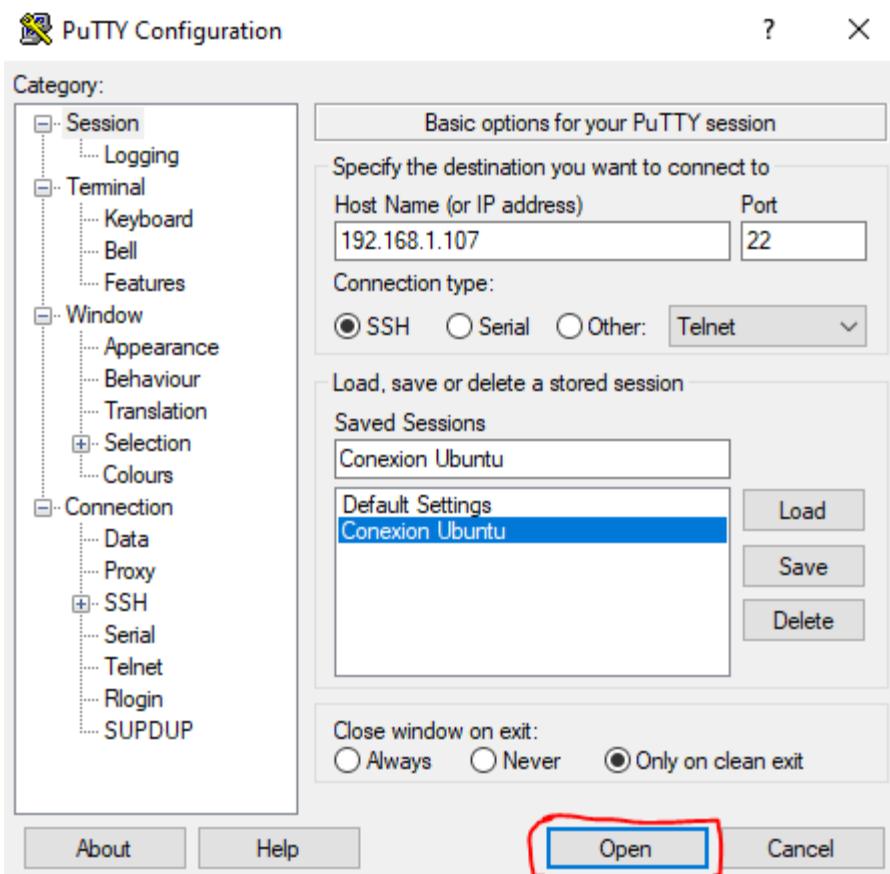


Le ponemos un nombre a la sesión para guardarla y no tener que introducir la configuración cada vez que tengamos que hacer la conexión. Le damos a **Save** para guardarla.

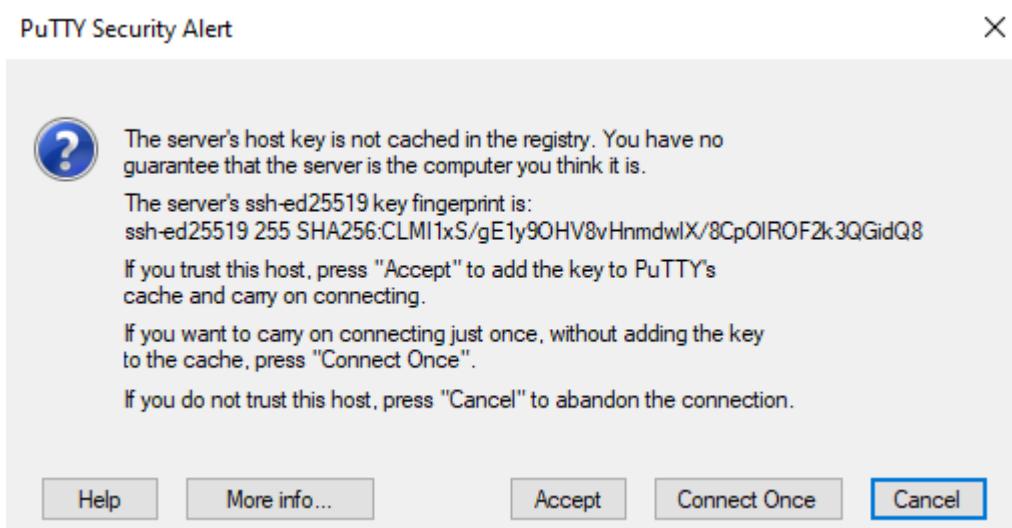


2.5.3 Ejemplo de uso

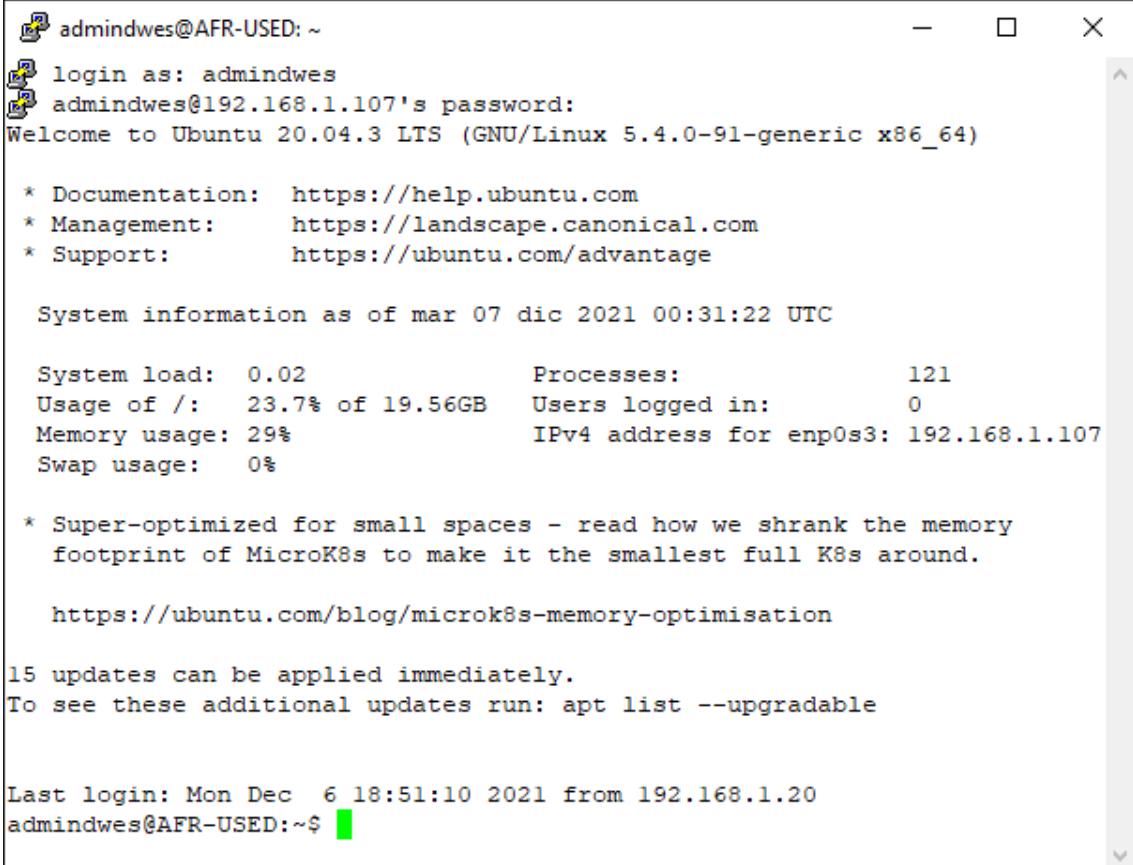
Para poder conectarnos a la conexión configurada anteriormente, solo tendremos que seleccionar la conexión que queremos en el apartado de Saved Sessions y darle a **Open**.



Nos dirá que la clave de la conexión no esta en el registro, en este caso como conocemos el host, le daremos a **Accept** para continuar y que nos guarde la clave de conexión en la cache para futuras conexiones.



Se nos abrirá la consola, aquí nos pedirá el login, solamente habrá que iniciar sesión con el usuario que necesitemos y ya estamos conectados mediante SSH con Ubuntu para poder realizar acciones sobre nuestro servidor remotamente.



```
adminwes@AFR-US: ~
login as: adminwes
adminwes@192.168.1.107's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-91-generic x86_64)

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

System information as of mar 07 dic 2021 00:31:22 UTC

System load:  0.02          Processes:           121
Usage of /:   23.7% of 19.56GB  Users logged in:      0
Memory usage: 29%            IPv4 address for enp0s3: 192.168.1.107
Swap usage:   0%

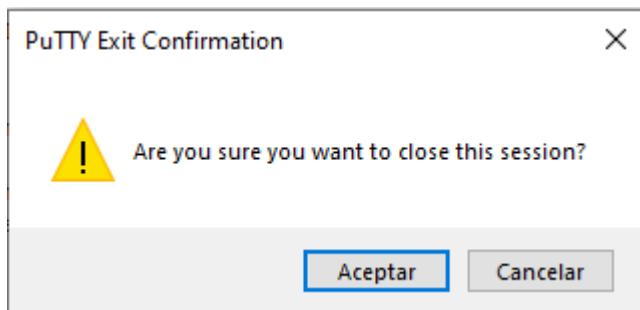
* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

15 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Mon Dec  6 18:51:10 2021 from 192.168.1.20
adminwes@AFR-US:~$
```

Para cerrar la sesión, solo hay que darle a la X para cerrar la ventana anterior, y nos preguntara si queremos cerrar la sesión, le daremos a **Aceptar** y ya estaría cerrada.



Para abrir una sesión nueva tendremos que abrir PuTTY otra vez y abrirla.

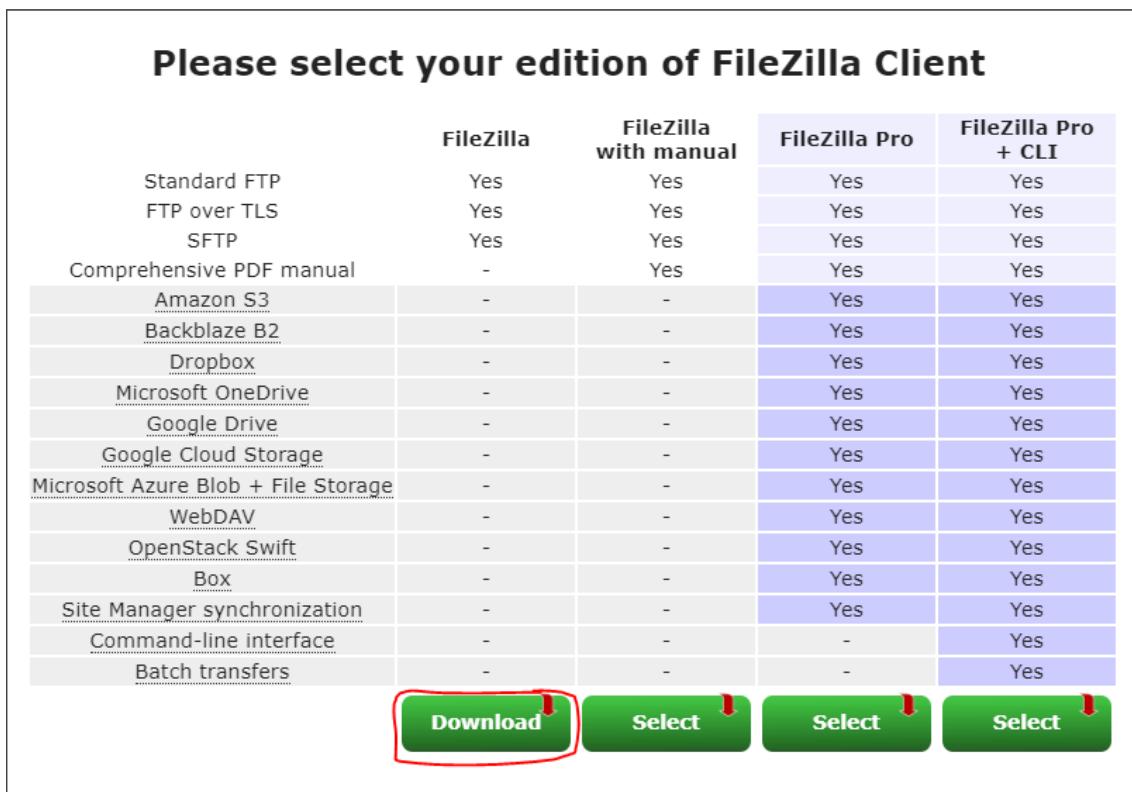
2.6 Instalación de FileZilla

2.6.1 Instalación

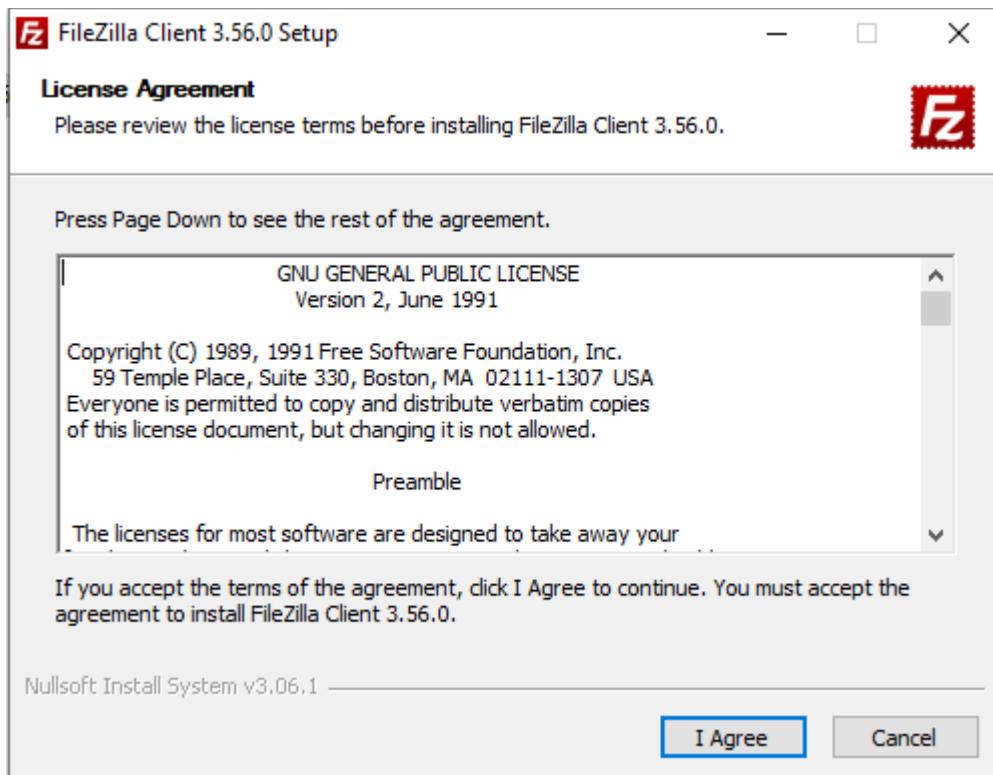
Primero nos vamos a la [página web de descarga](https://filezilla-project.org/download.php) y nos bajamos el instalador para Windows de 64bit dando en **Download FileZilla Client**.



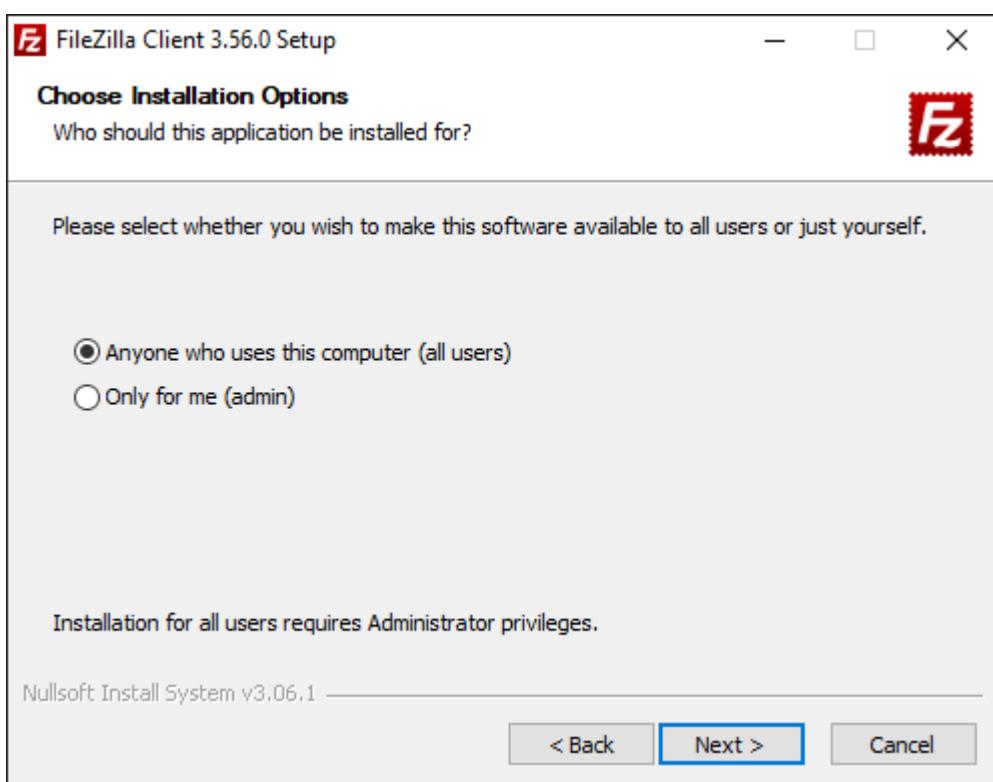
Cuando le damos a descargar, nos pregunta que edición de FileZilla, seleccionaremos la que se llama solo FileZilla y le damos a **Download**.



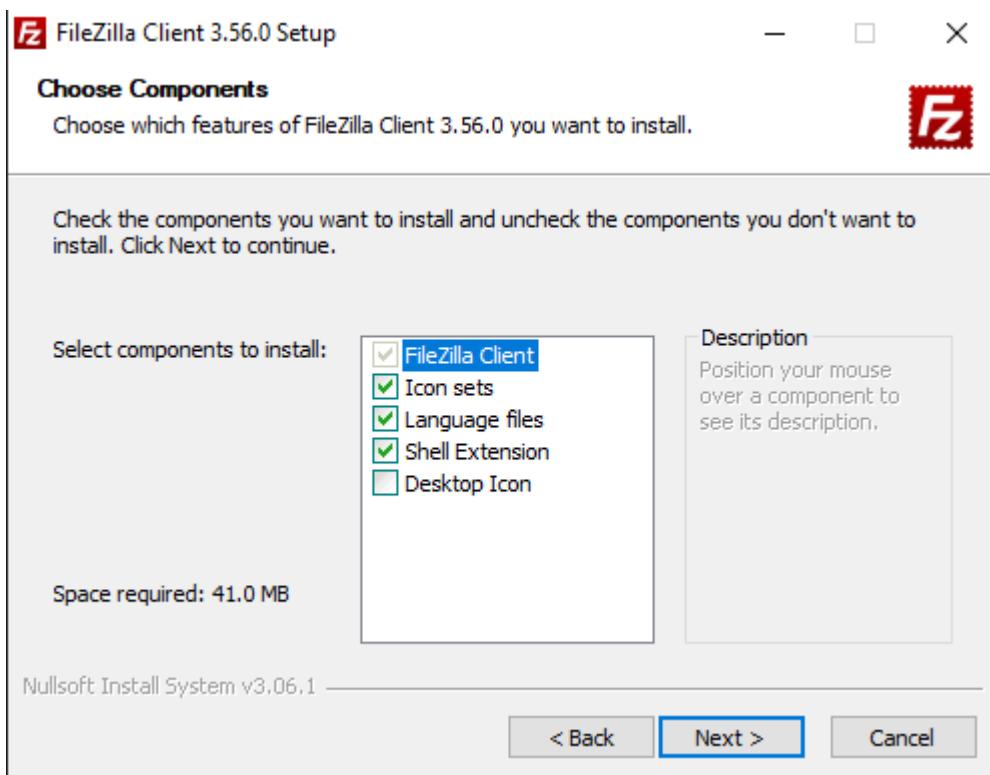
Una vez bajado el instalador lo ejecutaremos, lo primero será aceptar los términos de la licencia. Le daremos a **I Agree**.



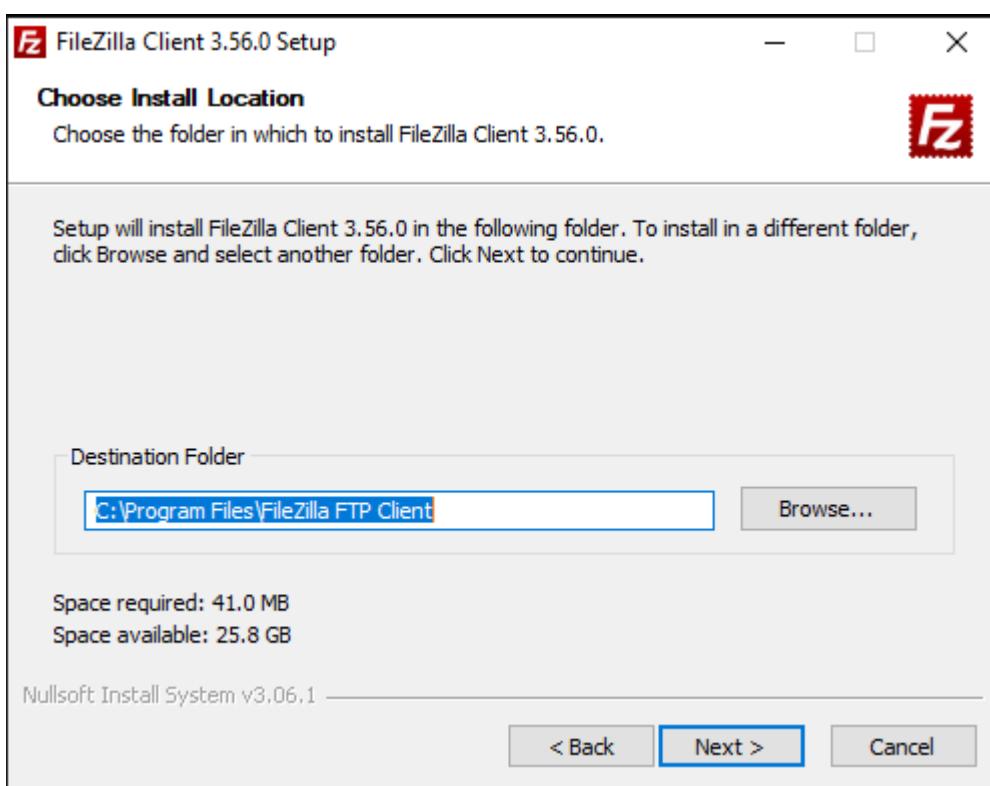
Ahora nos preguntara en que usuarios haremos la instalación, en este caso lo voy a hacer para todos los usuarios. Y le damos a **Next**.



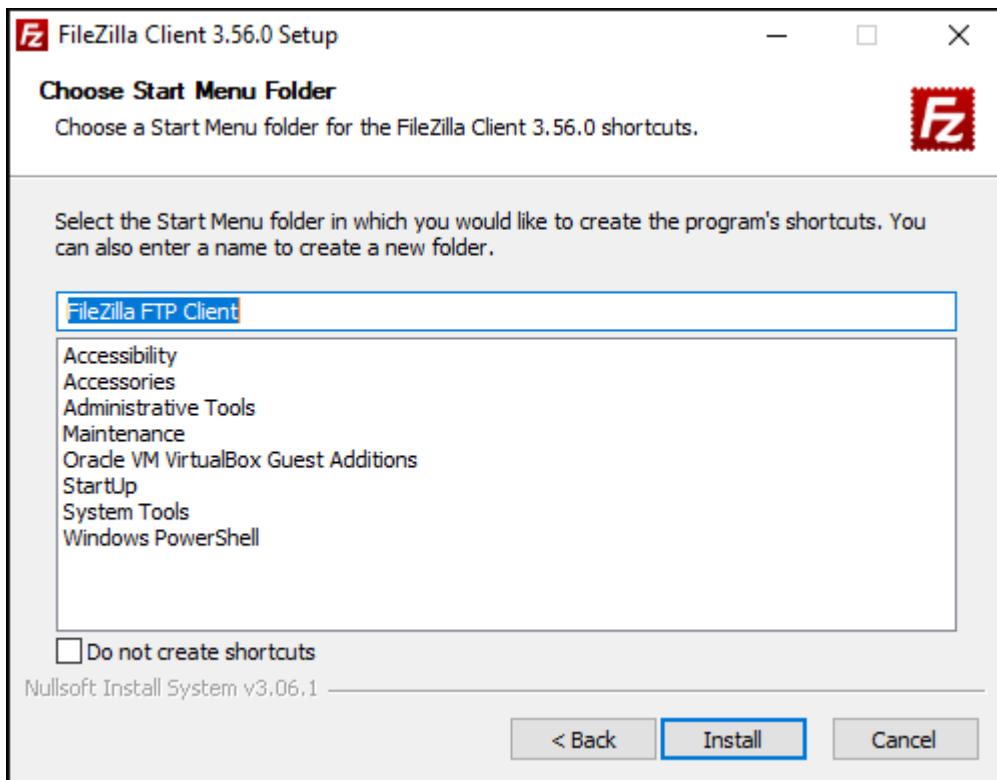
Lo siguiente será elegir los componentes que nos va a instalar, dejaremos marcado lo que viene por defecto, ya que es lo que necesitamos. Le damos a **Next**.



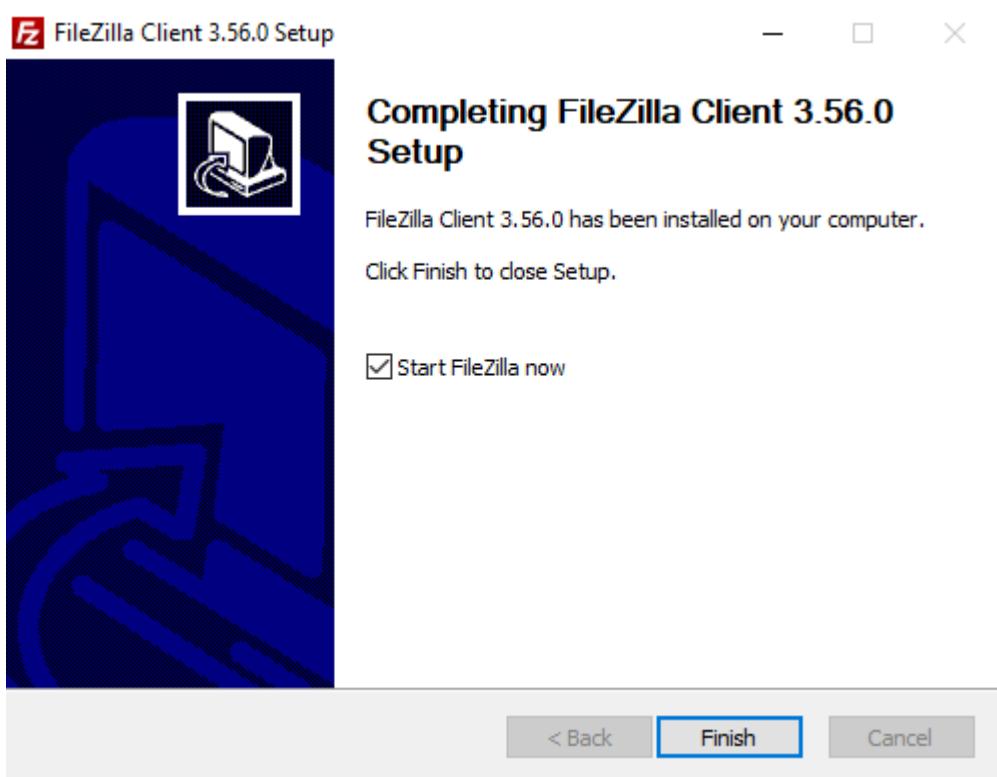
Ahora nos pedirá la ubicación de donde va a realizar la instalación en nuestro equipo, en mi caso lo he dejado por defecto, pero podemos elegir la ruta que queramos. Y le damos a **Next**.



Por último, nos pedirá elegir el menú de inicio, en este caso lo dejamos por defecto. Y ya le damos a **Install**.

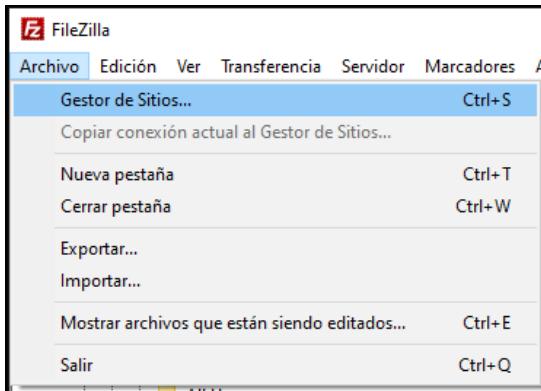


Una vez finalice la instalación nos mostrara esta ventana, ya estaría instalado el cliente de FileZilla. Dejaremos marcada la opción de Start FileZilla now para que se nos abra al darle a **Finish**.

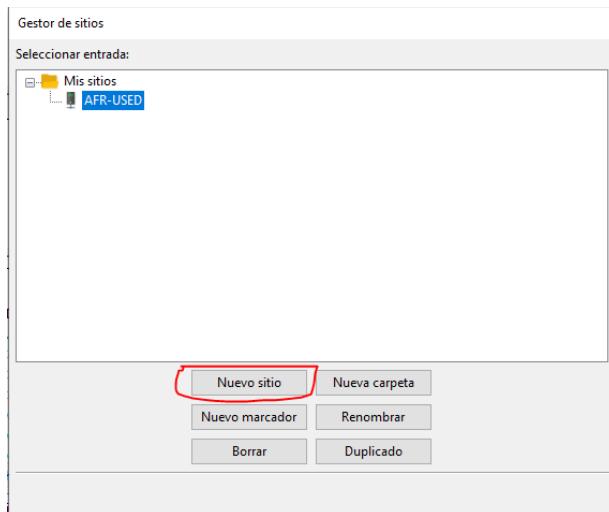


2.6.2 Configuración

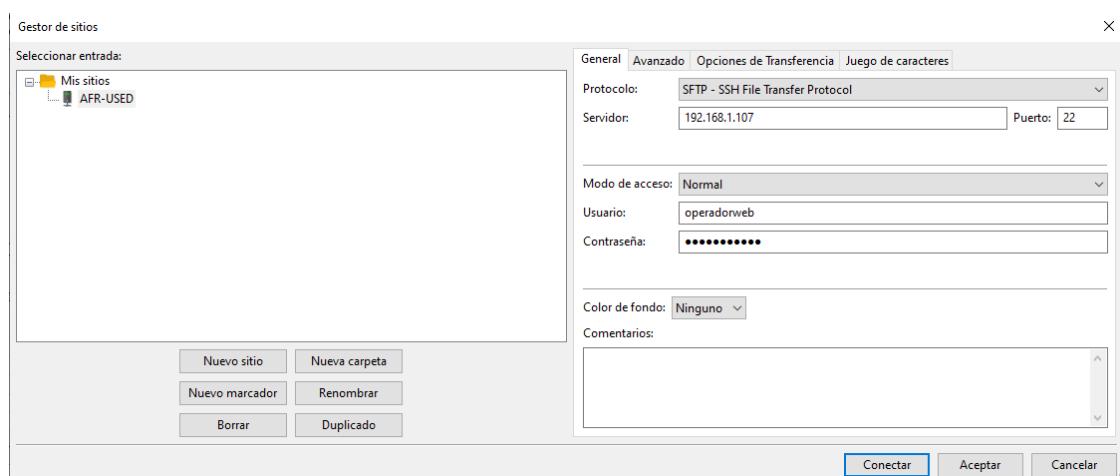
Una vez instalado el cliente de FileZilla, lo abrimos. Nos iremos a Archivo y Gestor de Sitios...



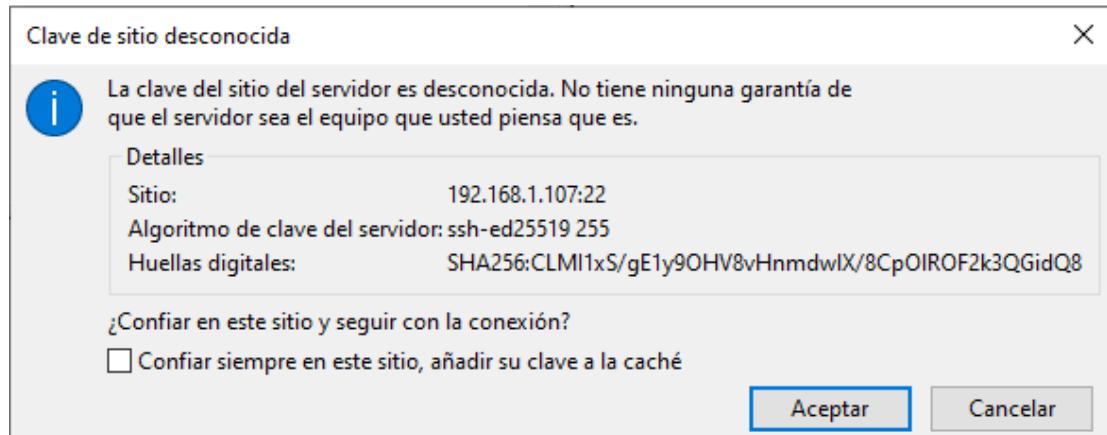
Una vez se nos abra el gestor de sitios, le daremos a **Nuevo Sitio** para configurar la conexión a nuestro servidor de Ubuntu y poder acceder más fácilmente.



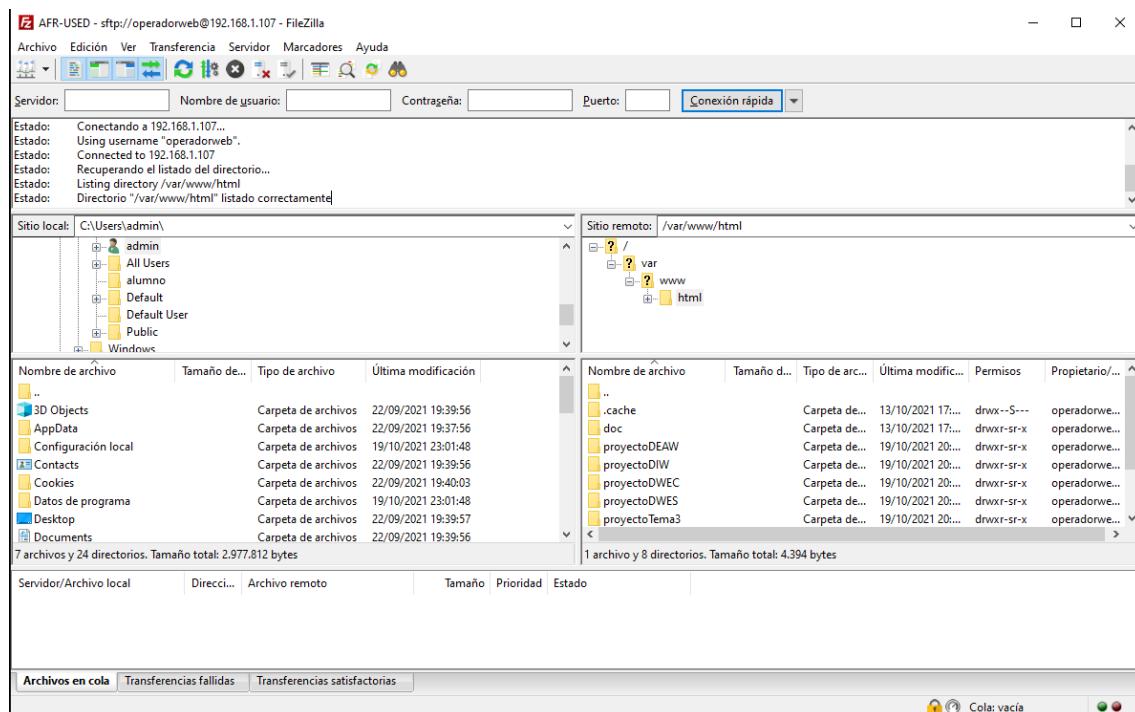
Una vez tengamos agregado un sitio con el nombre que queremos, configuraremos el apartado de General, el protocolo que usaremos para la conexión será SFTP, el servidor será la IP de nuestro Ubuntu Server, el puerto el 22, y el usuario y contraseña serán los de operadorweb que es el administrador del sitio. Una vez configurado le daremos a **Conectar**.



Nos dirá que la clave del servidor es desconocida, igualmente le daremos a **Aceptar** para conectarnos.

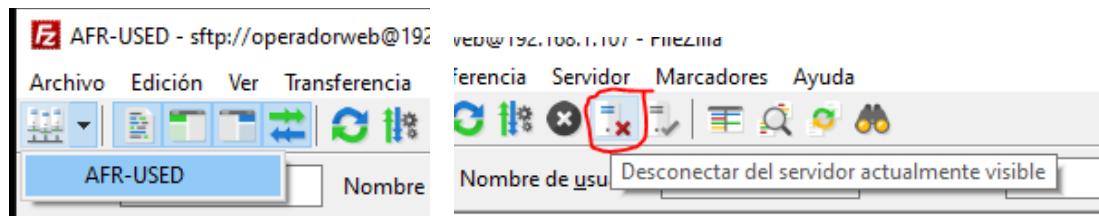


Y si todo está bien configurado, se realizará la conexión con nuestro servidor. Desde aquí ya podremos administrar los archivos y directorios de nuestro servidor web.



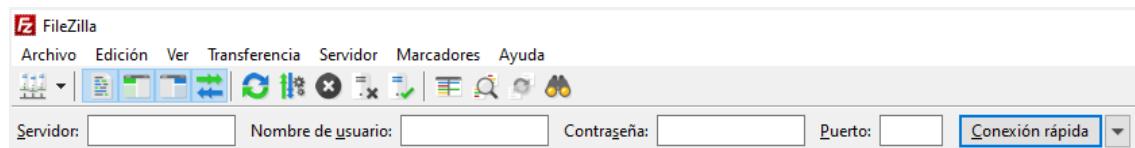
Para acceder a la conexión rápida que hemos configurado, solo tenemos que desplegar la pestaña de la izquierda superior y nos saldrá dicha conexión.

También podremos desconectarnos del servidor desde el botón de la imagen y realizar otra conexión.

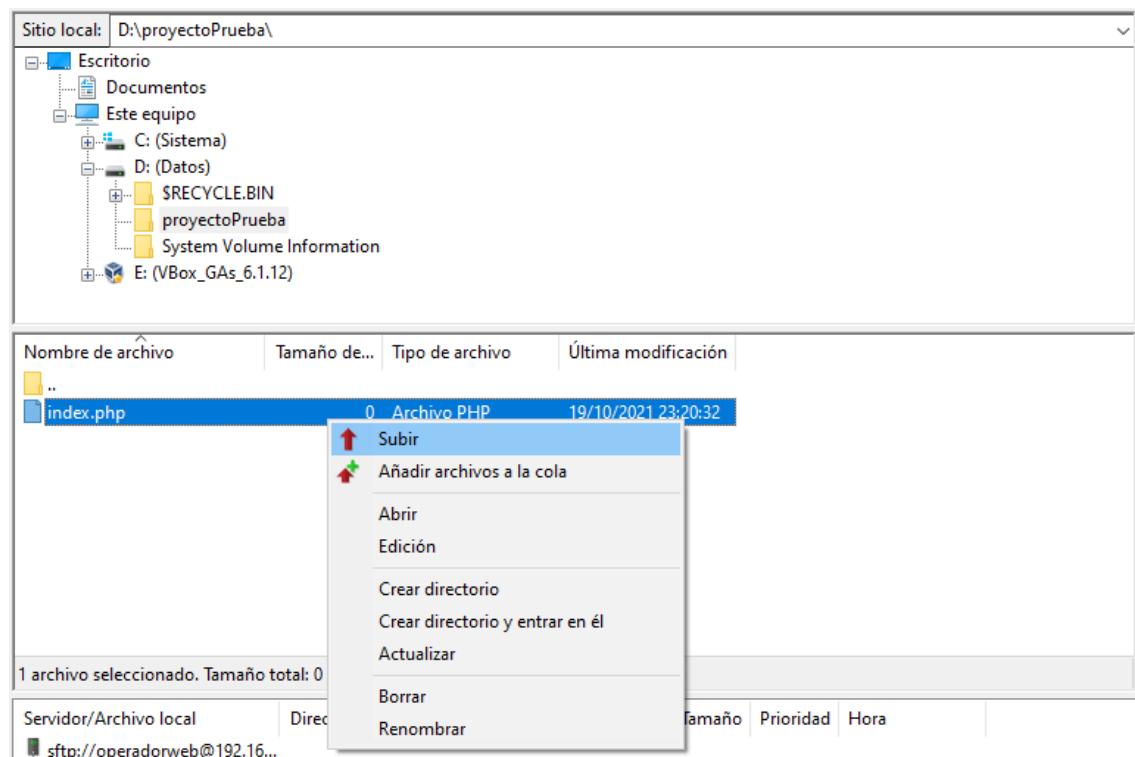


2.6.3 Ejemplo de uso

Nos conectamos a nuestro servidor, mediante la conexión que creamos anteriormente o a otra específica desde la parte superior metiendo los datos.



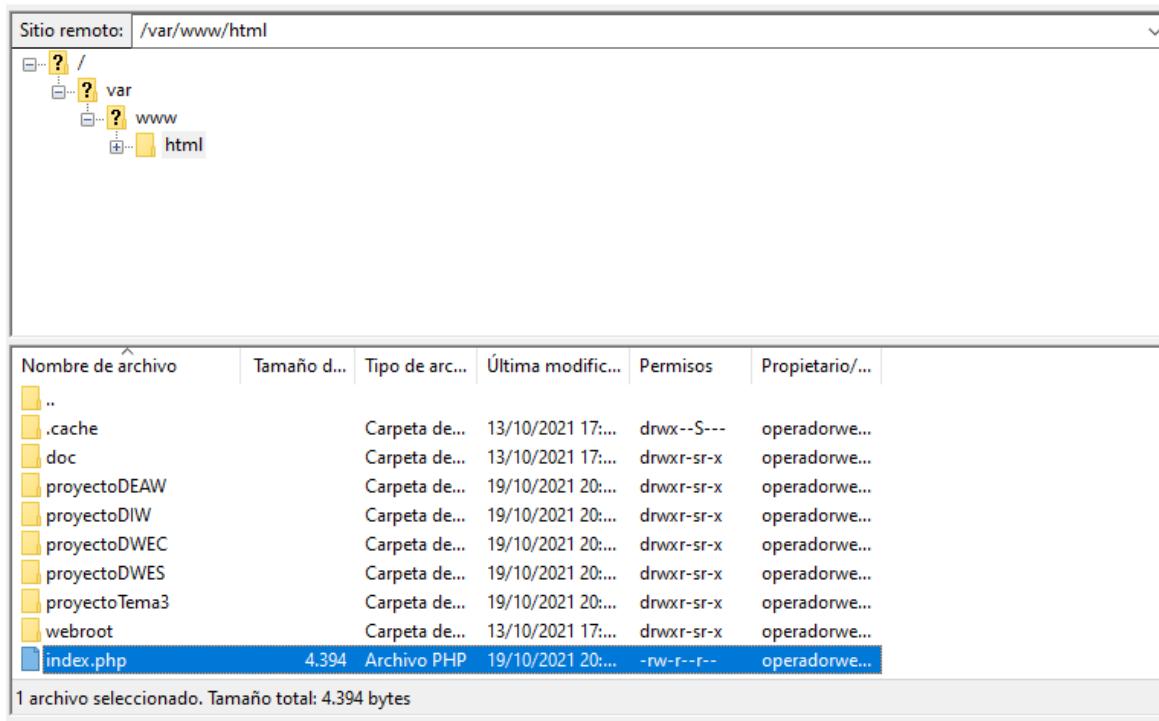
Para subir los archivos de nuestra web a el servidor, tendremos que irnos a el Sitio local: en la parte izquierda de la ventana y aquí seleccionar los archivos a subir, le damos clic derecho y a subir. **IMPORTANTE: Estar ubicados en el directorio remoto donde queremos subir los archivos en el lado del servidor.**



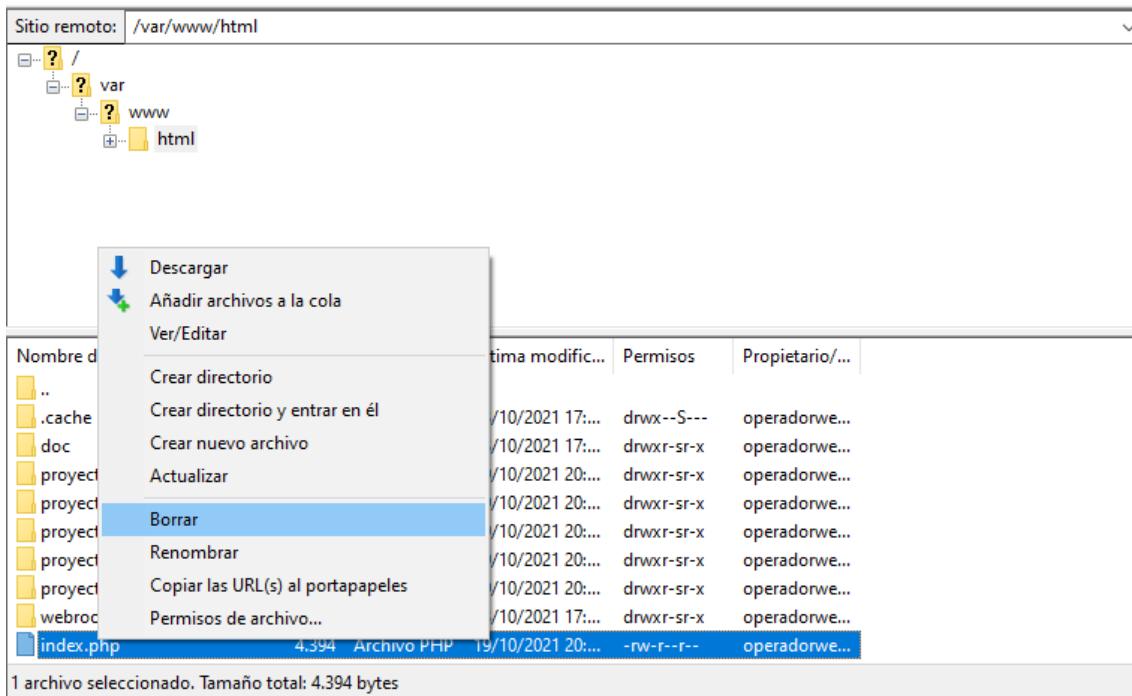
Si la transferencia se ha realizado con éxito, nos lo mostrara en la parte inferior de la ventana, aquí nos muestra donde se ha subido y en que directorio del sitio remoto.

Servidor/Archivo local	Direcci...	Archivo remoto	Tamaño	Prioridad	Hora
sftp://operadorweb@192.16...					
D:\ proyectoPrueba\index....	-->	/var/www/html/index.php	0	Normal	19/10/2021
D:\ proyectoPrueba\index....	-->	/var/www/html/index.php	0	Normal	19/10/2021

Si la transferencia se ha realizado con éxito, solo tendremos que comprobar en la parte derecha de la ventana donde nos sale el sitio remoto si están el archivo o archivos que hemos subido.



También podemos borrar del Sitio Remoto archivos con clic derecho y borrar.



2.7 Instalación de NetBeans

2.7.1 Instalación

Lo primero será buscar en el navegador “NetBeans” y entrar a la página oficial netbeans.apache.org/download.

También podemos ir directamente a la [página web de descarga](#).

Aproximadamente 55.000.000 resultados (0,41 segundos)

<https://netbeans.apache.org> › downl... ▾ Traducir esta página

Apache NetBeans Releases

Apache **NetBeans** 12.6. Latest version of the IDE, released on November 29, 2021. Features Download. Older releases.

Download Apache NetBeans 12.5 runs on JDK LTS releases 8 and 11 ...

Download NetBeans 8.2 Apache NetBeans 11.2 was released on October 25, 2019 ...

[Más resultados de apache.org »](#)

Una vez dentro de la pagina web de descarga de NetBeans, en este caso como no vamos a instalar la última versión de NetBeans nos iremos a “Find out more” para ver otras versiones.

Latest release

Apache NetBeans 12.6

[Download](#)

Apache NetBeans Releases

Apache NetBeans is released four times a year. For details, see [full release schedule](#).

Apache NetBeans 12.6

Latest version of the IDE, released on November 29, 2021.

[Features](#) [Download](#)

Older releases

Older Apache NetBeans releases and pre-Apache NetBeans releases can still be downloaded, but are no longer supported.

[Find out more](#) ↙

Daily builds and building from source

All Apache NetBeans source code is freely available to build yourself, or you can download (unsupported) daily development builds.

[Find out more](#)

Buscaremos en la página que se nos abra la versión 12.0 y le daremos a **Download**

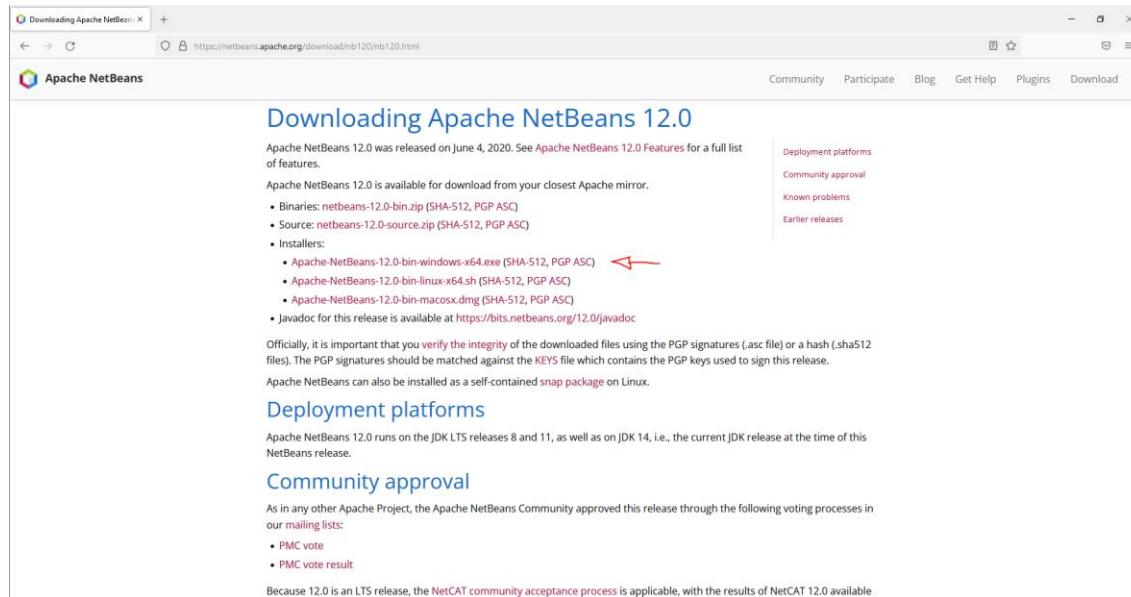
Apache NetBeans 12.0

Apache NetBeans 12.0 LTS was released on June 4, 2020.

[Features](#) [Download](#)

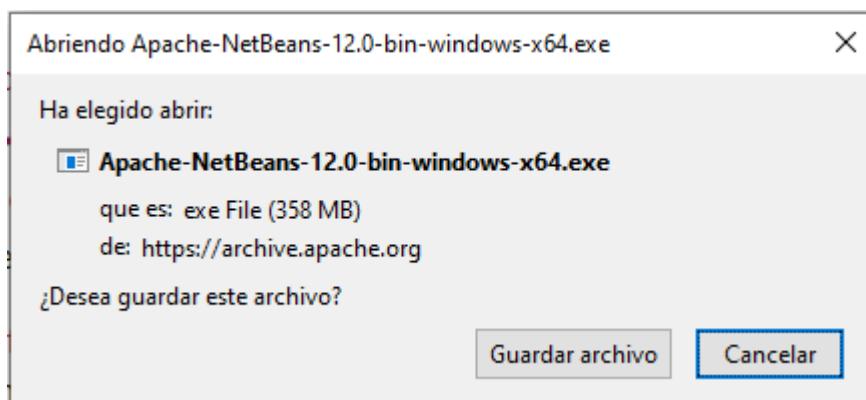


Cuando estamos en la página de descarga de la versión 12.0, seleccionaremos el instalador x64 para Windows.

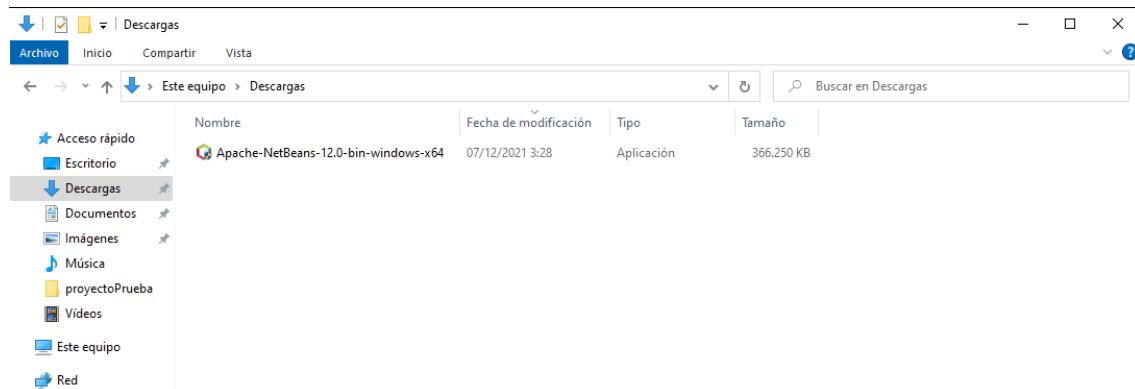


The screenshot shows the Apache NetBeans 12.0 download page. The URL in the address bar is <https://netbeans.apache.org/download/nb120/nb120.html>. The page title is "Downloading Apache NetBeans 12.0". It lists download links for different platforms, with the Windows x64 executable file being the primary focus. A red arrow points to the "Apache-NetBeans-12.0-bin-windows-x64.exe" link under the "Binaries" section.

Nos pedirá si queremos guardar el archivo, le daremos a **Guardar archivo** para que se guarde en descargas.

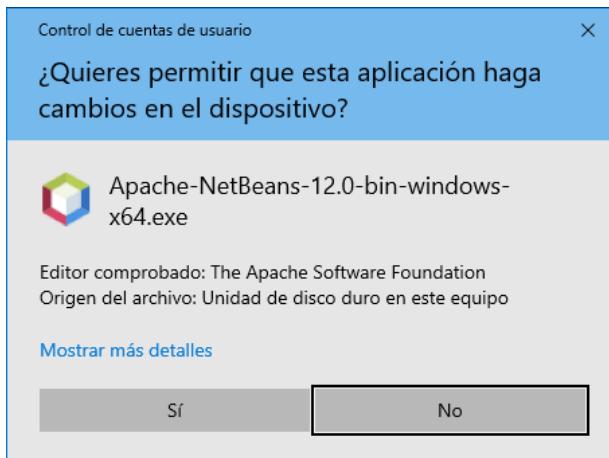


Ahora nos iremos a descargas y ejecutaremos el instalador de NetBeans.

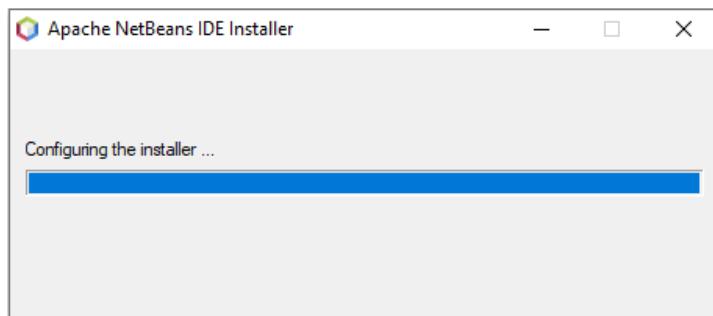


The screenshot shows the Windows File Explorer with the "Descargas" (Downloads) folder selected. The left sidebar shows navigation options like "Acceso rápido", "Escritorio", and "Este equipo". The main area displays a list of files in the "Descargas" folder. One file is listed: "Apache-NetBeans-12.0-bin-windows-x64" by Apache, modified on 07/12/2021 at 3:28, is an application file (Aplicación) with a size of 366.250 KB.

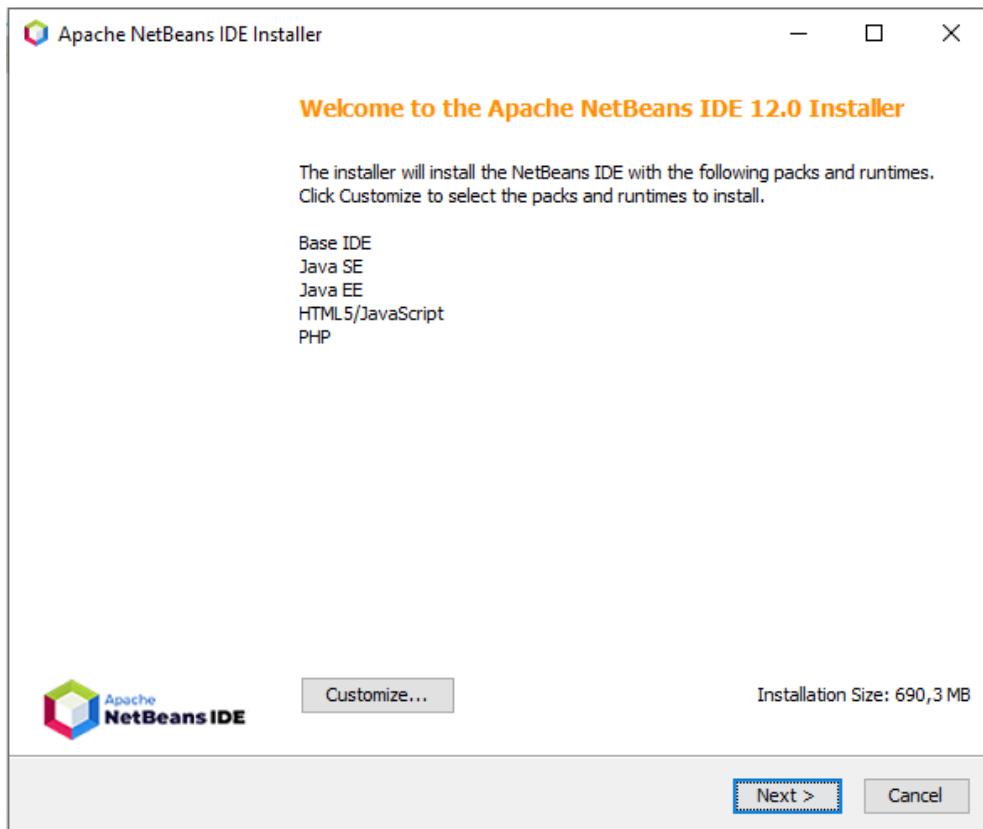
Nos pedirá permisos para iniciar el instalador, le daremos a que **sí**.



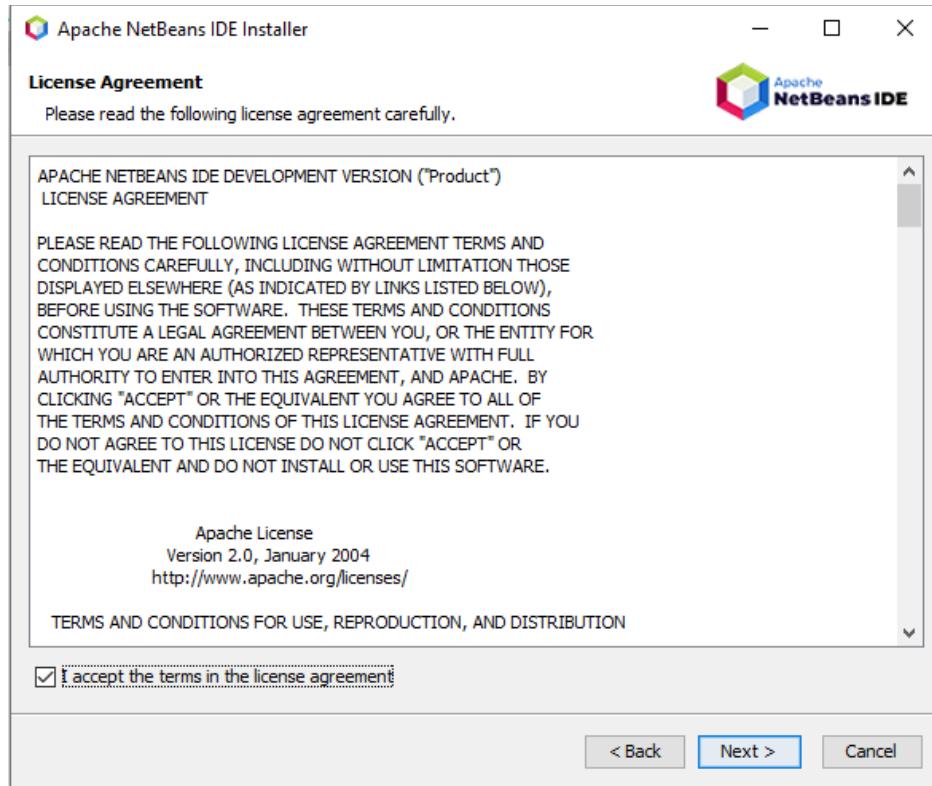
Se empezará a configurar el instalador. (Recordar que hay que tener instalado el Java SE Development kit (JDK) para poder instalar el IDE NetBeans).



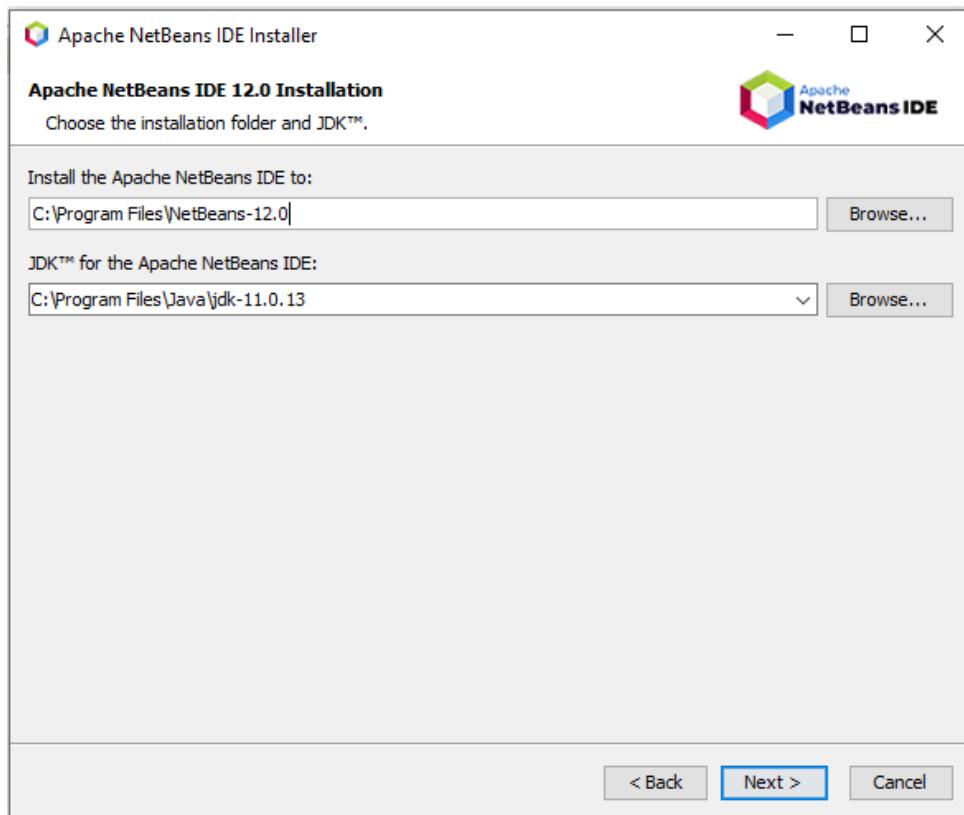
Una vez se configure se nos iniciara el instalador, para comenzar le daremos a **Next**.



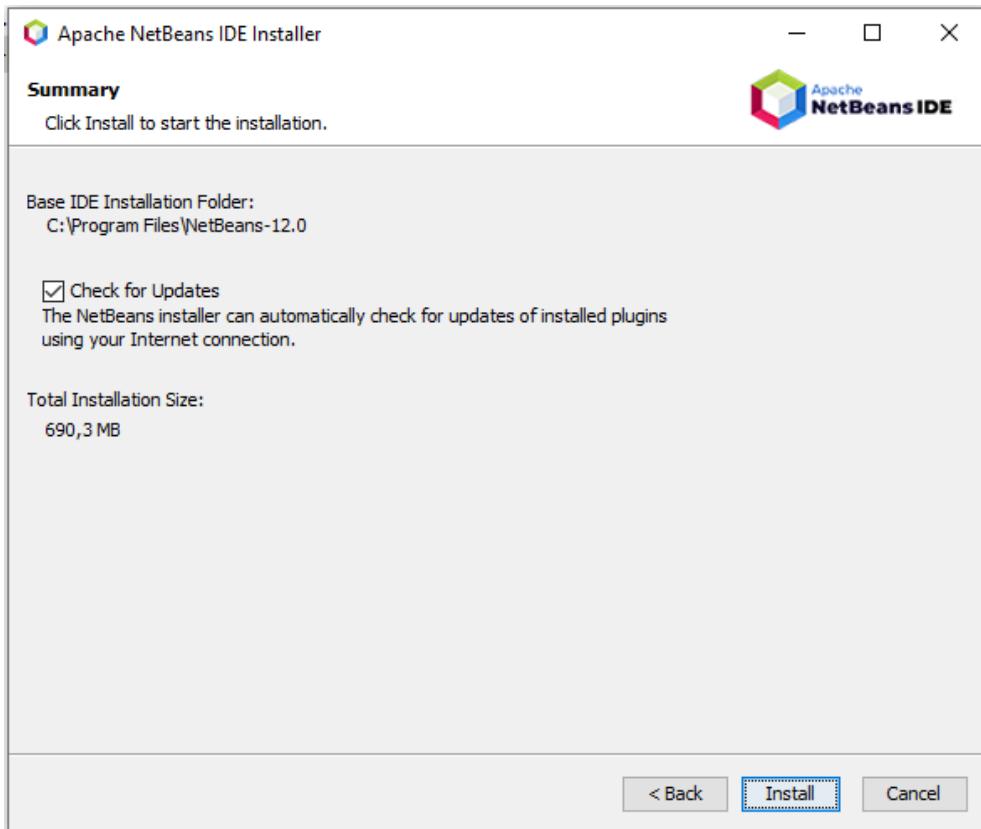
Nos pedirá que aceptemos los términos para poder continuar, los aceptamos y le damos a **Next**.



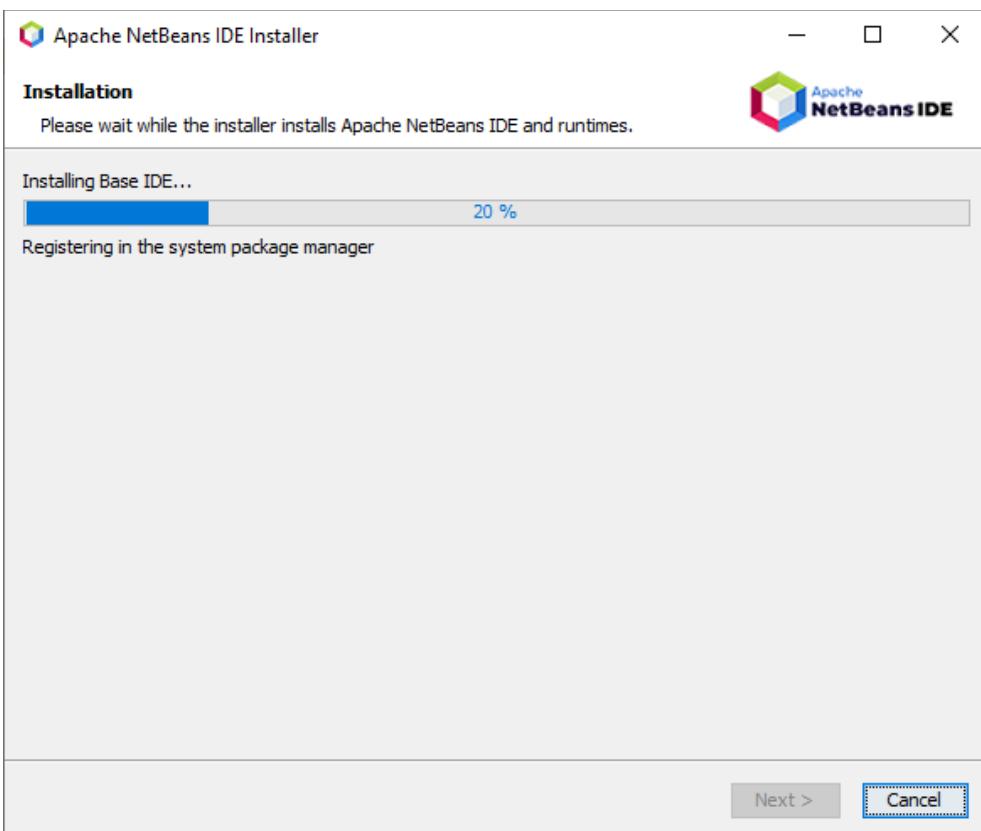
Nos preguntara el directorio de instalación, en mi caso lo dejare por defecto, también nos pedirá donde se encuentra el JDK que también sale por defecto la ruta. Una vez configurado le daremos a **Next**.



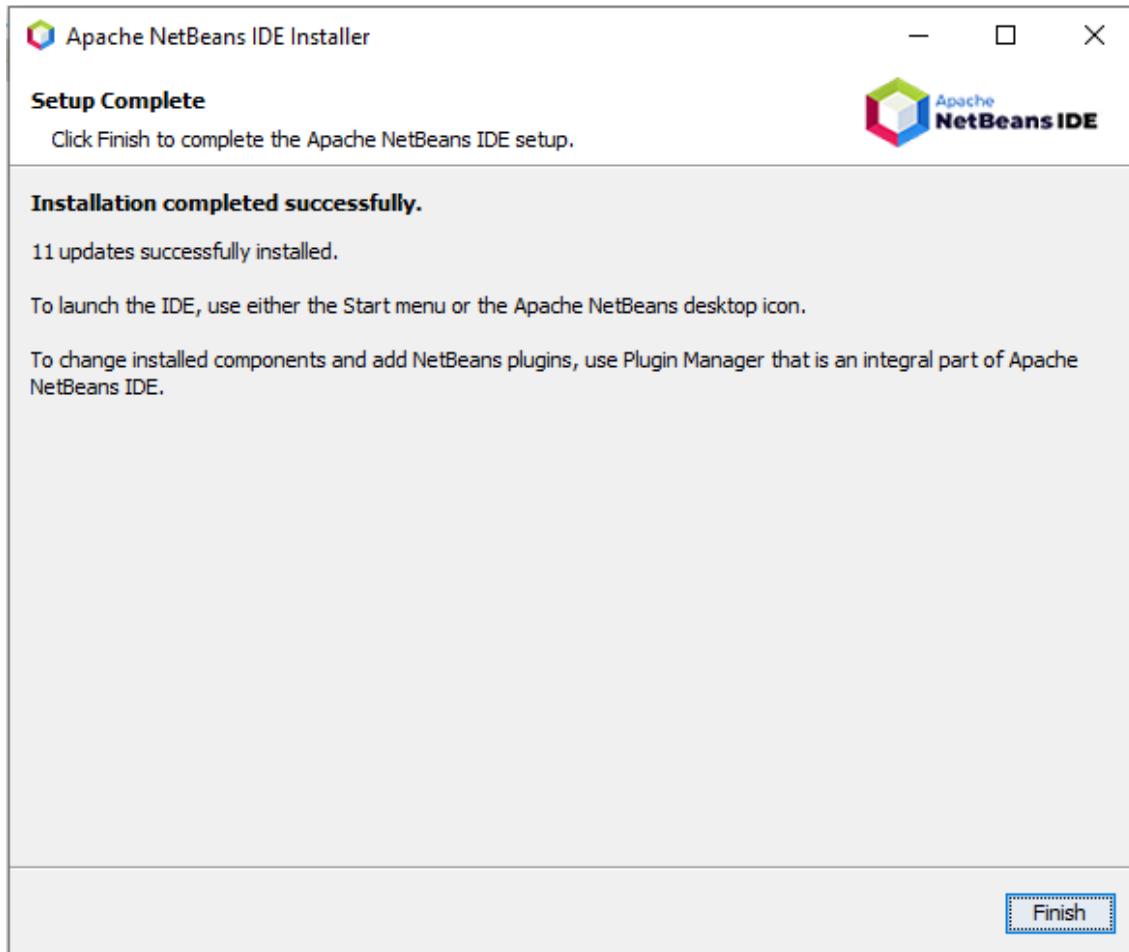
Por último, nos mostrara el sumario de la instalación, también nos da la opción de comprobar las Updates, en mi caso la voy a dejar marcada. Le daremos a **Install**.



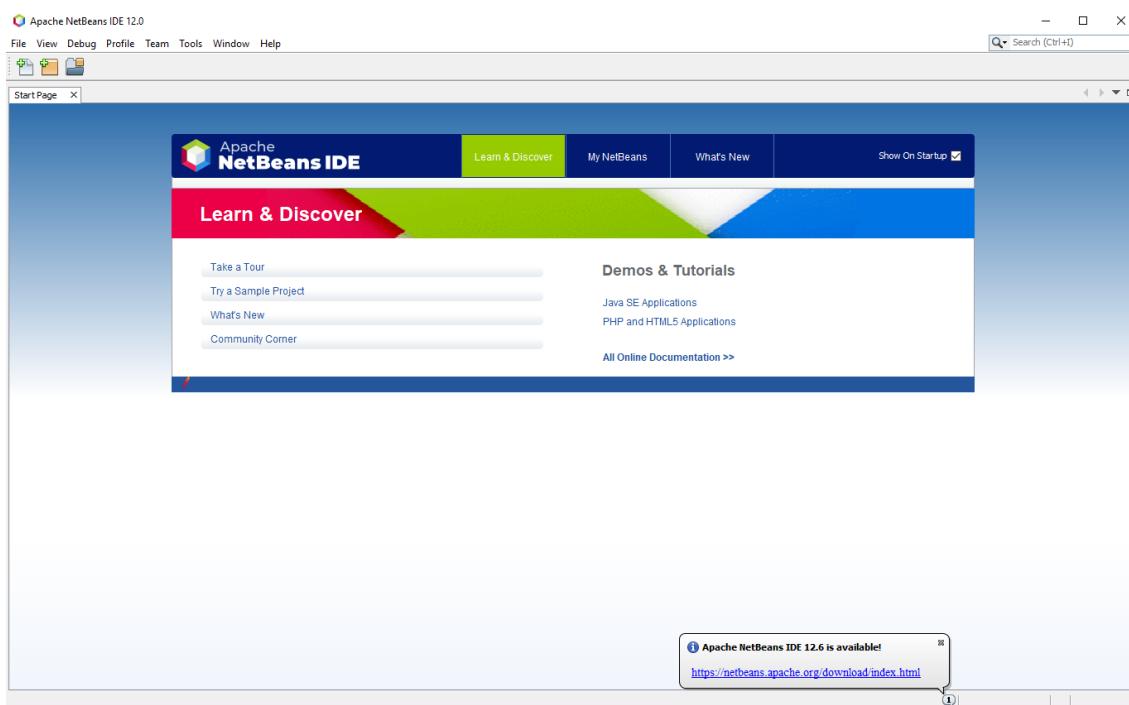
Empezara la instalación del IDE, dicha instalación tardara unos minutos.



Una vez finalice el proceso de instalación, nos mostrara que la instalación ha sido exitosa, ya podremos darle a **Finish**.



Una vez instalado ya podemos abrir el IDE NetBeans instalado con éxito.

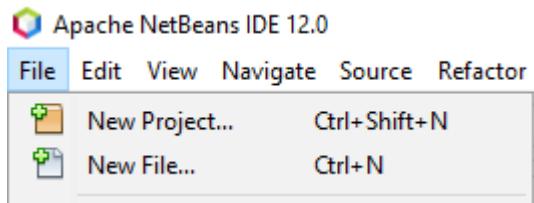


2.8. Conectar Proyecto de NetBeans con Ubuntu Server

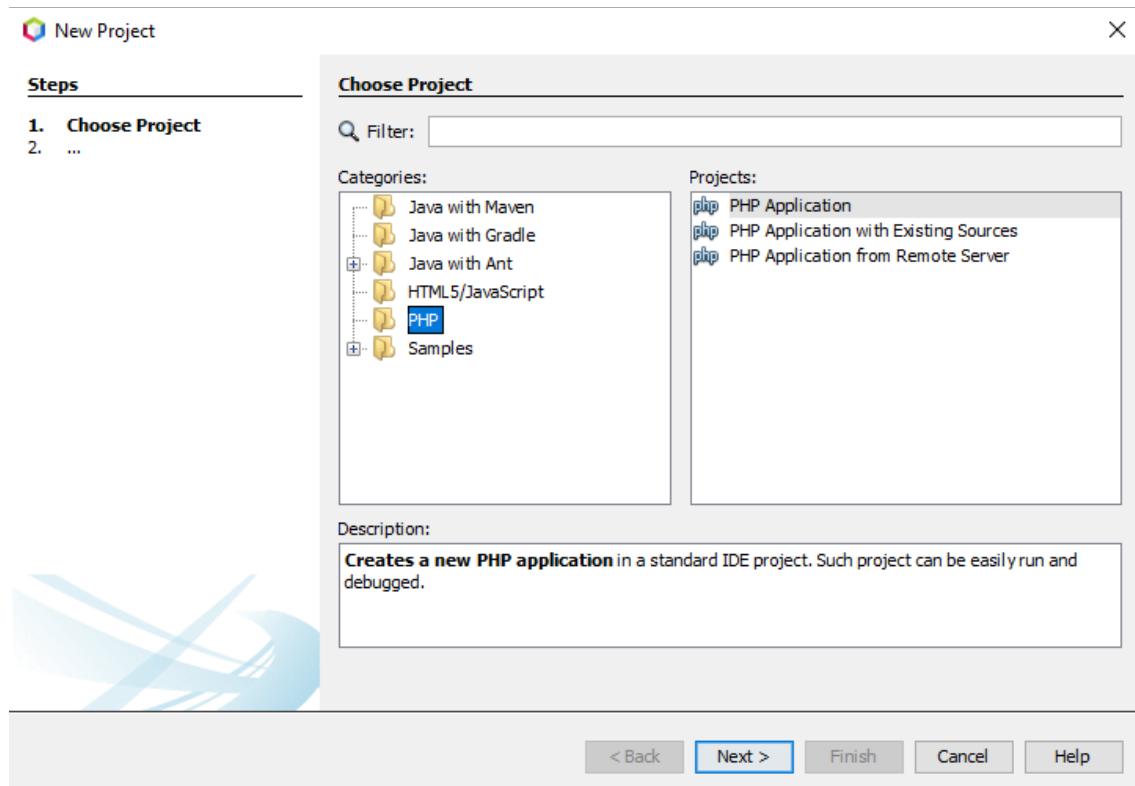
2.8.1 Configuración

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

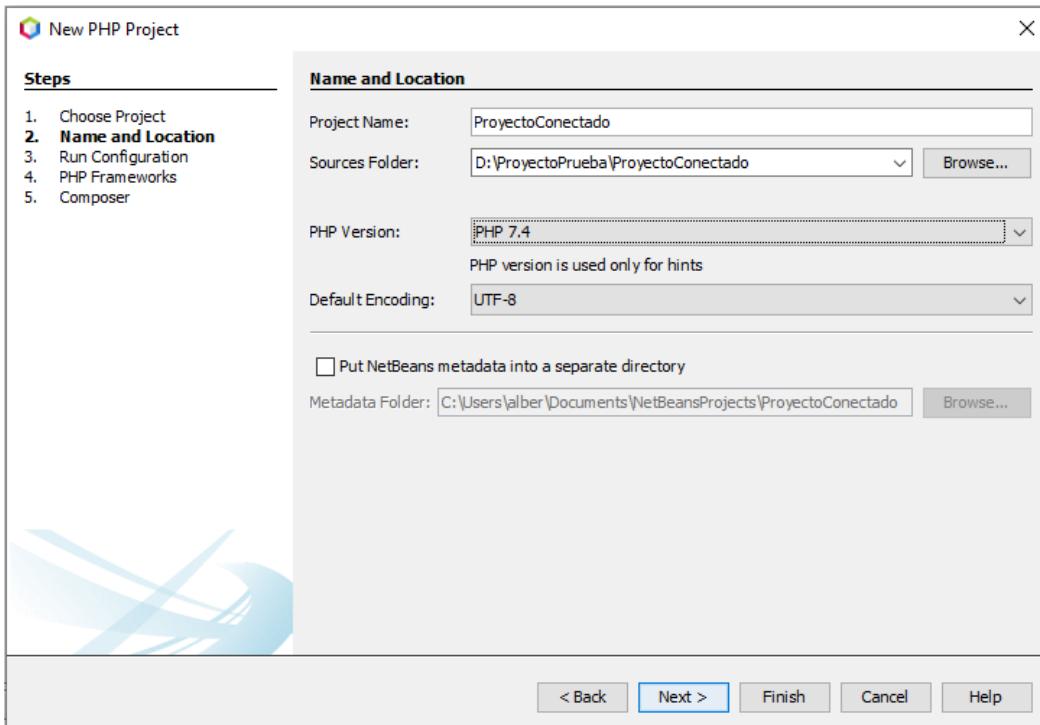
IMPORTANTE: Tener NetBeans sin ningún proyecto abierto, tener vacío el directorio /var/www/html en el servidor de Ubuntu y tener arrancado Ubuntu Server.



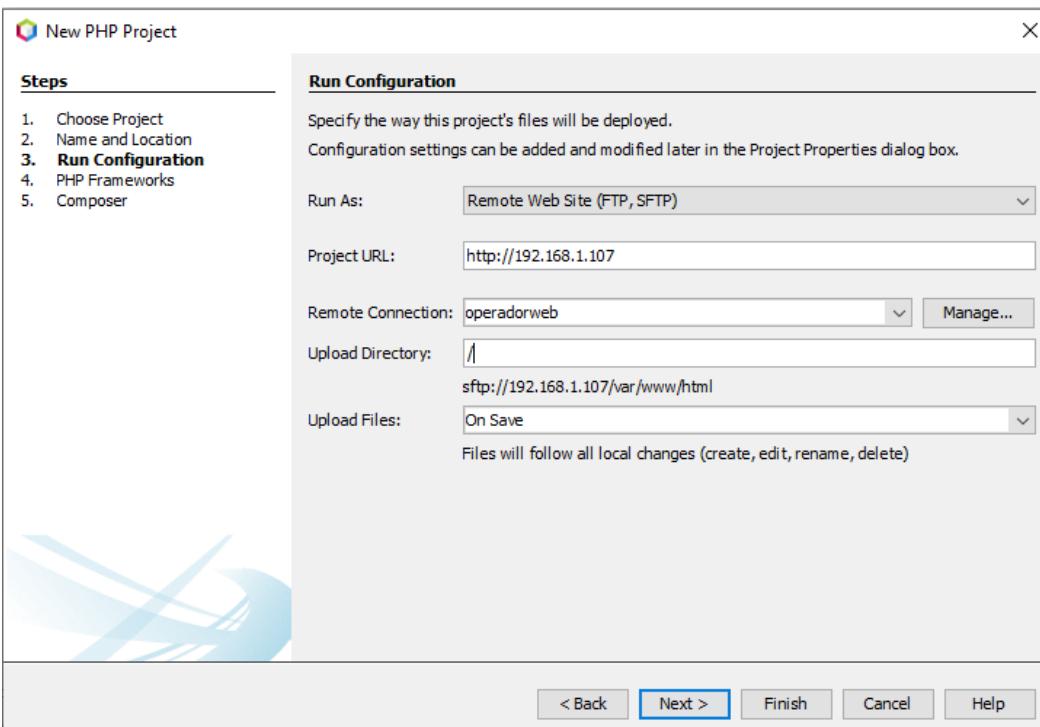
Se nos abrirá un cuadro para configurar el proyecto nuevo, dentro de este seleccionamos la categoría PHP y el proyecto PHP Application, a continuación, le damos a siguiente.



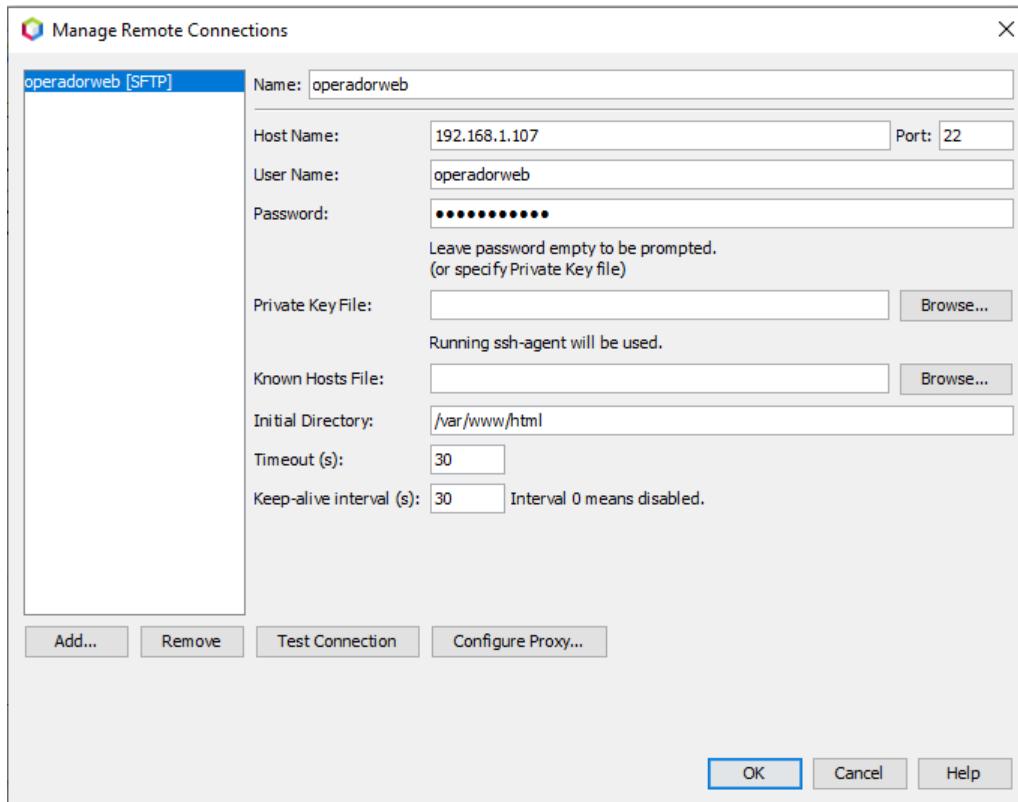
Ahora tenemos que darle un nombre al proyecto, también tendremos que seleccionar una carpeta vacía para meter el proyecto en dicha carpeta. Seleccionaremos la versión de PHP que usaremos en el proyecto y le daremos a siguiente.



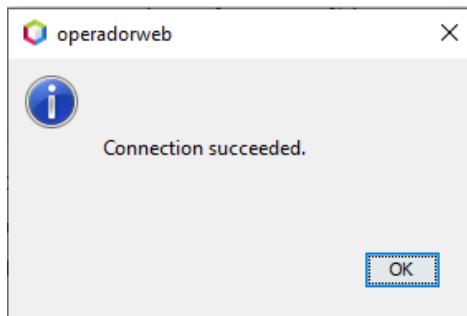
Lo siguiente es configurar como va a correr el proyecto, en este caso lo tendremos que poner en un sitio web remoto con FTP, SFTP. La URL será nuestra dirección IP de nuestro servidor de Ubuntu. Configuraremos que los archivos se suban al servidor al guardar. También tendremos que configurar la conexión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



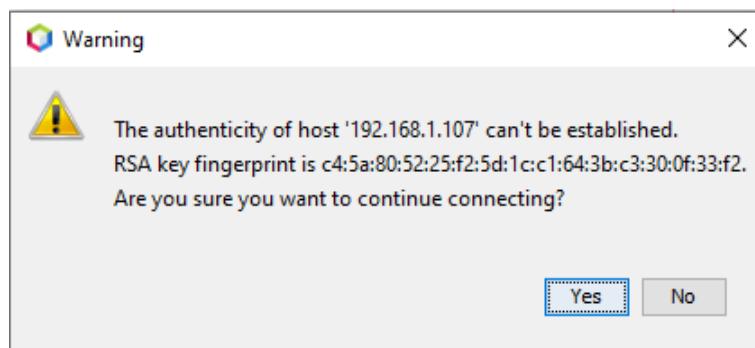
Una vez dentro de Manage, aquí pondremos la ip del host, el puerto 22, el usuario de conexión y su Password. Una vez introducidos los datos haremos un Test Connection dándole a dicho botón.



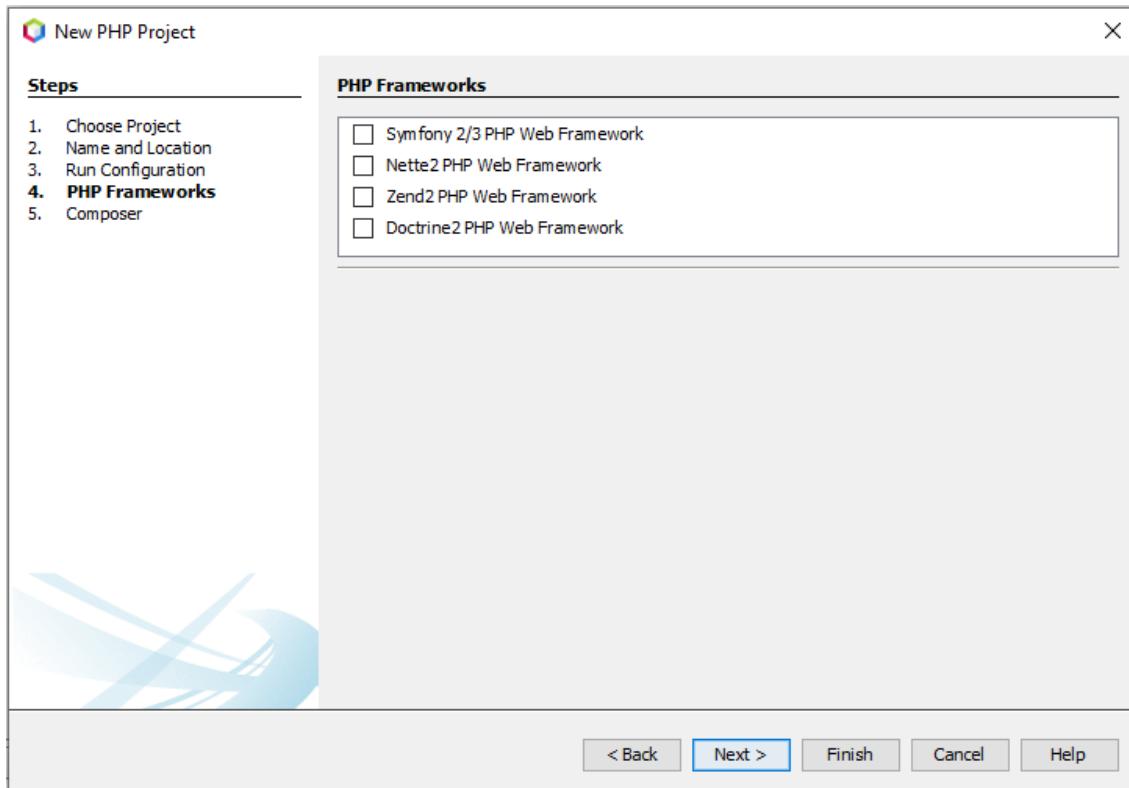
Una vez le damos al botón de Test Connection, probara que hay conexión con el Servidor de Ubuntu, si es correcta nos mostrara dicho mensaje.



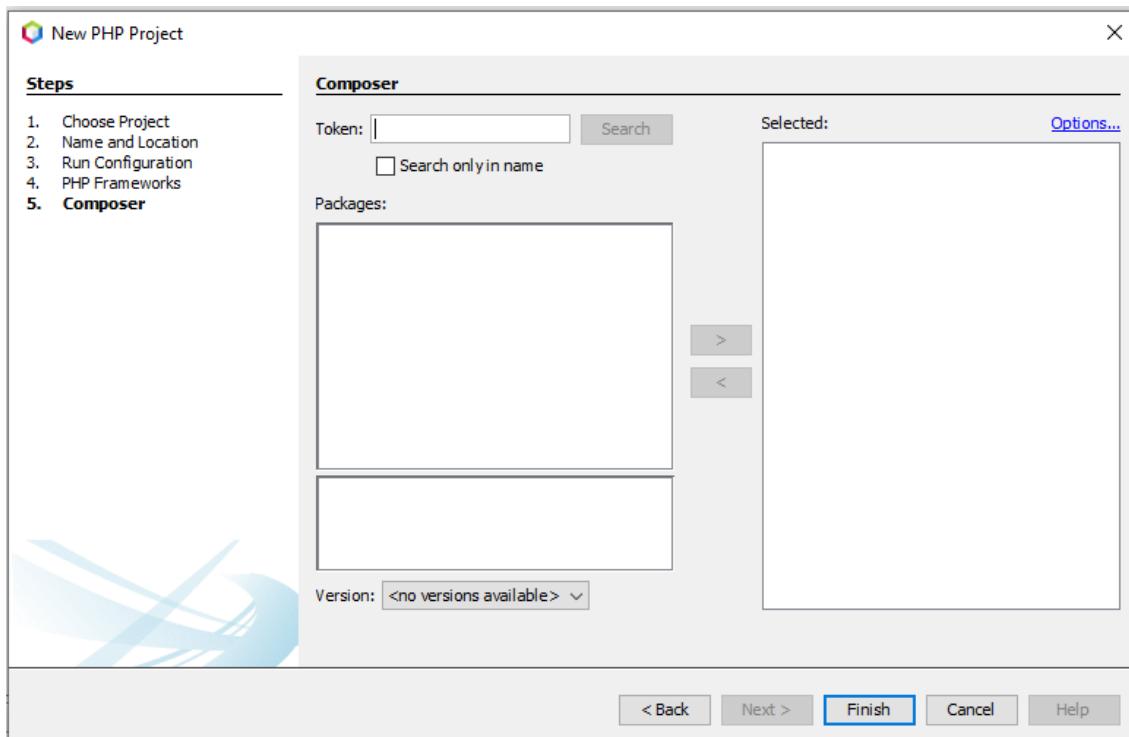
Una vez hecho este proceso le daremos a siguiente, y se nos mostrara dicho mensaje, el cual nos indica si estamos seguros de hacer la conexión, le daremos a que sí.



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

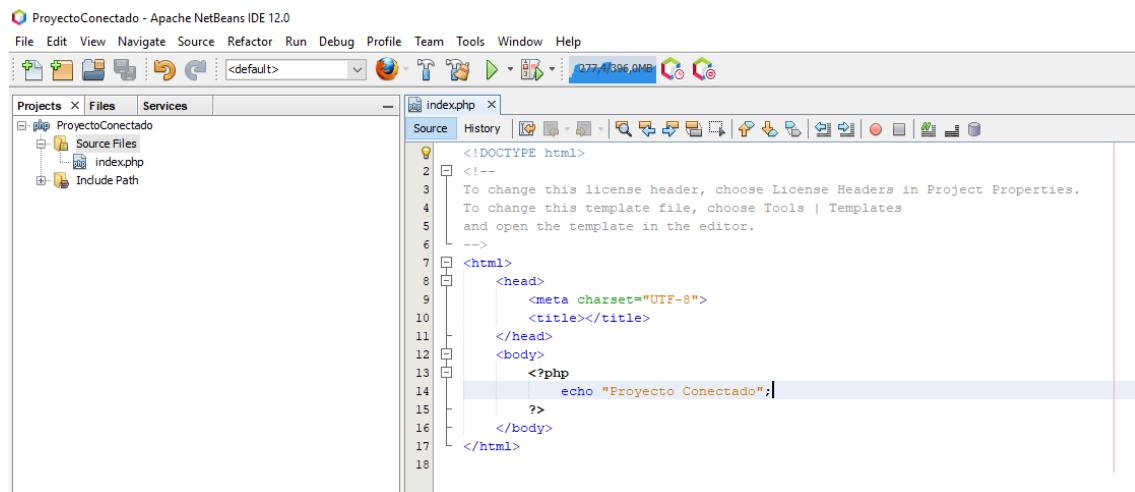


Por último, nos pide configurar un Composer, como no lo necesitamos tampoco por ahora le daremos a Finish.

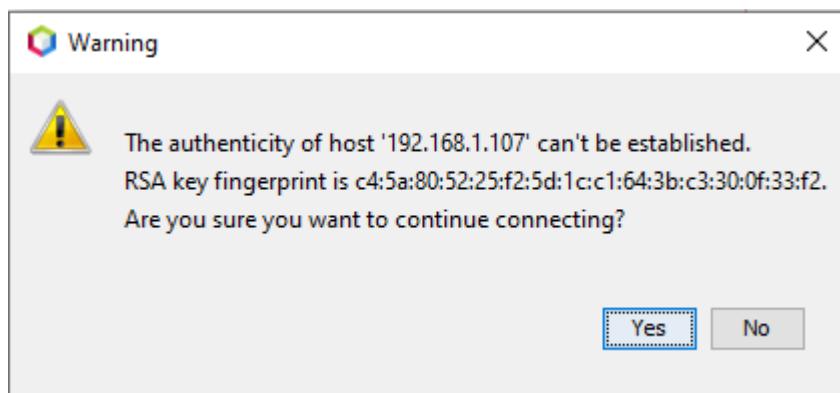


2.8.2 Ejemplo de uso

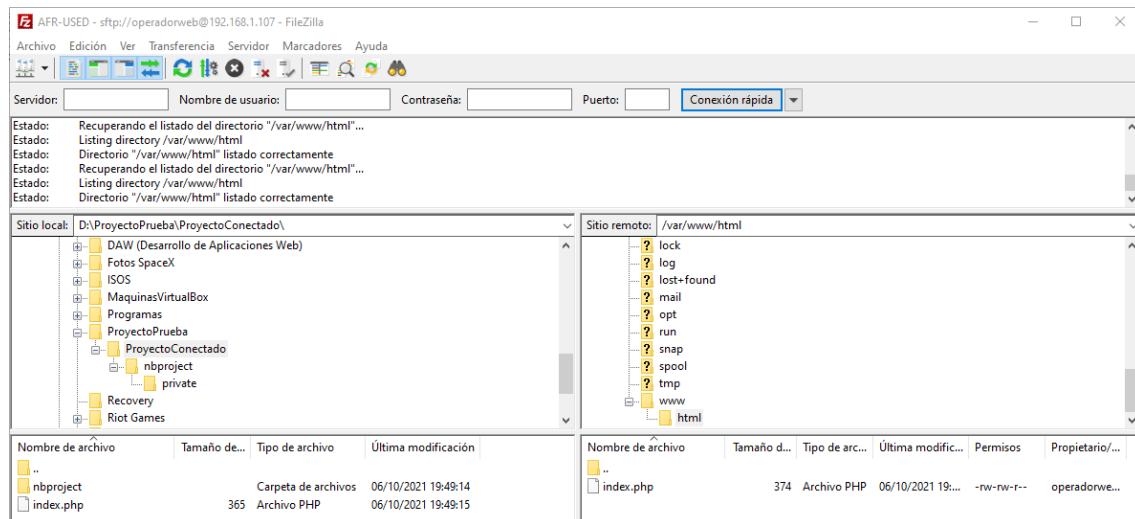
Una vez se acaba el menú de configuración el proyecto estará creado, dicho proyecto está conectado con el servidor de Ubuntu y cualquier cambio que se realice en él se cambiará en el servidor una vez guardemos.



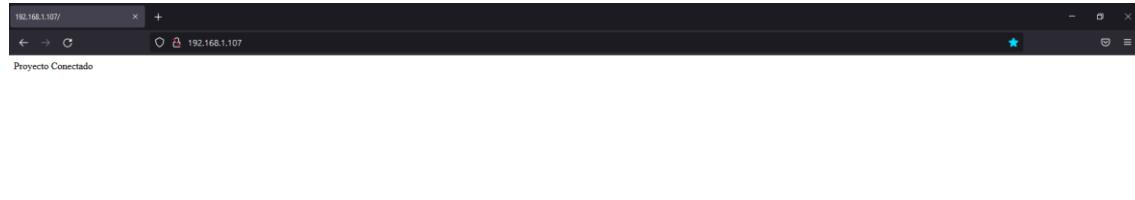
Para que se suba el Index, le pondremos una frase en PHP y le daremos a guardar, se nos abrirá dicho cuadro que pregunta si estás seguro de continuar para hacer los cambios, le daremos que sí.



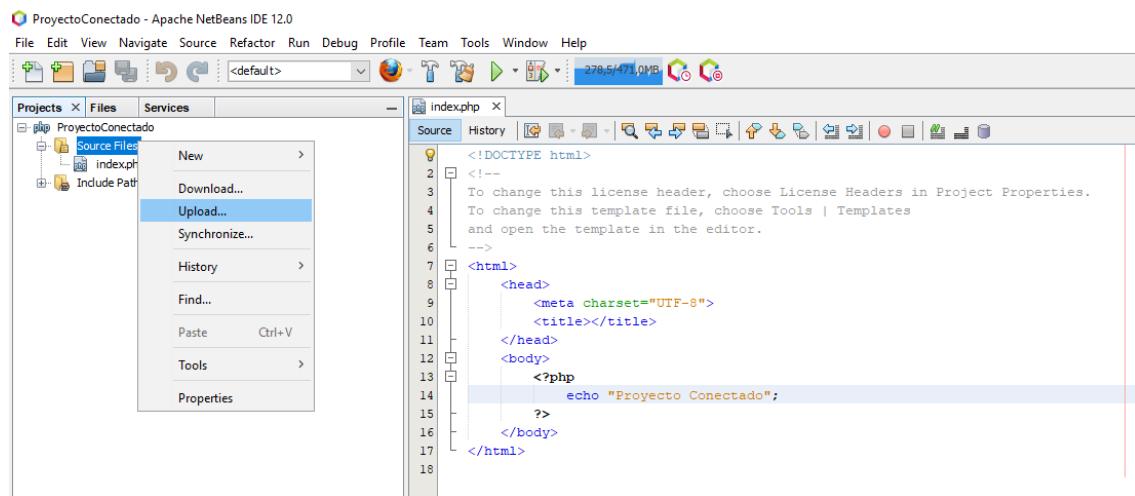
Una vez lo hemos hecho, podremos comprobar en FileZilla que se ha subido el archivo que hemos editado haciendo conexión con operadorweb.



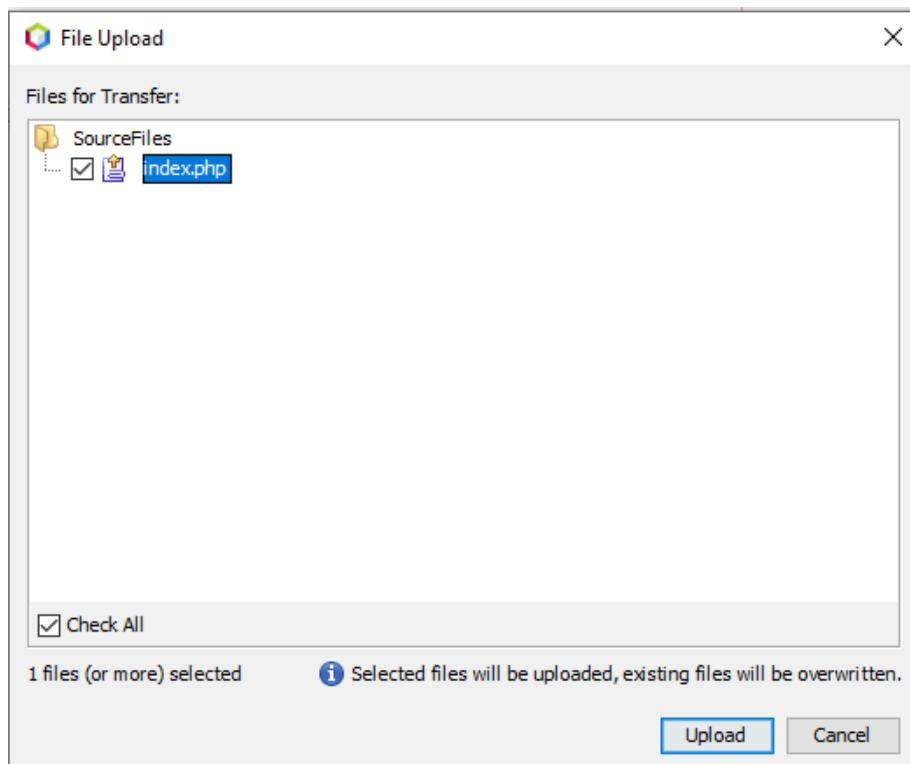
Para comprobar que funciona correctamente, abrimos cualquier navegador e introducimos la IP del servidor y veremos el mensaje escrito anteriormente en NetBeans, de haber hecho bien el proyecto y la conexión nos mostrara el mensaje de PHP.



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.

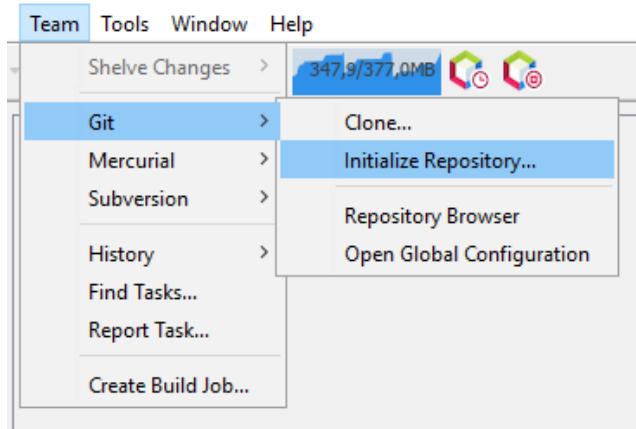


2.9 Administrar un repositorio con GitHub

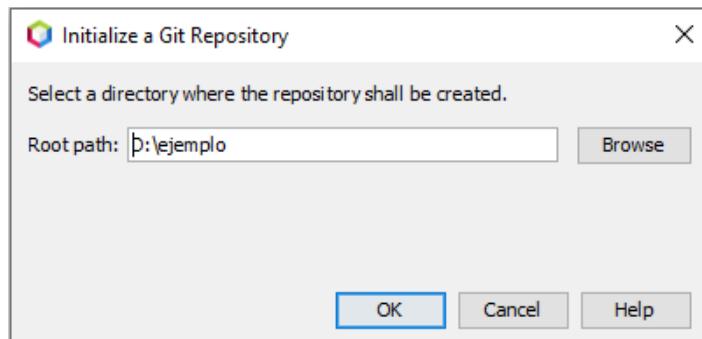
2.9.1 Configuración

2.9.1.1 Subir proyecto

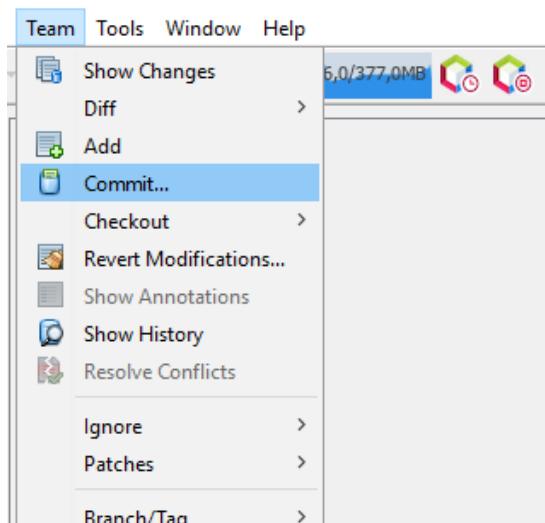
Para subir nuestro proyecto a GitHub, lo primero que tenemos que hacer es lo siguiente, con el proyecto abierto, nos vamos a el apartado Team, Git y Initialize Repository...



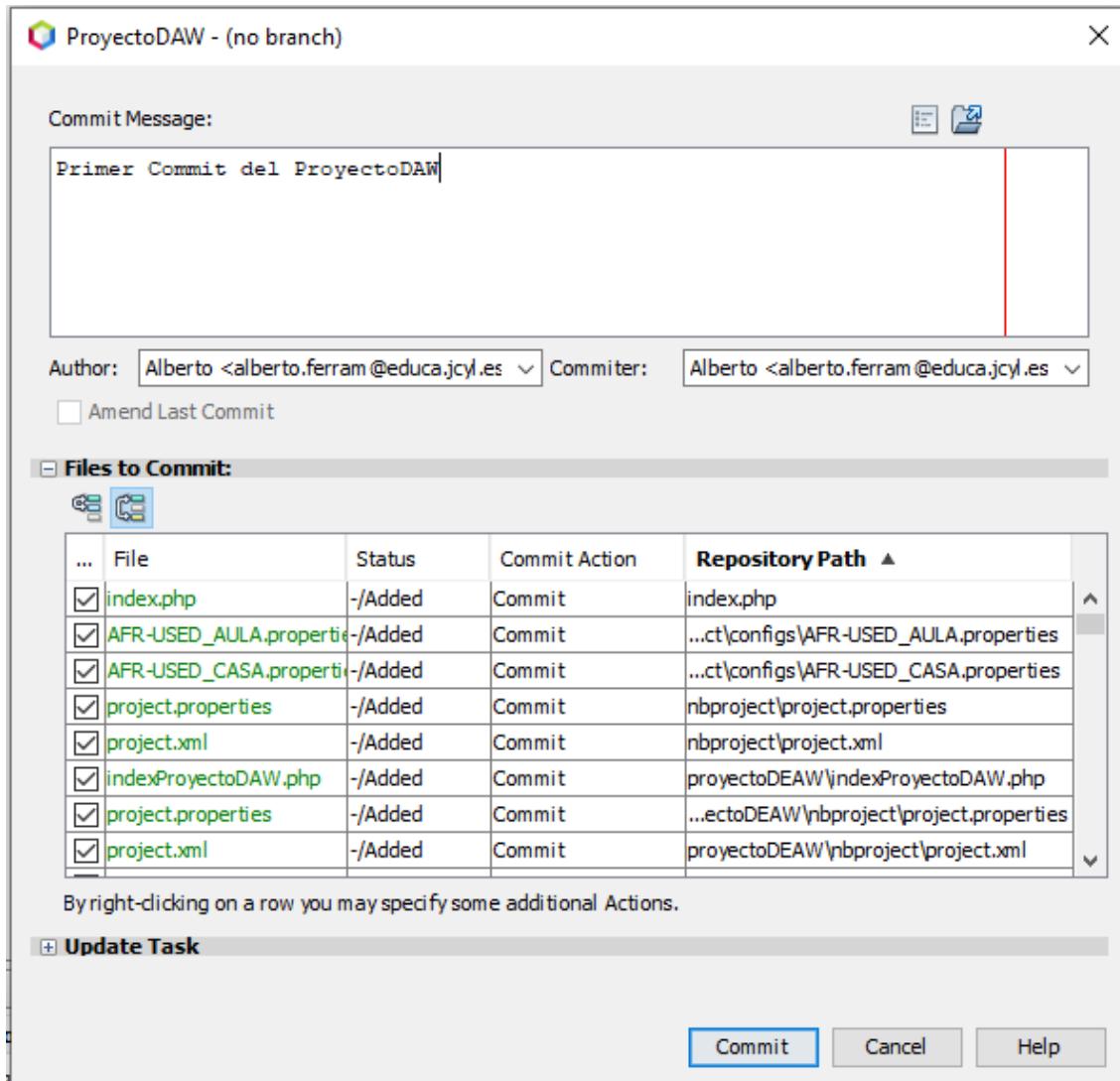
Se nos abrirá esta ventana, la cual nos pedirá que indiquemos el Root path, en mi caso deje la ruta que me dio por defecto ya que era la carpeta donde tengo mi proyecto.



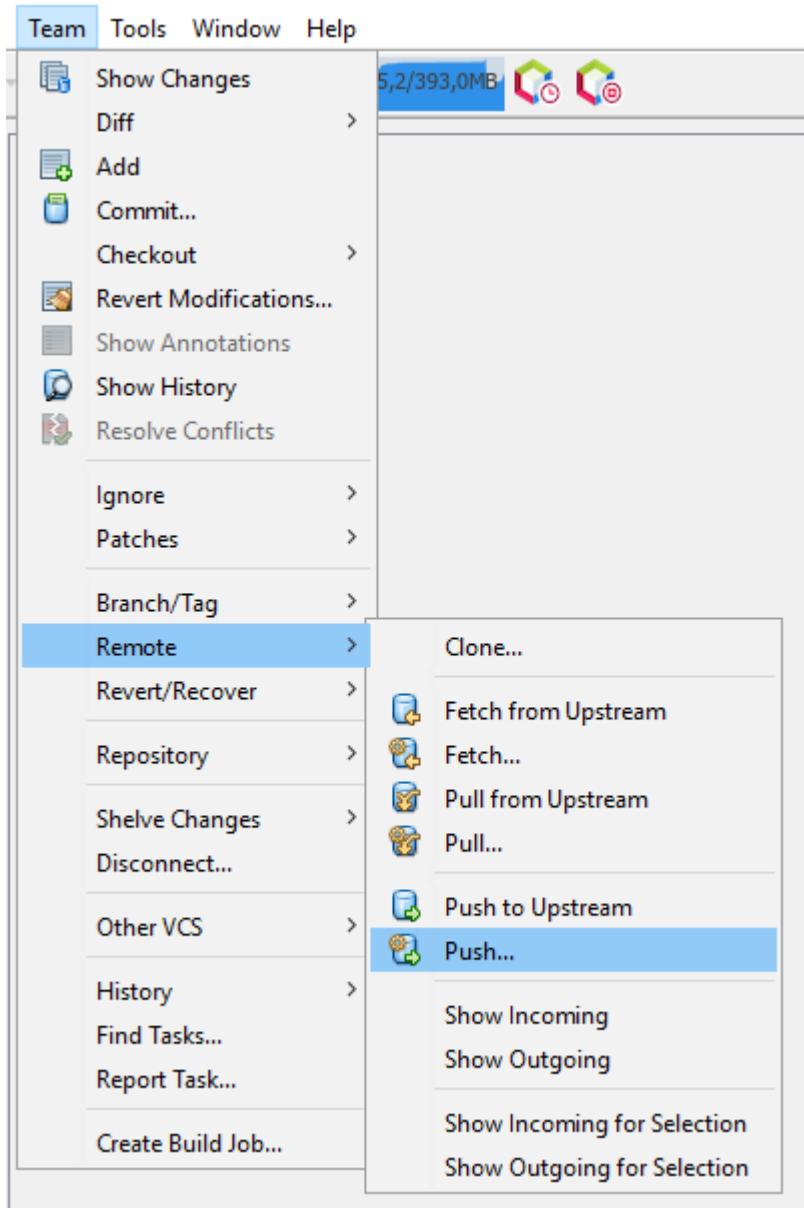
Una vez hecho lo anterior, ahora haremos un Commit de lo que vamos a subir, para ello nos iremos a Team, Commit...



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 GitHub interface for creating a new repository. At the top, it says "Create a new repository". Below that, there's a note: "A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository." The "Owner" dropdown is set to "AlbertoFRSauces". The "Repository name" field contains "proyectoDAW" with a green checkmark next to it. A note below says "Great repository names are short and memorable. Need inspiration? How about [fuzzy-octo-system](#)?" Under "Description (optional)", there is a large empty text area. Below that, there are two radio button options: "Public" (selected) and "Private". The "Public" option is described as "Anyone on the internet can see this repository. You choose who can commit." The "Private" option is described as "You choose who can see and commit to this repository." Under "Initialize this repository with:", there are three checkboxes: "Add a README file" (checked), "Add .gitignore" (unchecked), and "Choose a license" (unchecked). The "Add a README file" checkbox has a note: "This is where you can write a long description for your project. [Learn more](#)." The "Add .gitignore" checkbox has a note: "Choose which files not to track from a list of templates. [Learn more](#)." The "Choose a license" checkbox has a note: "A license tells others what they can and can't do with your code. [Learn more](#)." Below these options, a note says "This will set `main` as the default branch. Change the default name in your [settings](#)." At the bottom is a large green "Create repository" button.

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

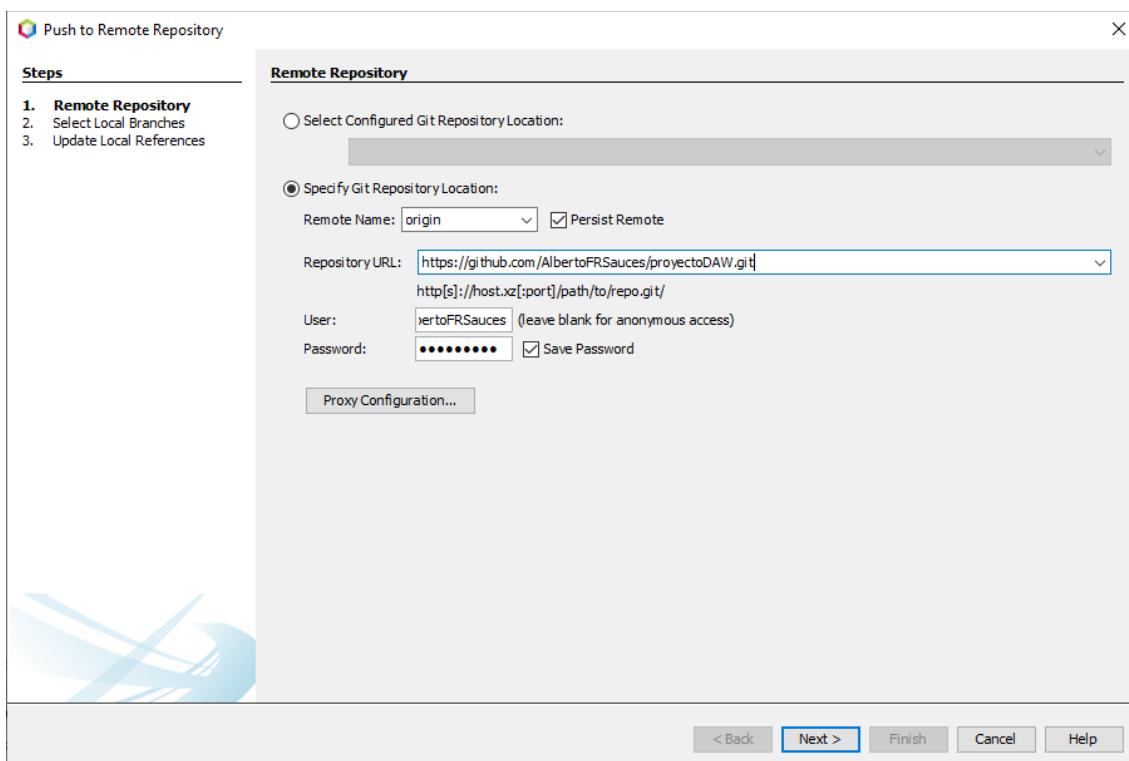
The screenshot shows the GitHub repository page for "proyectoDAW". The repository is public and has 1 branch and 0 tags. It contains one file, "README.md", with the content "Initial commit". On the right side, there is a "Code" tab with a dropdown menu. The "Clone" option is highlighted with a red box, showing the URL "https://github.com/AlbertoFRSauces/proyectoDAW". Other options in the dropdown include "HTTPS", "SSH", "GitHub CLI", and "None". Below the dropdown, there are links for "Go to file", "Add file", and "Code". To the right of the code area, there is an "About" section with the message "No description, website, or topics provided". There are also sections for "Releases" (no releases published) and "Packages" (no packages published). At the bottom, there are links for "Contact GitHub", "Pricing", "API", "Training", "Blog", and "About".

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 **Next**.

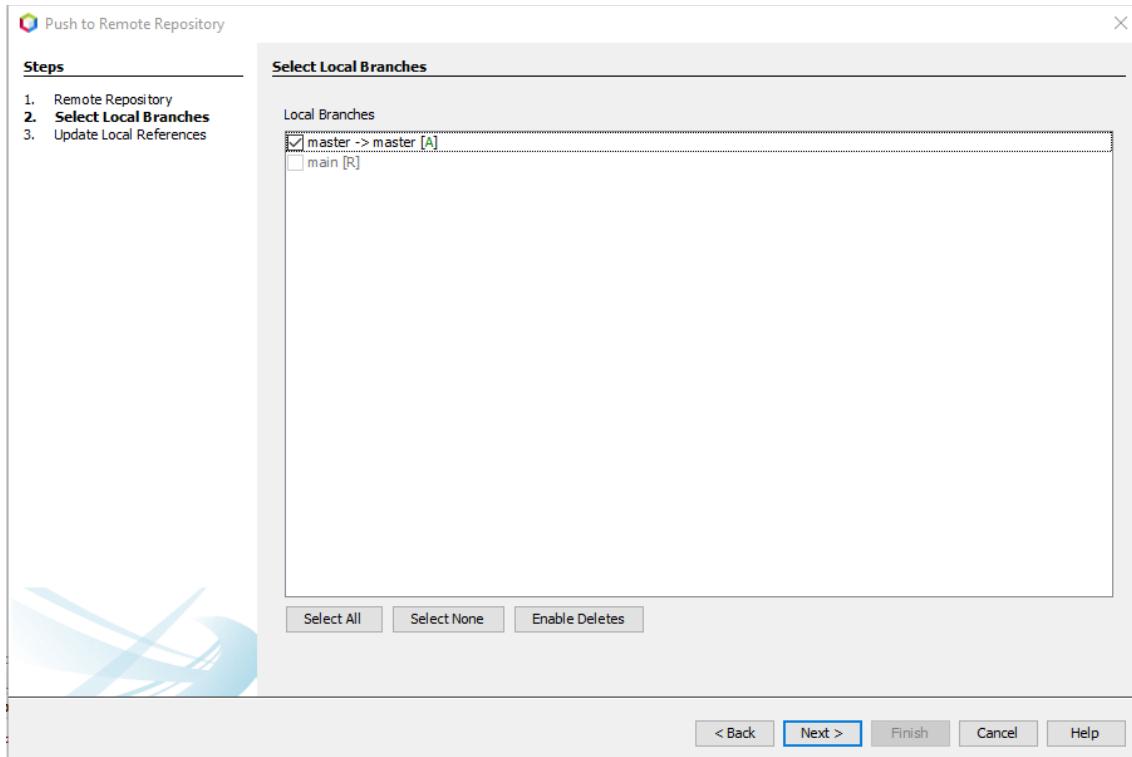
(IMPORTANTE: Si al finalizar todos los pasos nos salta continuamente una ventana en NetBeans de conexión que pide usuario y contraseña de GitHub para subir el proyecto al repositorio, se trata de un ERROR.

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.

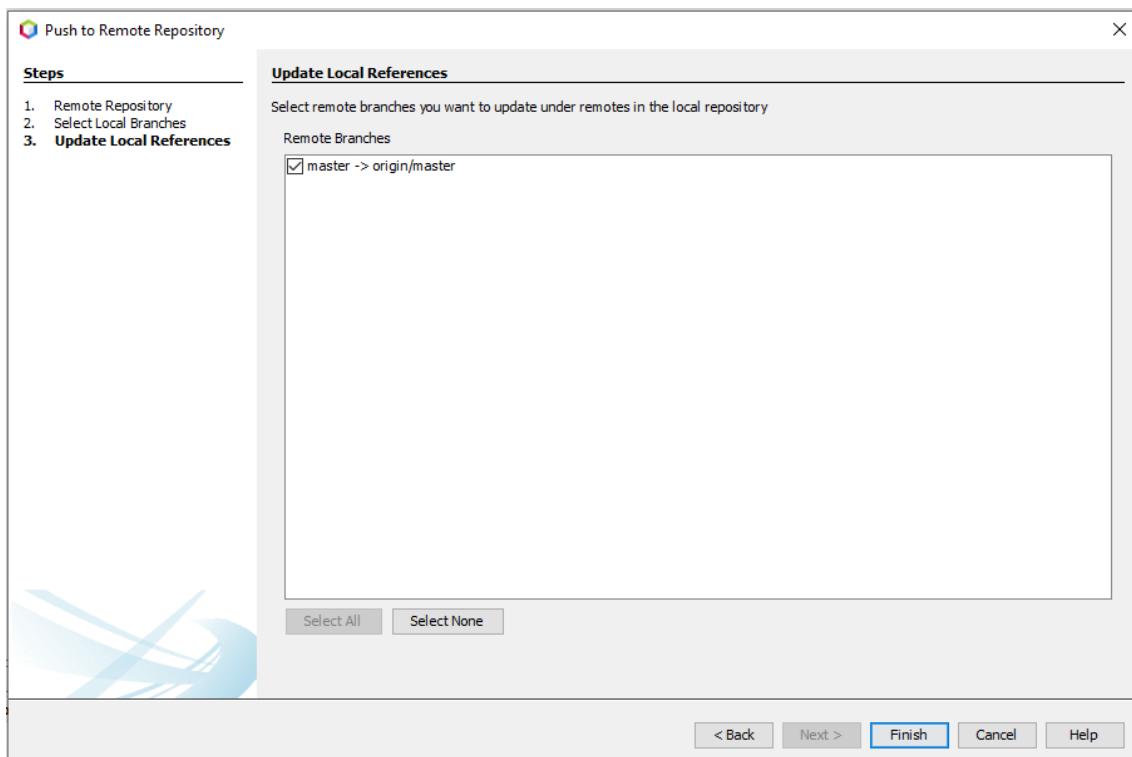
Para crear un Token: <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token>)



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



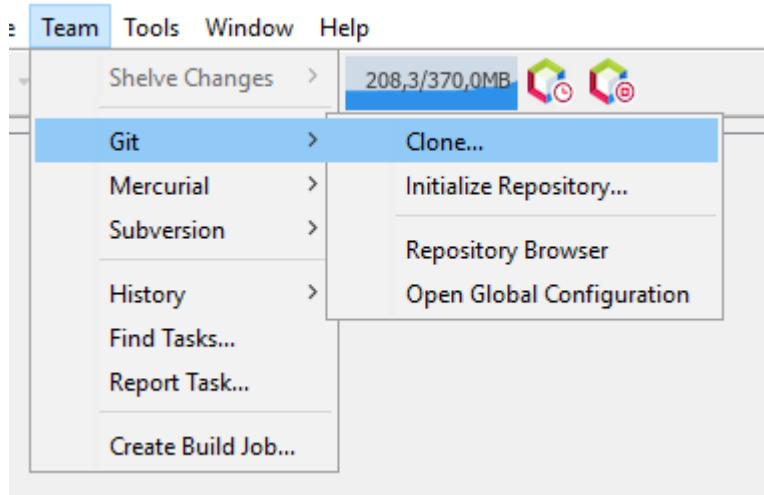
Y también nos pedirá la rama remota para actualizar el repositorio. En este caso la master. Una vez hecho esto le daremos a **Finish** y ya tendríamos el proyecto en nuestro repositorio de GitHub.



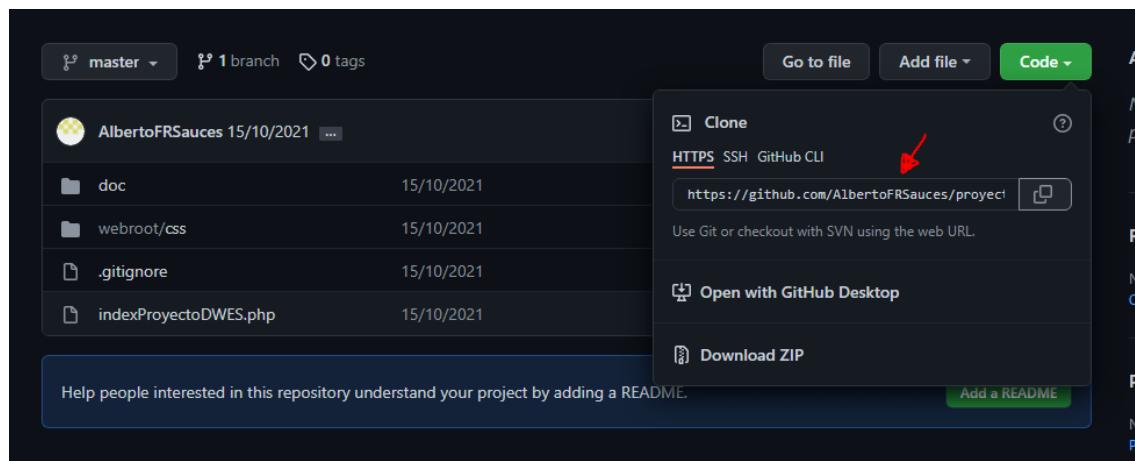
2.9.1.2 Bajar proyecto

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).

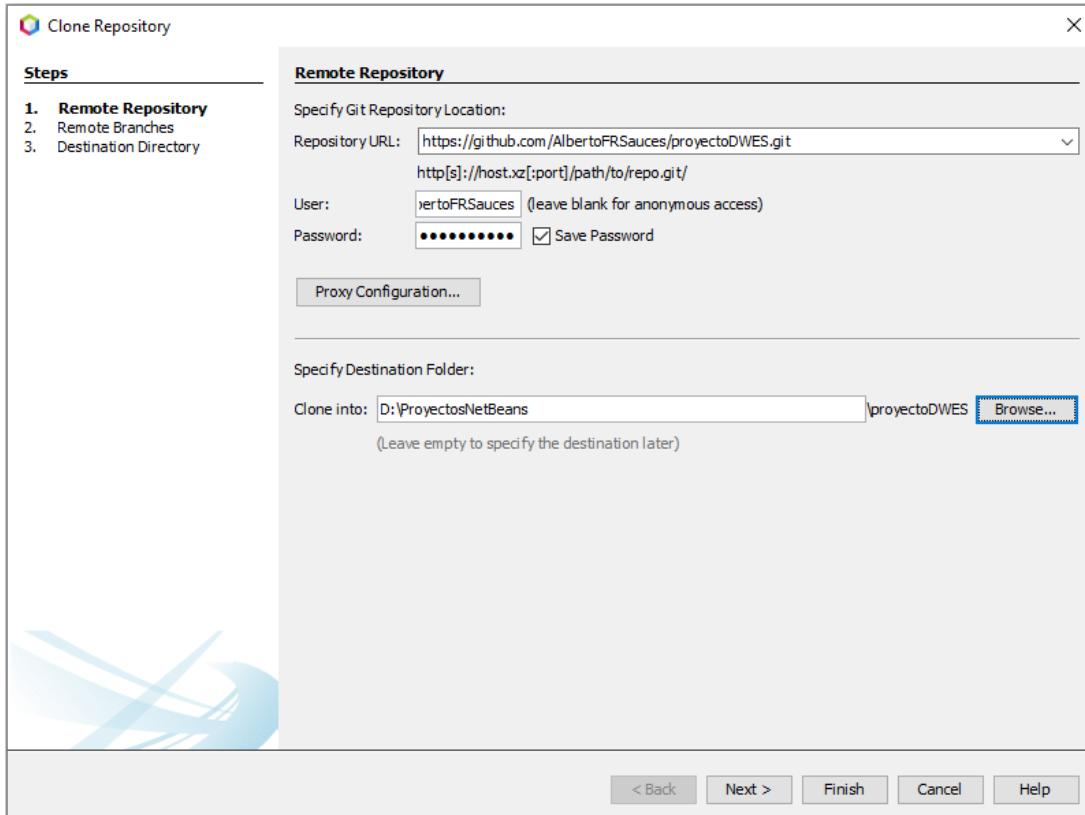
Lo primero será irnos en NetBeans a **Team**, después a **Git** y le daremos a **Clone...**



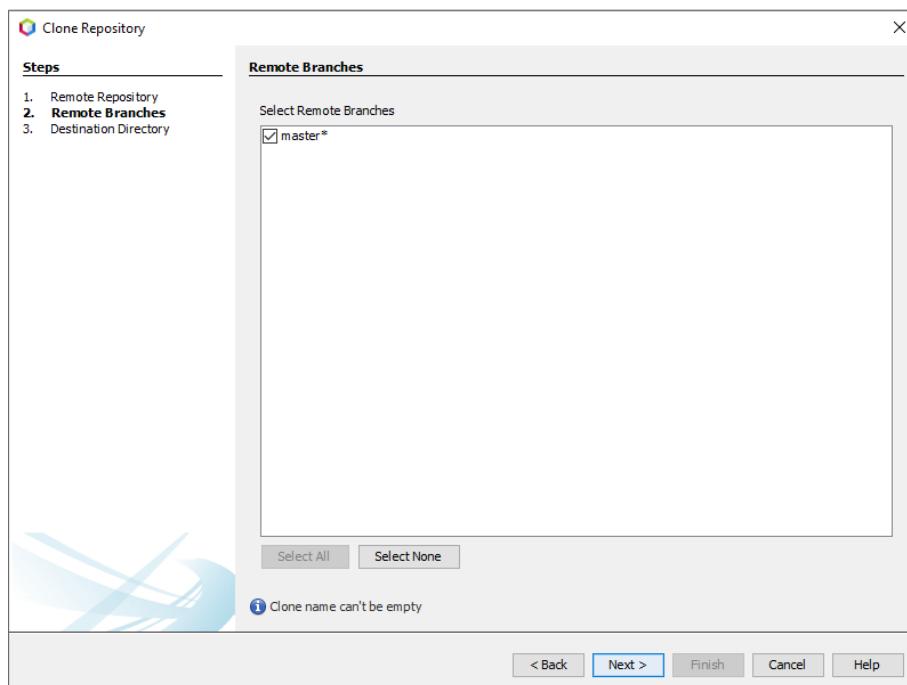
Antes de seguir, nos iremos a nuestro repositorio de GitHub, le daremos a **Code** y en el desplegable copiaremos la URL del apartado de HTTPS por que la vamos a necesitar a continuación.



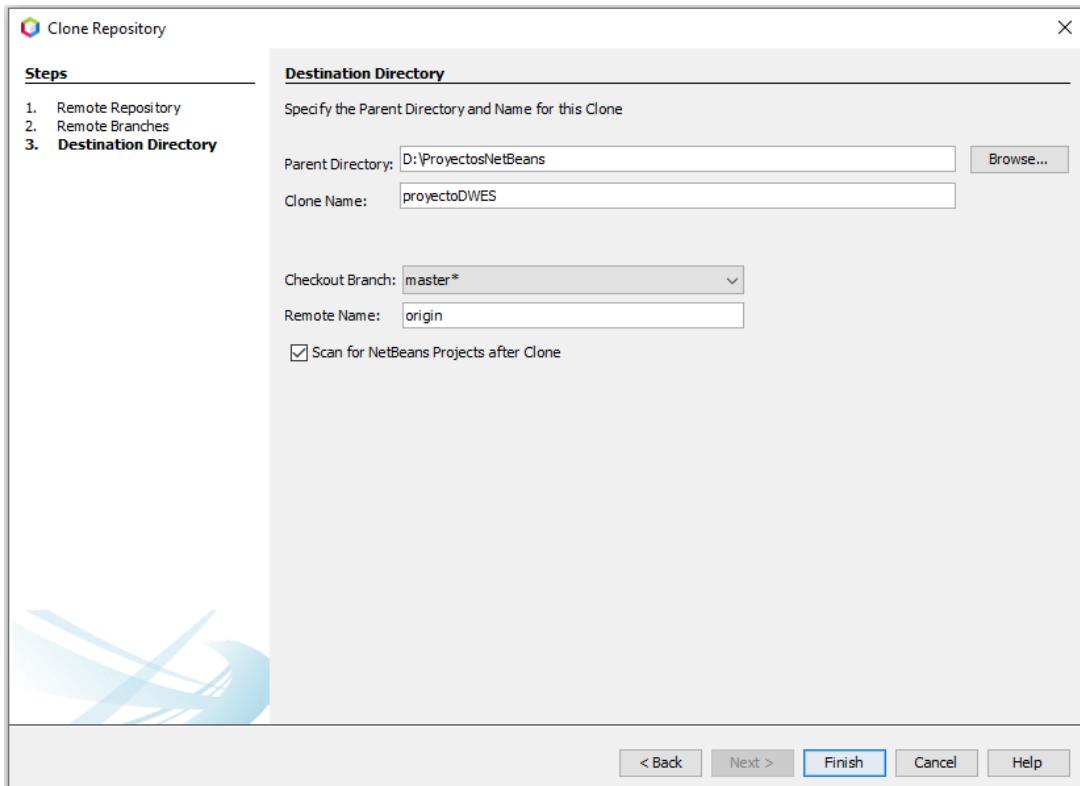
Una vez le hemos dado a la opción Clone, se nos abrirá esta ventana, aquí configuraremos el URL del repositorio que necesitamos (la URL la hemos copiado en el paso anterior). Introduciremos nuestros datos de nuestra cuenta de GitHub, y por último seleccionaremos donde se va a clonar el proyecto (en mi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



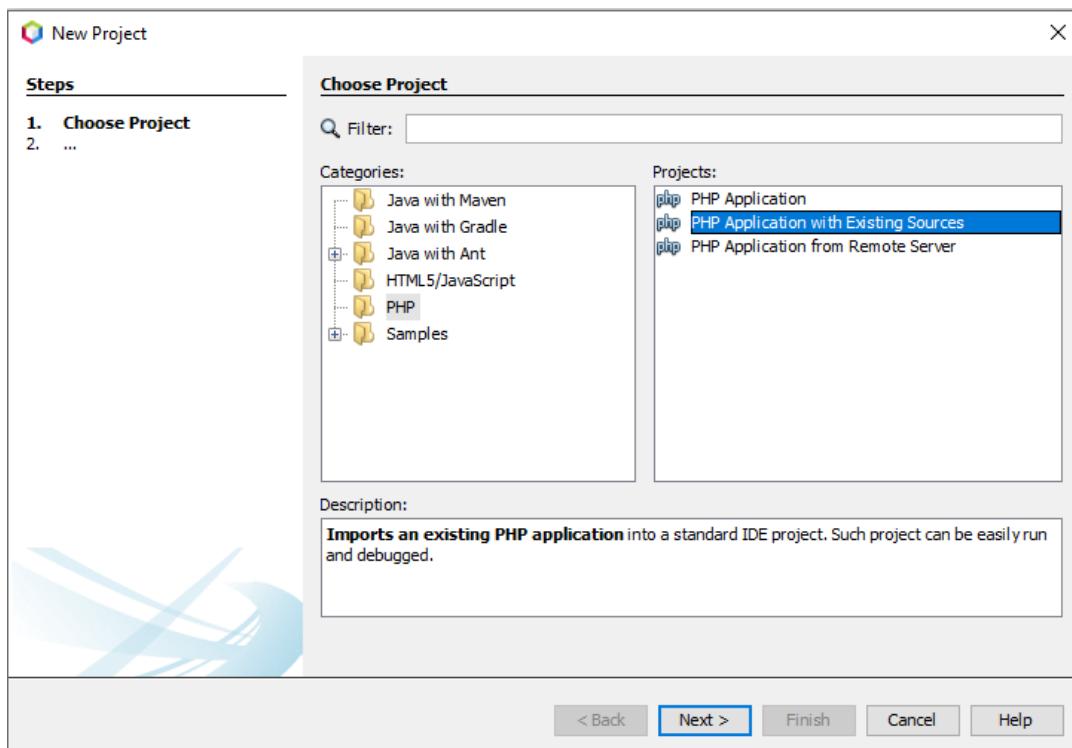
Lo siguiente será seleccionar la rama que vamos a bajar desde GitHub, en mi caso solo tengo la master, en caso de tener varias, seleccionamos la rama que queramos. Una vez seleccionada le daremos a **Next**.



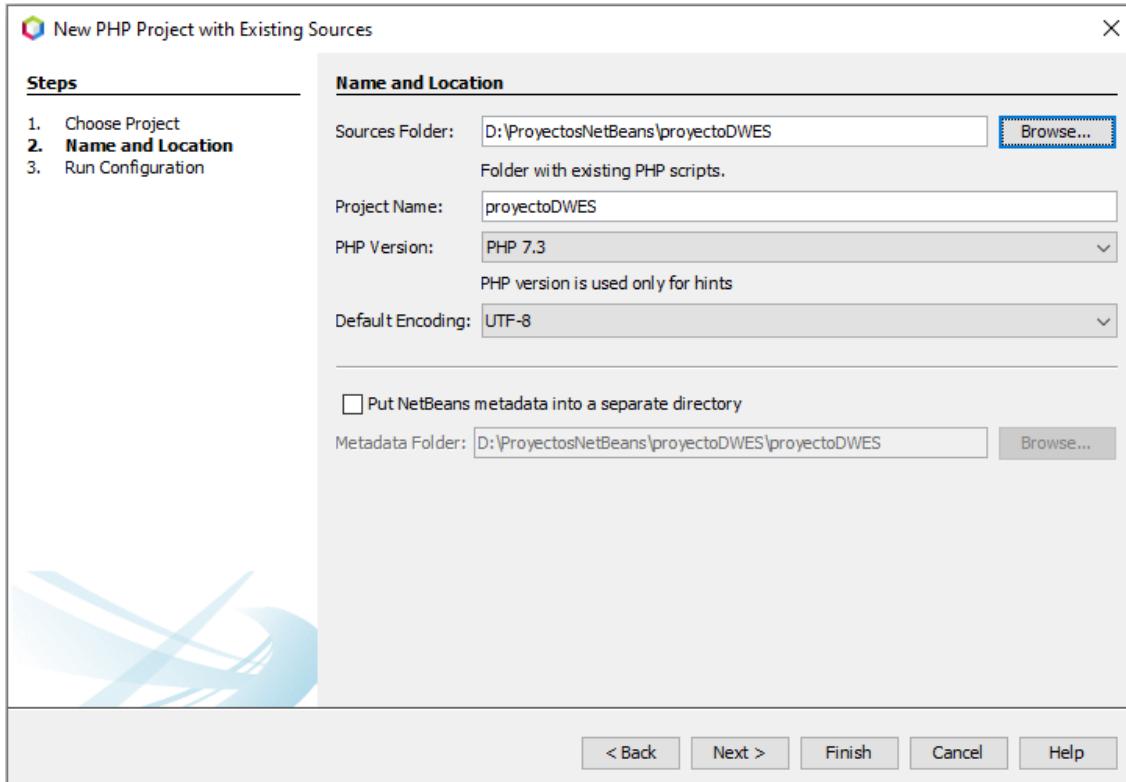
Por último, nos preguntará donde guardar los archivos y que nombre le vamos a dar a la carpeta, en mi caso elijo la carpeta creada anteriormente y le pongo el nombre del proyecto a el nombre de la clonación. Le damos a **Finish** por ultimo.



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 **Next**.



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 **Next**.



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 más el nombre de la carpeta del proyecto.

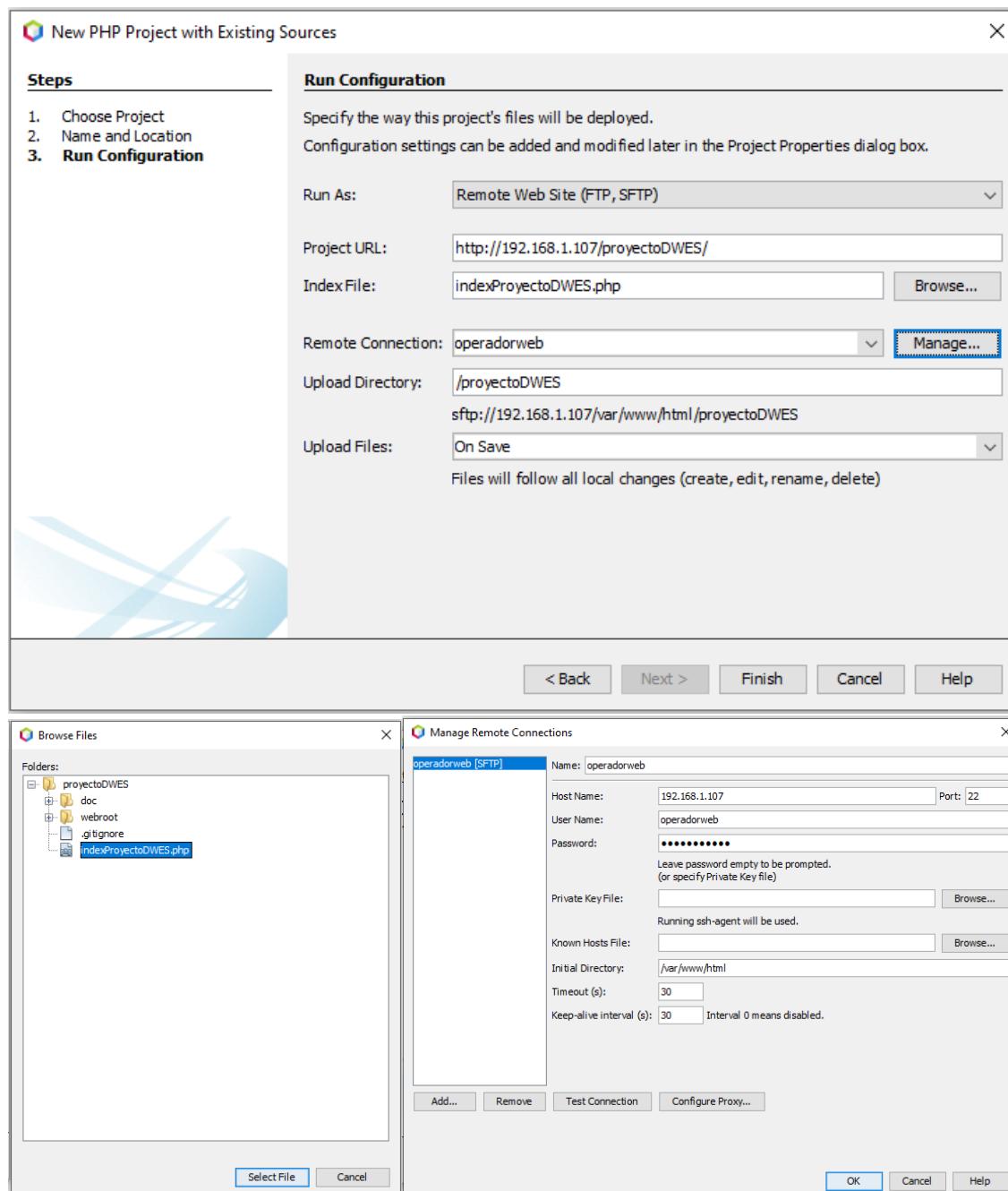
El index, le daremos a browse y seleccionaremos el archivo que es el index, en mi caso se llama **indexProyectoDWES.php**

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**.

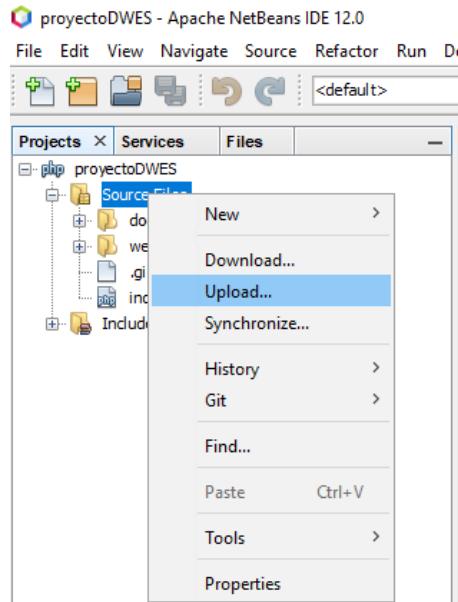
Y configuramos, en mi caso, que se suba a el servidor cuando se guarde algo.

Una vez configurado todo, le damos a **Finish** para finalizar el proceso.

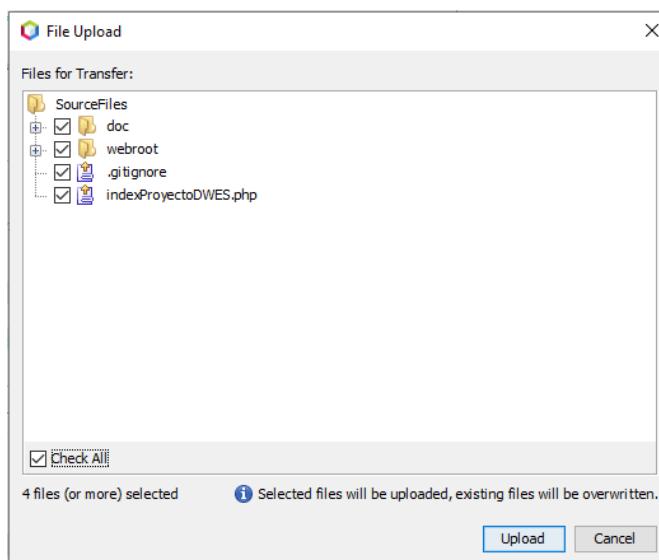


Una vez finalizado el proceso anterior, ya tenemos nuestro proyecto listo para trabajar en él, poder subirlo a nuestro servidor Ubuntu para ver los resultados y hacer Commit para seguir trabajando en él y subirlo a GitHub también.

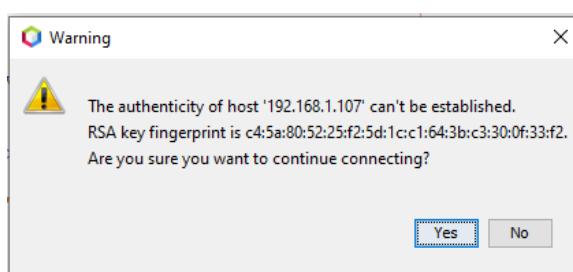
Ahora haremos un **Upload** de los archivos a nuestro servidor de Ubuntu ya conectado.



Nos pedirá que archivos subir, en este caso todos porque está vacío el servidor y le doy a **Upload** para que se suban.



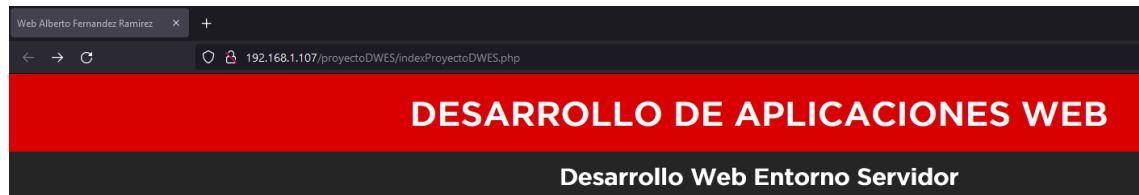
Nos pedirá que, si estamos seguros de seguir con la conexión con el servidor, le diremos que **si** para que se realice la subida de los archivos.



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



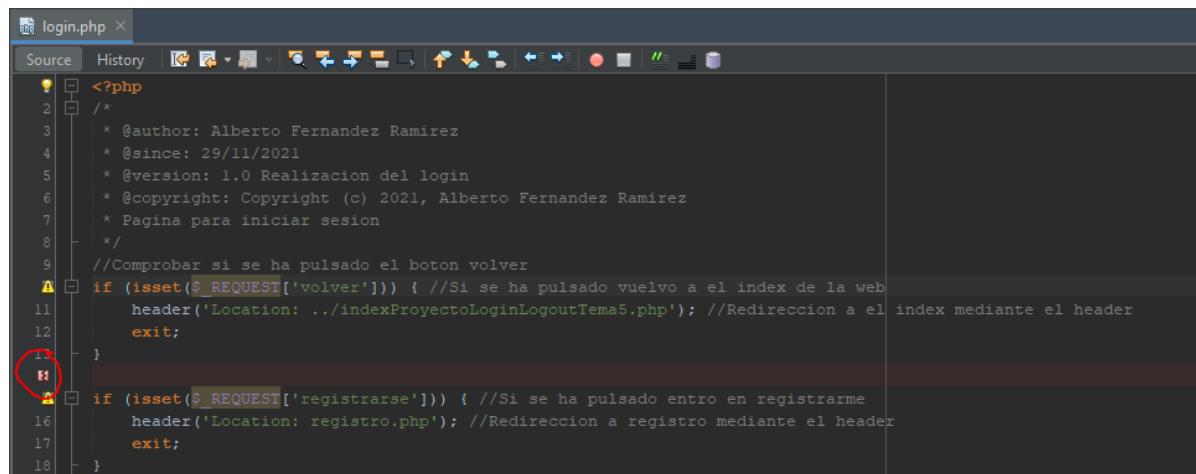
Si todo se ha hecho correctamente, veremos nuestro proyecto corriendo correctamente en el navegador.



2.10 XDebug

2.10.1 Ejemplo de uso

Para depurar código en PHP con XDebug lo primero que tenemos que hacer es marcar los puntos de interrupción donde queremos que el programa PHP se detenga. Es decir, la ejecución del programa PHP se detiene en cada línea donde se establece un punto de interrupción. Para ello solo hay que hacer clic en el número que indica la línea de código que es. Para desactivar un punto de interrupción bastaría con dar clic encima del cuadrado rojo que nos marca el punto.

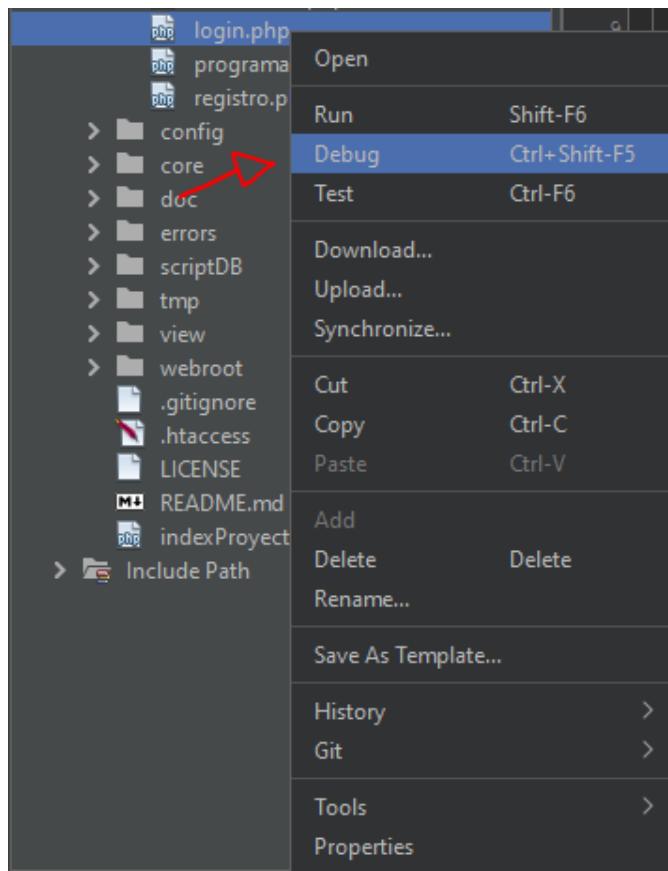


The screenshot shows a code editor window for a file named 'login.php'. The code is as follows:

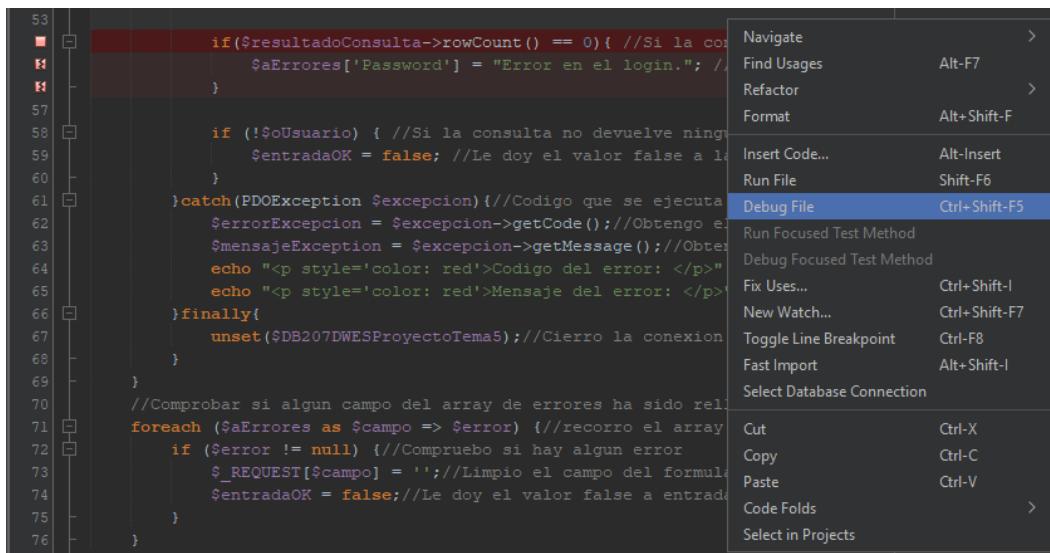
```
<?php
/*
 * @author: Alberto Fernandez Ramirez
 * @since: 29/11/2021
 * @version: 1.0 Realizacion del login
 * @copyright: Copyright (c) 2021, Alberto Fernandez Ramirez
 * Pagina para iniciar sesion
*/
//Comprobar si se ha pulsado el boton volver
if (isset($_REQUEST['volver'])) { //Si se ha pulsado vuelvo a el index de la web
    header('Location: ../indexProyectoLoginLogoutTema5.php'); //Redireccion a el index mediante el header
    exit;
}
if (isset($_REQUEST['registrarse'])) { //Si se ha pulsado entro en registrarme
    header('Location: registro.php'); //Redireccion a registro mediante el header
    exit;
}
```

A red circle highlights the number '13' at the start of the third line of code, indicating it is set as a breakpoint. The code editor interface includes tabs for 'Source' and 'History', and various toolbars and status bars.

A continuación, iniciaremos la sesión de depuración, para ello haremos clic derecho sobre el archivo y le daremos a **Debug**.



También podemos iniciar la sesión de depuración haciendo clic derecho en el propio código y dándole a **Debug File**.



Se nos abrirá un menú de botones, el cual podremos usar para realizar varias opciones.



Finalizar sesión ()	Finalizar la sesión de depuración
Pausa ()	Suspender sesión de depuración
Reanudar ()	Reanudar sesión de depuración
Paso a paso ()	Pasar por encima de una instrucción de ejecución
Entrar ()	Entrar en una llamada a una función
Salir ()	Salir de la llamada a la función actual
Ejecutar en el cursor ()	Ejecutar la ejecución en la posición del cursor

Cuando se alcanza una línea con un punto de interrupción, ejecutaremos el script una línea tras otra presionando los botones “**Paso a paso**” o “**Entrar**” también se puede hacer con **F7** y **F8**. Supervisaremos el estado de la aplicación en las ventanas del depurador.



Una vez terminemos, cerraremos la sesión de depuración en el siguiente botón.



[Guía de uso de XDebug con NetBeans.](#)

2.11 Uso de Git con Visual Studio Code

2.11.1 Configuración

Para poder hacer uso de nuestro repositorio de GitHub con Visual Studio, tendremos que instalar dos extensiones, para ello nos iremos al ícono siguiente que se encuentra en la barra izquierda.



Una vez dentro del Marketplace de extensiones, buscaremos la primera extensión que es “**GitHub Repositories**” y la instalaremos.

EXTENSIONES: MARKETPLACE index.html Extension: GitHub Repositories

GitHub

- GitHub Pull Requests a...** 5M ★ 3 Pull Request and Issue Provider for G...
- GitHub** Instalar
- GitHub Theme** 3.4M ★ 4.5 GitHub theme for VS Code
- GitHub** Instalar
- GitHub** 806K ★ 4.5 Integrates github and its workflows ... KnisterPeter
- GitHub** Instalar
- GitHub Codespaces** 1.1M ★ 5 Your instant dev environment
- GitHub** Instalar
- GitHub Copilot** 945K ★ 4.5 Your AI pair programmer
- GitHub** Instalar
- Markdown Preview ...** 562K ★ 4.5 Changes VS Code's built-in markdown...
- Matt Bierner** Instalar

GitHub Repositories Remotely browse and edit any GitHub...

GitHub

Detalles Contribuciones de características Registro de cambios Estado en tiempo de ejecución

GitHub Repositories

The **GitHub Repositories extension** lets you quickly browse, search, edit, and commit to any remote GitHub repository directly from within Visual Studio Code. Support for Azure Repos (part of Azure DevOps) is in preview, with only readonly operations currently supported.

Why do you need GitHub Repositories?

As developers, we often clone Git repos locally just to browse them or make small edits. We may want to look at the source code of a library we use, experiment with new tools, or just feel the desire to learn something new.

Una vez la tenemos instalada, pasaremos a instalar la otra, que será “**GitLens – Git supercharged**”, esta extensión nos ayudara con la revisión de código y tiene muchas opciones personalizables.

EXTENSIONES: MARKETPLACE index.html Extension: GitLens — Git supercharged

GitLens — Git supercharged v11.6.1

GitKraken | 13.131.360 | ★★★★★(478)

Supercharge the Git capabilities built into Visual Studio Code — Visualize code authorship at a glan...

Detalles Contribuciones de características Registro de cambios Estado en tiempo de ejecución

VS Marketplace v11.7.0 downloads 78.5M rating 4.81/5 (478) Live Share enabled

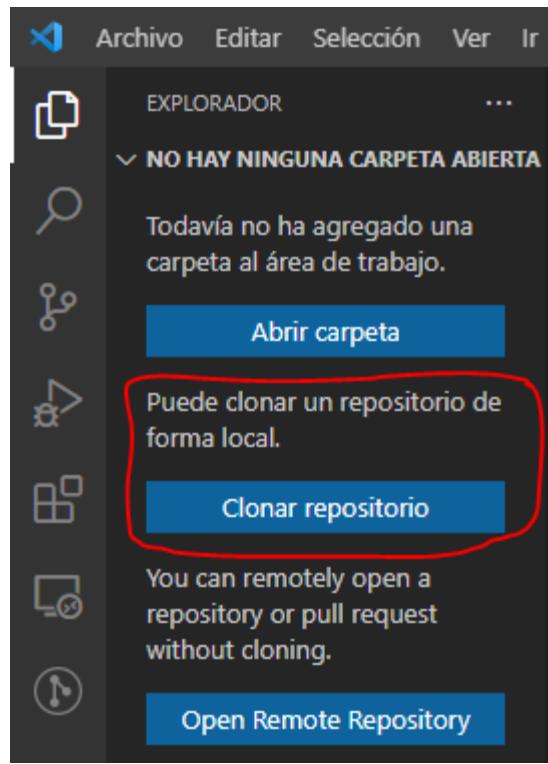
gitlens

GitLens — Git supercharged

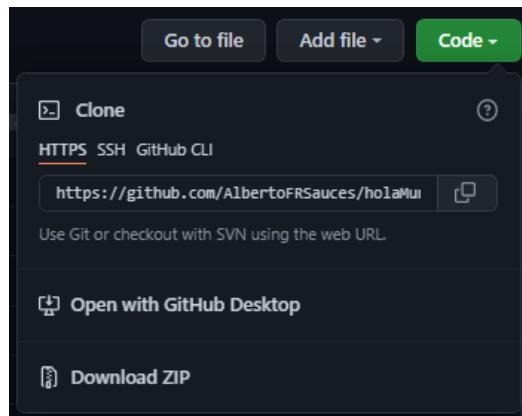
GitKraken + GitLens are joining forces!

2.11.2 Ejemplo de uso

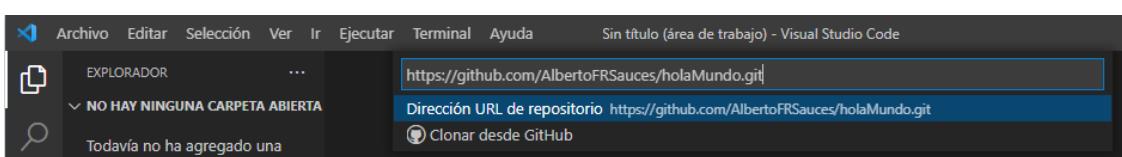
Ahora que ya tenemos instaladas las extensiones, nos iremos al explorador y le daremos a **Clonar repositorio**, para clonar el repositorio con el que queremos trabajar.



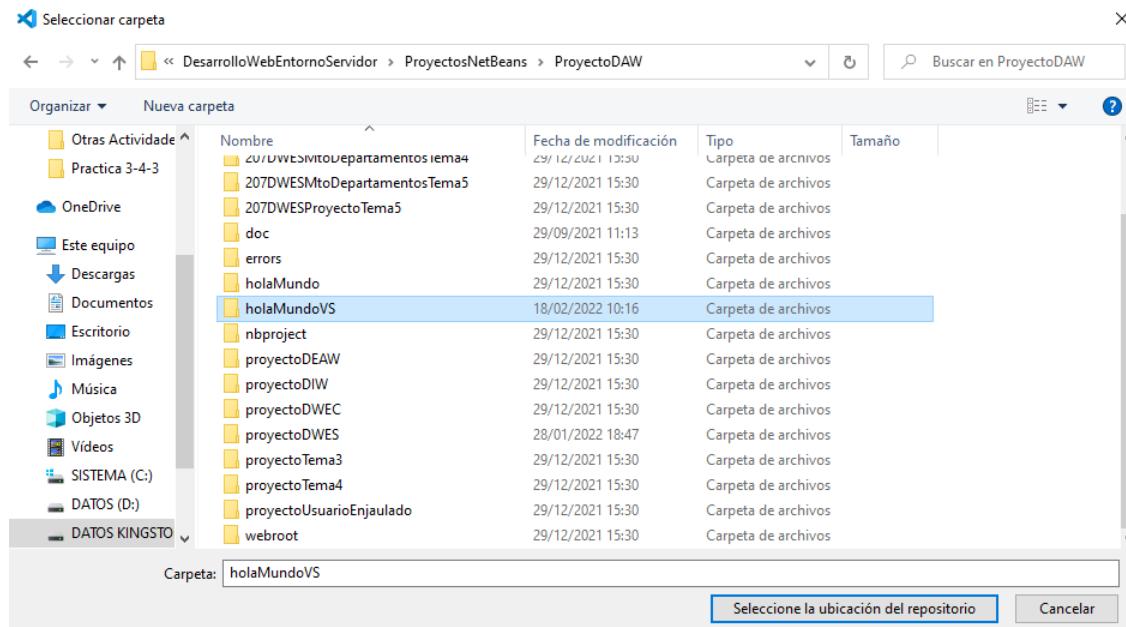
Nos pedirá el enlace del repositorio que queremos clonar, para obtener este enlace me he ido a mi repositorio de GitHub y he cogido el enlace HTTPS en Code.



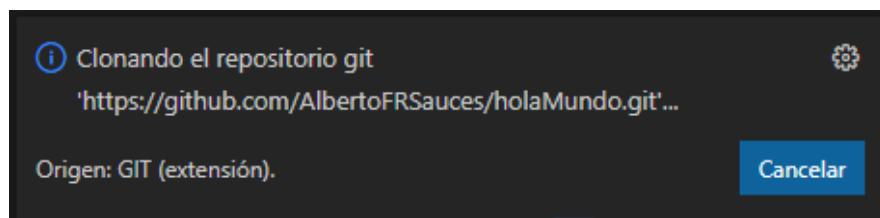
Una vez tenemos copiado el enlace, lo pegaremos y le damos a **Enter**.



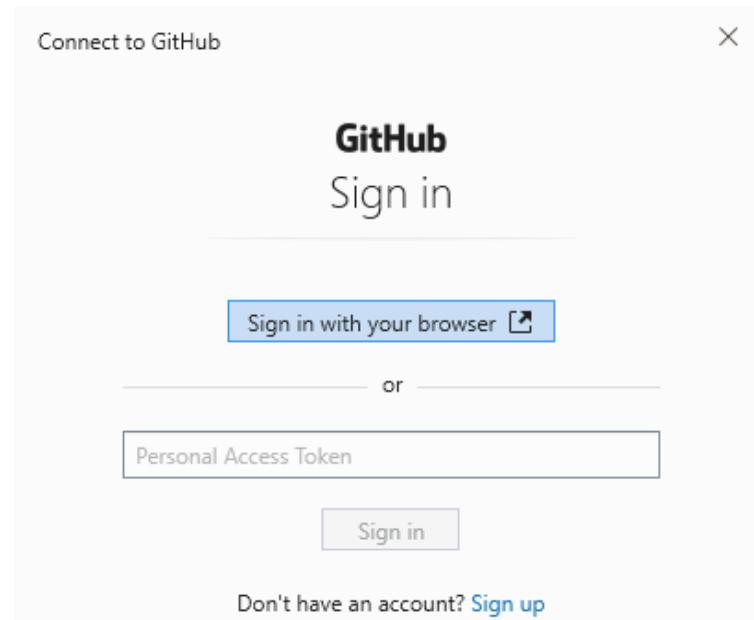
Nos preguntara donde queremos almacenar el proyecto que vamos a clonar, en mi caso he creado una carpeta para el proyecto llamada holaMundoVS que será donde se almacenara.



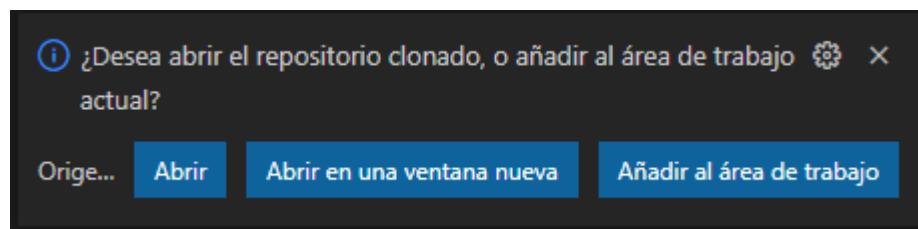
Una vez seleccionada la carpeta, se empezará a realizar la clonación del repositorio.



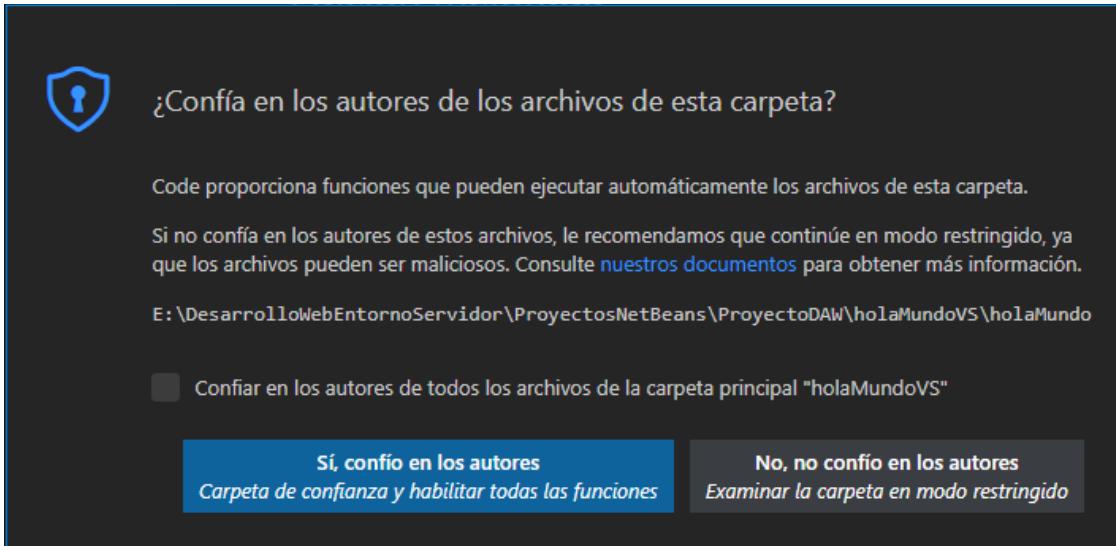
Se nos abrirá una ventana para **iniciar sesión en GitHub** y poder terminar con la clonación del repositorio. En mi caso inicie sesión desde el navegador, pero también se puede hacer con el token personal.



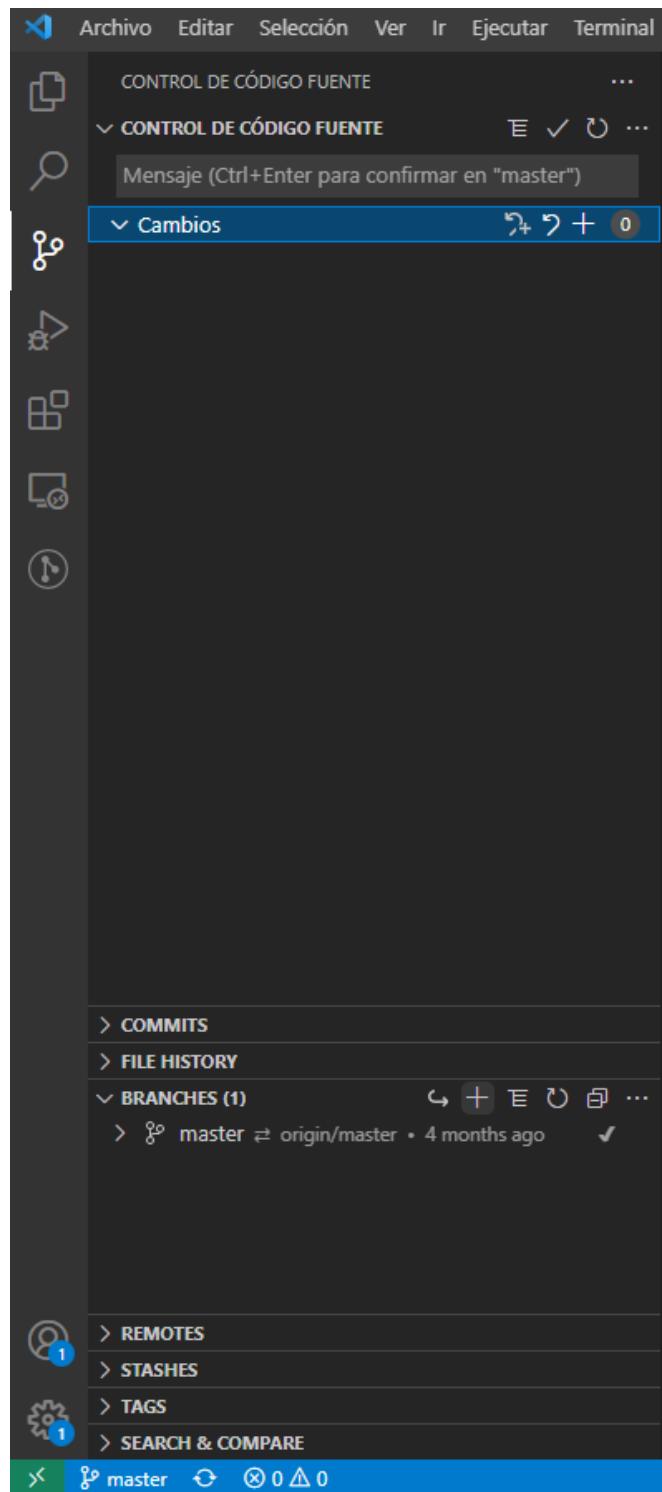
Una vez se termina de clonar, **abrimos en una ventana nueva** el repositorio clonado.



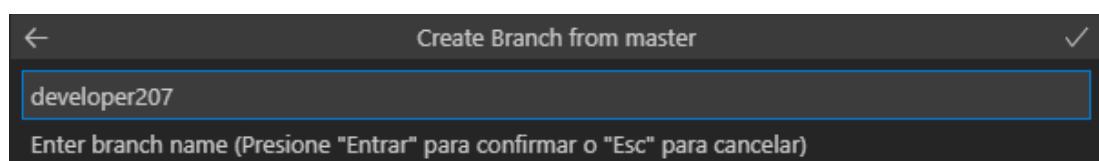
Daremos permiso de confianza en los autores ya que hemos creado nosotros la carpeta y hemos realizado la clonación.



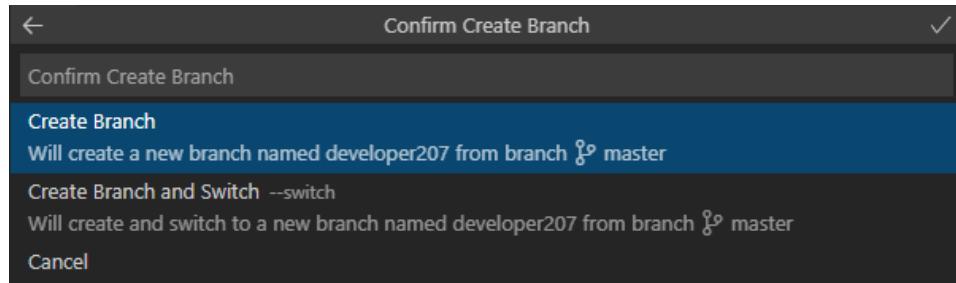
Lo primero será irnos al icono de Git y dentro de **BRANCHES** le daremos al **+** para crear una nueva rama y trabajar con ella.



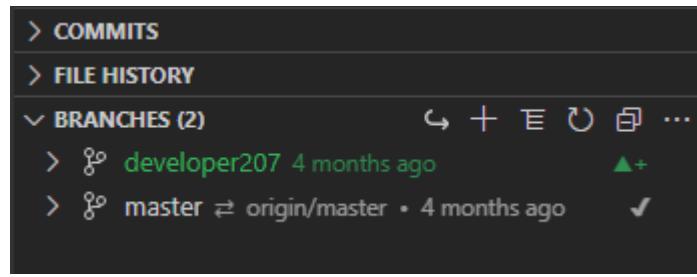
Introduciremos el nombre de la rama que queremos crear.



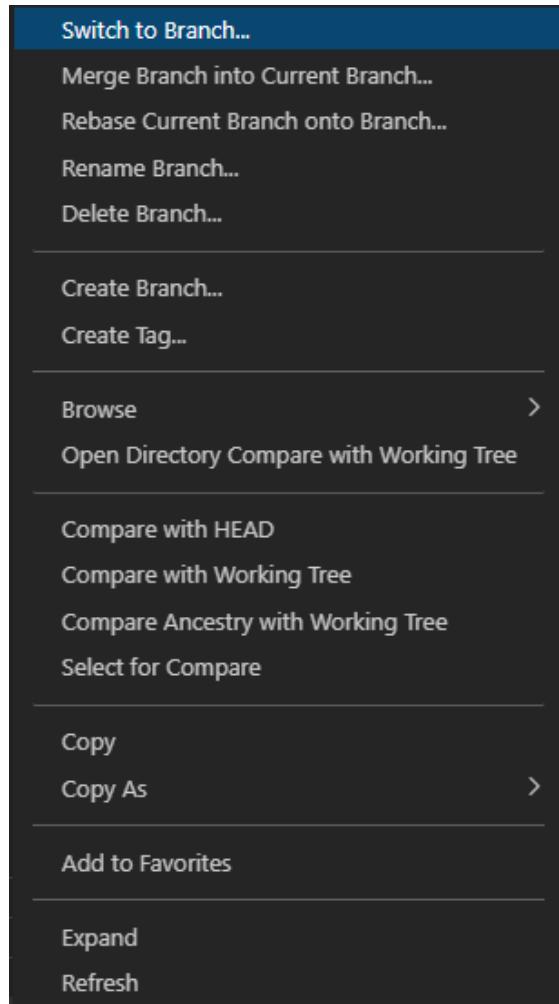
Confirmaremos que queremos crear la rama developer207 desde master dándole a la primera opción.



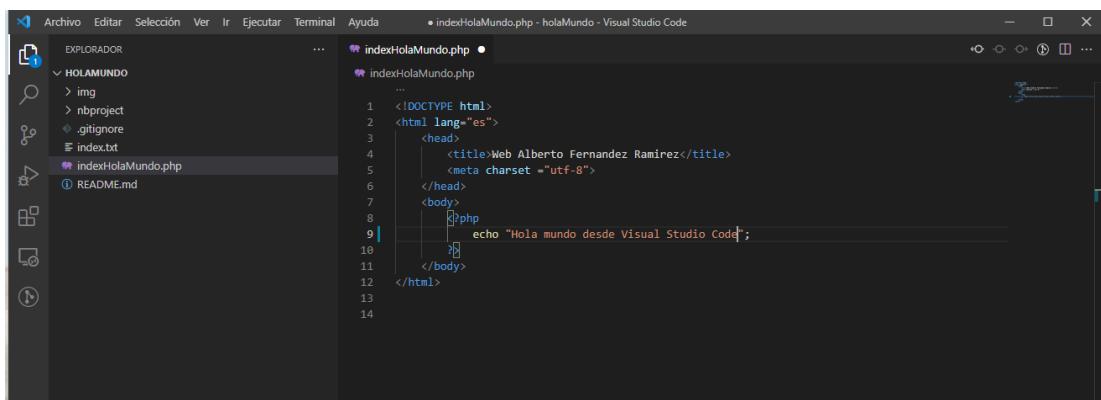
Una vez creada la rama, nos tendremos que mover a ella para hacer cambios.



Haremos **clic derecho en developer207** y le daremos a **Switch to Branch...** para movernos a la rama developer207.

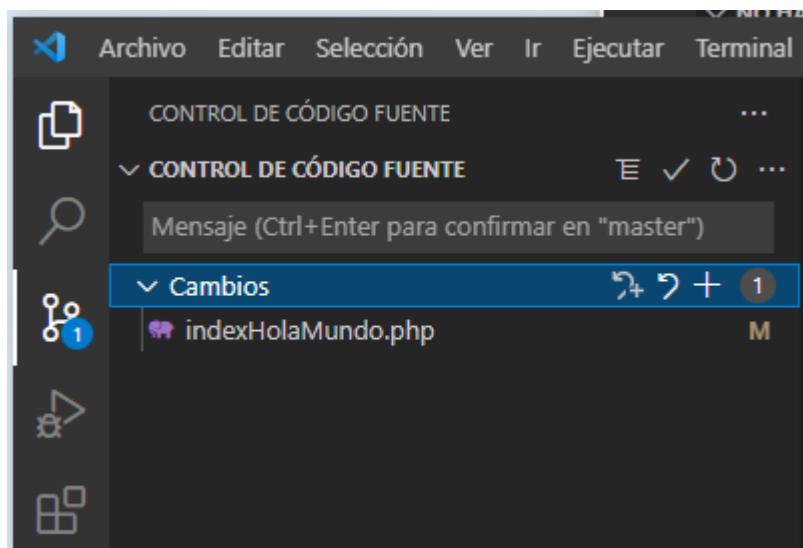


Haremos los cambios necesarios en el archivo y guardamos.

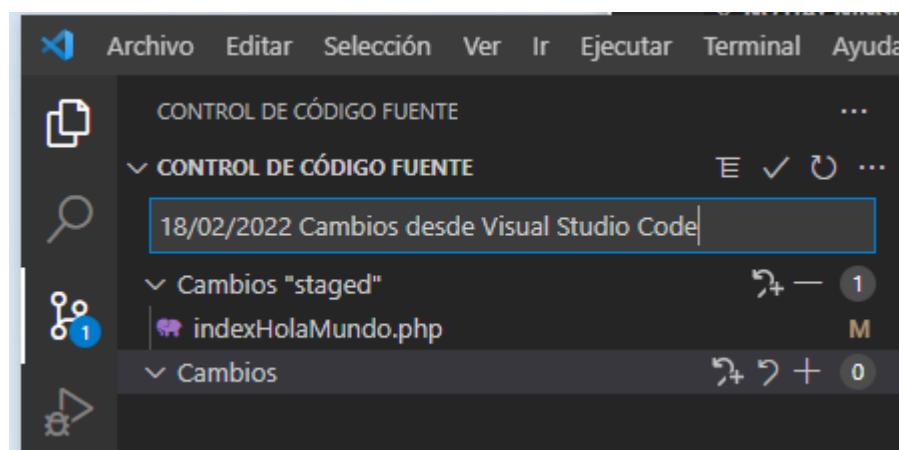


```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Web Alberto Fernandez Ramirez</title>
    <meta charset="utf-8">
</head>
<body>
<?php
echo "Hola mundo desde Visual Studio Code";
</body>
</html>
```

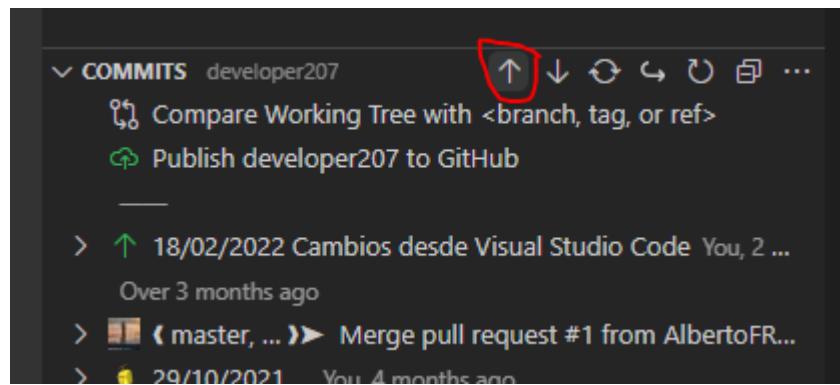
Nos moveremos a **Git** de nuevo y podremos ver que hay un apartado llamado Cambios que nos indica que archivos han sido modificados.



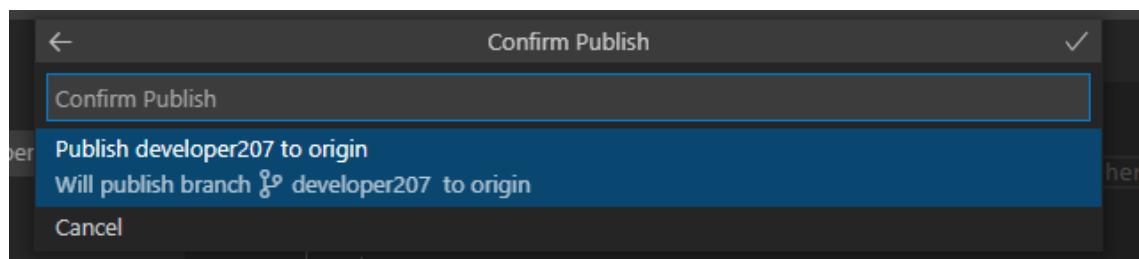
Para hacer el commit de los cambios realizados, introduciremos dentro del campo Mensaje, el mensaje del commit y le daremos a **Enter**.



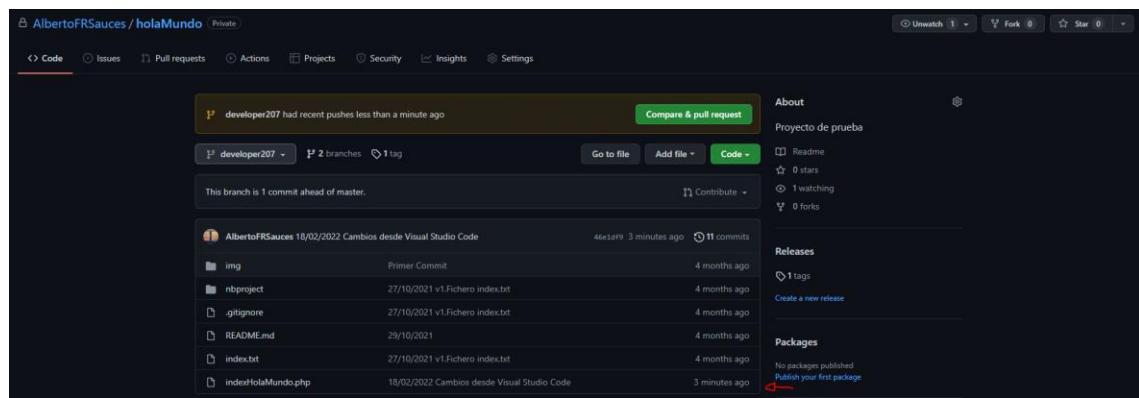
Una vez tenemos el commit, tendremos que hacer el **push**, para ello en el apartado de **COMMITs** le daremos a la **flecha hacia arriba** para hacer el push.



Nos preguntara si queremos publicar la rama developer207 en origin, le daremos a la **primera opcion** y se realizara el push del contenido.



Una vez hemos publicado la rama developer207 en origin ya podremos irnos al repositorio de GitHub y comprobar que los cambios se han realizado correctamente.



ULTIMA MODIFICACION 18/02/2022 – 20:10