

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

Sumario

USED – Ubuntu Server.....	3
Configuración inicial.....	3
- Configuración de red.....	3
- Particiones.....	5
- Cambio de nombre del equipo, zona horaria y actualización del sistema.....	6
Cuentas de administración.....	8
Apache.....	9
- Estructura de directorios y archivos de configuración de apache.....	9
- Instalación del servicio.....	10
- Apertura de puertos.....	10
- Conectividad con la máquina cliente.....	11
- Configurar archivo .htaccess.....	12
- Configurar logs.....	13
- Personalizar errores.....	13
- Configuración sitios virtuales.....	14
- Redirigir.....	16
DNS.....	17
- Instalación servidor DNS.....	17
- Declaración zona directa.....	19
- RR zona directa.....	20
- RR zona inversa.....	21
HTTPS.....	22
PHP.....	24
MySQL.....	25
Xdebug.....	28
PhpMyAdmin.....	29
Cuentas de desarrollo y hosting virtual.....	32
- Crear el usuario.....	32
- Cambiar permisos.....	32
- Configurar SSH.....	33
- Configurar sitios virtuales.....	34
- Enjaular usuarios.....	36
GITHUB – Internet.....	37

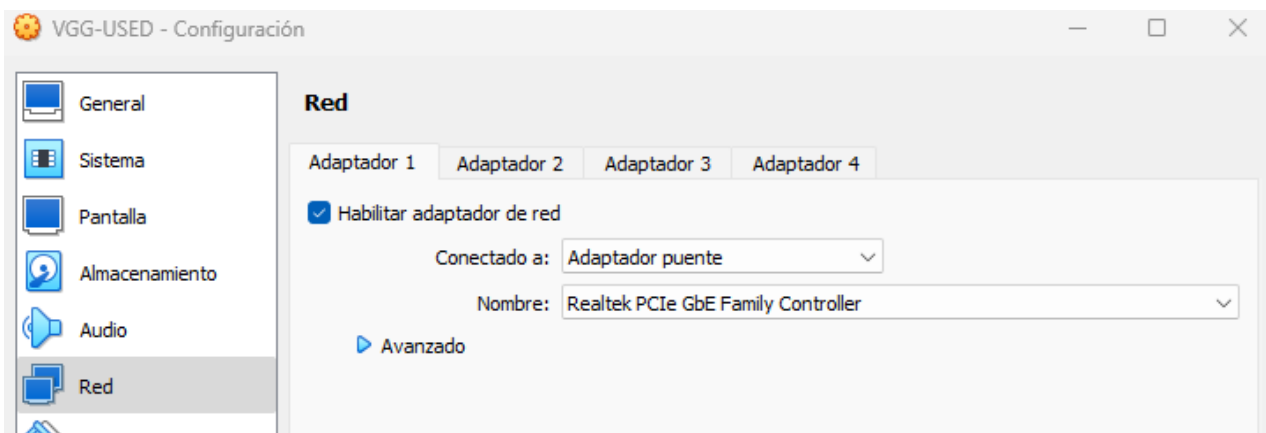
- Crear repositorio.....	37
- Comandos.....	38
WXED – Windows X.....	41
Instalación y configuración inicial de la máquina.....	41
Cuentas administradoras y cuenta de desarrollador.....	43
Navegadores.....	45
Filezilla.....	46
Notepad++.....	50
NetBeans.....	51
Instalación y configuración inicial (plugins).....	51
Creación de proyectos, modificación, borrado, prueba.....	52
Conexión al servidor remoto SFTP. (Almacenamiento local/almacenamiento remoto).....	55
Administración de la base de datos.....	56
Conexión al repositorio – versionado.....	56
Depuración – Configuración de la ejecución para la depuración.....	60
PHP Doc.....	61
CSS / JS / AJAX / XML /JSON.....	62
Paso a explotación – Plesk.....	63
Prestashop.....	67

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.

```
C:\Users\victo>ipconfig/all
```

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.

```
miadmin@vgg-used:~$ cd /etc/netplan
miadmin@vgg-used:/etc/netplan$ ls
enp0s3.yaml
miadmin@vgg-used:/etc/netplan$ sudo nano enp0s3.yaml
```

Aquí saldrán todos los datos que comprobamos con el comando **ipconfig/all** en el cmd en el caso de que fueran incorrectos o faltaran datos modificaremos el archivo y lo guardaremos.

```
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
```

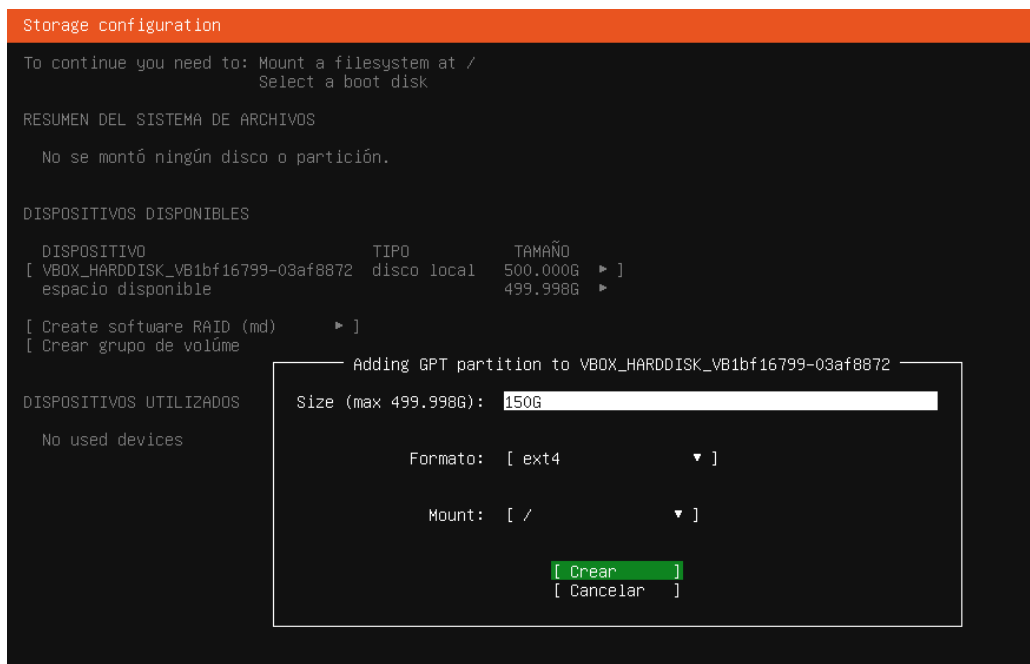
Para comprobar que la red funciona correctamente hacemos **ping** a Google **8.8.8.8**.

```
miadmin@vgg-used:~$ 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=20.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=21.2 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=19.5 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=19.6 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=117 time=21.3 ms
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4079ms
rtt min/avg/max/mdev = 19.533/20.352/21.303/0.752 ms
```

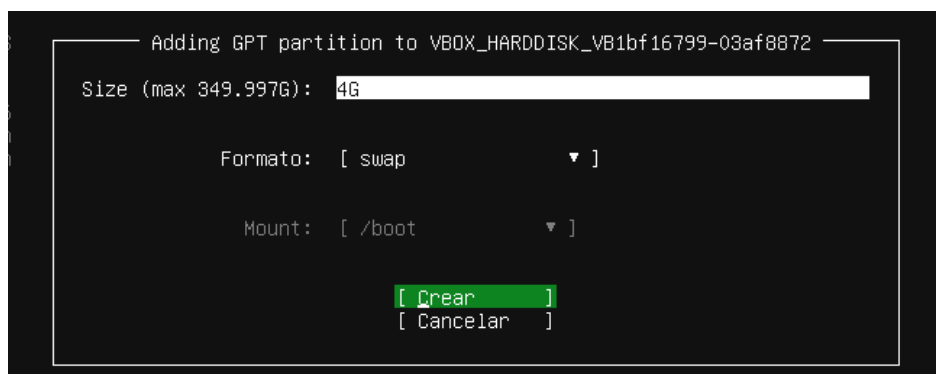
- Particiones

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

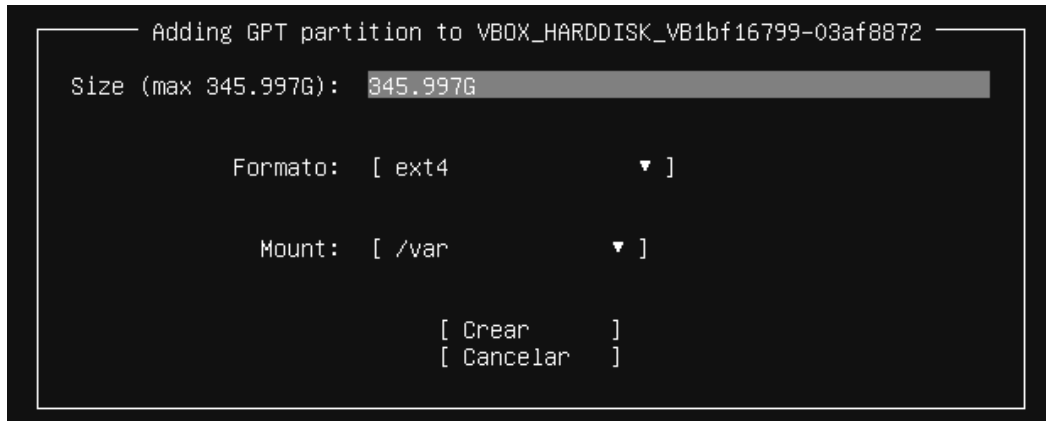
La primera es para el **sistema** y es de **150 GB**.



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
    
```

Para ver la hora con el comando **timedatectl**.

```
miadmin@vgg-used:~$ timedatectl
      Local time: dom 2024-11-10 13:26:27 CET
      Universal time: dom 2024-11-10 12:26:27 UTC
          RTC time: dom 2024-11-10 12:26:05
      Time zone: Europe/Madrid (CET, +0100)
System clock synchronized: yes
          NTP service: active
      RTC in local TZ: no
```

Si necesitáramos cambiar la zona horaria utilizaremos el comando **sudo timedatectl set-timezone [zona horaria]**.

```
miadmin@vqq-used:~$ sudo timedatectl set-timezone Europe/Madrid
```

Por último actualizaremos el sistema utilizando 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
Descargados 126 kB en 1s (214 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 18 paquetes. Ejecute «apt list --upgradable» para verlos.
miadmin@vgg-used:~$
```

```
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
Se instalarán los siguientes paquetes NUEVOS:
```

Cuentas de administración

Vamos a crear un usuario llamado **miadmin2** con el comando **sudo adduser miadmin2**.

```
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' ...
```

Luego utilizaremos el comando **sudo usermod -aG sudo miadmin2** para meterlo en el grupo de los **superusuarios (sudo)**.

```
miadmin@vqq-used:~$ sudo usermod -aG sudo miadmin2
```


Apache

- Estructura de directorios y archivos de configuración de apache

1. Directorio de Configuración Principal:

- Ruta: **/etc/apache2/**
- Descripción: Contiene la mayoría de los archivos de configuración de Apache.

2. Directorio de Configuración de Sitios Disponibles:

- Ruta: **/etc/apache2/sites-available/**
- Descripción: Almacena los archivos de configuración de los sitios web disponibles. No están activos por defecto.

3. Directorio de Configuración de Sitios Habilitados:

- Ruta: **/etc/apache2/sites-enabled/**
- Descripción: Contiene enlaces simbólicos a los archivos de configuración de los sitios activos, apuntando a los archivos en sites-available.

4. Directorio de DocumentRoot Predeterminado:

- Ruta: **/var/www/html/**
- Descripción: Directorio predeterminado para los archivos de los sitios web. Su contenido se muestra al acceder al servidor mediante la IP.

5. Archivo de Configuración Principal:

- Ruta: **/etc/apache2/apache2.conf**
- Descripción: Archivo principal de configuración de Apache, donde se pueden especificar configuraciones globales.

6. Archivo de Configuración de Módulos:

- Ruta: **/etc/apache2/mods-available/**
- Descripción: Contiene archivos de configuración para módulos de Apache, que se pueden habilitar o deshabilitar.

7. Archivo de Configuración de MIME:

- Ruta: **/etc/apache2/mime.types**
- Descripción: Define los tipos MIME que Apache utiliza para servir archivos.

8. Archivo de Configuración de Seguridad:

- Ruta: **/etc/apache2/conf-available/security.conf**
- Descripción: Contiene configuraciones de seguridad recomendadas para Apache.

- Instalación del servicio

Para instalar el servicio de apache se usa el comando **sudo apt install apache2**.

```
miadmin@vgg-used:~$ sudo apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
apache2 ya está en su versión más reciente (2.4.58-1ubuntu8.4).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
```

Luego para comprobar el estado del servicio usamos el comando **sudo service apache2 status**.

```
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 Sun 2024-11-10 11:19:46 CET; 2h 35min ago
     Docs: https://httpd.apache.org/docs/2.4/
```

- Apertura de puertos

Primero tenemos que activar el cortafuegos con el comando **sudo ufw enable**.

```
miadmin@vgg-used:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

Luego abrimos el puerto que usa apache por defecto que es el **80** con el comando **sudo ufw allow 80**.

```
miadmin@vgg-used:~$ sudo ufw allow 80
Skipping adding existing rule
Skipping adding existing rule (v6)
```

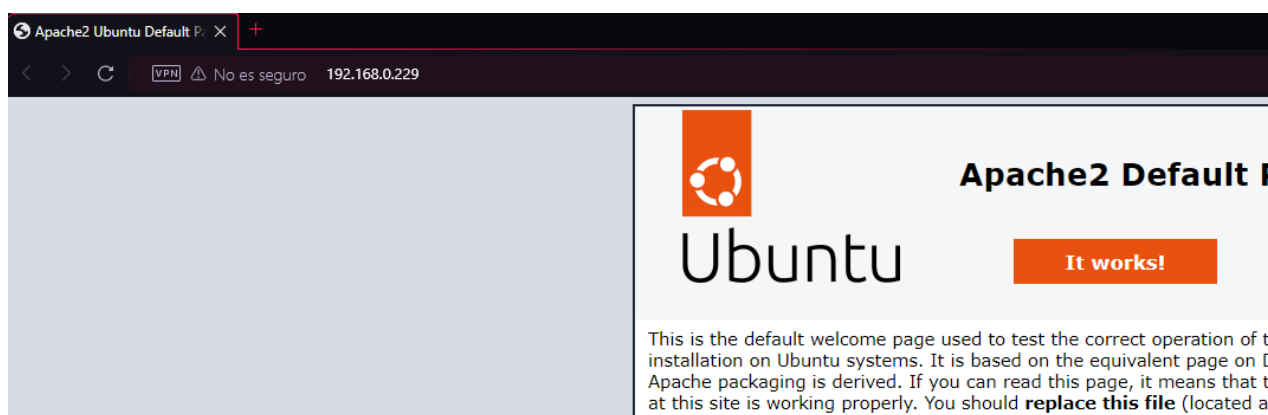
Luego comprobamos que está abierto el puerto **80** con el comando **sudo ufw status**.

```
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)

- Conectividad con la máquina cliente

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.



- Configurar archivo .htaccess

Para habilitar el uso del fichero .htaccess hay que habilitar el modulo **rewrite** primero con el comando **sudo a2enmod rewrite**.

```
miadmin@vgg-used:~$ sudo a2enmod rewrite
Module rewrite already enabled
```

Luego para activarlo hay que **reiniciar** apache con el comando **sudo service apache2 restart**.

```
miadmin@vgg-used:~$ sudo service apache2 restart
```

Luego abrimos el archivo **apache2.conf** con el comando **sudo nano /etc/apache2/apache2.conf**.

```
miadmin@vga-used:~$ sudo nano /etc/apache2/apache2.conf
```

Buscamos el directorio **/var/www** con **Ctrl+W** para encontrarlo antes y remplazamos el **None** del **AllowOverride** por **All** y ya tendŕamos los **indexes** configurados.

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

Después creamos un archivo **.htaccess** en el directorio raíz del servidor con el comando **sudo nano /var/www/html/.htaccess**.

```
miadmin@vgg-used:~$ sudo nano /var/www/html/.htaccess
```

Por último añadimos la línea siguiente al archivo para configurar la página de carga inicial del servidor.

```
DirectoryIndex index.php index.html
```

- Configurar logs

Editamos este archivo **sudo nano /etc/apache2/sites-enabled/000-default.conf** y añadimos estas líneas

```
miadmin@vgg-used:~$ sudo nano /etc/apache2/sites-enabled/000-default.conf
[sudo] password for miadmin:
miadmin@vgg-used:~$ miadmin@vgg-used:~$
```

Con estas líneas estamos indicando el nuevo directorio donde queremos guardar los mensajes de error y en las siguiente indicamos donde queremos guardar el archivo que almacena los registros de acceso

```
ErrorLog ${APACHE_LOG_DIR}/error.log
ErrorLog /var/www/html/logs/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
CustomLog /var/www/html/logs/access.log combined
```

- Personalizar errores

Para personalizar los errores tenemos que tener el archivo **.htaccess** y utilizaremos la directiva **ErrorDocument** añadiendo la ruta de los errores

```
ErrorDocument 403 /error/403.html
ErrorDocument 404 /error/404.html
```

- Configuración sitios virtuales

Para configurar un **sitio virtual** hay que editar el archivo siguiente:

```
miadmin@vgg-used:~$ sudo nano /etc/apache2/sites-available/000-default.conf
```

Pero como no vamos a editar ese archivo hacemos una copia de este por si acaso y editamos la copia

```
sudo cp 000-default.conf sitio-virtual.conf
```

Editamos el archivo que hemos copiado con **sudo nano sitio-virtual.conf** y ańadimos lo siguiente al archivo

```
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
Alias /archivos /home/miadmin/doc
<Directory /home/miadmin/doc>
    Options +FollowSymLinks +Indexes
    AllowOverride None
    Require all granted
</Directory>
```

1. ServerAdmin

Propósito: Especifica la dirección de correo electrónico del administrador del servidor.

Usos:

Apache utiliza esta dirección para enviar notificaciones o reportar problemas.

Aparece en las páginas de error predeterminadas como contacto del administrador.

2. DocumentRoot

Propósito: Define el directorio raíz del sitio web.

Usos:

Los archivos servidos al acceder al servidor estarán en este directorio.

Coloca archivos como index.html aquí para que se muestren automáticamente.

3. Alias

Propósito: Crea un alias llamado /archivos que apunta al directorio físico /home/miadmin/doc.

Usos:

Permite acceder al contenido de /home/miadmin/doc usando la URL <http://tusitio/archivos>.

Organiza y sirve archivos externos al directorio raíz del sitio.

4. <Directory>

Propósito: Define configuraciones específicas para el directorio /home/miadmin/doc.

Options:

+FollowSymLinks: Permite seguir enlaces simbólicos en este directorio.

+Indexes: Habilita listados de archivos en el navegador si no hay un archivo índice (como index.html).

AllowOverride:

Especifica que los archivos .htaccess no pueden modificar configuraciones del servidor en este directorio.

Require:

Permite que todos los usuarios tengan acceso al contenido del directorio sin restricciones.

Ahora nos colocamos en el **home** y hacemos un **sudo mkdir miadmin/doc** y dentro creamos un archivo cualquiera

```
miadmin@vgg-used:~$ cd /home
miadmin@vgg-used:/home$ sudo mkdir miadmin/doc

miadmin@vgg-used:~/doc$ ls
archivo.txt
```

Cambiamos los **permisos** del directorio

```
miadmin@vgg-used:~$ sudo chmod 755 /home/miadmin
```

Activamos el sitio que hemos creado y **reiniciamos Apache**

```
miadmin@vgg-used:~$ sudo a2ensite sitio-virtual.conf
Enabling site sitio-virtual.
To activate the new configuration, you need to run:
    systemctl reload apache2
miadmin@vgg-used:~$ systemctl reload apache2
```

- Redirigir

Primero tenemos que habilitar el **módulo rewrite** y **reiniciamos el servicio**

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

Ahora ańadimos en el **.htaccess** esta línea por ejemplo que **redirige permanentemente** una **URL** a otra

```
Redirect 301 /google https://www.google.es
```


DNS

- Instalación servidor DNS

Instalamos el paquete

```
miadmin@vgg-used:~$ sudo apt install bind9
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
```

Comprobamos el **estado**

```
miadmin@vgg-used:~$ sudo service bind9 status
● named.service - BIND Domain Name Server
   Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: enabled)
   Active: active (running) since Mon 2024-12-09 16:54:12 UTC; 1min 34s ago
     Docs: man:named(8)
  Main PID: 4721 (named)
    Status: "running"
     Tasks: 8 (limit: 2276)
  Memory: 7.1M (peak: 7.5M)
     CPU: 52ms
    CGroup: /system.slice/named.service
            └─4721 /usr/sbin/named -f -u bind
```

Abrimos el **puerto** necesario que en este caso es el 53

```
miadmin@vgg-used:~$ sudo ufw allow 53
Skipping adding existing rule
Skipping adding existing rule (v6)
miadmin@vgg-used:~$ sudo ufw status
Status: active
```

To	Action	From
80	ALLOW	Anywhere
Apache	ALLOW	Anywhere
3306	ALLOW	Anywhere
22	ALLOW	Anywhere
53	ALLOW	Anywhere
80 (v6)	ALLOW	Anywhere (v6)
Apache (v6)	ALLOW	Anywhere (v6)
3306 (v6)	ALLOW	Anywhere (v6)
22 (v6)	ALLOW	Anywhere (v6)
53 (v6)	ALLOW	Anywhere (v6)

Ahora editamos el archivo con extensión **.yaml** que contiene la configuración de red

```
miadmin@vgg-used:~$ cd /etc/netplan
miadmin@vgg-used:/etc/netplan$ ls
50-cloud-init.yaml
miadmin@vgg-used:/etc/netplan$ sudo nano 50-cloud-init.yaml
```

Cambiamos el **DNS** anterior por nuestra **IP** y añadimos en **search: [tunombre.local]**

```
# /etc/cloud/cloud.cfg.d/99-disable-network-con
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      addresses:
        - 192.168.0.229/24
      nameservers:
        addresses:
          - 192.168.0.229
        search: [victor.local]
      routes:
        - to: default
          via: 192.168.0.1
  version: 2
```

Habilitamos la nueva configuración

```
miadmin@vgg-used:/etc/netplan$ sudo netplan apply
miadmin@vgg-used:/etc/netplan$
```

Hacemos **ping** a google por ejemplo para probar a ver si funciona

```
miadmin@vgg-used:/etc/netplan$ ping www.google.es
PING www.google.es (142.250.184.163) 56(84) bytes of data.
64 bytes from mad07s23-in-f3.1e100.net (142.250.184.163): icmp_seq=1 ttl=117 time=18.5 ms
64 bytes from mad07s23-in-f3.1e100.net (142.250.184.163): icmp_seq=2 ttl=117 time=19.0 ms
64 bytes from mad07s23-in-f3.1e100.net (142.250.184.163): icmp_seq=3 ttl=117 time=17.5 ms
```

- Declaración zona directa

Tendremos que modificar el archivo siguiente

```
miadmin@vqq-used:~$ sudo nano /etc/bind/named.conf.local
```

Añadimos estas líneas

```
GNU nano 7.2 /etc/bind/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 "victor.local" {
    type master;
    file "/etc/bind/db.victor.local;
};
```

Para chequear los ficheros de configuración

```
miadmin@vgg-used:~$ sudo named-checkconf
```

- RR zona directa

Hago primero una copia de seguridad del archivo

```
miadmin@vgg-used:~$ sudo cp /etc/bind/db.local /etc/bind/db.victor.local
```

Editamos el archivo y tendría que verse así para ello tendremos que añadir ciertas líneas y añadir otras

```
GNU nano 7.2 /etc/bind/db.victor.local
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      vgg-used.victor.local. marodriguez.educa.jcyl.es. (
                        1          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        3600 ) ; Negative Cache TTL
;
@         IN      NS       vgg-used.victor.local.
@         IN      A        192.168.0.229
vgg-used  IN      A        192.168.0.229
daw202    IN      CNAME    vgg-used.victor.local.
```

Comprobamos la sintaxis

```
miadmin@vgg-used:~$ sudo named-checkzone victor.local /etc/bind/db.victor.local
zone victor.local/IN: loaded serial 1
OK
miadmin@vgg-used:~$
```

Y reiniciamos el servicio

```
miadmin@vgg-used:~$ sudo service bind9 restart
```

- RR zona inversa

Hago primero una copia de seguridad del archivo

```
miadmin@vgg-used:~$ sudo cp /etc/bind/db.victor.local /etc/bind/db.3.168.192.in-addr.arpa
```

Editamos el archivo para que se vea así

```
GNU nano 7.2 /etc/bind/db.3.168.192.in-addr.arpa
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      vgg-used.victor.local. marodriguez.educa.jcyl.es. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                3600      ) ; Negative Cache TTL
;
@         IN      NS       vgg-used.victor.local.
214       IN      PTR      vgg-used.victor.local.
214       IN      PTR      victor.local.
```

Comprobamos la sintaxis y volvemos a reiniciar el servicio

```
miadmin@vgg-used:~$ sudo named-checkzone 229.168.192.in-addr.arpa /etc/bind/db.3.168.192.in-addr.arpa
zone 229.168.192.in-addr.arpa/IN: loaded serial 1
OK
miadmin@vgg-used:~$ sudo service bind9 restart
```

Por último para verificar la zona inversa se utiliza el comando

```
miadmin@vgg-used:~$ nslookup victor.local
```

HTTPS

Primero creamos la **clave privada**

```
miadmin@vgg-used:~$ openssl genrsa 2048 > clavePrivada.key
```

Y creamos también el **certificado** y rellenamos los **datos** que nos pide los que he dejado en blanco son **opcionales**

```
miadmin@vgg-used:~$ openssl req -new -key clavePrivada.key > certificado.csr
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]:IES Los Sauces
Organizational Unit Name (eg, section) []:Departamento de Informática
Common Name (e.g. server FQDN or YOUR name) []:daw202.victor.local
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
miadmin@vgg-used:~$
```

Luego **autofirmamos** nuestro certificado

```
miadmin@vgg-used:~$ openssl x509 -req -days 365 -in certificado.csr -singkey clavePrivada.key > certificado.crt
```

Tenemos que habilitar el módulo **SSL**, nos pedirá reiniciar **Apache**

```
miadmin@vgg-used:~$ sudo a2enmod ssl
[sudo] password for miadmin:
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
miadmin@vgg-used:~$ systemctl restart apache2
```

Hacemos una **copia** de seguridad del archivo

```
miadmin@vgg-used:~$ sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/default-ssl.conf.backup
miadmin@vgg-used:~$
```

Editamos el archivo de esta manera

```
GNU nano 7.2 default-ssl.conf
<VirtualHost *:443>
    ServerAdmin webmaster@localhost
    ServerName daw202.victor.local
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., tracel, 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}/errordaw202ssl.log
    CustomLog ${APACHE_LOG_DIR}/accessdaw202ssl.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
    SSLCertificateFile /etc/ssl/certs/certificado.crt
    SSLCertificateKeyFile /etc/ssl/private/clavePrivada.key
```

PHP

Para instalar el módulo de PHP hay que utilizar el 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 2 no actualizados.
```

Luego ejecutamos el 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
libapache2-mod-php ya está en su versión más reciente (2:8.3+93ubuntu2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
```

Luego iremos al directorio **`/etc/php/8.3/apache2`** y ejecutaremos el comando **`sudo cp php.ini php.ini.backup`** para tener una copia por si tuviéramos errores al cambiar el archivo `php.ini`.

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

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

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

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


MySQL

Para instalar **MySQL** utilizaremos el comando **sudo apt install mysql-server**.

```
miadmin@vgg-used:~$ sudo apt install mysql-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
```

Luego empleamos el comando **sudo cp /etc/mysql/mysql.conf.d/mysqld.cnf mysqld.cnf.backup** para hacer una copia de seguridad del archivo que posteriormente vamos a editar.

```
miadmin@vgg-used:~$ sudo cp /etc/mysql/mysql.conf.d/mysqld.cnf mysqld.cnf.backup
```

Utilizaremos el comando **sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf** y comentaremos las dos líneas que se muestran a continuación con #.

```
bind-address          = 127.0.0.1
mysqlx-bind-address   = 127.0.0.1
```

Reiniciamos el servicio con el comando **sudo service mysql restart**.

```
miadmin@vgg-used:~$ sudo service mysql restart
miadmin@vgg-used:~$
```

Abrimos el puerto **3306** que es el que usa **MySQL** por defecto con **sudo ufw allow 3306**.

```
miadmin@vgg-used:~$ sudo ufw allow 3306
Rule added
Rule added (v6)
```

Comprobamos el que lo hemos abierto con **sudo ufw status**.

```
miadmin@vgg-used:~$ sudo ufw status
Status: active
```

To	Action	From
--	-----	----
80	ALLOW	Anywhere
22	ALLOW	Anywhere
3306	ALLOW	Anywhere
80 (v6)	ALLOW	Anywhere (v6)
22 (v6)	ALLOW	Anywhere (v6)
3306 (v6)	ALLOW	Anywhere (v6)

Ahora tenemos que hacer la instalación **segura** de **MySQL** con el comando **sudo mysql_secure_installation**. Nos pedirá si queremos validar la contraseña y le diremos que si y en el nivel 0 y aceptamos todo lo demás que nos pregunta.

```
miadmin@vgg-used:~$ sudo mysql_secure_installation
[sudo] password for miadmin:

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?
```

Descargamos el módulo de la librería de apache de **PHP** para **MySQL** con el comando **sudo apt install libapache2-mod-php8.3 php8.3-mysql**.

```
miadmin@vgg-used:~$ sudo apt install libapache2-mod-php8.3 php8.3-mysql
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
```

Ahora tendremos que entrar a **MySQL** con el comando **sudo mysql** y poner este comando **SET GLOBAL validate_password.length = 4;** para que no nos de error el paso siguiente de creación del usuario.

```
miadmin@vgg-used:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.39-0ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2024, 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> SET GLOBAL validate_password.length = 4;
Query OK, 0 rows affected (0,00 sec)
```

Ahora por último crearemos el **usuario** y volveremos a cambiar la variable que modificamos antes a su valor por defecto en el caso de **LOW** sería **8**.

```
mysql> CREATE USER 'adminsqli'@'%' IDENTIFIED BY 'paso';
Query OK, 0 rows affected (0,02 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'adminsqli'@'%' ;
Query OK, 0 rows affected (0,02 sec)

mysql> SET GLOBAL validate_password.length = 8;
Query OK, 0 rows affected (0,00 sec)

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:~$ 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 NUEVOS:
  php-xdebug
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 25 no actualizados.
Se necesita descargar 3.444 B de archivos.
Se utilizarán 16,4 kB de espacio de disco adicional después de esta operación.
Des:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 php-xdebug amd64 3.2.0+3.1.6+2.9
44 B]
Descargados 3.444 B en 0s (8.256 B/s)
Seleccionando el paquete php-xdebug previamente no seleccionado.
```

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

Luego entramos en el archivo y escribimos lo siguiente:

```
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
```

PhpMyAdmin

Primero miramos los módulos instalados

```
miadmin@vgg-used:~$ dpkg --get-selections | grep php
libapache2-mod-php8.3      install
php-bz2                    install
php-common                 install
php-composer-ca-bundle    install
php-curl                   install
php-fig-http-message-util install
php-gd                     install
php-getallheaders          install
php-google-recaptcha       install
php-twig-til8n-extension  install
php-webmozart-assert       install
php-xml                     install
php-zip                     install
php8.3-bz2                  install
php8.3-cli                  install
php8.3-common               install
php8.3-curl                 install
php8.3-gd                   install
php8.3-mbstring             install
php8.3-mcrypt               install
php8.3-mysql                install
php8.3-opcache              install
php8.3-readline             install
php8.3-xml                  install
php8.3-zip                  install
phpmyadmin                  install
miadmin@vgg-used:~$ |
```

Yo como tengo todos y es una lista muy larga pues se instalan los necesarios

Antes de instalar tenemos que entrar a mysql con **mysql -u adminsql -p** y desinstalar el componente de validación de contraseña para que no nos de error la instalación

```
miadmin@vgg-used:~$ mysql -u adminsql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.40-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2024, 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.

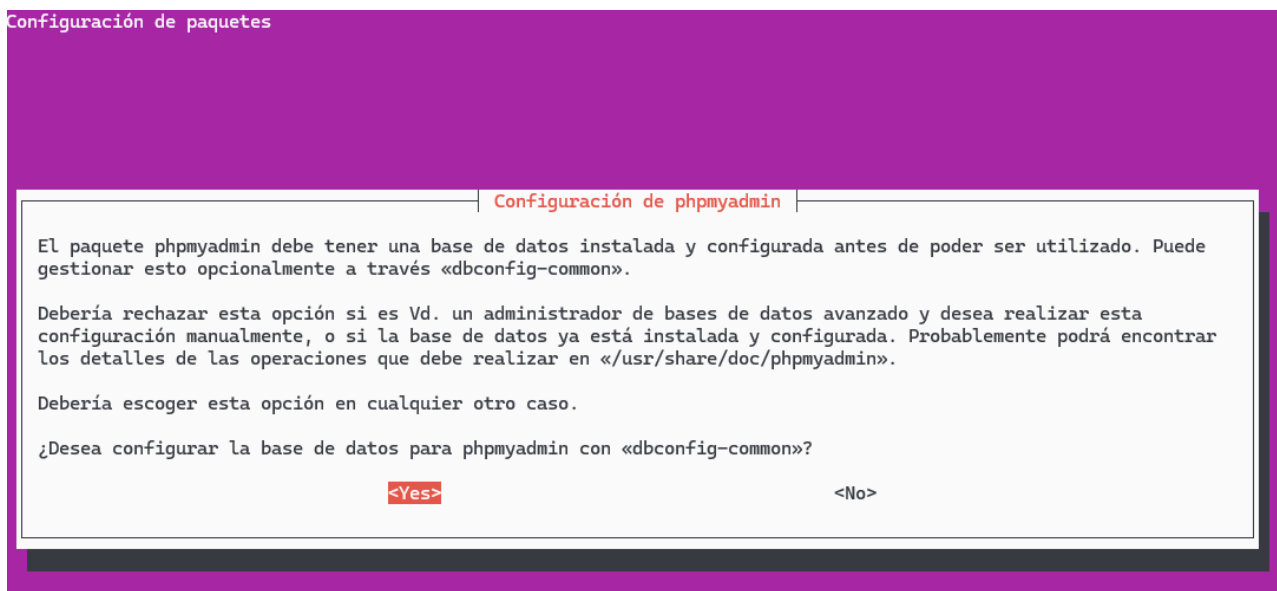
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> uninstall component "file://component_validate_password";
```

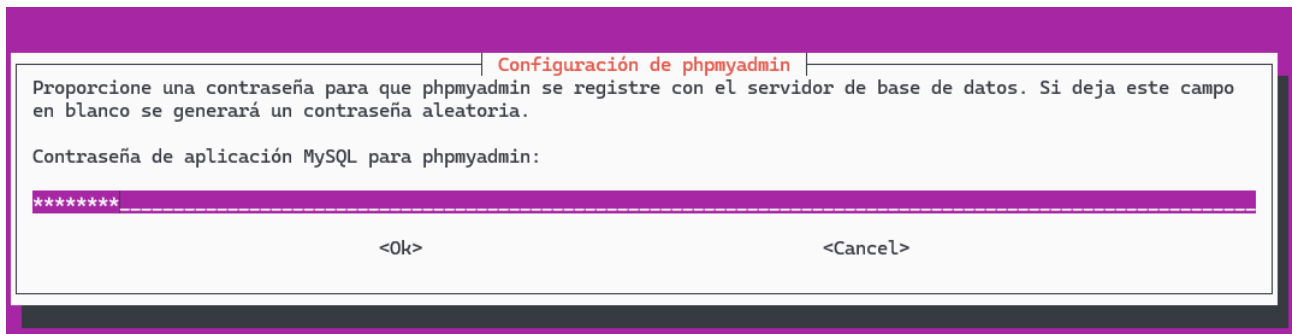
Procedemos a instalar

```
miadmin@vgg-used:~$ sudo apt install phpmyadmin
```

Le damos a yes a configurar la BD



Escogemos una contraseña



The screenshot shows a dialog box titled "Configuración de phpmyadmin". It contains the following text: "Proporcione una contraseña para que phpmyadmin se registre con el servidor de base de datos. Si deja este campo en blanco se generará un contraseña aleatoria." Below this, it says "Contraseña de aplicación MySQL para phpmyadmin:". There is a text input field containing "*****". At the bottom, there are two buttons: "<Ok>" and "<Cancel>".

Por último nos pide escoger el servidor web en nuestro caso **Apache** y poner en nuestro servidor **/phpmyadmin** en la **URL** y nos saldrá lo siguiente y ahí ya está listo



The screenshot shows the phpMyAdmin login page. At the top, there is a logo with the text "phpMyAdmin" and "Bienvenido a phpMyAdmin". Below the logo, there is a section for language selection with the label "Idioma - Language" and a dropdown menu showing "Español - Spanish". Below this, there is a section for login with the label "Iniciar sesión" and a button with a plus icon. There are two input fields: "Usuario:" and "Contraseña:". At the bottom right, there is a button labeled "Continuar".

Cuentas de desarrollo y hosting virtual

- Crear el usuario

Para crear el usuario que lo llamaremos operadorweb utilizaremos el comando

sudo adduser --no-create-home --home /var/www/html --ingroup www-data operadorweb

```
miadmin@vgg-used:~$ miadmin@vgg-used:~$ sudo adduser --no-create-home --home /var/www/html --ingroup www-data operadorweb
fatal: The user `operadorweb' already exists.
miadmin@vgg-used:~$
```

Para comprobar que esta bien creado usaremos **cat /etc/passwd | grep operadorweb**

```
miadmin@vgg-used:~$ cat /etc/passwd | grep operadorweb
operadorweb:x:1002:33:,,,:/var/www/html:/bin/bash
```

- Cambiar permisos

A continuación cambiamos el propietario del fichero con el comando **sudo chown -R operadorweb:www-data /var/www/html**

```
miadmin@vgg-used:~$ sudo chown -R operadorweb:www-data /var/www/html
miadmin@vgg-used:~$
```

Para cambiar los permisos al fichero para poder editarlo utilizaremos el comando **sudo chmod -R 2775 /var/www/html**

```
miadmin@vgg-used:~$ sudo chmod -R 2775 /var/www/html
miadmin@vgg-used:~$
```


Comprobamos los permisos con **ll /var/www/html**

```
miadmin@vgg-used:~$ ll /var/www/html
total 44
drwxrwsr-x  9 operadorweb www-data 4096 oct 25 09:15 ./
drwxr-xr-x  4 root          root    4096 oct 24 10:40 ../
drwxrwsr-x 12 operadorweb www-data 4096 oct 11 12:25 202DAWProyectoDAW/
drwxrwsr-x  8 operadorweb www-data 4096 oct 11 13:32 202DWESProyectoDWES/
drwxrwsr-x 12 operadorweb www-data 4096 oct 18 09:02 202DWESProyectoTema3/
drwxrwsr-x  2 operadorweb www-data 4096 sep 20 09:19 .cache/
drwxrwsr-x  2 operadorweb www-data 4096 sep 26 12:18 doc/
-rwxrwsr-x  1 operadorweb www-data  98 oct 25 09:18 .htaccess*
-rwxrwsr-x  1 operadorweb www-data 1237 oct  8 11:43 index.html*
drwxrwsr-x  2 operadorweb www-data 4096 oct  3 10:29 logs/
drwxrwsr-x  3 operadorweb www-data 4096 sep 20 09:20 webroot/
```

- Configurar SSH

Luego procedemos a configurar la conexión SSH así que modificaremos el siguiente archivo

sudo nano /etc/ssh/sshd_config

```
miadmin@vgg-used:~$ sudo nano /etc/ssh/sshd_config
```

Añadimos estas líneas al final

```
# Example of overriding settings on a per-user basis
#Match User anoncvs
#      X11Forwarding no
#      AllowTcpForwarding no
#      PermitTTY no
#      ForceCommand cvs server
Match Group ftpusers
ChrootDirectory %h
ForceCommand internal-sftp -u 2
AllowTcpForwarding yes
PermitTunnel no
X11Forwarding no
```

- Configurar sitios virtuales

Ahora vamos a configurar los sitios virtuales. Tenemos que ir al directorio

cd /etc/apache2/sites-available y creamos este archivo **sudo nano DAW202.conf**

```
niadmin@vgg-used:/etc/apache2/sites-available$ sudo nano DAW202.conf
```

Lo rellenamos con lo siguiente

```
GNU nano 7.2 DAW202.conf *
<VirtualHost *:80>
    ServerName daw202.isauces.local
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/DAW202/public_html
    ErrorLog /var/log/apache2/error-daw214.log
    CustomLog /var/log/apache2/access-daw214.log combined
</VirtualHost>
```

Para comprobar que no exista ningún error en los archivos ejecutamos este comando **sudo apache2ctl configtest**

```
miadmin@vgg-used:/etc/apache2/sites-available$ sudo apache2ctl configtest
Syntax OK
```

Ahora para habilitar el fichero utilizaremos el comando **sudo a2ensite DAW202.conf** y utilizamos el comando que nos dice **systemctl reload apache2** y nos pide autenticarnos

```
miadmin@vgg-used:/etc/apache2/sites-available$ sudo a2ensite DAW202.conf
Enabling site DAW202.
To activate the new configuration, you need to run:
  systemctl reload apache2
miadmin@vgg-used:/etc/apache2/sites-available$ systemctl reload apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to reload 'apache2.service'.
Multiple identities can be used for authentication:
 1. Victor (miadmin)
 2. miadmin2
Choose identity to authenticate as (1-2): 1
Password:
==== AUTHENTICATION COMPLETE ====
miadmin@vgg-used:/etc/apache2/sites-available$
```

- Enjaular usuarios

Primero creamos un **grupo** para los usuarios de **SFTP** o **FTP**

```
miadmin@vgg-used:~$ sudo groupadd ftpusers
```

Creamos un **usuario** y le cambiamos la **contraseña**

```
miadmin@vgg-used:~$ sudo useradd -g www-data -G ftpusers -m -d /var/www/DAW202 DAW202
miadmin@vgg-used:~$ passwd DAW202
passwd: You may not view or modify password information for DAW202.
miadmin@vgg-used:~$ sudo passwd DAW202
New password:
Retype new password:
passwd: password updated successfully
```

Cambiamos el propietario del directorio que tiene que ser **root** y eliminamos el **permiso de escritura**

```
miadmin@vgg-used:~$ sudo chown root:root /var/www/DAW202
miadmin@vgg-used:~$ sudo chmod 555 /var/www/DAW202
```

Por último creamos la **carpeta** a la que puede acceder el usuario y le indicamos los **permisos** y el **propietario**

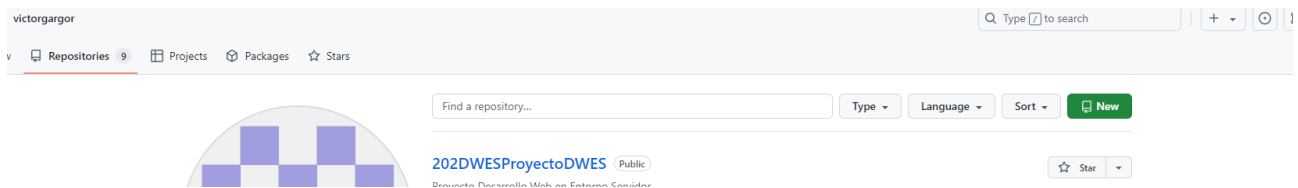
```
miadmin@vgg-used:~$ sudo mkdir /var/www/DAW202/public_html
```

```
miadmin@vgg-used:~$ sudo chmod 2775 -R /var/www/DAW202/public_html
miadmin@vgg-used:~$ sudo chown DAW202:www-data -R /var/www/DAW202/public_html
```

GITHUB – Internet

- Crear repositorio

Iniciamos sesión en **github** o nos creamos una cuenta y vamos a donde pone **repositories** y le damos a **new**



Nos pide ahora introducir datos como **nombre**, **descripción**, si queremos que sea **público** o **privado**, añadir un **archivo README**, **licencia** etc.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * / Repository name *

repositorio is available.

Great repository names are short and memorable. Need inspiration? How about [scaling-memory](#) ?

Description (optional)

- ☒ ☐ Public
Anyone on the internet can see this repository. You choose who can commit.
- ☐ ☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:

- ☒ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set [main](#) as the default branch. Change the default name in your [settings](#).

☐ You are creating a public repository in your personal account.

Create repository

Activar W
Ve a Configu

- Comandos

Configuraciones Globales

1. Configurar el usuario y el correo electrónico:

```
git config --global user.name "Tu Nombre"
```

```
git config --global user.email "tuemail@ejemplo.com"
```

2. Ver configuraciones actuales:

```
git config --list
```

Creación de Repositorios

1. Inicializar un repositorio nuevo:

```
git init
```

2. Clonar un repositorio existente:

```
git clone <URL_del_repositorio>
```

Descarga un repositorio remoto y crea una copia local.

Estado del Repositorio

1. Verificar el estado actual:

```
git status
```

Muestra información sobre los cambios en el repositorio.

2. Ver diferencias en los archivos:

```
git diff
```

Muestra las diferencias entre los archivos modificados y el último commit.

Actualización y Control de Cambios

1. Agregar archivos al área de preparación:

git add <archivo>

git add .

Agrega archivos específicos o todos los cambios al área de preparación.

2. Confirmar cambios (commit):

git commit -m "Mensaje descriptivo"

Guarda los cambios en el historial del repositorio.

3. Combinar agregar y confirmar:

git commit -am "Mensaje descriptivo"

Agrega y confirma cambios de archivos ya rastreados.

4. Actualizar el repositorio local con cambios remotos:

git pull <origen> <rama>

Descarga y fusiona cambios desde el repositorio remoto.

5. Subir cambios al repositorio remoto:

git push <origen> <rama>

Sube los commits locales al repositorio remoto.

Gestión de Ramas

1. Crear una nueva rama:

git branch <nombre_rama>

Crea una nueva rama local.

2. Cambiar a otra rama:

git checkout <nombre_rama>

Cambia a la rama especificada.

3. Crear y cambiar de rama en un paso:

git checkout -b <nombre_rama>

Crea y cambia a una nueva rama en un solo comando.

4. Fusionar ramas:

git merge <nombre_rama>

Combina los cambios de una rama en la rama activa.

Etiquetas (Tags)

1. Crear una nueva etiqueta:

git tag <nombre_etiqueta>

Crea una etiqueta simple.

2. Crear una etiqueta con mensaje:

git tag -a <nombre_etiqueta> -m "Mensaje descriptivo"

Crea una etiqueta anotada con un mensaje.

3. Ver todas las etiquetas:

git tag

Lista todas las etiquetas existentes.

4. Subir etiquetas al repositorio remoto:

git push <origen> <nombre_etiqueta>

Sube una etiqueta específica al repositorio remoto.

5. Subir todas las etiquetas:

git push --tags

Sube todas las etiquetas locales al repositorio remoto.

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

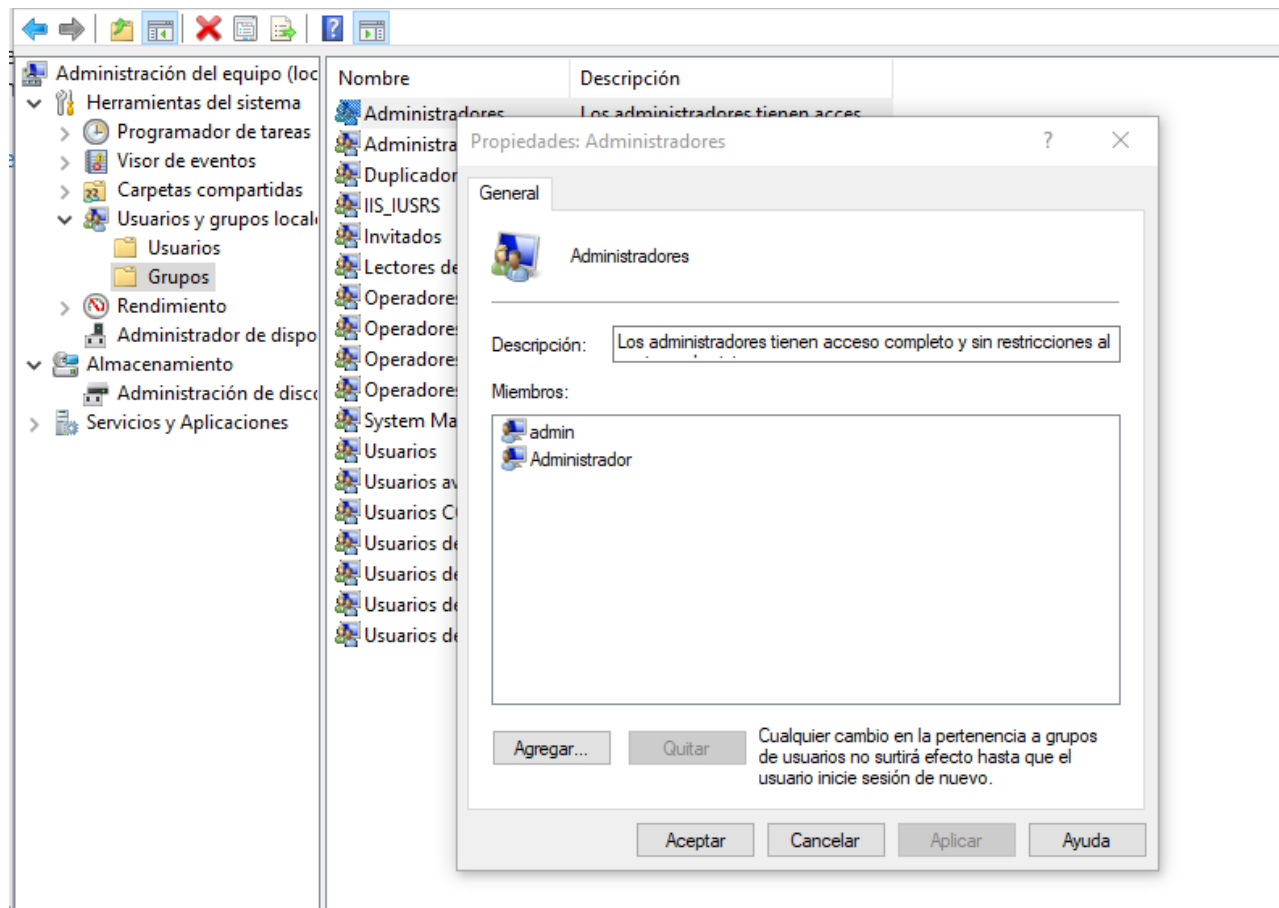
Copiar

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



Nombre	Descripción
Administradores	Los administradores tienen acceso a todos los recursos del sistema.
Administradores de tareas	
Usuarios	
Duplicador	
IIS_IUSRS	
Invitados	
Lectores de discos	
Operadores de disco	
Operadores de impresora	
Operadores de red	
Operadores de sistema	
System Maintenance	
Usuarios	
Usuarios de red	
Usuarios de sistema	
Usuarios de sistema	
Usuarios de sistema	
Usuarios de sistema	

Propiedades: Usuarios

General

Usuarios

Descripción: Los usuarios no pueden hacer cambios accidentales o

Miembros:

- bachillerato
- daw1
- daw2
- eso
- NT AUTHORITY\INTERACTIVE (S-1-5-4)
- NT AUTHORITY\Usuarios autenticados (S-1-5-11)
- smr1
- smr2

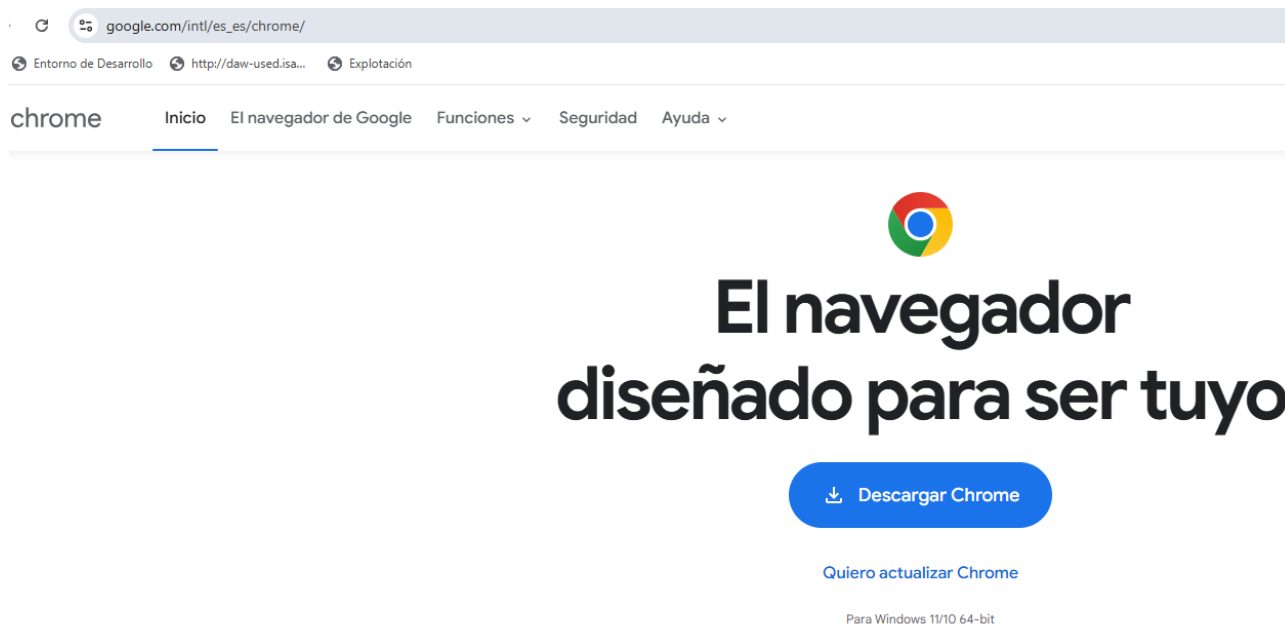
Agregar... Quitar

Cualquier cambio en la pertenencia a grupos de usuarios no surtirá efecto hasta que el usuario inicie sesión de nuevo.

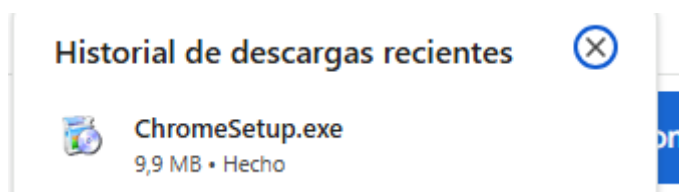
Aceptar Cancelar Aplicar Ayuda

Navegadores

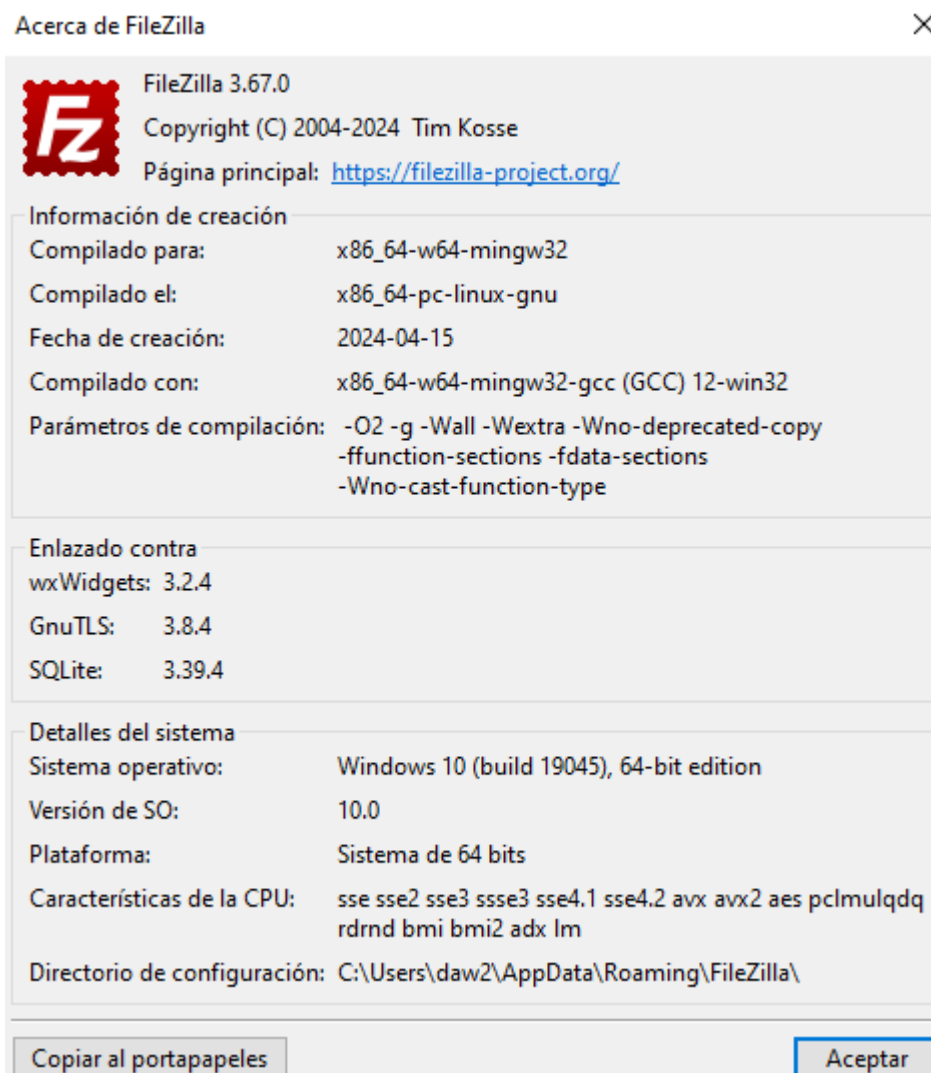
En mi caso voy a utilizar **Google Chrome** y para descargarlo lo descargamos por el enlace siguiente https://www.google.com/intl/es_es/chrome/



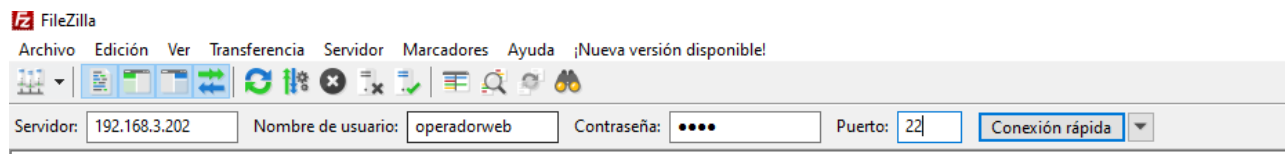
Le damos al botón de descargar y se nos descarga el ejecutable lo abrimos, seguimos las instrucciones del instalador y nos abre el navegador y nos crea un acceso directo en el escritorio



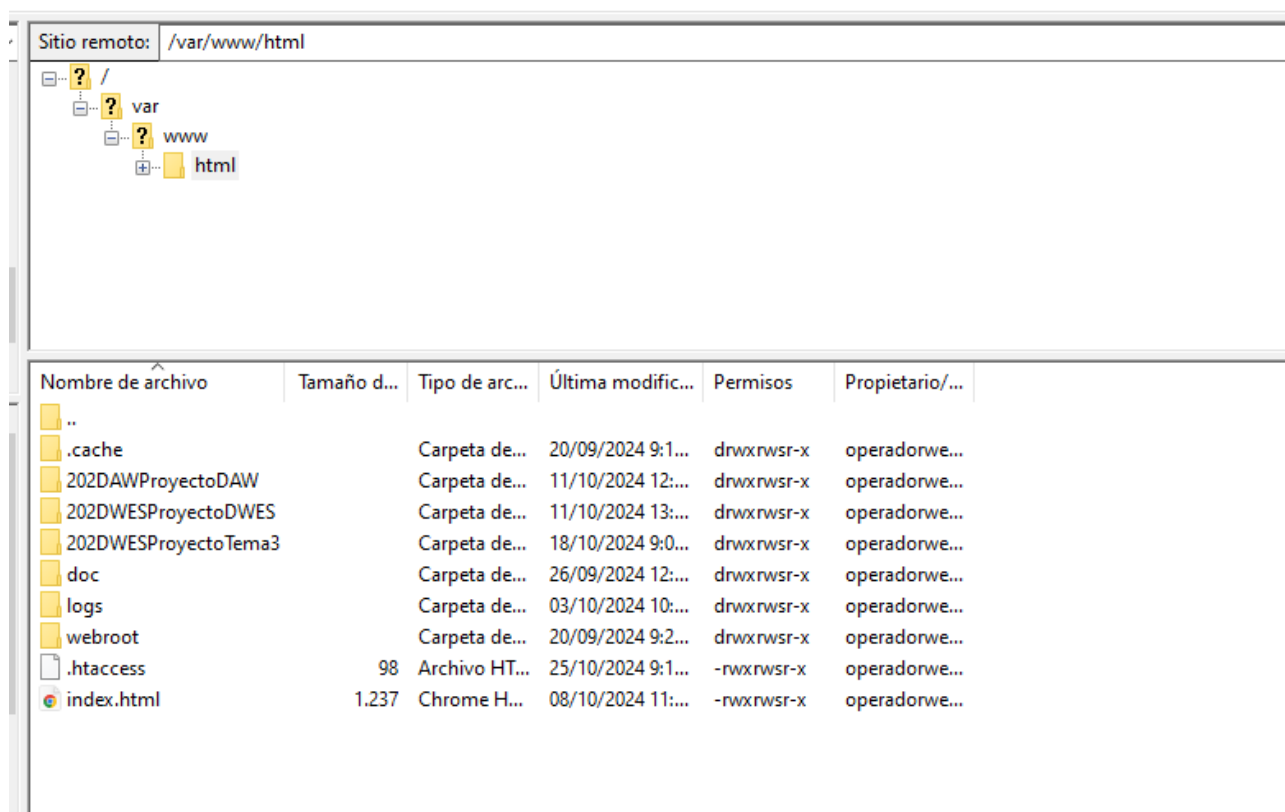
Filezilla



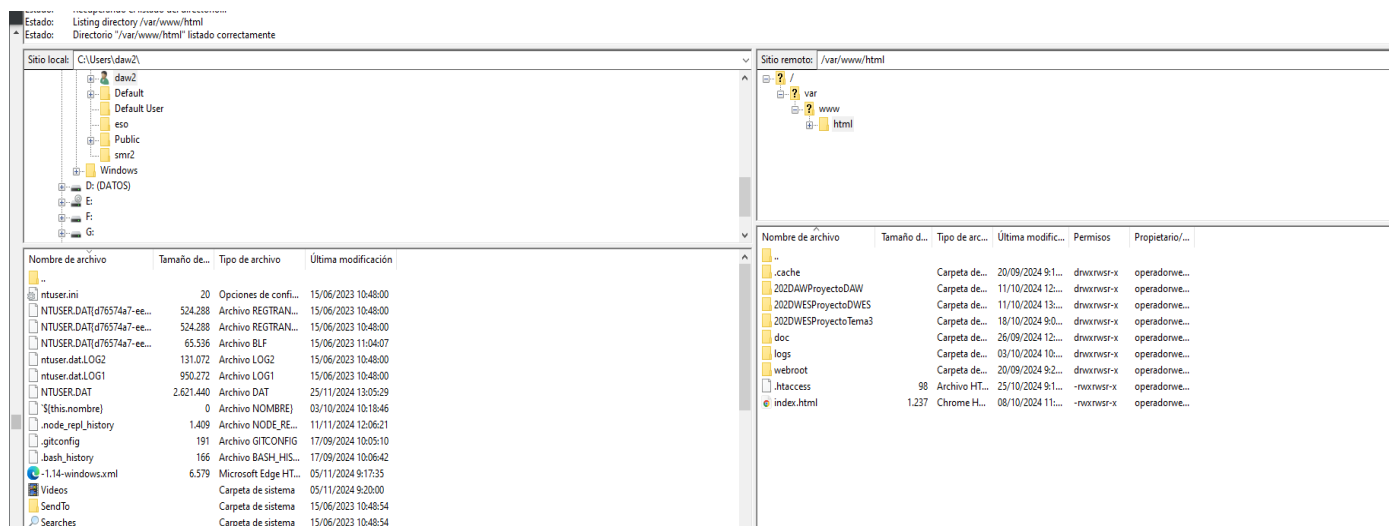
Al abrir **FileZilla** introducimos los parámetros de conexión



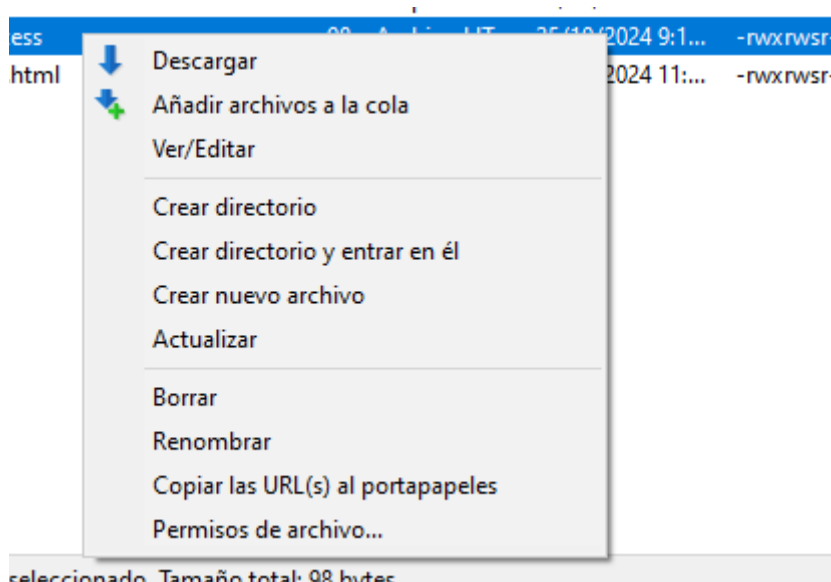
Nos aparecerá el directorio remoto y su contenido

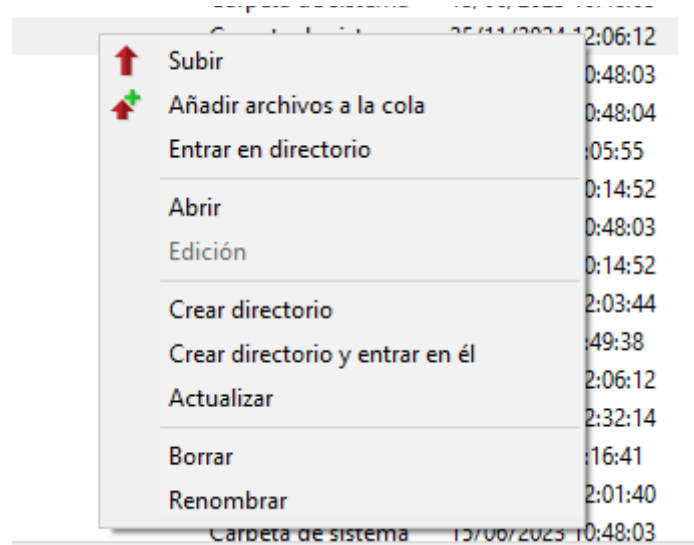


Para subir y bajar archivos se puede arrastrar lo que se necesita subir al servidor o lo que se quiere descargar de él



O también se puede hacer click derecho sobre el archivo o directorio e indicar lo que se quiere hacer. (La primera captura es la parte del **Servidor** y la segunda del **Cliente**)





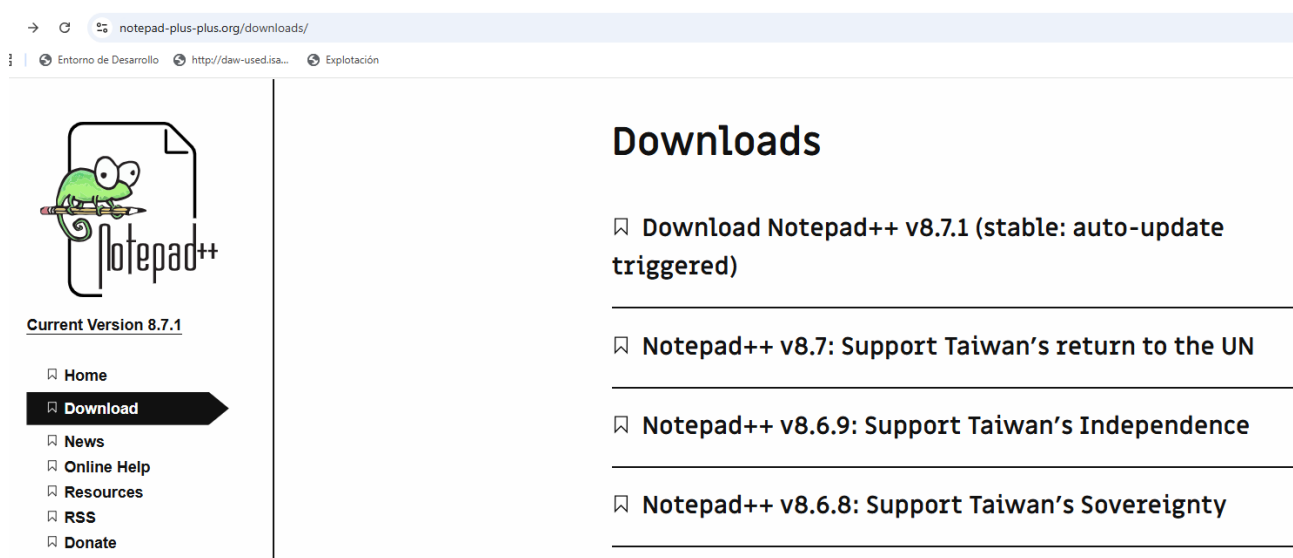
Notepad++

Notepad++ es un editor de texto y de código fuente **libre** con soporte para varios **lenguajes de programación**. Con soporte nativo para Microsoft Windows.

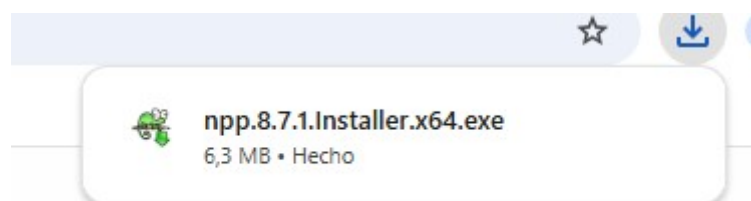
Se parece al Bloc de notas en cuanto al hecho de que puede editar texto sin formato y de forma simple. No obstante, incluye opciones más avanzadas que pueden ser útiles para usuarios avanzados como desarrolladores y programadores.

Para instalarlo vamos a la página oficial con el enlace siguiente

<https://notepad-plus-plus.org/downloads/>

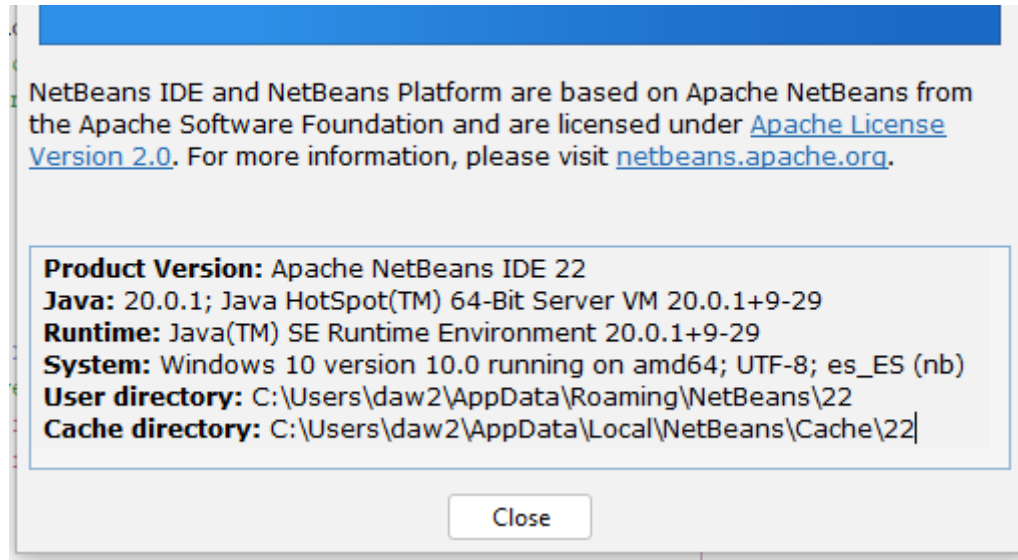


Entramos a la última versión le damos a **descargar** y nos descarga el **ejecutable**. Simplemente hacemos caso al asistente de instalación y se nos creará un acceso directo en el escritorio y se ejecutará



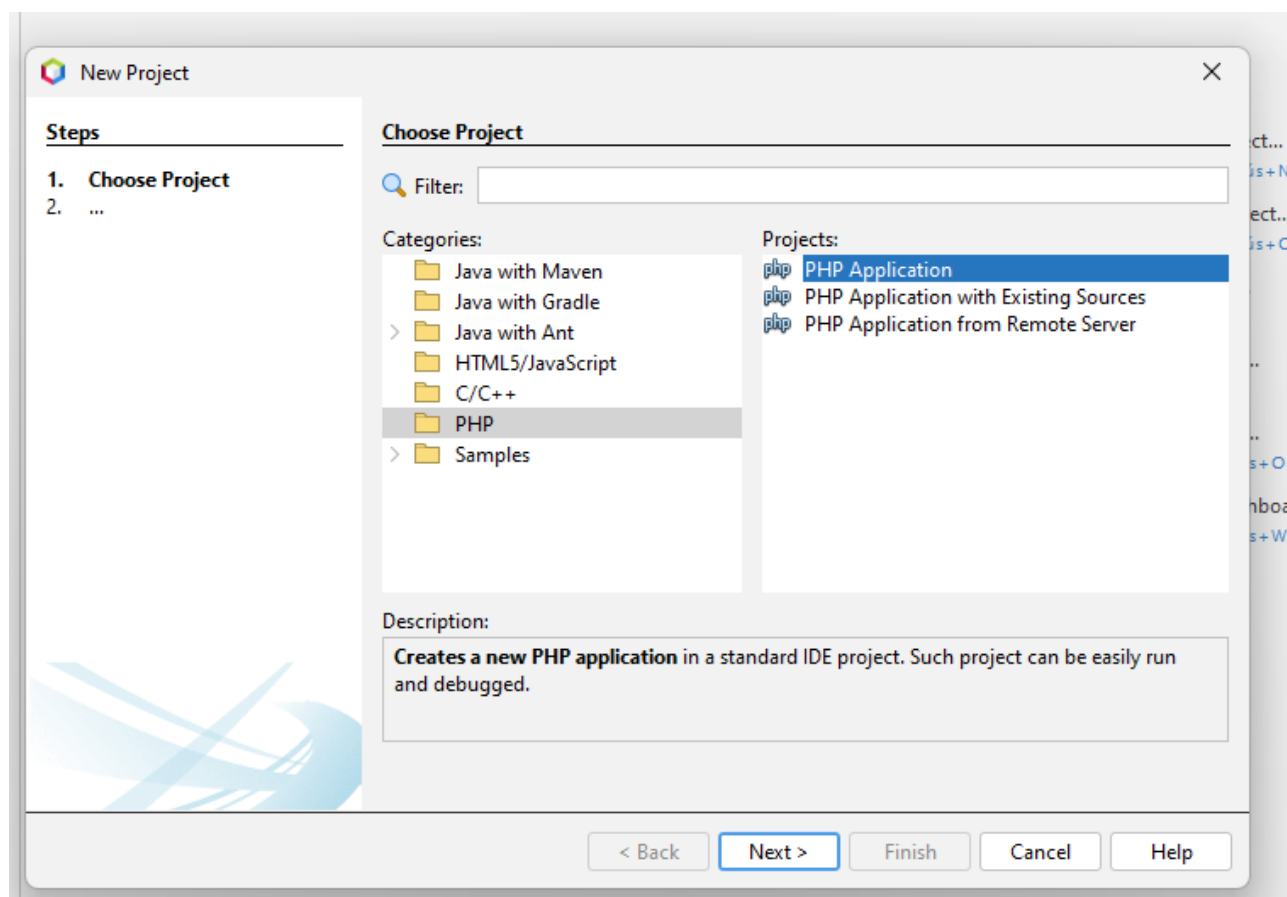
NetBeans

Instalación y configuración inicial (plugins)

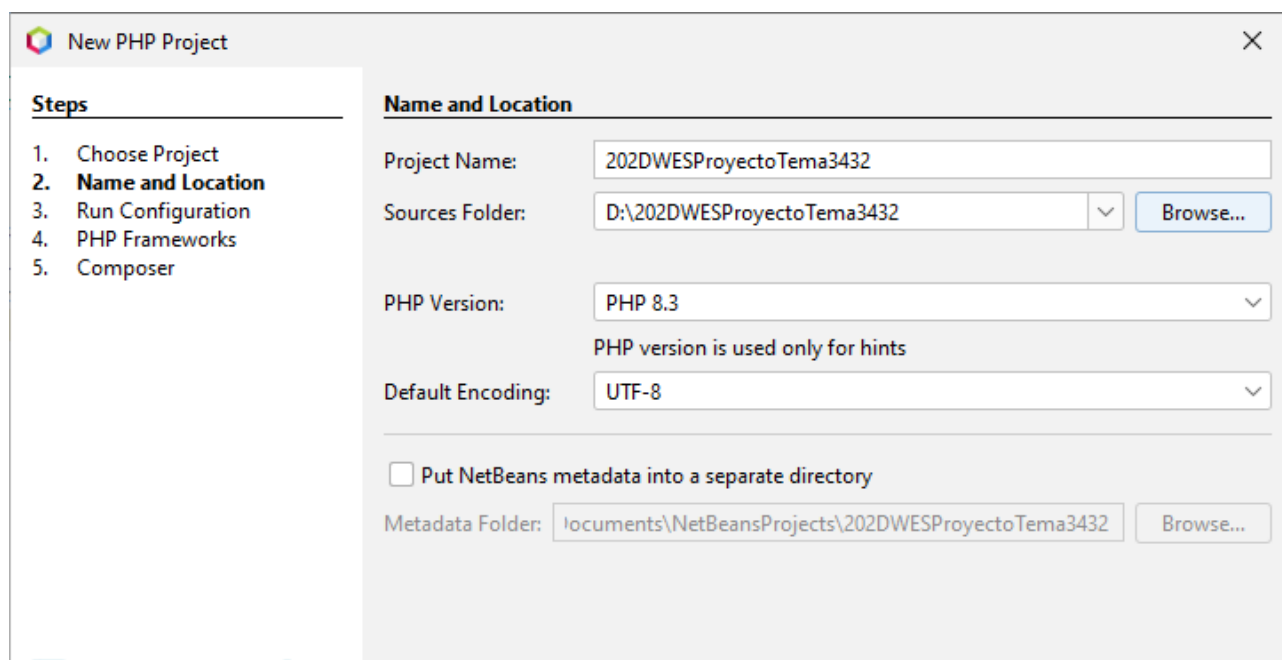


Creación de proyectos, modificación, borrado, prueba

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



New PHP Project

Steps

1. Choose Project
2. **Name and Location**
3. Run Configuration
4. PHP Frameworks
5. Composer

Name and Location

Project Name:

Sources Folder:

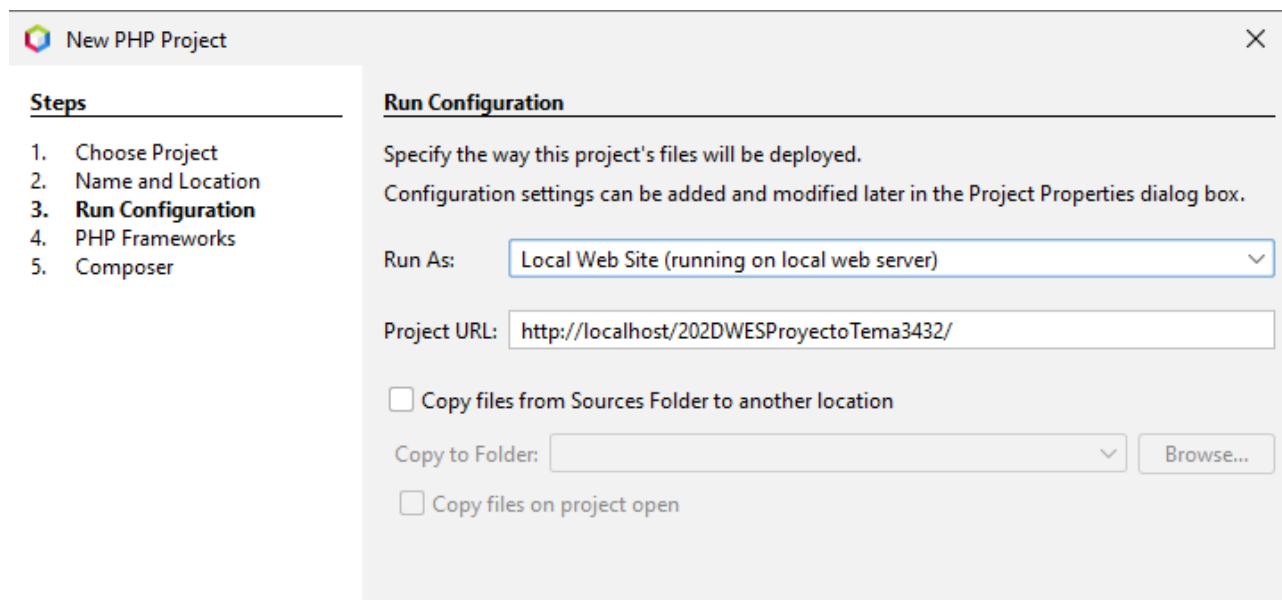
PHP Version:
PHP version is used only for hints

Default Encoding:

☐ Put NetBeans metadata into a separate directory

Metadata Folder:

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



New PHP Project

Steps

1. Choose Project
2. Name and Location
3. **Run Configuration**
4. PHP Frameworks
5. Composer

Run Configuration

Specify the way this project's files will be deployed.
 Configuration settings can be added and modified later in the Project Properties dialog box.

Run As:

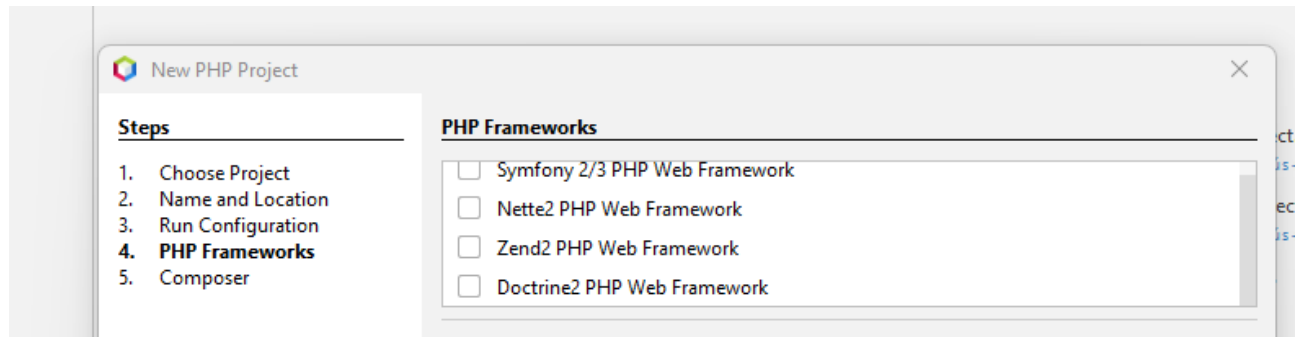
Project URL:

☐ Copy files from Sources Folder to another location

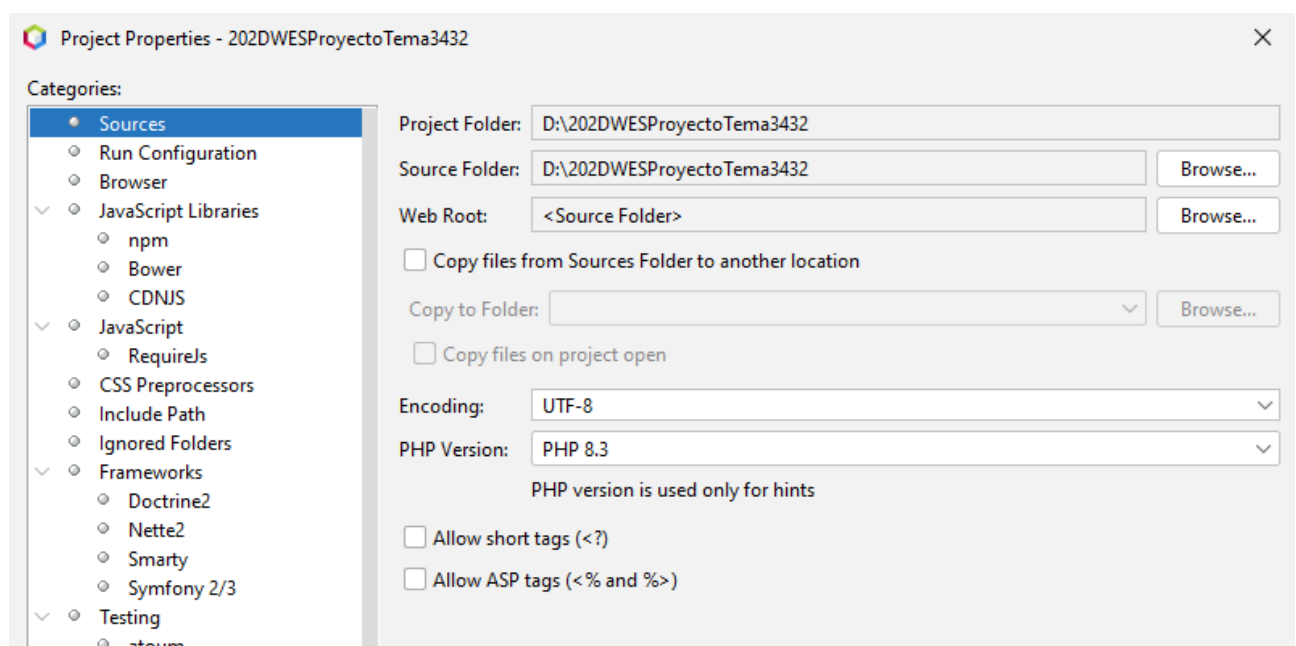
Copy to Folder:

☐ Copy files on project open

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.

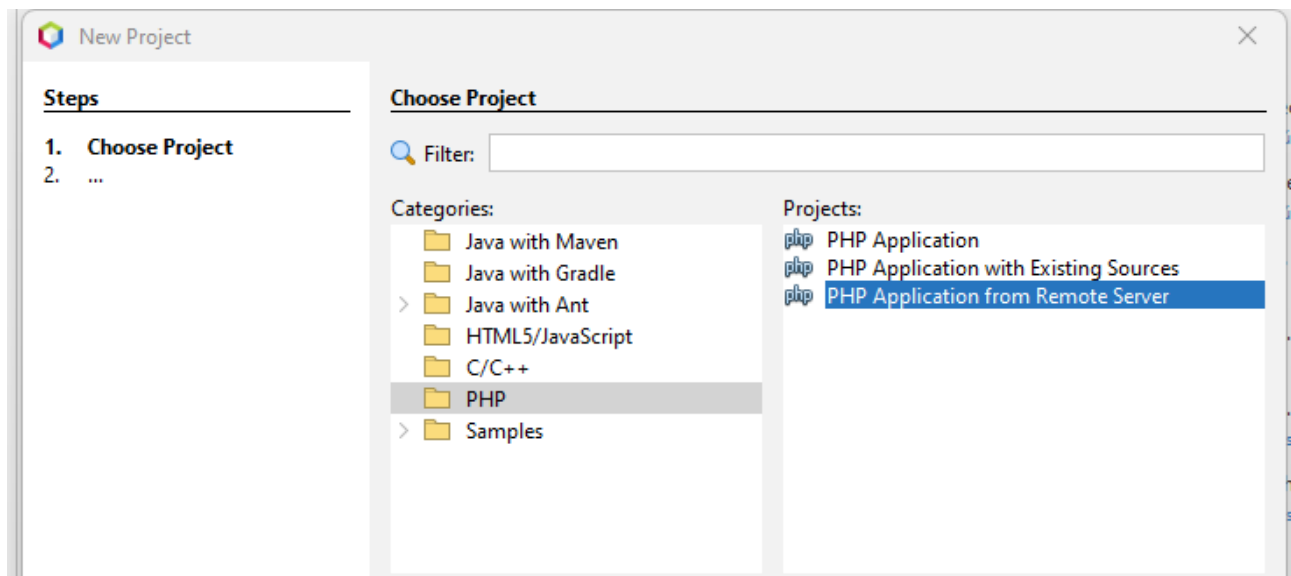


Para eliminar sería tan sencillo como hacer click derecho y darle a delete y se borrará el proyecto de nuestro equipo en local,

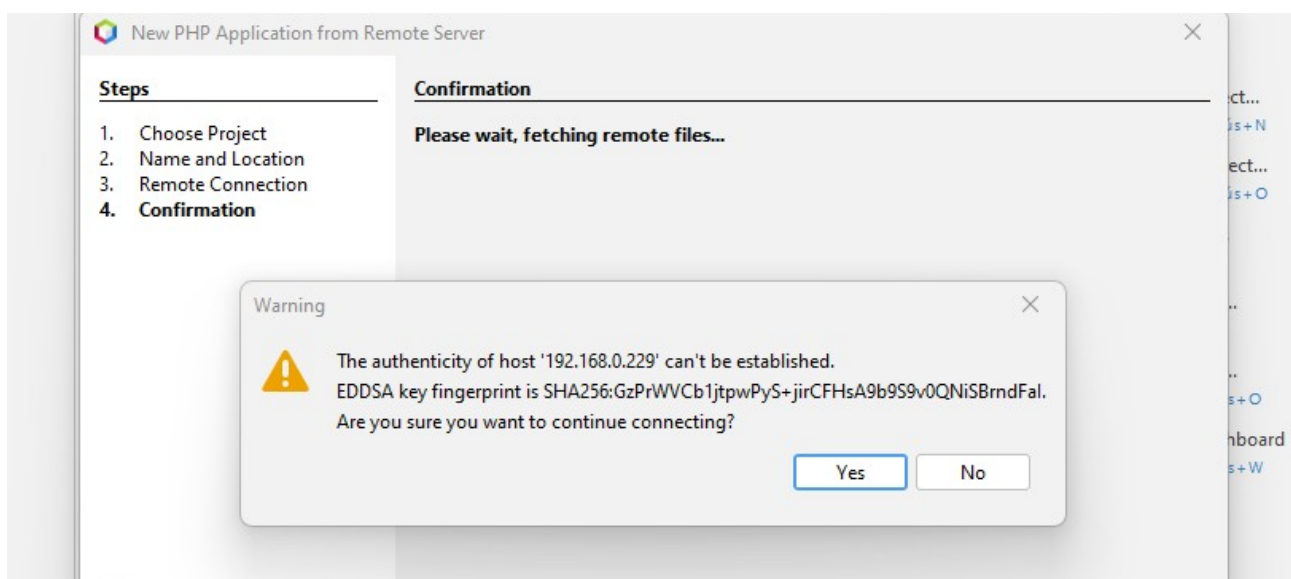
Conexión al servidor remoto SFTP. (Almacenamiento local/almacenamiento remoto)

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

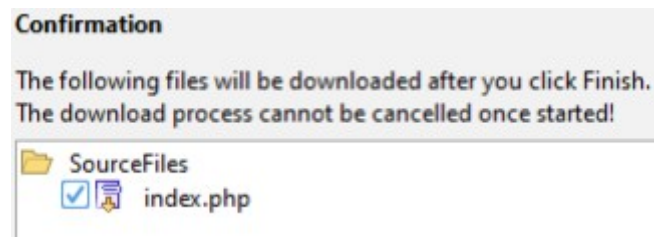
Al hacer el New Project elegiremos la opción de Remote Server.



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.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

victorgargor

Repository name *

202DWESProyectoTema34

202DWESProyectoTema3432 is available.

Great repository names are short and memorable. Need inspiration? How about [reimagined-octo-telegram](#) ?

Description (optional)

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.

Clone repository

Steps

1. Remote Repository
2. Remote Branches
3. Destination Directory

Remote Repository

Specify Git Repository Location:

Repository URL: ⌵

http[s]://host.xz[:port]/path/to/repo.git/

User: (leave blank for anonymous access)

Password: ☒ Save Password

[Proxy Configuration...](#)

Specify Destination Folder:

Clone into: [Browse...](#)

(Leave empty to specify the destination later)

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.

Destination Directory

Specify the Parent Directory and Name for this Clone

Parent Directory: [Browse...](#)

Clone Name:

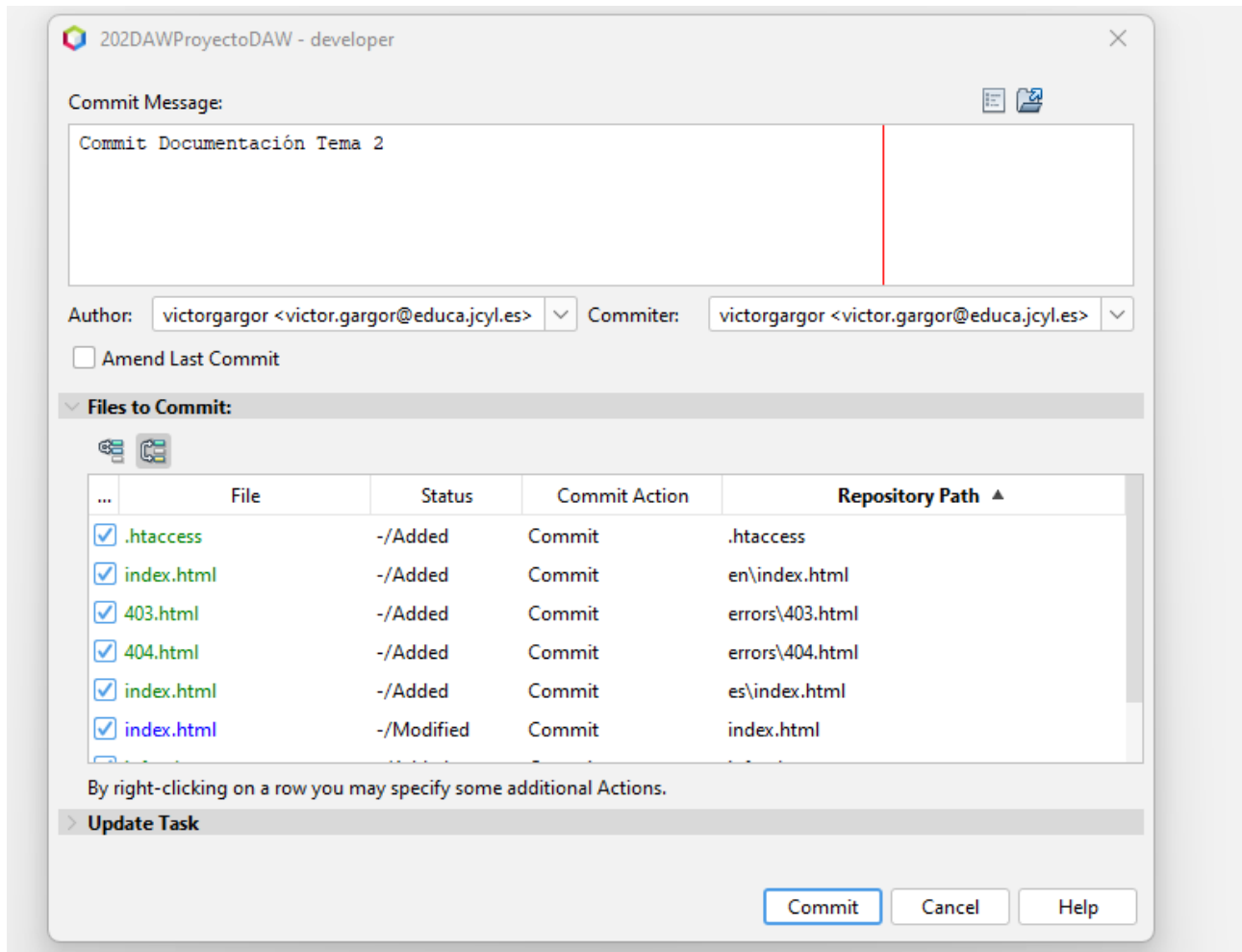
Checkout Branch: No Branch ⌵

Remote Name:

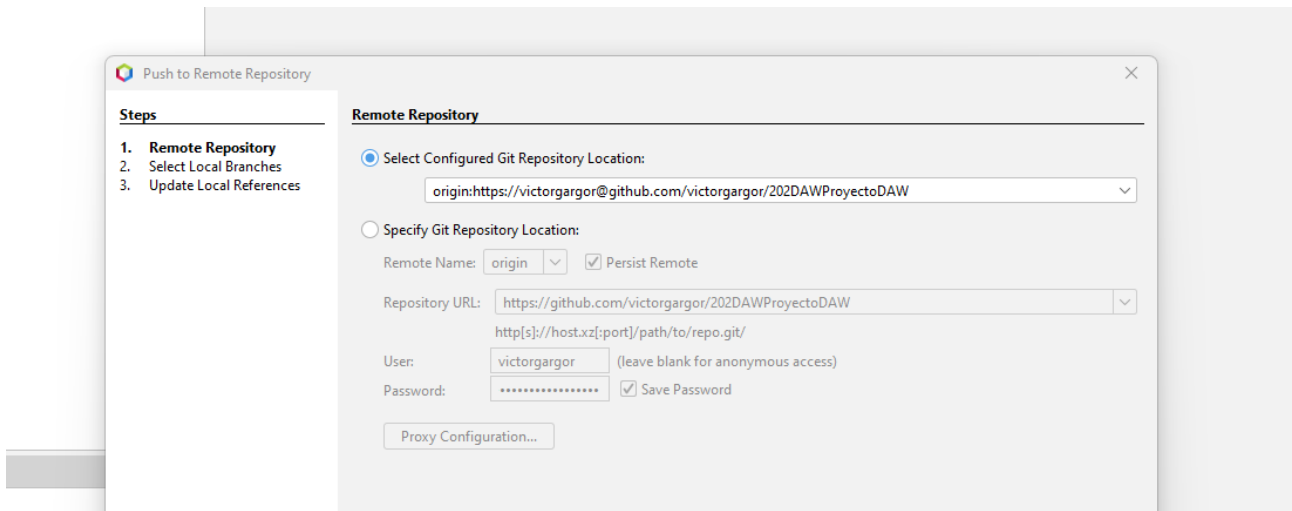
☒ Scan for NetBeans Projects after Clone

Le damos a finish y ahí estaría conectado el repositorio.

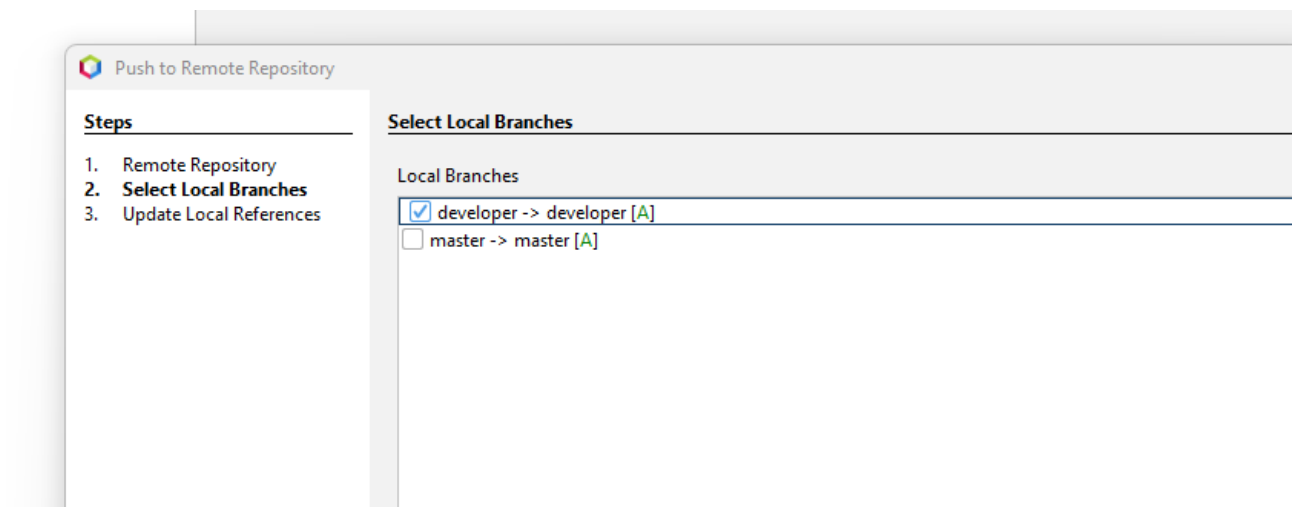
Ahora para versionar el contenido del proyecto lo primero que tendremos que hacer es hacer click derecho en el proyecto ir a Git y a Commit. Ahí ponemos el mensaje que queramos y seleccionamos los archivos correspondientes.



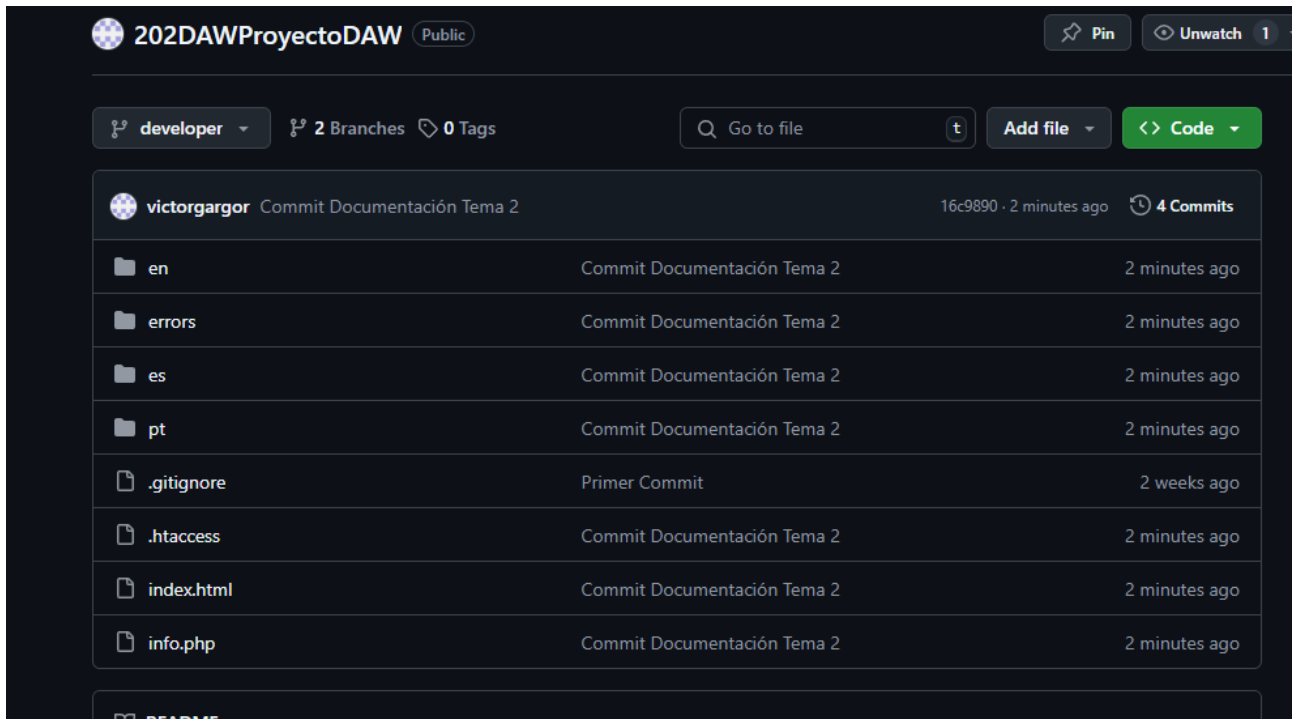
Para subir el contenido al repositorio tendremos que hacer click derecho Git ir a Remote y hacer un Push. Te pedirá la localización del repositorio remoto.



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



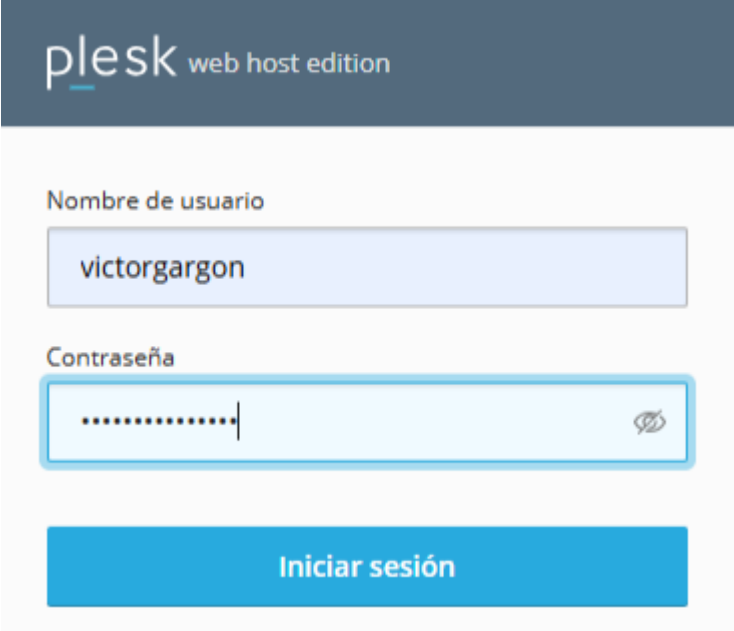
Revisamos que se haya actualizado el repositorio en la rama seleccionada.



Depuración – Configuración de la ejecución para la depuración

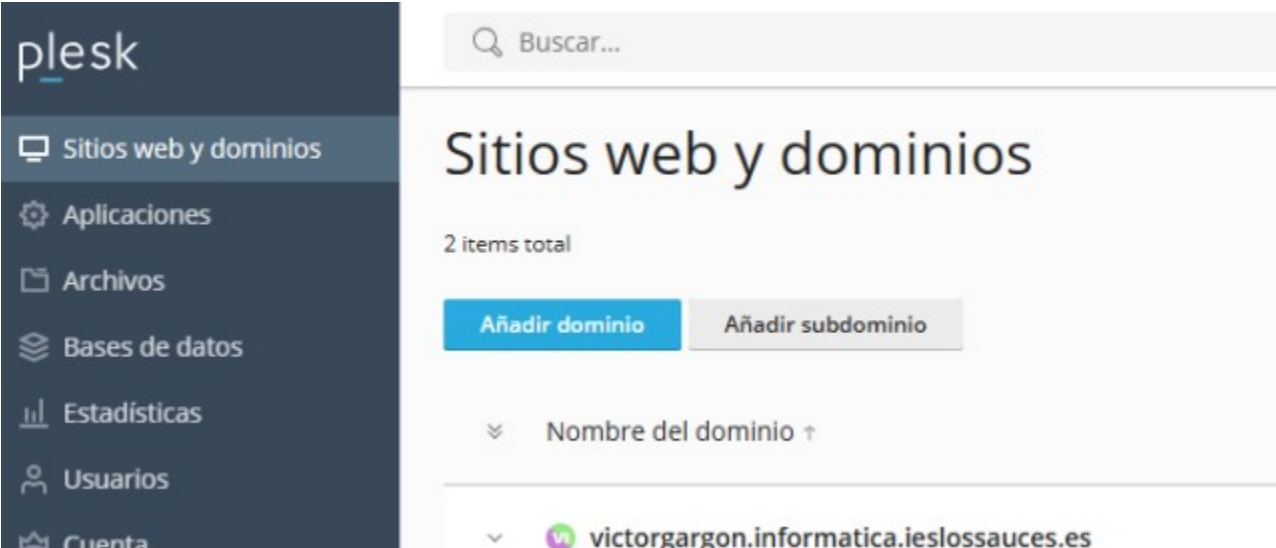
Paso a explotación – Plesk

Primero iniciamos sesión en **plesk** con el usuario y contraseña que se nos ha proporcionado



The image shows the Plesk login interface. At the top, there is a dark blue header with the 'plesk' logo and 'web host edition' text. Below this, the login form is centered. It has two input fields: 'Nombre de usuario' (Username) with the value 'victorgargon' and 'Contraseña' (Password) with masked characters. A blue 'Iniciar sesión' (Login) button is at the bottom.

Tenemos que ir al apartado de **Sitios web y dominios** y le damos a añadir subdominio



The image shows the Plesk dashboard for 'Sitios web y dominios' (Websites and domains). On the left is a dark sidebar with the Plesk logo and a menu: 'Sitios web y dominios' (selected), 'Aplicaciones', 'Archivos', 'Bases de datos', 'Estadísticas', 'Usuarios', and 'Cuenta'. The main area has a search bar at the top. Below it, the title 'Sitios web y dominios' is followed by '2 items total'. There are two buttons: 'Añadir dominio' (Add domain) and 'Añadir subdominio' (Add subdomain). Below these is a table with one header 'Nombre del dominio ↑' and one row showing a subdomain 'victorgargon.informatica.ieslossauces.es' with a dropdown arrow to its left.

Ahora nos pide el nombre del subdominio que será el nombre del proyecto, y el raíz del documento que tiene que ser **httpdocs/nombredelproyecto** y le damos a aceptar

Añadir un subdominio

Los subdominios son direcciones de Internet para las distintas secciones de su sitio web. Estos utilizan su nombre de dominio. Puede crear un subdominio wildcard introduciendo el símbolo * en vez del nombre. En este caso, los visitantes del sitio serán redirigidos a la carpeta correspondiente.

Nombre del subdominio * .

Introduzca * para crear un subdominio wildcard.

Configuración de hosting

Raíz del documento *

Ruta al directorio principal del sitio web.

* Campos obligatorios

Vamos ahora al subdominio creado y tienes distintas maneras de cargar archivos, bien por **Filezilla** (documentado en el apartado **Filezilla**), mediante carga directa en plesk con el **.zip** de la **release** o con la opción de **desplegar mediante Git** que es la que voy a emplear yo

Estadísticas

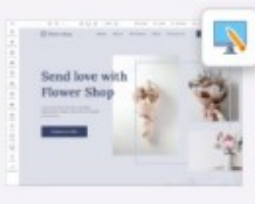
Espacio en disco 0 MB

Tráfico este mes 0 MB

[Más estadísticas](#) ▼

Panel de información Hosting y DNS Correo **Primeros pasos**


Crear un sitio web o una aplicación en este dominio




Sitejet Builder

Creador de sitios web de última generación: + de 100 plantillas de diseño, sin código, generador de texto con IA, SEO, E-commerce y mucho más.


[Más info](#)




Cargar archivos




Desplegar mediante Git




WordPress




Importe una app o un sitio



Laravel



Noc



Más apps

Nos pedirá introducir la **url del repositorio**, nuestro **usuario y contraseña de GitHub** y el modo de despliegue lo dejamos en **Automático** y le damos a crear

URL del repositorio *

Se soportan los protocolos HTTP(S) y SSH

Si el acceso al repositorio está protegido mediante credenciales HTTP básicas, indíquelas a conti

Nombre de usuario

Contraseña

Nombre del repositorio *

Indique un nombre único en un dominio.

Configuración de despliegue

Modo de despliegue *

☒ Automático

☐ Manual

☐ Desactivado

Los archivos se desplegarán en el sitio de producción cuando estén disponibles en el repositorio de Plesk.

Ruta de acceso al servidor *

Directorio del servidor donde se desplegarán los archivos.

☐

Habilite acciones de despliegue adicionales

Especifique comandos shell a ejecutar cada vez que se produzca un despliegue.

Luego comprobamos que desplegamos la rama **master** y le damos a **desplegar ahora**

Desplegar clave

Commits más recientes

- 2024-11-07 11:52 Merge developer
- 2024-11-07 11:51 .htaccess modificado
- ⋮ [mostrar más](#)

 Pull ahora

Despliegue

master rama automáticamente en </httpdocs/202DWESProyectoTema3>

 Desplegar ahora

Comprobamos que funciona y listo

 victorgarcia.informatica.ieslossauces.es/202DWESProyectoTema3/indexProyectoTema3.php

mo de Desarrollo  <http://daw-used.ia...>  Explotación

TEMA 3 CARACTERÍSTICAS DEL LENGUAJE PHP

Enunciado

Hola mundo y phpinfo().

Prestashop