

TEMA 2 – INSTALACIÓN, CONFIGURACIÓN Y DOCUMENTACIÓN DEL ENTORNO DE DESARROLLO Y DEL ENTORNO DE EXPLOTACIÓN

Sumario

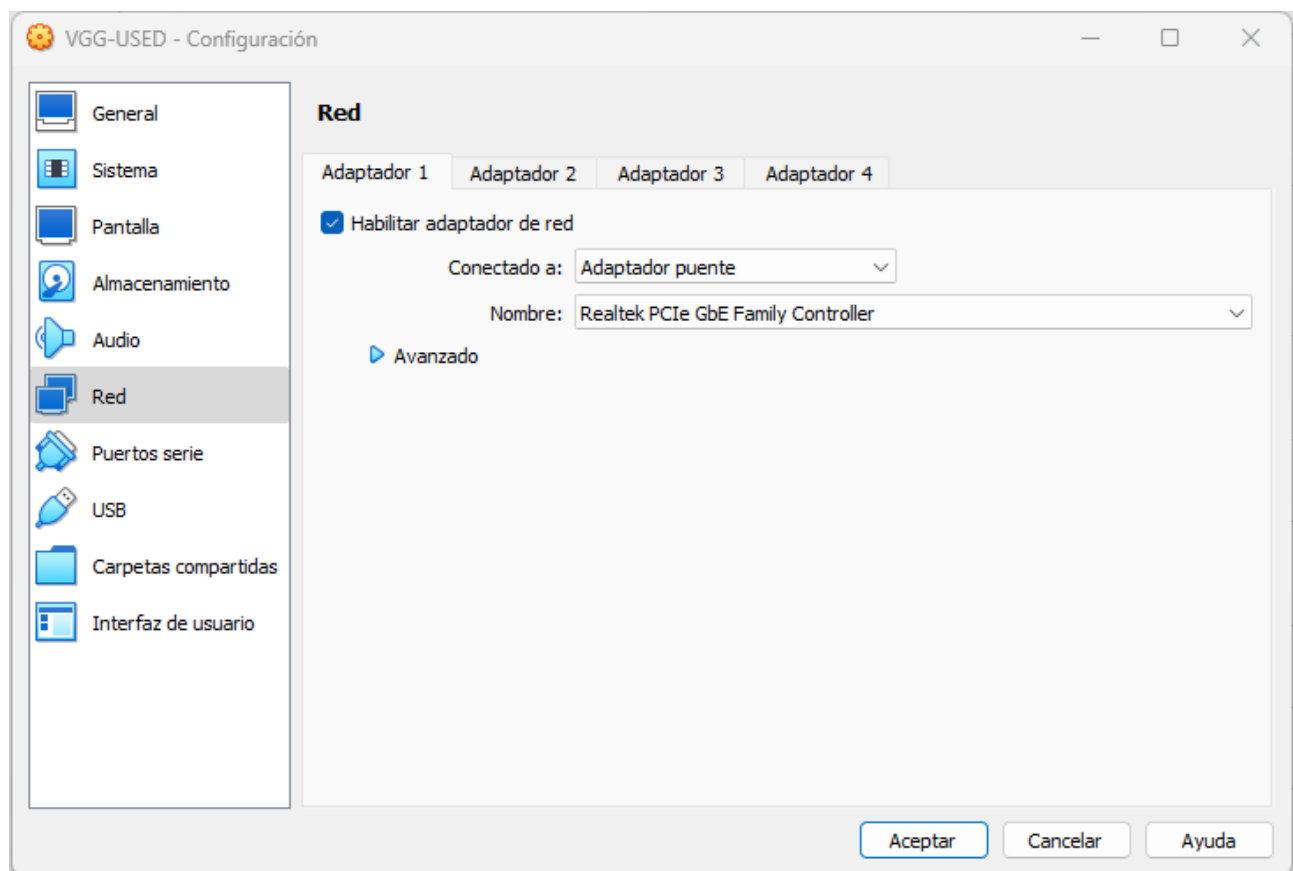
USED – Ubuntu Server.....	2
Configuración inicial.....	2
Cuentas de administración.....	8
Apache.....	8
PHP.....	11
MySQL.....	12
Xdebug.....	12
Cuentas de desarrollo y hosting virtual.....	13
GITHUB – Internet.....	13
Cuentas de desarrollador.....	13
WXED – Windows X.....	14
Instalación y configuración inicial de la máquina.....	14
Cuentas administradoras y cuenta de desarrollador.....	15
Navegadores.....	16
Filezilla.....	16
Notepad++.....	16
NetBeans.....	17
Instalación y configuración inicial (plugins).....	17
Creación de proyectos, modificación, borrado, prueba.....	18
Conexión al servidor remoto SFTP. (Almacenamiento local/almacenamiento remoto).....	21
Administración de la base de datos.....	22
Conexión al repositorio – versionado.....	22
Depuración – Configuración de la ejecución para la depuración.....	26
Paso a explotación.....	26
PHP Doc.....	26
CSS / JS / AJAX / XML /JSON.....	26

USED – Ubuntu Server

Configuración inicial

- Configuración de red

Lo primero que hay que hacer es poner la máquina virtual en adaptador puente.



Ahora hay que abrir el cmd en el equipo cliente y poner el comando `ipconfig /all` para ver la ip, la puerta de enlace y todo lo necesario para configurar la red.

Una vez instalada entramos al directorio `/etc/netplan` y abrimos el archivo que contiene ya sea para comprobar que está todo correcto o para cambiar algún dato.

```

Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
miadmin@daw-limpia:~$ cd /etc/netplan
miadmin@daw-limpia:/etc/netplan$ sudo nano 50-cloud-init.yaml

```

```

Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 7.2
# This file is generated from information provided
# to it will not persist across an instance reboot
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      addresses:
        - 192.168.0.229/24
      nameservers:
        addresses:
          - 8.8.8.8
        search: []
      routes:
        - to: default
          via: 192.168.0.1
  version: 2

```

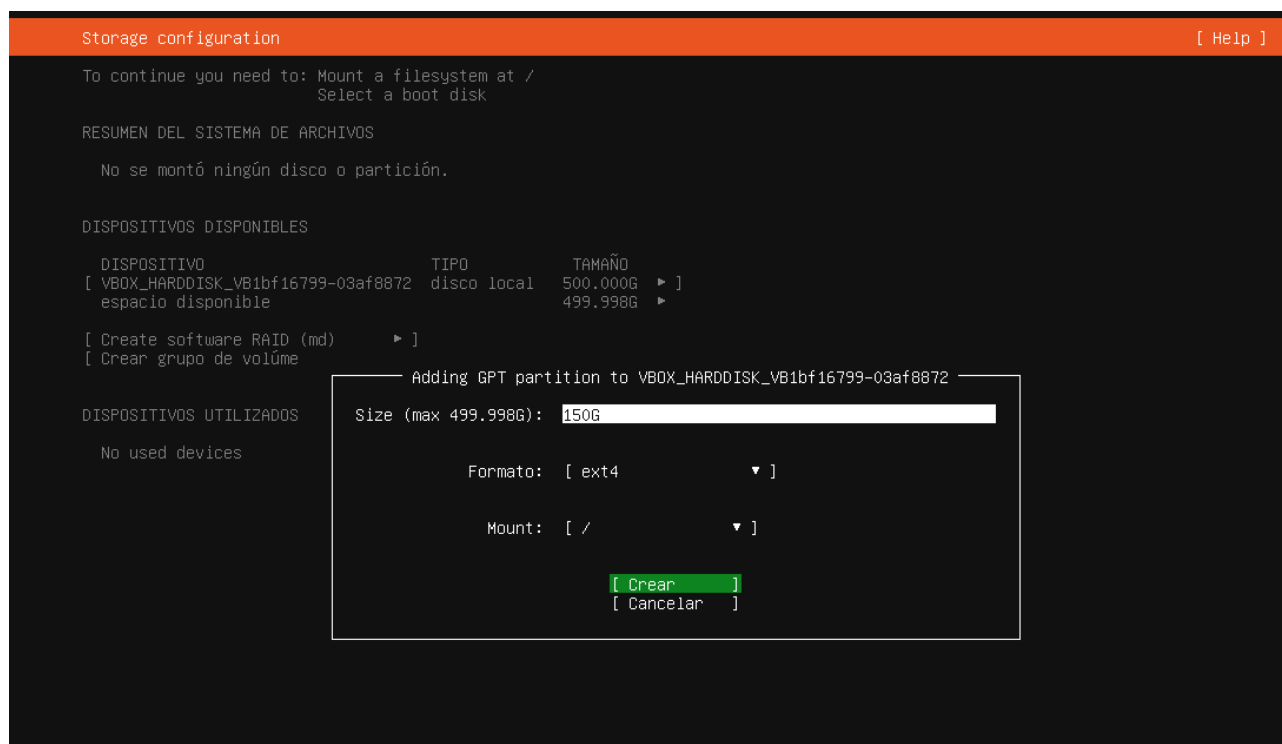
Para comprobar que la red funciona correctamente hacemos ping a la Google (8.8.8.8).

```
miadmin@daw-limpia:~$ 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=117 time=19.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=21.3 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=20.3 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=19.8 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=117 time=19.8 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=117 time=19.8 ms
^C
--- 8.8.8.8 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5083ms
rtt min/avg/max/mdev = 19.759/20.114/21.274/0.549 ms
miadmin@daw-limpia:~$ _
```

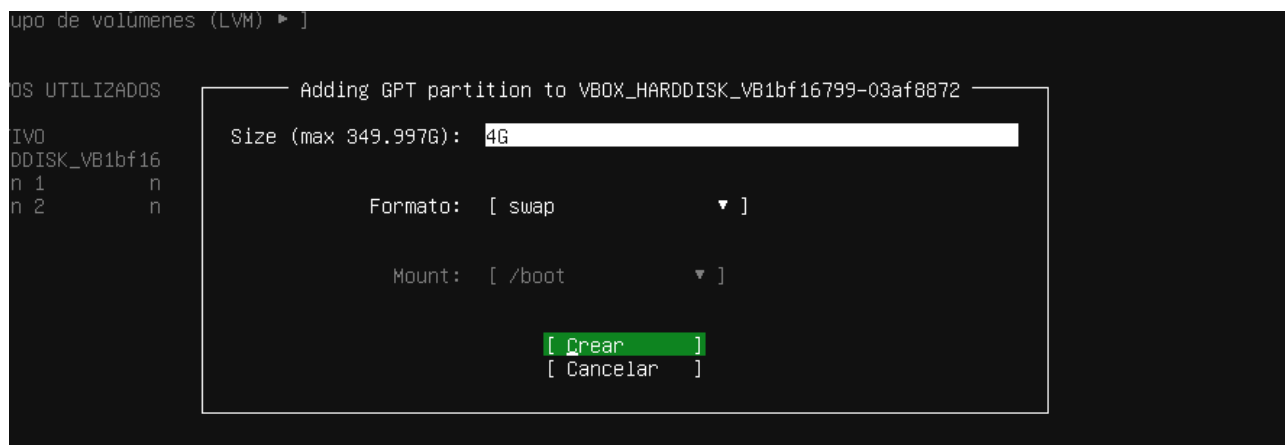
- Particiones

De los 500 GB que le hemos puesto a nuestra máquina tendremos que hacer tres particiones.

La primera es para el sistema y es de 150 GB.



La segunda es la swap y es del tamaño de la RAM * 2 así que en nuestro caso como hemos puesto 2 GB de RAM pues entonces de 4 GB.

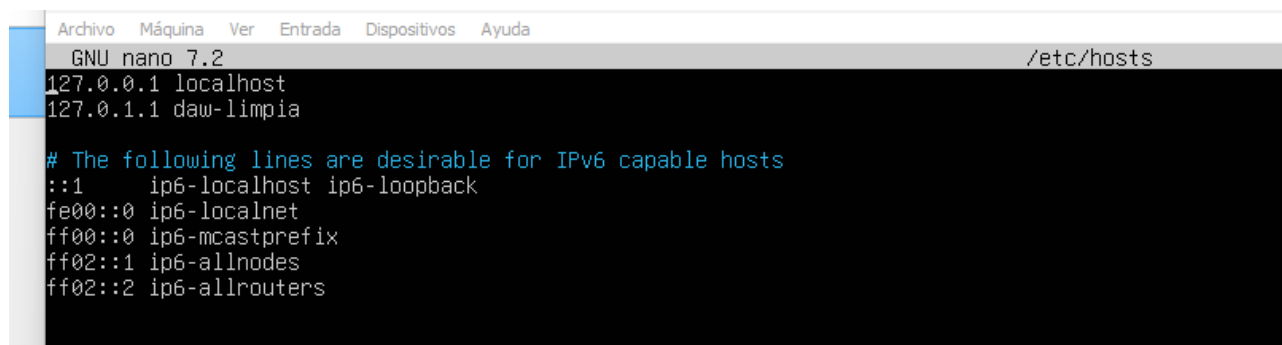


La ultima es para los datos (/var) y es del resto del espacio del disco.



- Cambio de nombre del equipo, zona horaria y actualización del sistema

Para ver el nombre del equipo en el fichero /etc/hosts y lo cambiaremos con el comando `sudo hostnamectl hostname [nombre]`.

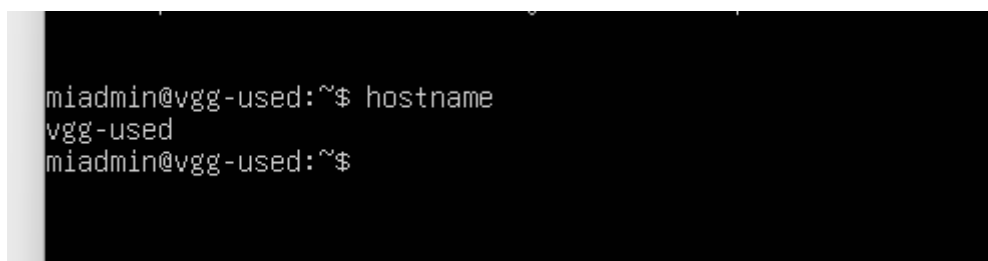


```

GNU nano 7.2 /etc/hosts
127.0.0.1 localhost
127.0.1.1 daw-limpia

# 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
    
```

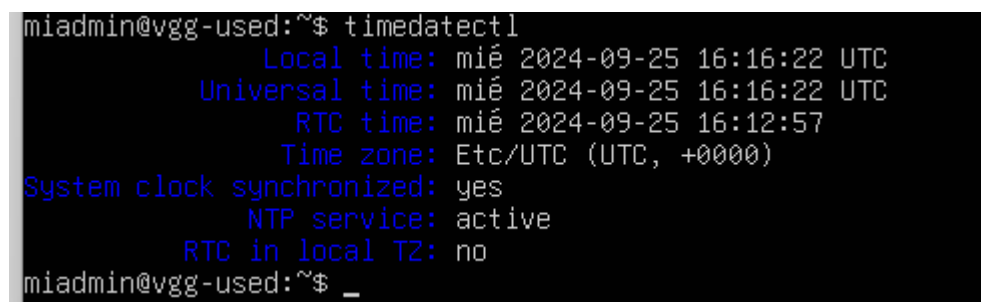
Después de usar el comando, hacemos un reboot y ponemos el comando `hostname` o bien volviendo a ver el fichero mencionado anteriormente.



```

miadmin@vgg-used:~$ hostname
vgg-used
miadmin@vgg-used:~$
    
```

Para ver la hora con el comando `timedatectl`.



```

miadmin@vgg-used:~$ timedatectl
          Local time: mié 2024-09-25 16:16:22 UTC
          Universal time: mié 2024-09-25 16:16:22 UTC
             RTC time: mié 2024-09-25 16:12:57
            Time zone: Etc/UTC (UTC, +0000)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
miadmin@vgg-used:~$ _
    
```

Ya que está en una zona horaria distinta a la nuestra habrá que cambiarla empleando este comando `sudo timedatectl set-timezone Europe/Madrid`.

```
miadmin@vgg-used:~$ sudo timedatectl set-timezone Europe/Madrid
[sudo] password for miadmin:
miadmin@vgg-used:~$ timedatectl
          Local time: mié 2024-09-25 18:19:29 CEST
          Universal time: mié 2024-09-25 16:19:29 UTC
             RTC time: mié 2024-09-25 16:13:20
            Time zone: Europe/Madrid (CEST, +0200)
System clock synchronized: yes
              NTP service: active
            RTC in local TZ: no
miadmin@vgg-used:~$
```

Por último actualizaremos el sistema y utilizaremos los comandos `sudo apt update` y `sudo apt upgrade`.

```
miadmin@vgg-used:~$ sudo apt update
Obj:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Obj:2 http://archive.ubuntu.com/ubuntu noble InRelease
Des:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Obj:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Des:5 http://archive.ubuntu.com/ubuntu noble/main Translation-es [325 kB]
Des:6 http://archive.ubuntu.com/ubuntu noble/restricted Translation-es [816 B]
Des:7 http://archive.ubuntu.com/ubuntu noble/universe Translation-es [1.371 kB]
Des:8 http://archive.ubuntu.com/ubuntu noble/multiverse Translation-es [63,1 kB]
Des:9 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [531 kB]
Des:10 http://archive.ubuntu.com/ubuntu noble-updates/main Translation-en [129 kB]
Des:11 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8.600 B]
Des:12 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [374 kB]
Des:13 http://archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [154 kB]
Descargados 3.082 kB en 1s (2.337 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 34 paquetes. Ejecute «apt list --upgradable» para verlos.
miadmin@vgg-used:~$ _
```

```
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
miadmin@vgg-used:~$ sudo apt upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
The following upgrades have been deferred due to phasing:
  lxd-agent-loader python3-distupgrade ubuntu-release-upgrader-core
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 3 no actualizados.
miadmin@vgg-used:~$ _
```

Cuentas de administración

Creamos un usuario llamado miadmin2 con el comando `sudo adduser miadmin2`. Luego utilizaremos el comando `sudo usermod -aG sudo miadmin2` para meterlo en el grupo de los superusuarios (sudo).

```
miadmin@vgg-used:~$ sudo adduser miadmin2
info: Adding user `miadmin2' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `miadmin2' (1001) ...
info: Adding new user `miadmin2' (1001) with group `miadmin2 (1001)' ...
info: Creating home directory `/home/miadmin2' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for miadmin2
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] Y
info: Adding new user `miadmin2' to supplemental / extra groups `users' ...
info: Adding user `miadmin2' to group `users' ...
miadmin@vgg-used:~$ sudo usermod -aG sudo miadmin2
miadmin@vgg-used:~$ _
```

Apache

Lo primero que tendremos que hacer antes de instalar Apache sería activar el cortafuegos con el comando `sudo ufw enable`, luego ver si está este activo con el comando `sudo ufw status` y luego abrir el puerto 80 con `sudo ufw allow 80` ya que es el puerto que usar Apache (HTTP).


```
miadmin@vgg-used:~$ sudo ufw enable
Firewall is active and enabled on system startup
miadmin@vgg-used:~$ sudo ufw status
Status: active
miadmin@vgg-used:~$ sudo ufw allow 80
Rule added
Rule added (v6)
miadmin@vgg-used:~$ sudo ufw allow 22
Rule added
Rule added (v6)
miadmin@vgg-used:~$ sudo ufw status
Status: active
```

To	Action	From
--	-----	----
80	ALLOW	Anywhere
22	ALLOW	Anywhere
80 (v6)	ALLOW	Anywhere (v6)
22 (v6)	ALLOW	Anywhere (v6)

```
miadmin@vgg-used:~$
```

Luego instalamos apache con el comando `sudo apt install apache2`. Y una vez instalado con `sudo service apache2 status` vemos si está activo.

```
miadmin@vgg-used:~$ sudo service apache2 status
• apache2.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
  Active: active (running) since Wed 2024-09-25 18:42:11 CEST; 1min 13s ago
    Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 9763 (apache2)
    Tasks: 55 (limit: 2276)
  Memory: 5.4M (peak: 5.7M)
     CPU: 55ms
  CGroup: /system.slice/apache2.service
          └─9763 /usr/sbin/apache2 -k start
            └─9974 /usr/sbin/apache2 -k start
              └─9975 /usr/sbin/apache2 -k start
```

Por último tendríamos que comprobar la conectividad desde el cliente en mi caso tendría que poner en el navegador <http://192.168.0.229> y se debería ver la página por defecto de Apache.



Luego habilitamos el uso del fichero `.htaccess` para ello primero habilitamos `mod_rewrite` con el comando: **`sudo a2enmod rewrite`**.

```
miadmin@vgg-used:~$ sudo a2enmod rewrite
[sudo] password for miadmin:
Enabling module rewrite.
To activate the new configuration, you need to run:
    systemctl restart apache2
miadmin@vgg-used:~$ sudo service apache2 restart
miadmin@vgg-used:~$
```

Luego abrimos el archivo **apache2.conf** y remplazamos en la sección siguiente el None del AllowOverride por All en el directorio /var/www así tendríamos ya los indexes configurados.

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride All_
    Require all granted
</Directory>
```

PHP

Para instalar el módulo de PHP hay que poner el siguiente comando: **sudo apt install php**.

```
miadmin@vgg-used:~$ sudo apt install php
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
php ya está en su versión más reciente (2:8.3+93ubuntu2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 7 no actualizados.
miadmin@vgg-used:~$ _
```

Luego ejecutamos este comando: **sudo apt install libapache2-mod-php** para conectar Apache con PHP.

```
miadmin@vgg-used:~$ sudo apt install libapache2-mod-php
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-php
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 7 no actualizados.
Se necesita descargar 4.224 B de archivos.
Se utilizarán 15,4 kB de espacio de disco adicional después de esta operación.
Des:1 http://archive.ubuntu.com/ubuntu noble/main amd64 libapache2-mod-php all 2:8.3+93ubuntu2 [4.224 B]
Descargados 4.224 B en 0s (27,7 kB/s)
Seleccionando el paquete libapache2-mod-php previamente no seleccionado.
(Leyendo la base de datos ... 90619 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libapache2-mod-php_2%3a8.3+93ubuntu2_all.deb ...
Desempaquetando libapache2-mod-php (2:8.3+93ubuntu2) ...
Configurando libapache2-mod-php (2:8.3+93ubuntu2)
```

Luego iremos al directorio `/etc/php/8.3/apache2` y ejecutaremos: **`sudo cp php.ini php.ini.backup`**.

```
miadmin@vgg-used:~$ cd /etc/php/8.3/apache2
miadmin@vgg-used:/etc/php/8.3/apache2$ sudo cp php.ini php.ini.backup
miadmin@vgg-used:/etc/php/8.3/apache2$ _
```

Luego en el **php.ini** hay que revisar que la sección **display_errors = On** y **memory_limit = 256M** sino habría que cambiarlo.

```
; Production value: Off
; https://php.net/display-errors
display_errors = On
```

```
; https://php.net/memory-limit
memory_limit = 256M
```

MySQL

Xdebug

Para instalar **Xdebug** hay que emplear este comando: **`sudo apt install php-xdebug`** y luego ir al directorio `/etc/php/8.3/apache2/conf.d` y revisar que está el archivo **20-xdebug.ini**.

```
miadmin@vgg-used:/etc/php/8.3/apache2$ sudo apt install php-xdebug
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  php8.3-xdebug
Se instalarán los siguientes paquetes NUEVOS:
  php-xdebug php8.3-xdebug
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 7 no actualizados.
Se necesita descargar 632 kB de archivos.
Se utilizarán 1.953 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
Des:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 php8.3-xdebug amd64 3.2
```

```
miadmin@vgg-used:/etc/php/8.3/apache2/conf.d$ ls
10-mysqlnd.ini  20-bz2.ini      20-dom.ini      20-ftp.ini      20-mbstring.ini  20-phar.ini      20-simplexml.ini  20-sysvsem.ini
10-opcache.ini  20-calendar.ini 20-exif.ini     20-gd.ini       20-mcrypt.ini    20-posix.ini     20-soap.ini       20-sysvshm.ini
10-pdo.ini      20-ctype.ini    20-ffi.ini      20-gettext.ini  20-mysqli.ini    20-readline.ini  20-sockets.ini    20-tokenizer.ini
15-xml.ini      20-curl.ini     20-fileinfo.ini 20-iconv.ini    20-pdo_mysql.ini 20-shmop.ini     20-sysvmsg.ini    20-xdebug.ini
miadmin@vgg-used:/etc/php/8.3/apache2/conf.d$ _
```

Luego entramos en el archivo y escribimos lo siguiente:

```
GNU nano 7.2
zend_extension=xdebug.so
xdebug.discover_client_host=1
xdebug.mode=debug
xdebug.client_host=localhost
xdebug.client_port=9003
xdebug.idekey="netbeans-xdebug"
xdebug.show_error_trace=1
xdebug.remote_autostart=on
xdebug.start_with_request=yes
```

Cuentas de desarrollo y hosting virtual

GITHUB – Internet

Cuentas de desarrollador

WXED – Windows X

Instalación y configuración inicial de la máquina

Especificaciones del dispositivo

Nombre del dispositivo	IS32WX02
Procesador	Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz 3.00 GHz
RAM instalada	16,0 GB (15,9 GB usable)
Identificador de dispositivo	8E500A52-01CD-41F1-98F9-DEA49041B031
Id. del producto	00328-20460-00000-AA718
Tipo de sistema	Sistema operativo de 64 bits, procesador basado en x64
Lápiz y entrada táctil	La entrada táctil o manuscrita no está disponible para esta pantalla

Copiar

Cambiar el nombre de este equipo

Especificaciones de Windows

Edición	Windows 10 Education
Versión	22H2
Instalado el	04/09/2020
Compilación del sistema operativo	19045.4894
Experiencia	Windows Feature Experience Pack 1000.19060.1000.0

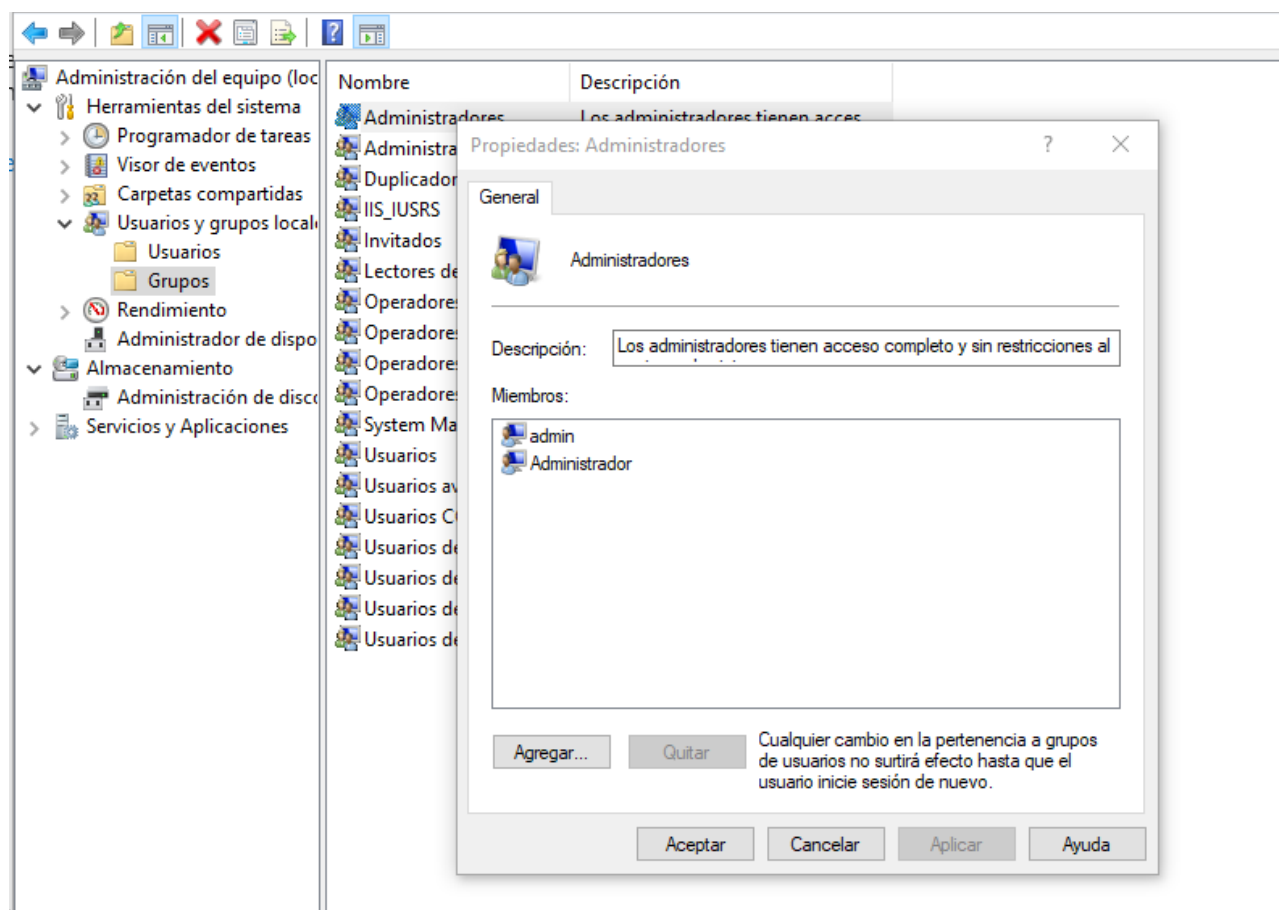
Copiar

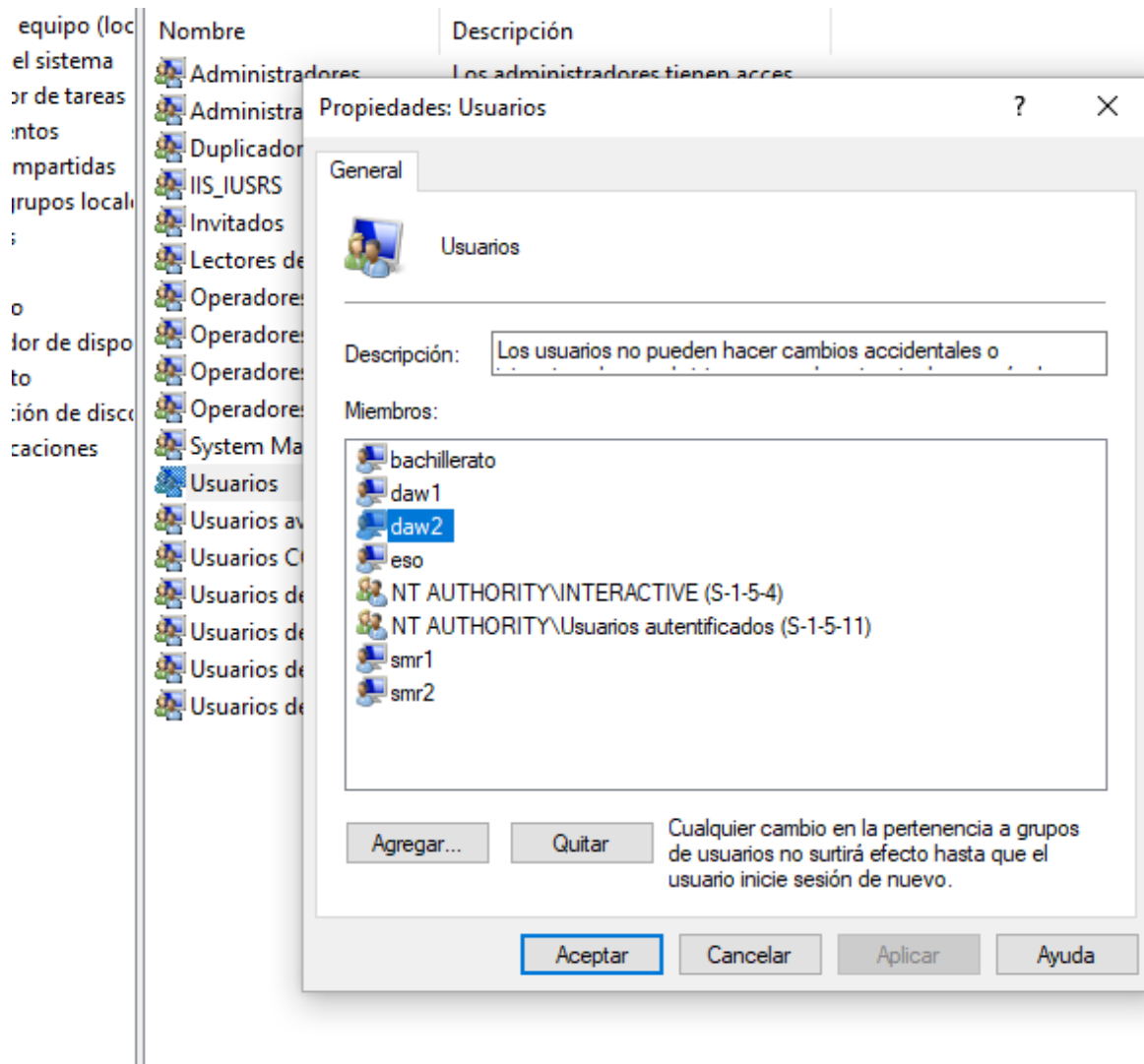
Configuración de IP

Asignación de IP:	Manual
Dirección IPv4:	192.168.3.2
Longitud del prefijo de subred IPv4	24
Puerta de enlace de IPv4:	192.168.3.1
Servidores DNS IPv4:	192.168.20.30 8.8.8.8

Editar

Cuentas administradoras y cuenta de desarrollador





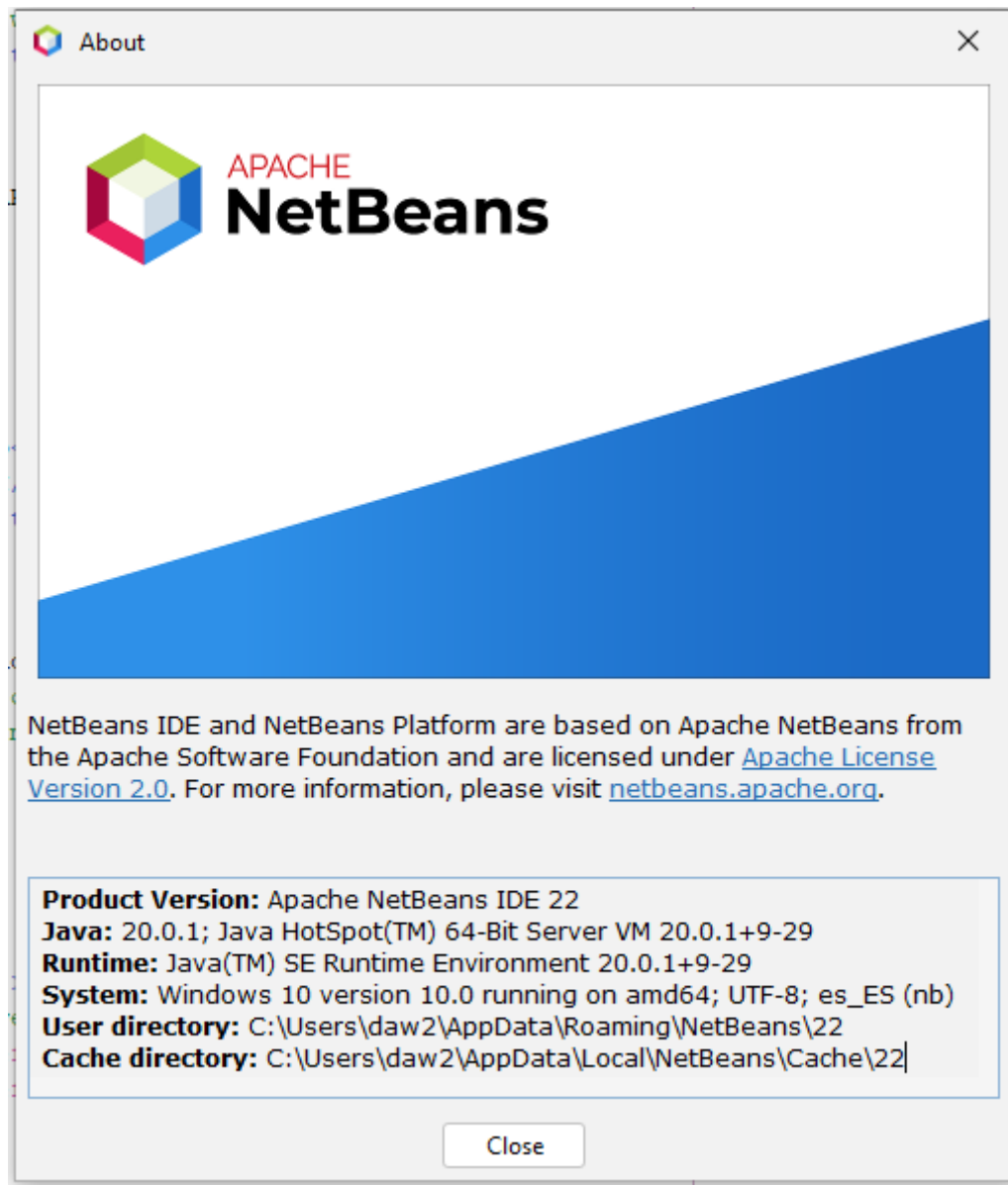
Navegadores

Filezilla

Notepad++

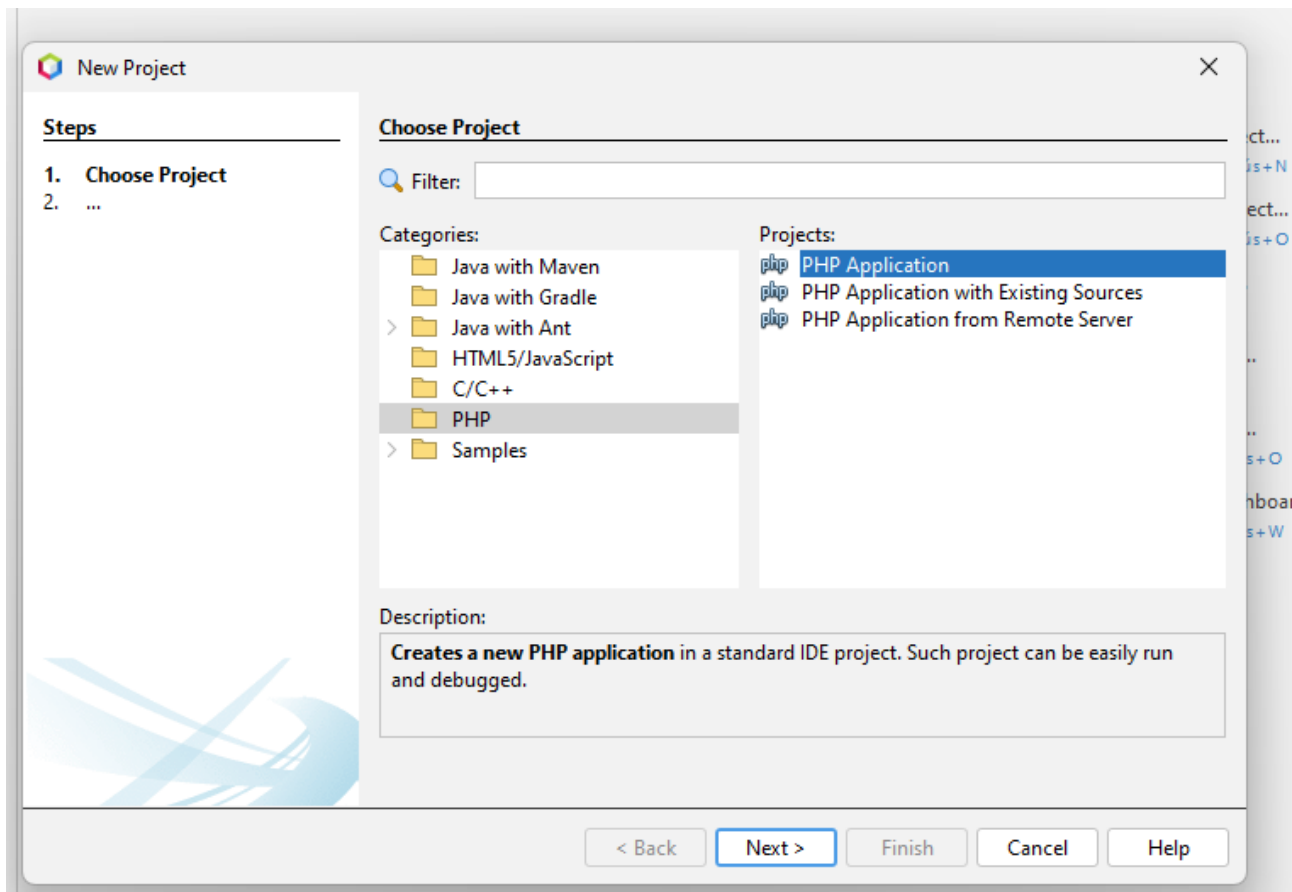
NetBeans

Instalación y configuración inicial (plugins)

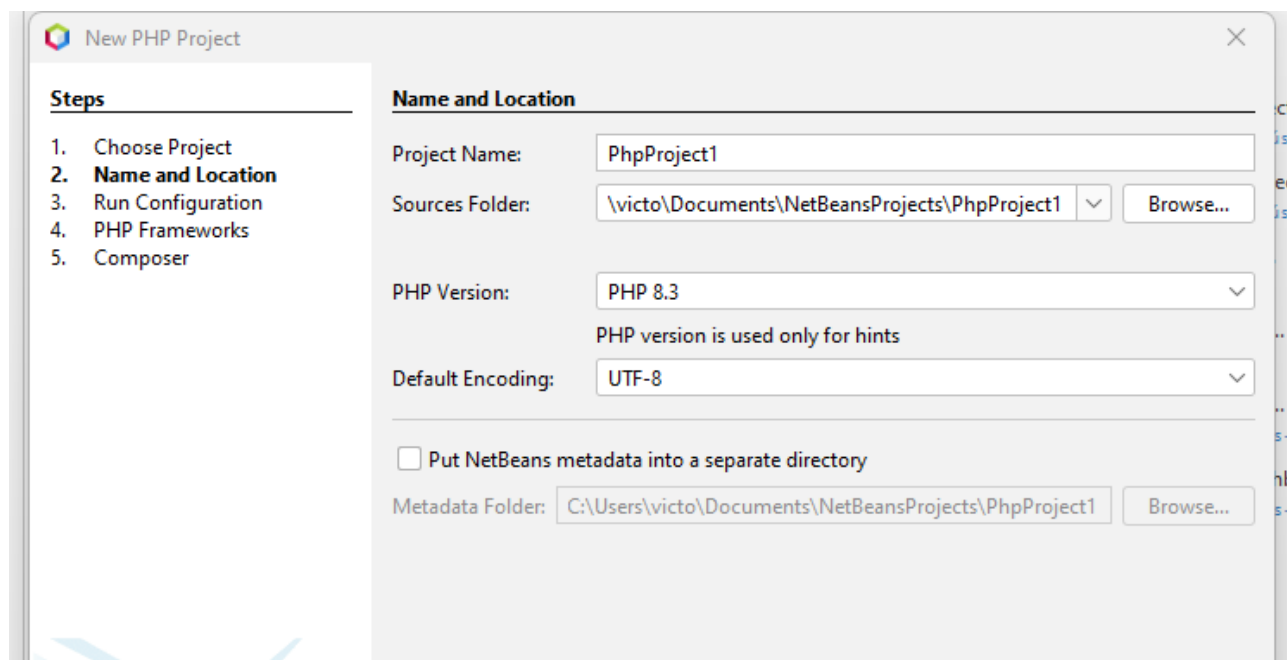


Creación de proyectos, modificación, borrado, prueba

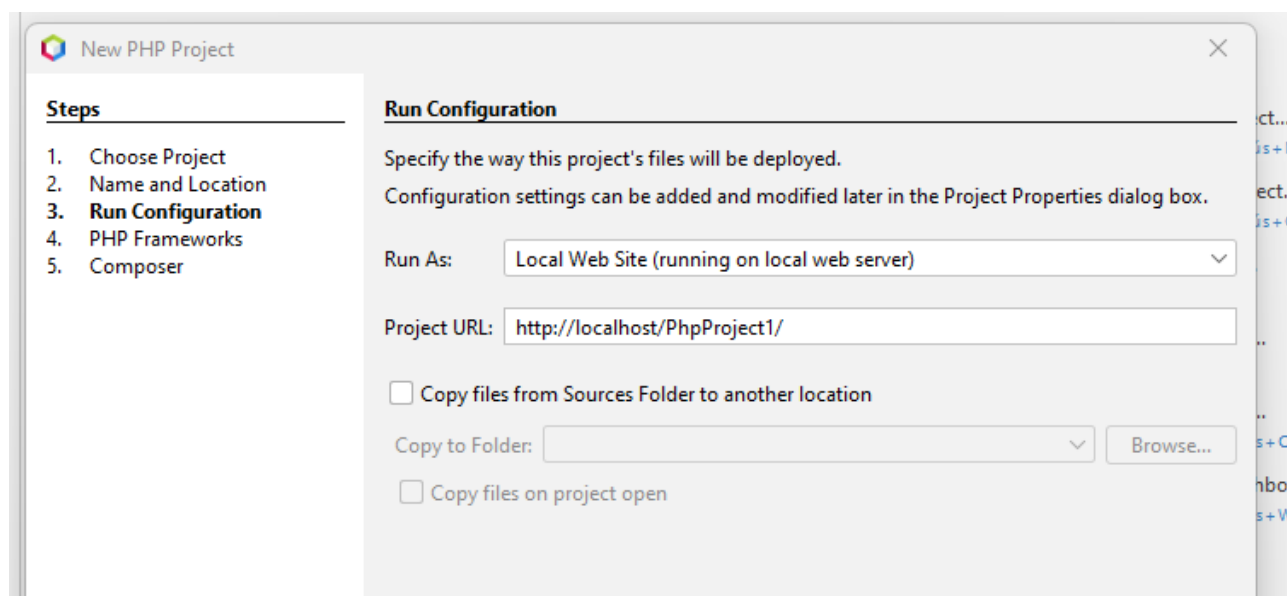
Para crear un proyecto nos vamos a File New Project y seleccionamos el lenguaje de programación que queramos.



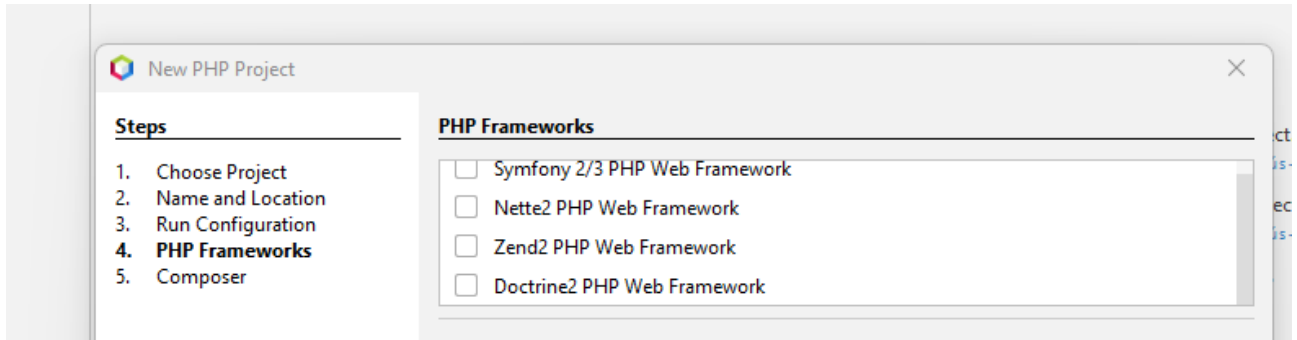
Ahora se elige donde lo quieres guardar el proyecto y la versión de PHP



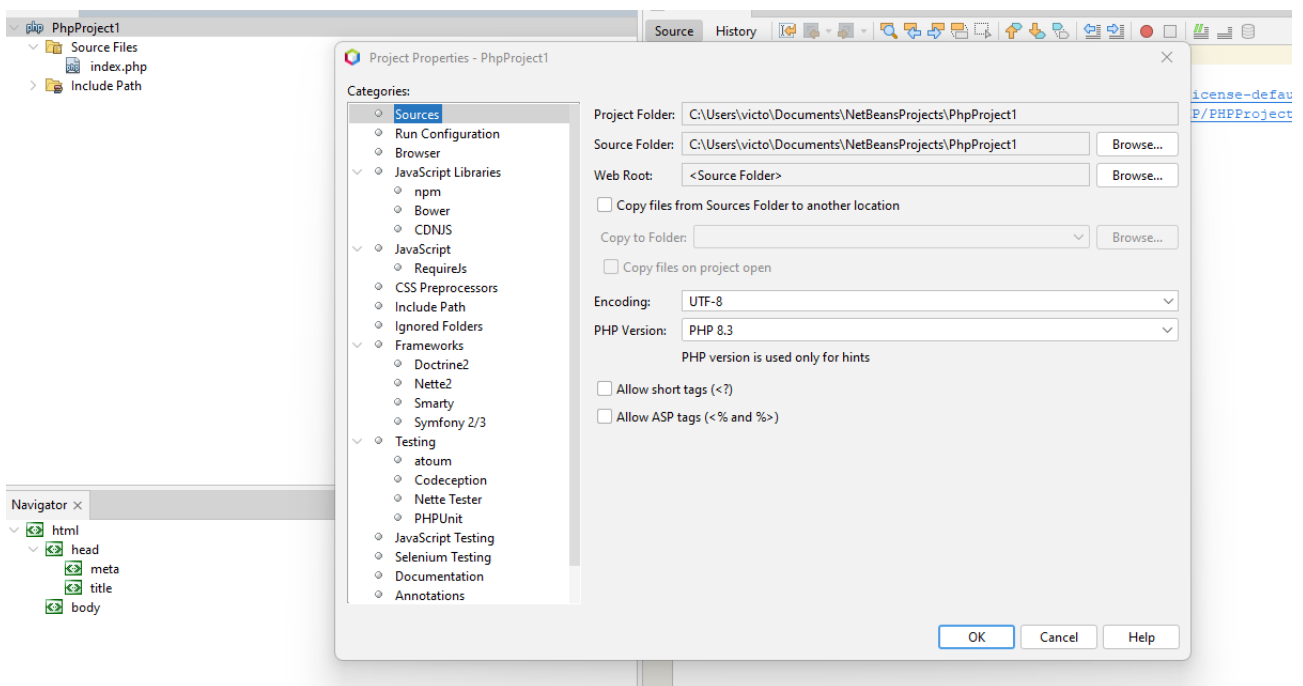
Luego te dice si quieres guardar el proyecto en local o en remoto.



Luego te pide si quieres utilizar un framework para PHP pero como no quiero no lo selecciono y finalizo la creación del proyecto.



Para modificar cualquier cosa del proyecto click derecho en el nombre del proyecto y nos vamos a propiedades.

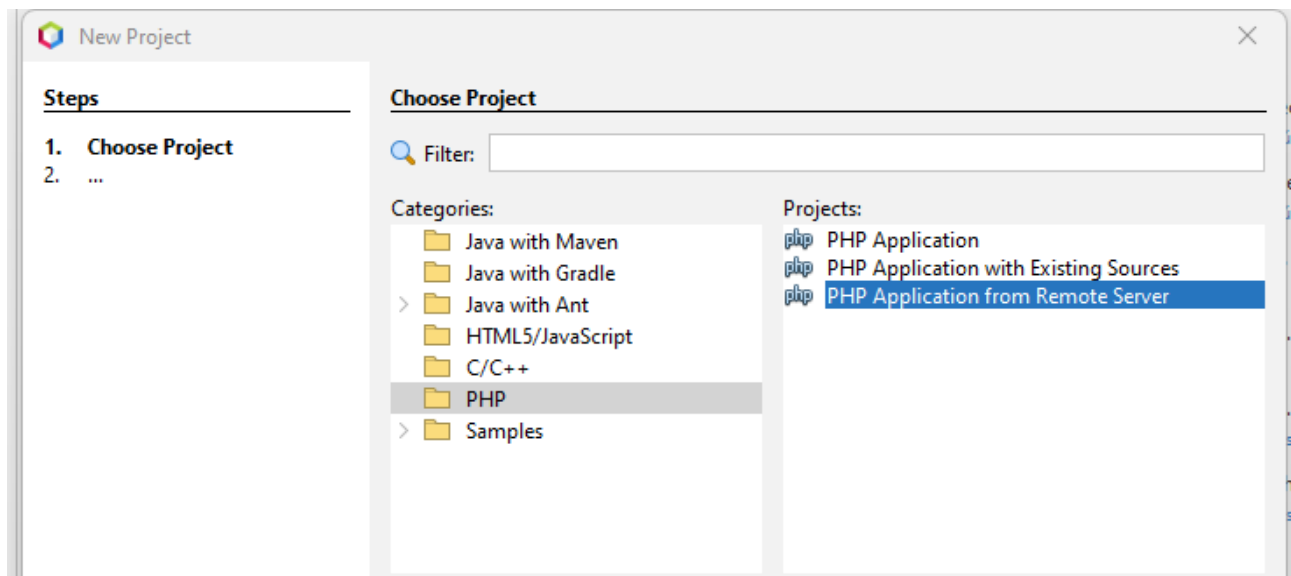


Para eliminar sería tan sencillo como hacer click derecho y darle a delete y se borrará el proyecto de nuestro equipo en local,

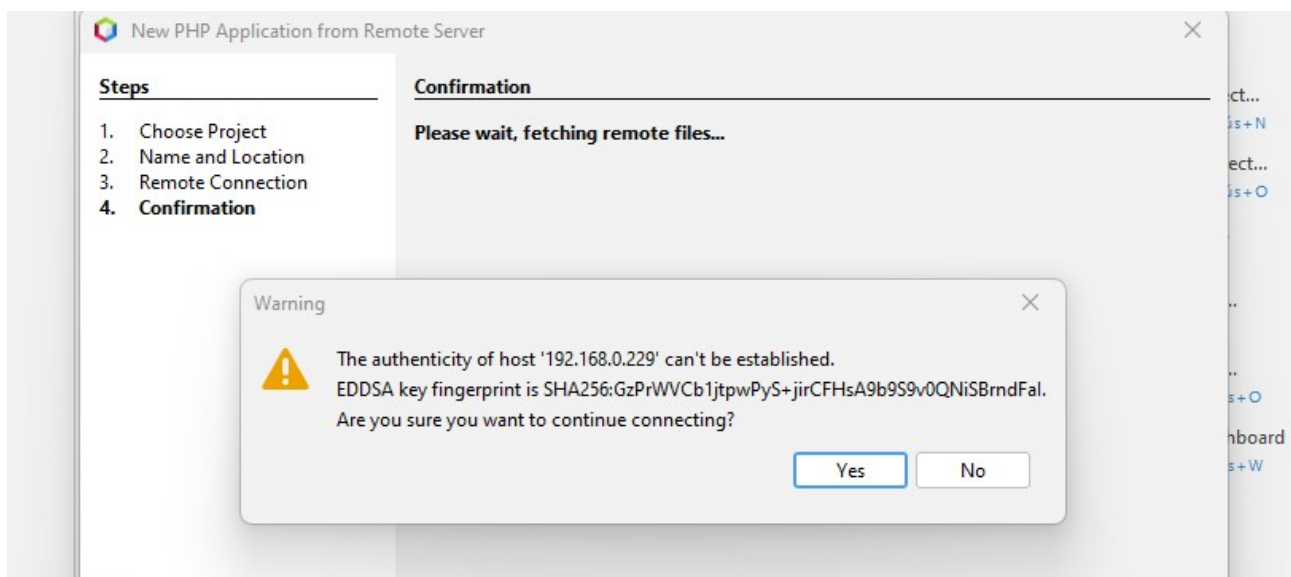
Conexión al servidor remoto SFTP. (Almacenamiento local/almacenamiento remoto)

Esto se puede hacer en el proyecto que hemos creado en properties pero vamos a hacerlo con un proyecto nuevo.

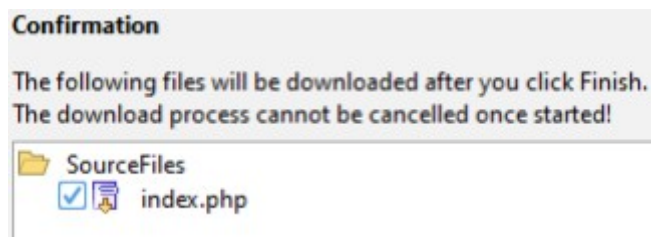
Al hacer el New Project elegiremos la opción de Remote Server.



Después pones la IP del servidor y el directorio para guardar el proyecto y al continuar tendrás que aceptar este mensaje.



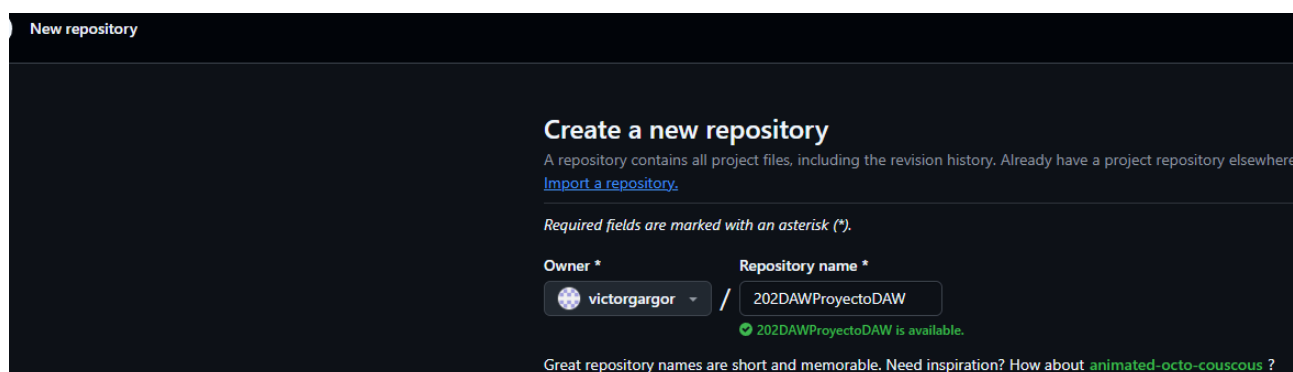
Al aceptar el mensaje te tienen que salir los ficheros de tu servidor y le damos a finish.



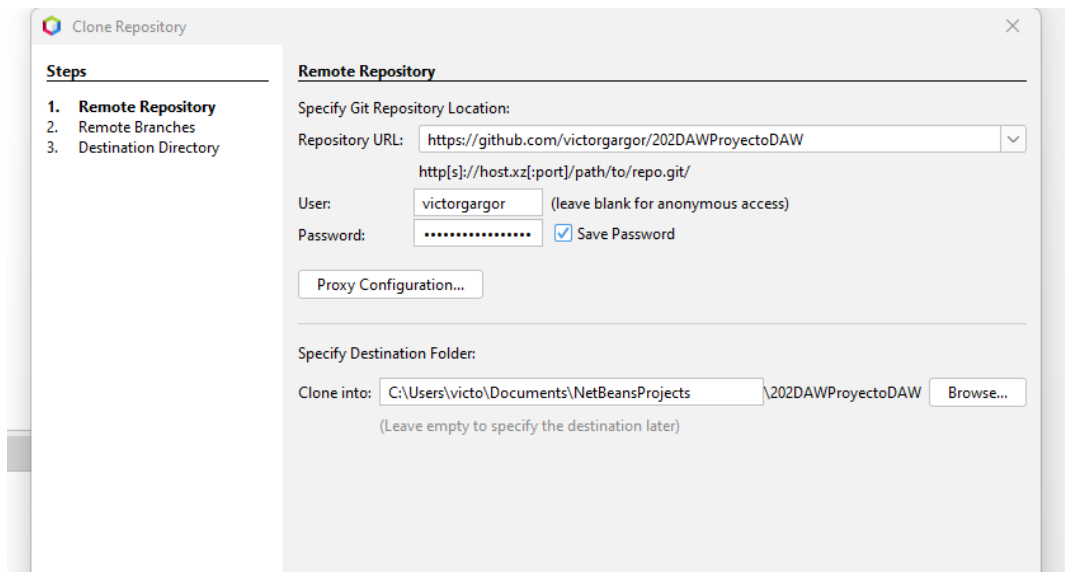
Administración de la base de datos

Conexión al repositorio – versionado

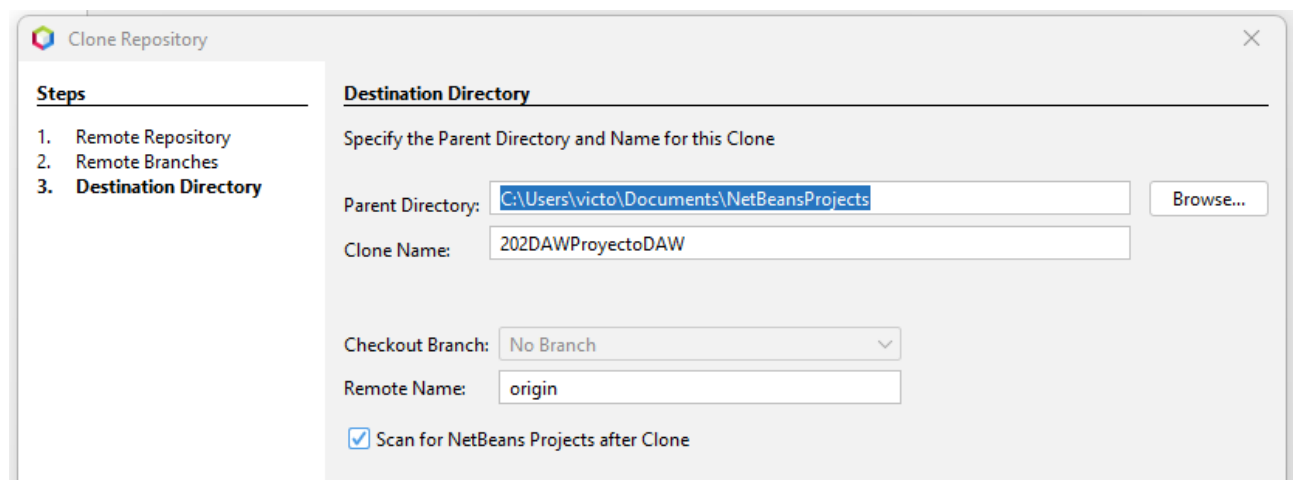
Primero creamos un repositorio nuevo en GitHub empleando el mismo nombre del proyecto en mi caso.



Para conectar al repositorio desde NetBeans le damos click derecho al proyecto luego nos vamos al apartado Git a Remote y le damos a Clone. Ahí indicamos la URL del repositorio que hemos creado el usuario de Git, nuestro token y podemos elegir donde lo queremos guardar en local.

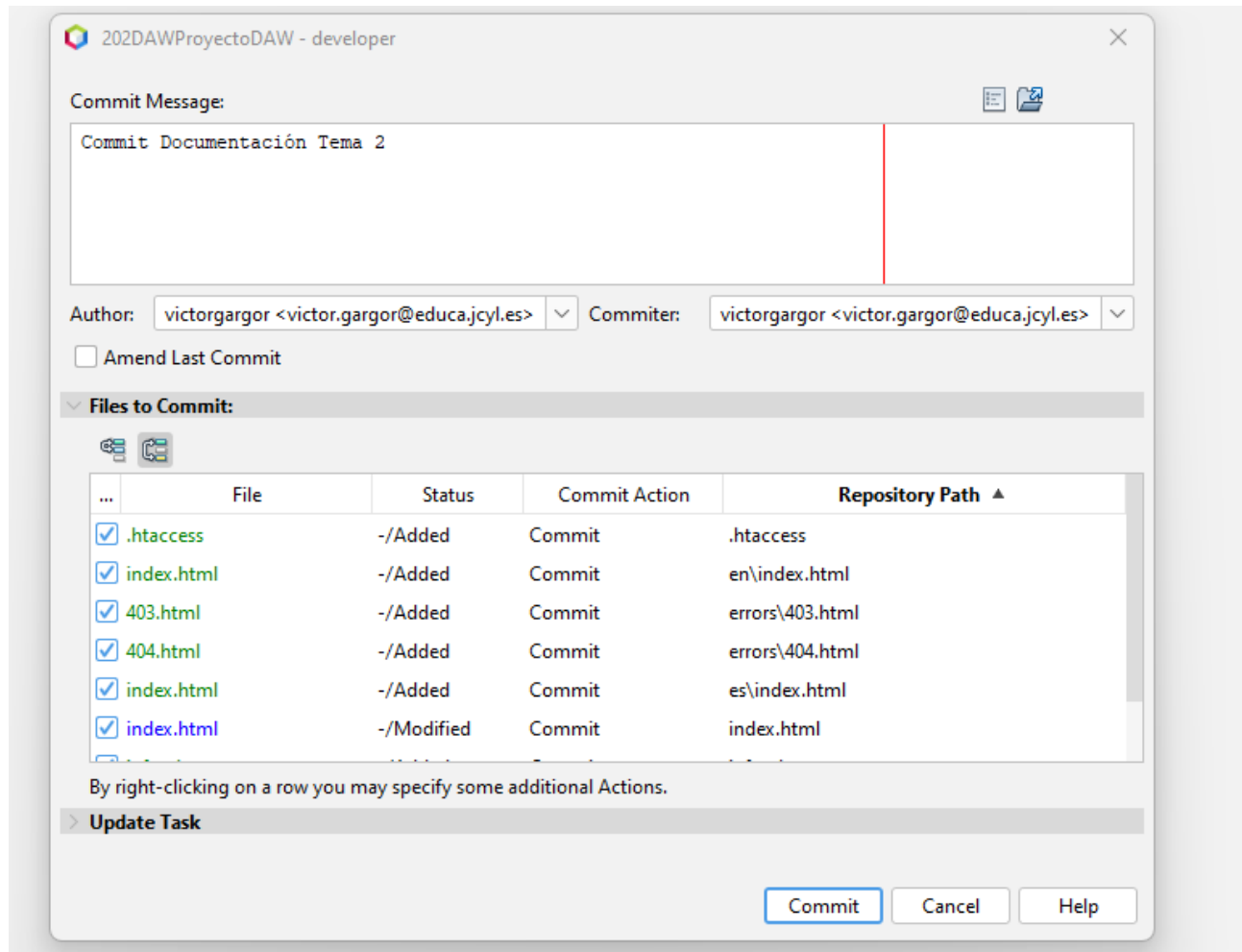


Luego nos pide seleccionar que rama quieres clonar del proyecto y el nombre del proyecto en local y el destino que quieres al igual del nombre de la rama principal.

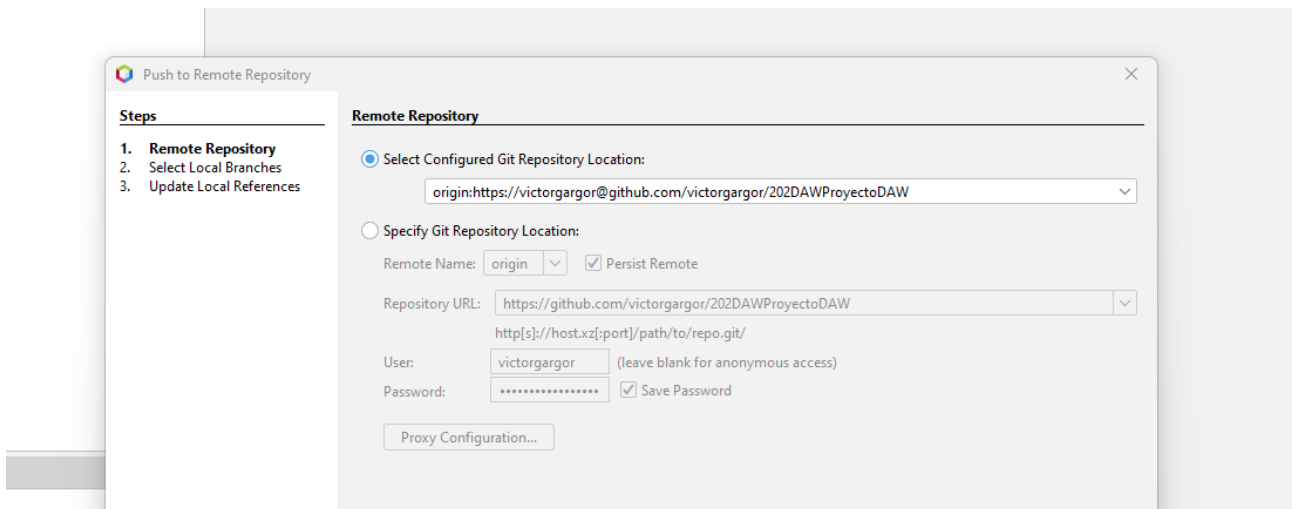


Le damos a finish y ahí estaría conectado el repositorio.

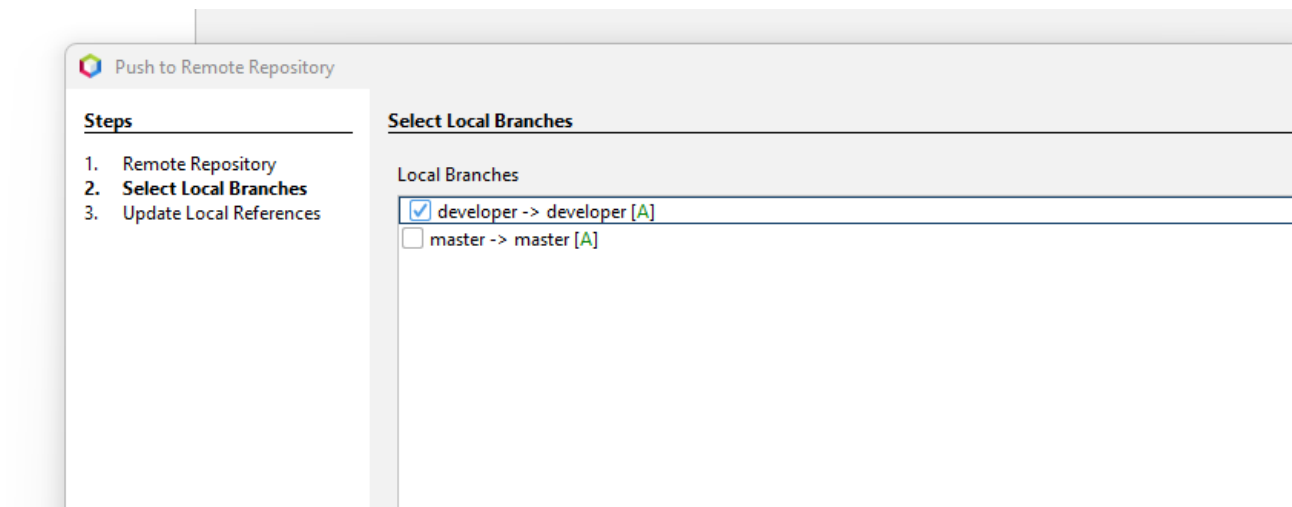
Ahora para versionar el contenido del proyecto lo primero que tendremos que hacer es hacer click derecho en el proyecto ir a Git y a Commit. Ahí ponemos el mensaje que queramos y seleccionamos los archivos correspondientes.



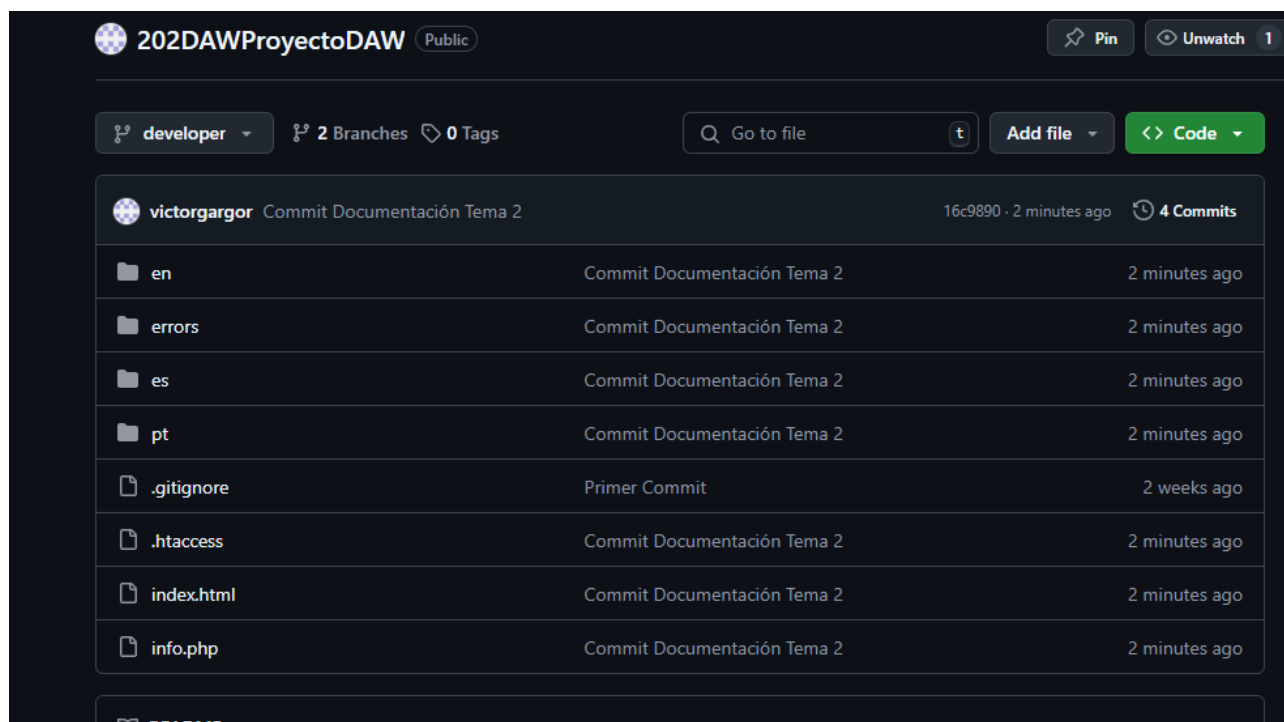
Para subir el contenido al repositorio tendremos que hacer click derecho Git ir a Remote y hacer un Push. Te pedirá la localización del repositorio remoto.



Seleccionamos la rama o ramas que queramos subir al repositorio y le damos a finish.



Revisamos que se haya actualizado el repositorio en la rama seleccionada.



Depuración – Configuración de la ejecución para la depuración

Paso a explotación

PHP Doc

CSS / JS / AJAX / XML /JSON