

---

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

---

## Tabla de contenido

BNR-USED .....	3
1. Configuración inicial.....	3
1.1. Características de la máquina virtual.....	3
1.2. Configuración de red.....	3
1.3. Nombre de la máquina.....	4
1.4. Actualización del sistema.....	5
1.5. Configuración zona horaria.....	5
1.6. Cortafuegos local.....	6
2. Cuentas de administración.....	7
3. Apache.....	8
3.1. ¿Qué es y para qué sirve?.....	8
3.2. Estructura de directorios/archivos de Apache 2.....	8
3.3. Comandos de control del servicio .....	8
3.4. Instalación y configuración En la maquina RSP-USED instalaremos el servicio Apache2, pero antes de eso tenemos que actualizar los repositorios de Linux con el comando “sudo apt-get update”. .....	9
3.5. Configuración de Logs de errores y acceso .....	10
3.6. Habilitar el uso de .htaccess .....	11
3.7. Modulo Alias.....	13
3.8. Modulo Rewrite.....	14
3.9. HTTPS .....	15
4. PHP .....	20
4.1. Instalación.....	20
4.2. Módulos .....	23

5. XDebug .....	23
5.1. Instalación .....	23
6. MySQL.....	24
6.1. Instalación y configuración.....	24
6.2. PHPMyAdmin.....	26
8. Cuentas de desarrollo y hosting virtual. ....	29
8.1. Creación de operadorweb.....	29
IS32WX07.....	30
5. NetBeans.....	30
5.2. Creación de proyectos, modificación, borrado, prueba. ....	30
<b>5.2.1. Creación.</b> .....	30
<b>5.2.2. Modificación.</b> .....	34
<b>5.2.3. Borrado</b> .....	35
<b>5.2.4. Prueba</b> .....	36
<b>5.2.5. Configuración.</b> .....	36
5.5. Depuración - Configuración de la ejecución para la depuración.....	39
<b>5.5.1. Ejecución de la depuración.</b> .....	39
<b>5.5.2 Configuración del debug.</b> .....	41
Commit .....	45

## BNR-USED

### 1. Configuración inicial.

#### 1.1. Características de la máquina virtual.

EQUIPO	SISTEMA OPERATIVO:	CONFIGURACIÓN DE RED: IP/Mascara DNS GW MAC	CUENTAS LOCALES (administrativas): usuario/password	RAM / DISCO PARTICIONES
<b>BNR-USED</b>	Ubuntu Server 22.04 x64	<b>Enp0s3:</b> <b>IP:</b> 192.168.3.17 <b>DNS:</b> 8.8.8.8 <b>Máscara:</b> 255.255.255.0/24 <b>GW:</b> 192.168.3.1 <b>MAC:</b> 08:00:27: cb:bd: ff	miadmin/paso miadmin2/paso operadorweb/paso	<b>RAM:</b> 2GB <b>Disco:</b> 500GB <b>Particiones:</b> -150GB / -350GB /var

#### 1.2. Configuración de red.

Si queremos acceder a la configuración de la red, nos dirigimos a la ruta de los archivos de configuración escribiendo “cd /etc/netplan”. Para modificar el archivo de red usaremos en comando “sudo nano 00-installer-config.yaml”.

```
miadmin@bnr-uslimpia:~$ sudo nano /etc/netplan/00-installer-config.yaml
```

Aparecerá esta pantalla con el contenido de ese archivo. Si queremos cambiar la IP de nuestra maquina basta con editar la que se muestra en el archivo (es muy importante respetar la tabulación y el espaciado de este archivo). Para aplicar los cambios usamos el comando “sudo netplan apply” y se cambiará la dirección IP por la nueva.

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      addresses:
        - 192.168.3.107/24
      dhcp4: false
      routes:
        - to: default
          via: 192.168.3.1
      nameservers:
        addresses:
          - 8.8.8.8
          - 8.8.4.4
  version: 2
```

```
miadmin@bnr-uslimpia:~$ sudo netplan apply
```

### 1.3. Nombre de la máquina

Vamos al fichero “/etc/hostname” y ponemos el nombre de la máquina que queramos.

```
miadmin@bnr-uslimpia:~$ sudo nano /etc/hostname
```

```
GNU nano 6.2  
bnr-used_
```

Ahora debemos ir al fichero “/etc/hosts” y también cambiar el nombre de la máquina.

```
miadmin@bnr-uslimpia:~$ sudo nano /etc/hosts
```

```
miadmin@bnr-uslimpia: ~
```

```
GNU nano 6.2  
127.0.0.1 localhost  
127.0.1.1 bnr-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
```

Ahora reiniciamos la máquina.

```
miadmin@bnr-uslimpia:~$ sudo reboot_
```

#### 1.4. Actualización del sistema.

Si tenemos conexión a internet ejecutamos los comandos “sudo apt update” y “sudo apt upgrade” para actualizar los repositorios y librerías.

```
miadmin@bnr-used:~$ sudo apt update
```

```
miadmin@bnr-used:~$ sudo apt upgrade
```

#### 1.5. Configuración zona horaria.

Para configurar la fecha y la hora antes debemos conocer cuál es la fecha y hora activa en nuestro server, para ello escribimos “date” y para cambiar a la franja horaria de Madrid usaremos el comando “sudo timedatectl set-timezone Europe/Madrid”

```
miadmin@bnr-used:~$ date  
jue 21 mar 2024 09:01:47 CET
```

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

## 1.6. Cortafuegos local.

En primer lugar, tenemos que saber cuál es el estado del cortafuegos con el comando “sudo ufw status”, si esta inactivo lo activamos con “sudo ufw enable”.

```
miadmin@bnr-used:~$ sudo ufw status
Status: inactive
```

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

Cuando tenemos el cortafuegos activo abrimos el puerto 22 para la conexión ssh con “sudo ufw allow 22” y comprobamos que está abierto con “sudo ufw status”.

```
miadmin@bnr-used:~$ sudo ufw allow 22
```

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

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

## 2. Cuentas de administración.

Para la creación de un usuario nuevo usamos el comando “sudo adduser miadmin2”, rellenamos los campos que queramos yo solo he rellenado el de la contraseña para ponerle la contraseña “paso”.

```
miadmin@bnr-used:~$ sudo adduser miadmin2
Adding user `miadmin2' ...
Adding new group `miadmin2' (1001) ...
Adding new user `miadmin2' (1001) with group `miadmin2' ...
Creating home directory `/home/miadmin2' ...
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
```

Añadimos el nuevo usuario al grupo de super usuarios (es decir, el grupo de los admins o root) escribiremos el comando “sudo usermod -aG sudo miadmin2”.

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

Para comprobar que usuarios se encuentran en el grupo sudo ejecutaremos el comando “cat /etc/group | grep sudo”.

```
miadmin@bnr-used:~$ cat /etc/group | grep sudo
sudo:x:27:miadmin,miadmin2
```

### 3. Apache.

#### 3.1. ¿Qué es y para qué sirve?

Apache es un servidor web HTTP de código abierto que se usa para poder desarrollar y acceder a aplicaciones web con modelos cliente/servidor a través de los protocolos HTTP y HTTPS. La funcionalidad principal de este servicio web es abastecer a los usuarios de todos los ficheros necesarios para visualizar una web. Las solicitudes de los usuarios se hacen normalmente mediante un navegador (Chrome, Firefox, Safari, etc.).

#### 3.2. Estructura de directorios/archivos de Apache 2.

#### 3.3. Comandos de control del servicio

- `sudo apache2ctl -S`: muestra los sitios activos.
- `sudo apache2ctl -M`: muestra los módulos activos.
- `sudo apache2ctl {start|stop|restart|gracefullconfigtest}`: en orden arranca, para, reinicia, reinicia de forma breve y valida los ficheros de configuración (este último comando, te muestra los posibles errores o warning que has podido cometer en algún fichero de configuración).
- `a2ensite nombre-del-sitio`: activa el sitio virtual que se le especifica.
- `a2dissite nombre-del-sitio`: desactiva el sitio virtual que se le especifica.
- `a2enmod nombre-del-modulo`: activa el módulo que se le especifica.
- `a2dismod nombre-del-modulo`: desactiva el módulo que se le especifica.
- `a2enconf mi nombre-del-archivo`: habilita archivos de configuración.



### 3.4. Instalación y configuración

En la maquina RSP-USED instalaremos el servicio Apache2, pero antes de eso tenemos que actualizar los repositorios de Linux con el comando “sudo apt-get update”.

Para instalar Apache escribiremos “sudo apt-get install apache2”:

```
miadmin@bnr-used:~$ sudo apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
```

Una vez instalado Apache, tenemos que permitir el acceso a Apache al puerto 80 del cortafuegos de nuestro server mediante el comando “sudo ufw allow apache” y para comprobarlo escribimos “sudo ufw status”.

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

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

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

Al comprobar nos saldrá este warning.

```
miadmin@bnr-used:~$ apache2ctl configtest
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 message
Syntax OK
```

Para solucionarlo debemos entrar en el fichero de configuración global “/etc/apache2/apach2.conf” y poner esta directiva y al volver a comprobarlo no nos dará el warning.

```
# vim: syntax=apache2 conf=on
```

```
ServerName bnr-used
```

```
■
```

```
miadmin@bnr-used:~$ sudo apache2ctl configtest
Syntax OK
```

Para instalar un navegador de texto usamos el comando “sudo apt install lynx” y para activar el módulo de apache status usamos “sudo a2enmod status” y reiniciamos apache2.

```
miadmin@bmr-used:~$ sudo apt install lynx
```

```
miadmin@bmr-used:~$ sudo a2enmod status
Module status already enabled
```

- El módulo mod\_status sirve para saber el rendimiento de nuestro servidor Apache de forma remota a través de una página HTML que accedemos desde el navegador.

Para ver los informes escribimos en la terminal “apache2ctl fullstatus”

```
miadmin@bmr-used:~$ sudo apache2ctl fullstatus
Apache Server Status for localhost (via 127.0.0.1)
```

### 3.5. Configuración de Logs de errores y acceso

Modificamos con “sudo nano /etc/apache2/sites-available/000-default.conf” el fichero de configuración para grabar los registros logs en un directorio específico. Para ello escribimos estas 2 líneas:

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

- La línea ErrorLog /var/www/html/log/error.log sirve para generar un archivo (error.log) con registros logs de los errores en la ruta que se especifica (en /var/www/html/log).
- La línea CustomLog /var/www/html/log/access.log combined sirve para generar una serie de registros de solicitudes o acceso a archivos en un servidor web.

Después de esto creamos el directorio para almacenar los logs que hemos especificado, para ello usamos “sudo mkdir /var/www/html/log” y reiniciamos el servicio de apache.

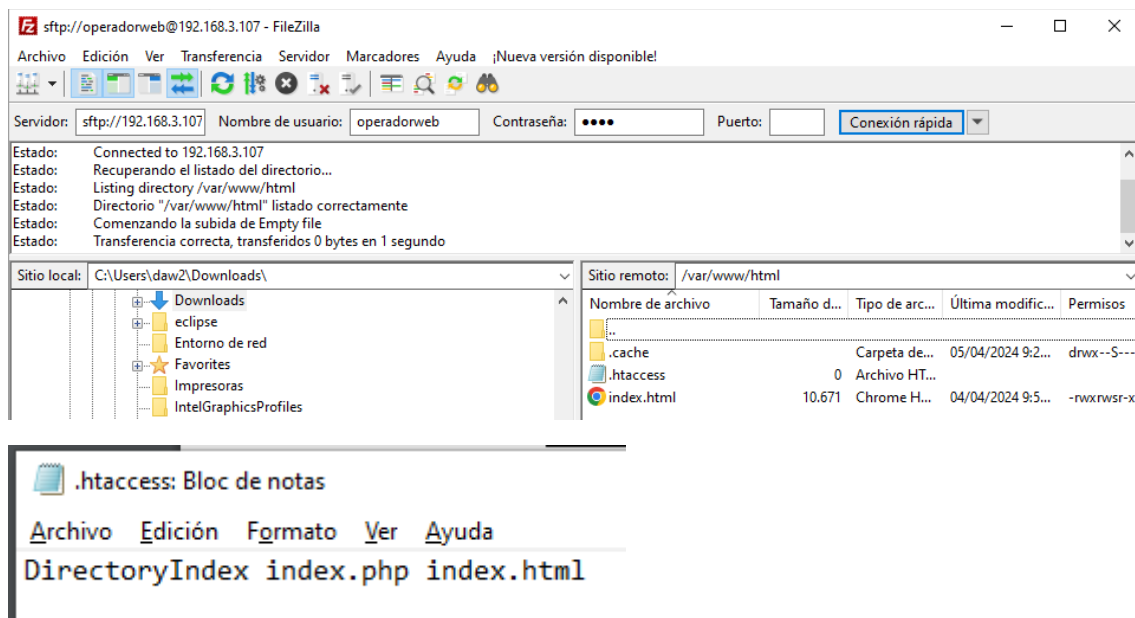
### 3.6. Habilitar el uso de .htaccess

Configuraremos el archivo de configuración de apache con “sudo nano /etc/apache2/apache2.conf” y escribimos lo siguiente:

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

- Options -Indexes +FollowSymLinks: Define las opciones para el directorio. Significa que se desactiva la función de mostrar automáticamente el listado de archivos si no hay un archivo de índice en el directorio (-Indexes) y se permite el seguimiento de enlaces simbólicos (+FollowSymLinks)
- AllowOverride All: Permite la anulación de configuraciones mediante archivos .htaccess en ese directorio. Esto permite a los usuarios locales del directorio personalizar algunas configuraciones para sus propios fines.
- Require all granted: Esto establece que se permite el acceso a este directorio a cualquier usuario o cliente que intente acceder.

Ahora creo el htaccess desde filezilla




La directiva “DirectoryIndex index.php index.html” indica el nombre del archivo que se debe servir como página de inicio cuando se accede a un directorio en el servidor web, tiene preferencia el primer archivo (index.php) y en el caso de que no se encuentre el fichero, se buscara el segundo nombre (index.html) y así consecutivamente.

Para validar los archivos de configuración de Apache escribimos el comando “sudo apache2ctl configtest”.

```
miadmin@bnr-used:~$ sudo apache2ctl configtest
Syntax OK
```

Para la personalización de errores podemos editar el fichero .htaccess (con “sudo nano /var/www/html/.htaccess”)

 .htaccess: Bloc de notas

Archivo Edición Formato Ver Ayuda

DirectoryIndex index.php index.html

ErrorDocument 404 "lo siento, esta página no esta disponible"

- La directiva “ErrorDocument 404 “lo siento...”” sirve para personalizar los mensajes de error. En primer lugar, se especifica el código de error HTTP que queremos personalizar (404) y a continuación escribimos el mensaje de error que queremos que aparezca en el navegador entre comillas (“lo siento, esta página no está disponible”).

### 3.7. Modulo Alias

Instalaremos un módulo de apache llamado Alias que nos permite el acceso que a directorios que están por encima del host del usuario. En primer lugar, comprobamos que el módulo Alias esté activo con “ls /etc/apache2/mods-enabled/ | grep alias”.

```
miadmin@bnr-used:~$ ls /etc/apache2/mods-enabled/ | grep alias
alias.conf
alias.load
```

Si está activo, modificamos el fichero de configuración del sitio virtual (“sudo nano /etc/apache2/sites-enabled/000-default.conf”) escribiendo las siguientes líneas:

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




- La directiva “Alias /documentos /home/miadmin/doc” vincula una carpeta existente en nuestro proyecto (en nuestro caso /documentos) con un directorio virtual o externo al proyecto especificando su ruta absoluta (/home/miadmin/doc).

Ahora cambiamos los permisos y comprobamos.

```
miadmin@bnr-used:~$ sudo chmod 775 /home/miadmin
```

← → ↻ ⚠ No es seguro 192.168.3.107/errors/

## Index of /errors

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 <a href="#">Parent Directory</a>		-	
 <a href="#">file1.txt</a>	2024-04-11 09:07	0	
 <a href="#">file2.txt</a>	2024-04-11 09:08	0	

Apache/2.4.52 (Ubuntu) Server at 192.168.3.107 Port 80

### 3.8. Modulo Rewrite

El módulo Rewrite de Apache nos permite crear direcciones URL alternativas a las generadas por nuestro sitio web. Habilitamos el módulo con “sudo a2enmod rewrite” y reiniciamos el servicio

```
miadmin@bnr-used:~$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
    systemctl restart apache2
miadmin@bnr-used:~$ systemctl restart apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to restart 'apache2.service'.
Multiple identities can be used for authentication:
 1. miadmin
 2. ,,, (miadmin2)
Choose identity to authenticate as (1-2): 1
Password:
==== AUTHENTICATION COMPLETE ====
```

Como ejemplo, editaremos un archivo .htaccess y le escribiremos la siguiente linea:

```
Redirect 301 /educa https://www.educa.jcyl.es/es
```

- Esta directiva redirige la primera ruta (/educa) automáticamente a la segunda (https://www.educa.es) como un enlace.

A continuación, escribimos en el navegador la URL de nuestro proyecto seguido de /educa y cuando carguemos la página, nos redimirá automáticamente a la página web de educa.



### 3.9. HTTPS

Generamos la clave privada y el certificado ya firmado directamente.

[illegible]

A continuación, pasaremos a configurar apache para poder usar la clave privada y el certificado que hemos generado. Para ello, en primer lugar, activamos el módulo SSL con el comando “sudo a2enmod ssl”.

```
--
miadmin@bmr-used:~$ 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
miadmin@bmr-used:~$ systemctl restart apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to restart 'apache2.service'.
Multiple identities can be used for authentication:
 1. miadmin
 2. ,, (miadmin2)
Choose identity to authenticate as (1-2): 1
Password:
==== AUTHENTICATION COMPLETE ====
```

Para la clave privada cambiamos el propietario para que pertenezca al grupo root:ssl-cert con “sudo chown root:ssl-cert /etc/ssl/private/claveprivada.key”, cambiamos los permisos para que tenga 640 con “sudo chmod 640 /etc/ssl/private/claveprivada.key” y comprobamos listando el contenido de ese directorio.

```
miadmin@bnr-used:~$ sudo chown root:ssl-cert /etc/ssl/private/claveprivada.key
miadmin@bnr-used:~$ sudo chmod 640 /etc/ssl/private/claveprivada.key
```

Para el certificado cambiamos el propietario para que pertenezca a root:root con “sudo chown root:root /etc/ssl/certs/certificado.crt” y comprobamos listando el contenido de ese directorio.

```
miadmin@bnr-used:~$ sudo chown root:root /etc/ssl/certs/certificado.crt
miadmin@bnr-used:~$ sudo ls -l /etc/ssl/certs/certificado.crt
-rw-r--r-- 1 root root 1383 abr 11 10:03 /etc/ssl/certs/certificado.crt
```

A continuación, vamos a /etc/apache2/sites-available y copiamos el sitio “default-ssl.conf” creando uno nuevo llamado “rebeca-ssl.conf” con el comando “sudo cp default-ssl.conf borja-ssl.conf”.

```
miadmin@bnr-used:/etc/apache2/sites-available$ sudo cp default-ssl.conf borja-ssl.conf
[sudo] password for miadmin:
miadmin@bnr-used:/etc/apache2/sites-available$ ls
000-default.conf  borja-ssl.conf  default-ssl.conf
```



Editamos “rebeca-ssl.conf”: insertamos SSL Engine on, modificamos las líneas SSLCertificateFile y SSLCertificateKeyFile especificando la ruta donde se encuentran el certificado y la clave privada respectivamente y añadimos un dominio que tenga de puerto el 443 (HTTPS).

```
<VirtualHost *:443>
    ServerAdmin webmaster@localhost
    ServerName borja.local
    ServerAlias www.borja.local
    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-ssl.log
    CustomLog ${APACHE_LOG_DIR}/access-ssl.log combined
    ErrorLog /var/www/html/log/error-ssl.log
    CustomLog /var/www/html/log/access-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/certificado.crt
    SSLCertificateKeyFile /etc/ssl/private/claveprivada.key
```

Guardamos el documento y comprobamos que no hay errores con “sudo apache2ctl configtest”.

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

Si todo está bien activamos el sitio con “sudo a2ensite borja-ssl.conf” y reiniciamos apache.

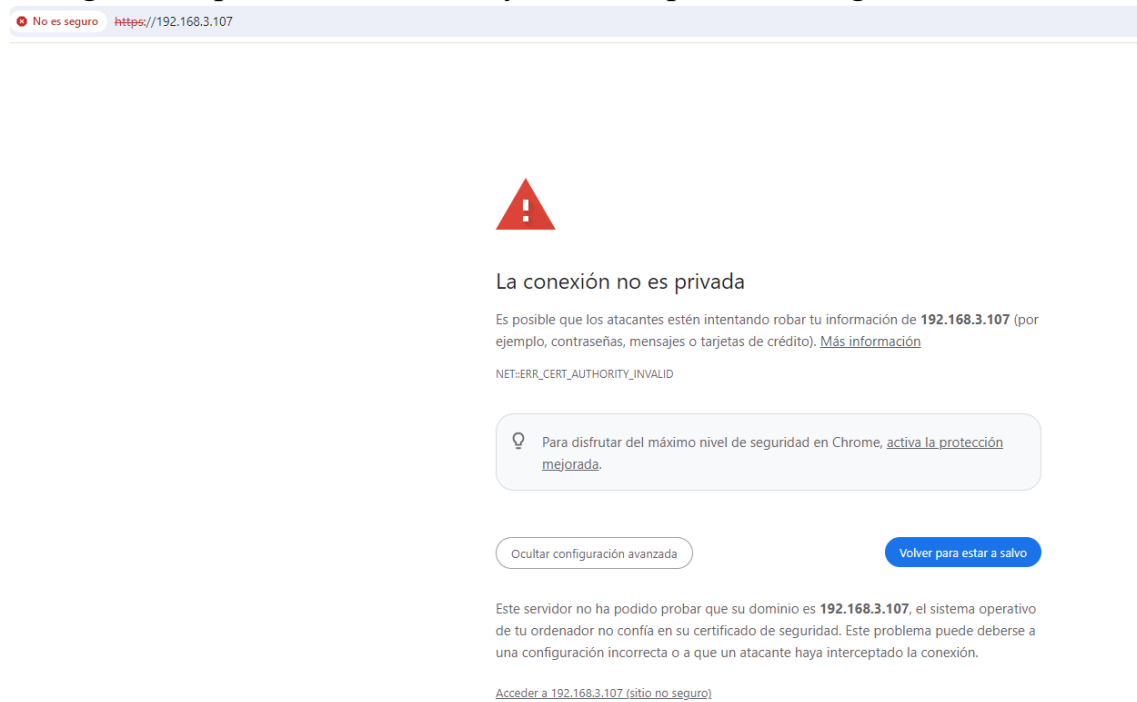
```
miadmin@bnr-used:/etc/apache2/sites-available$ sudo a2ensite borja-ssl.conf
Enabling site borja-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
miadmin@bnr-used:/etc/apache2/sites-available$ systemctl reload apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units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

Abrimos el puerto 443 para poder trabajar con https con “sudo ufw allow 443” y comprobamos que este abierto con “sudo ufw status”.

```
miadmin@bnr-used:/etc/apache2/sites-available$ sudo ufw allow 443
Rule added
Rule added (v6)
miadmin@bnr-used:/etc/apache2/sites-available$ sudo ufw status
Status: active
```

To	Action	From
--	-----	----
22	ALLOW	Anywhere
Apache	ALLOW	Anywhere
443	ALLOW	Anywhere
22 (v6)	ALLOW	Anywhere (v6)
Apache (v6)	ALLOW	Anywhere (v6)
443 (v6)	ALLOW	Anywhere (v6)

Para comprobar que hemos activado el certificado escribimos en el navegador `https://192.168.3.107` y debería aparecer lo siguiente:



Modificamos el fichero `000-default` y escribimos estas líneas:

```
ServerName borja.local
ServerAlias www.borja.local
```

Nos vamos a `/var/www/html/.htaccess` y escribimos las siguientes líneas en el archivo para que cuando escribamos el dominio en el navegador nos redireccione automáticamente a `https`.

```
RewriteEngine On
RewriteCond %{SERVER_PORT} 80
RewriteRule ^(.*)$ https://borja.local/$1 [R,L]
```

Guardamos el fichero y comprobamos que el módulo `rewrite` este activo con “`sudo a2enmod rewrite`” y reiniciamos `apache` y hacemos un `configtest` para ver que todo ha ido bien.

```
miadmin@bnr-used:/etc/apache2/sites-available$ sudo a2enmod rewrite
Module rewrite already enabled
miadmin@bnr-used:/etc/apache2/sites-available$ sudo apache2ctl configtest
Syntax OK
```

## 4. PHP

### 4.1. Instalación

Antes de instalar php tendremos que actualizar el sistema con “sudo apt update” y para instalarlo escribimos “sudo apt -y install php8.1” (instalamos la versión 8.1 porque es la más nueva que es estable) y para comprobar la versión podemos utilizar “sudo php -v”.

```
miadmin@bmr-used:~$ sudo apt -y install php8.1
```

```
miadmin@bmr-used:~$ sudo php -v
PHP 8.1.2-1ubuntu2.15 (cli) (built: Feb 23 2024 17:26:53) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.15, Copyright (c), by Zend Technologies
```

Instalamos la librería de módulos de apache con “sudo apt install libapache2-mod-php” y para crear un archivo info.php usamos filezilla.

```
miadmin@bmr-used:~$ sudo apt install libapache2-mod-php
```


..					
.cache		Carpeta de...	05/04/2024 9:2...	drwx--S---	
documentos		Carpeta de...	09/04/2024 9:1...	drwxr-sr-x	
log		Carpeta de...	12/04/2024 9:0...	drwxr-sr-x	
.htaccess	247	Archivo HT...	12/04/2024 9:2...	-rw-r--r--	
index.html	10.671	Chrome H...	04/04/2024 9:5...	-rwxrwsr-x	
info.php	0	Archivo PHP	16/04/2024 8:5...	-rw-r--r--	

```
<?php
phpinfo();
?>
```

- `phpinfo()` es una función que muestra información detallada sobre la configuración de PHP y el entorno del servidor web.

A continuación, reiniciamos el servicio de apache con “sudo service apache2 restart” y para comprobar que todo está bien escribimos en nuestro navegador 192.168.3.204/info.php y si aparece esta pantalla significa que se ha creado correctamente:

No es segurohttps://192.168.3.107/info.php

PHP Version 8.1.2-1ubuntu2.15

System	Linux bnr-used 5.15.0-102-generic #112-Ubuntu SMP Tue Mar 5 16:50:32 UTC 2024 x86_64
Build Date	Feb 23 2024 17:26:53
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902.NTS
PHP Extension Build	API20210902.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:  
Zend Engine v4.1.2, Copyright (c) Zend Technologies  
with Zend OPcache v8.1.2-1ubuntu2.15, Copyright (c), by Zend Technologies

zendengine

Instalamos los módulos de SOAP y de xml con el comando “sudo apt install -y php8.1-xml php8.1-soap” y reiniciamos el servicio de apache2. En 192.168.3.204/info.php podemos comprobar que se han instalado correctamente.

```
miadmin@bnr-used:~$ sudo apt install -y php8.1-xml php8.1-soap
```

SOAP	Brad Lafountain, Shane Caraveo, Dmitry Stogov
------	---

Ahora cambiaremos el fichero de configuración php.ini pero antes de esto haremos una copia de seguridad. Cambiamos de directorio a “cd /etc/php/8.1/apache2” y usamos el comando “sudo cp php.ini php.ini.backup” para hacer la copia.

```
miadmin@bnr-used:/etc/php/8.1/apache2$ sudo cp php.ini php.ini.backup
miadmin@bnr-used:/etc/php/8.1/apache2$ ls
conf.d  php.ini  php.ini.backup
```

Usamos el comando “sudo nano php.ini” para modificar el fichero de configuración de php. Buscamos en el archivo la línea “display\_errors” y cambiamos la línea de display\_errors=Off y la ponemos en On

```
display_errors = On
```

En ese mismo archivo buscamos la línea “display\_startup\_errors” y lo activamos igual que el anterior.

```
display_startup_errors = On
```

Guardamos el archivo, reiniciamos el servicio de apache y para comprobarlo volvemos al navegador escribiendo 192.168.3.204/info.php y buscamos las configuraciones que acabamos de activar para comprobar que se han cambiado correctamente.

display_errors	On	On
display_startup_errors	On	On

Por último, volvemos a modificar el php.ini y cambiamos los valores de “memory\_limit” para que tenga una memoria de 256M.

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

Como hemos hecho anteriormente hacemos “sudo service apache2 restart” para reiniciar apache y comprobamos que se haya cambiado la memoria en nuestro navegador.

memory_limit	256M	256M
--------------	------	------

## 4.2. Módulos

A continuación, se muestran los módulos instalados de php mostrados por el comando “dpkg --get-selections | grep php8.1”:

```
miadmin@bnr-used:/etc/php/8.1/apache2$ dpkg --get-selections | grep php8.1
libapache2-mod-php8.1          install
php8.1                        install
php8.1-cli                    install
php8.1-common                 install
php8.1-opcache                install
php8.1-readline               install
php8.1-soap                   install
php8.1-xml                    install
```

## 5. XDebug

### 5.1. Instalación

Antes de instalar XDebug comprobaremos que no esté instalado anteriormente con “php -m | grep sdebug”, si la terminal no devuelve nada significa que ese modulo no está instalado. Para instalarlo escribimos “sudo apt install php8.1-xdebug”.

```
miadmin@bnr-used:~$ php -m | grep sdebug
miadmin@bnr-used:~$ sudo apt install php8.1-xdebug
```

A continuación, cambiaremos de directorio a “cd /etc/php/8.1/mods-available/” y modificamos el archivo 20-xdebug.ini con “sudo nano xdebug.ini” y escribimos las siguientes líneas:

```
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
```

Para que se pueda conectar por el puerto 9003 tendremos que abrirlo mediante “sudo ufw allow 9003” y con “sudo ufw status” vemos los puertos que están abiertos

```
miadmin@bnr-used:~$ sudo ufw allow 9003
Rule added
Rule added (v6)
```

Para terminar, reiniciamos el servicio con “sudo service apache2 restart”.

## 6. MySQL

### 6.1. Instalación y configuración

Primero actualizamos las librerías de Linux con “sudo apt update” y “sudo apt upgrade” y después instalamos el servicio de MySQL con “sudo apt -y install mysql-server”. Para visualizar la versión que se ha instalado, usamos el comando “mysql --version”

```
miadmin@bnr-used:~$ sudo apt -y install mysql-server

miadmin@bnr-used:~$ mysql --version
mysql Ver 8.0.36-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
```

Para permitir la conexión desde cualquier IP necesitamos entrar al fichero de configuración de MySQL (“sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf”) y comentar las siguientes líneas: “bind-address = 127.0.0.1” y “mysqlx-bind-address = 127.0.0.1”

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

Guardamos el archivo y reiniciamos mysql con “sudo service mysql restart” y después abrimos el puerto 3306 con “sudo ufw allow 3306”.

```
miadmin@bnr-used:~$ sudo service mysql restart
miadmin@bnr-used:~$ sudo ufw allow 3306
Rule added
Rule added (v6)
miadmin@bnr-used:~$ sudo ufw status
Status: active
```

To	Action	From
--	-----	----
22	ALLOW	Anywhere
Apache	ALLOW	Anywhere
443	ALLOW	Anywhere
9003	ALLOW	Anywhere
3306	ALLOW	Anywhere
22 (v6)	ALLOW	Anywhere (v6)
Apache (v6)	ALLOW	Anywhere (v6)
443 (v6)	ALLOW	Anywhere (v6)
9003 (v6)	ALLOW	Anywhere (v6)
3306 (v6)	ALLOW	Anywhere (v6)



A continuación, ejecutaremos un script para realizar una configuración inicial de mysql con el comando “sudo mysql\_secure\_installation” y responderemos “yes” a todos los pasos excepto el de nivel de seguridad de contraseña que elegiremos el “LOW”.

```
miadmin@bnr-used:~$ sudo mysql_secure_installation
```

En el caso de que no nos permita crear el usuario saltando el error de que la contraseña no cumple con los requisitos, tendremos que cambiar las variables globales de mysql para admitir contraseñas con 4 caracteres como mínimo. Para ello dentro de la consola de mysql escribiremos “show variables like ‘validate\_password%’” y se mostrara la tabla de las variables que controlan los requisitos de las contraseñas de los usuarios.

```
mysql> SHOW VARIABLES LIKE 'validate_password%';
```

Variable_name	Value
validate_password.changed_characters_percentage	0
validate_password.check_user_name	ON
validate_password.dictionary_file	
validate_password.length	8
validate_password.mixed_case_count	1
validate_password.number_count	1
validate_password.policy	MEDIUM
validate_password.special_char_count	1

```
8 rows in set (0,05 sec)
```

Una vez aquí, cambiaremos el valor de “length” y ”policy” con “set global validate\_password.length=4;” y “set global validate\_password.policy=LOW;”

```
mysql> set global validate_password.length=4;
Query OK, 0 rows affected (0,00 sec)
```

```
mysql> set global validate_password.policy=LOW;
Query OK, 0 rows affected (0,00 sec)
```

```
mysql> show variables like 'validate_password%';
```

Variable_name	Value
validate_password.changed_characters_percentage	0
validate_password.check_user_name	ON
validate_password.dictionary_file	
validate_password.length	4
validate_password.mixed_case_count	1
validate_password.number_count	1
validate_password.policy	LOW
validate_password.special_char_count	1

```
8 rows in set (0,01 sec)
```

## 6.2. PHPMyAdmin

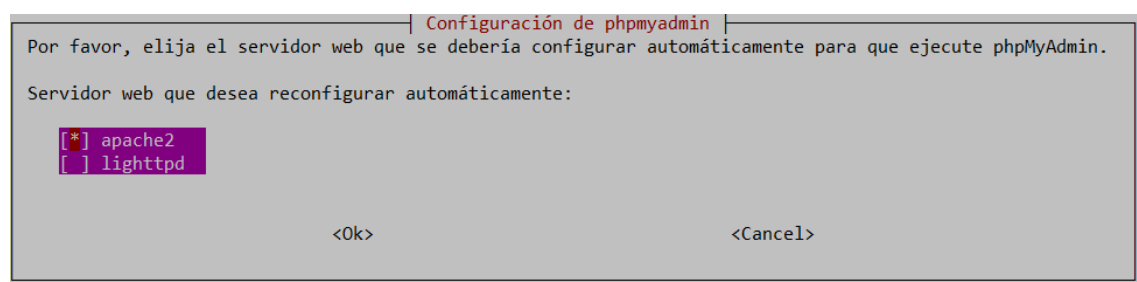
Creamos el usuario “create user ‘adminsql’@’%’ identified by ‘paso’;” y le damos los permisos “grant all privileges on \*.\* to ‘adminsql’@’%’ with grant option;”

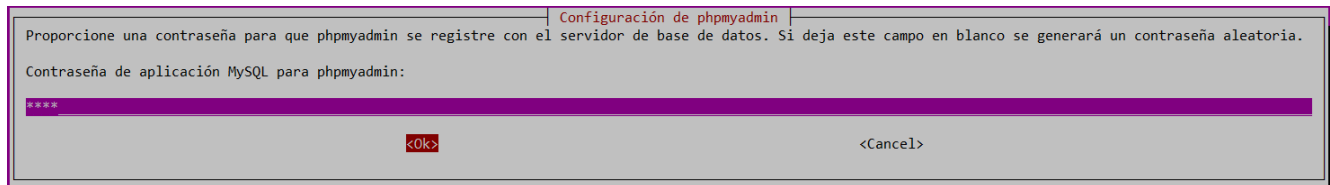
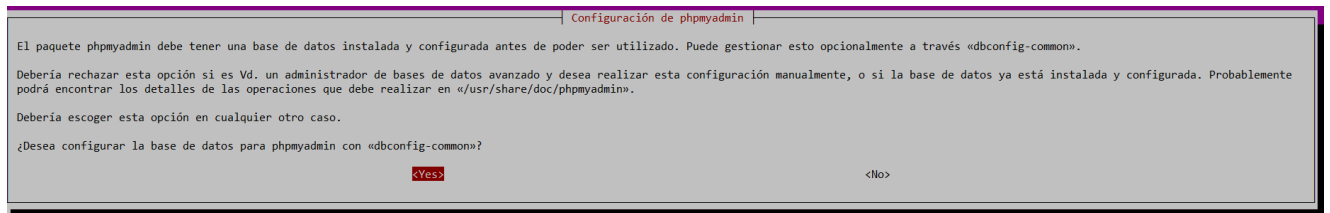
```
mysql> create user 'adminsql'@'%' identified by 'paso';
Query OK, 0 rows affected (0,02 sec)
```

```
mysql> grant all privileges on *.* to 'adminsql'@'%' with grant option;
Query OK, 0 rows affected (0,01 sec)
```

Instalamos PHPMyAdmin y los módulos que necesitamos escribiremos el comando “sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl”. Cuando lo ejecutemos, aparecerá una interfaz del Ubuntu diferente en la que nos harán una serie de preguntas:

```
miadmin@bnr-used:~$ sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```





Una vez terminada la instalación, nos volvemos a conectar a mysql pero con el nuevo usuario “mysql -u adminsql -p”.

```
miadmin@bnr-used:~$ mysql -u adminsql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 33
Server version: 8.0.36-0ubuntu0.22.04.1 (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>

Ejecutamos el comando "uninstall component "file://component\_validate\_password";" después hacer el comando "install component "file://component\_validate\_password";". Para comprobar que todo ha ido bien, escribiremos en el navegador la IP de nuestro servidor seguida de "phpmyadmin" y debería aparecer esta ventana.

```
mysql> uninstall component "file://component_validate_password";  
Query OK, 0 rows affected (0,06 sec)
```

```
mysql> install component "file://component_validate_password";  
Query OK, 0 rows affected (0,01 sec)
```

---



**Bienvenido a phpMyAdmin**

**Idioma - Language**

Español - Spanish ▼

**Iniciar sesión** ⓘ

Usuario:

Contraseña:

**Continuar**

## 8. Cuentas de desarrollo y hosting virtual.

### 8.1. Creación de operadorweb.

Tenemos que crear un usuario que se encargue de conectarse al servidor mediante http/s desde la maquina anfitriona que usaremos de cliente. Para eso usamos el comando “sudo adduser --home /var/www/html --no-create-home --ingroup www-data operadorweb” que creara el usuario operadorweb en el grupo www-data (grupo por defecto de usuarios de apache) y con el home en el directorio html.

```
miadmin@bnr-used:~$ sudo adduser --home /var/www/html --no-create-home --ingroup www-data operadorweb
Adding user `operadorweb' ...
Adding new user `operadorweb' (1002) with group `www-data' ...
Not creating home directory `/var/www/html'.
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for operadorweb
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
```

Para confirmar que se ha creado usamos el comando “id operadorweb” y se mostrará la información sobre ese usuario.

```
miadmin@bnr-used:~$ id operadorweb
uid=1002(operadorweb) gid=33(www-data) groups=33(www-data)
```

A continuación, cambiamos el propietario del directorio /var/www/html para que sea operadorweb con “sudo chown -R operadorweb:www-data /var/www/html” y seguido de esto cambiamos los permisos del archivo que se encuentra en ese directorio para que pueda ser legible, modificado y ejecutado por ese usuario con “sudo chmod -R 2775 /var/www/html”.

```
miadmin@bnr-used:~$ sudo chown -R operadorweb:www-data /var/www/html
miadmin@bnr-used:~$ sudo chmod -R 2775 /var/www/html

miadmin@bnr-used:~$ ll /var/www/html/
total 20
drwxrwsr-x 2 operadorweb www-data 4096 abr  4 09:52 ./
drwxr-xr-x 3 root         root    4096 abr  4 09:52 ../
-rwxrwsr-x 1 operadorweb www-data 10671 abr  4 09:52 index.html*
```

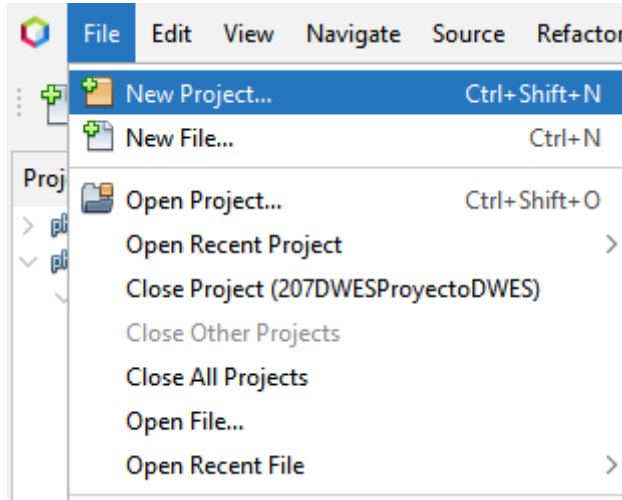
## IS32WX07

### 5. NetBeans.

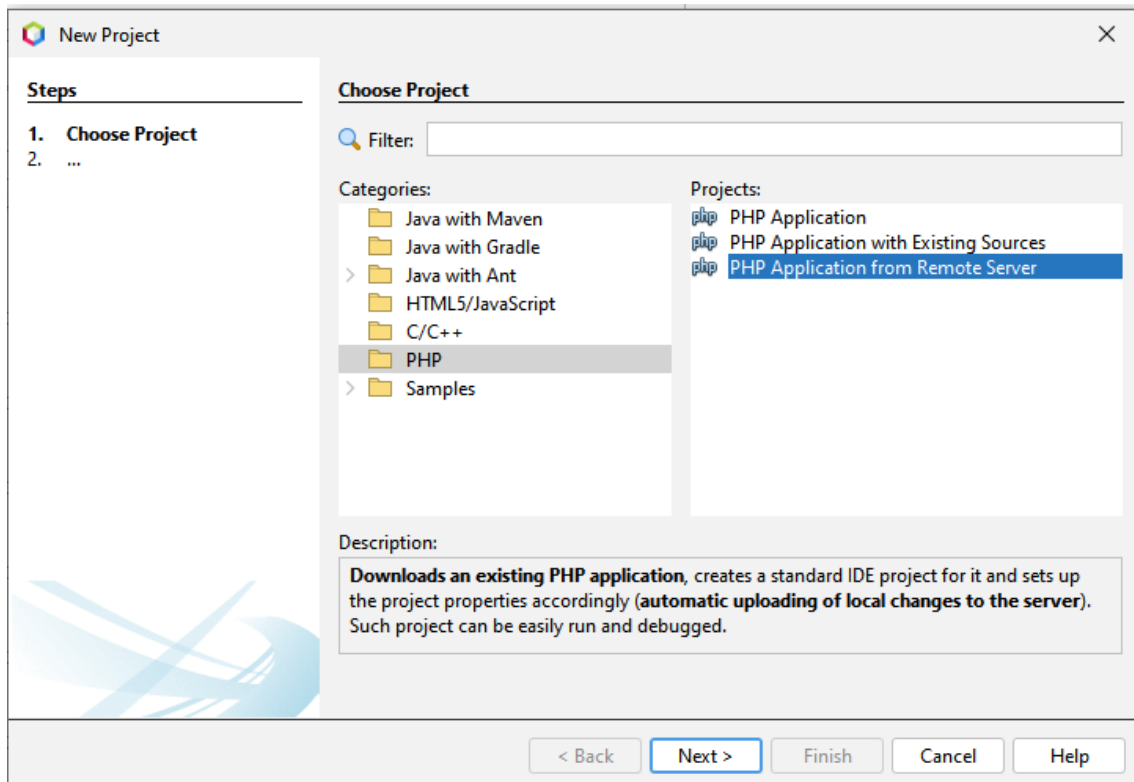
#### 5.2. Creación de proyectos, modificación, borrado, prueba.

##### 5.2.1. Creación.

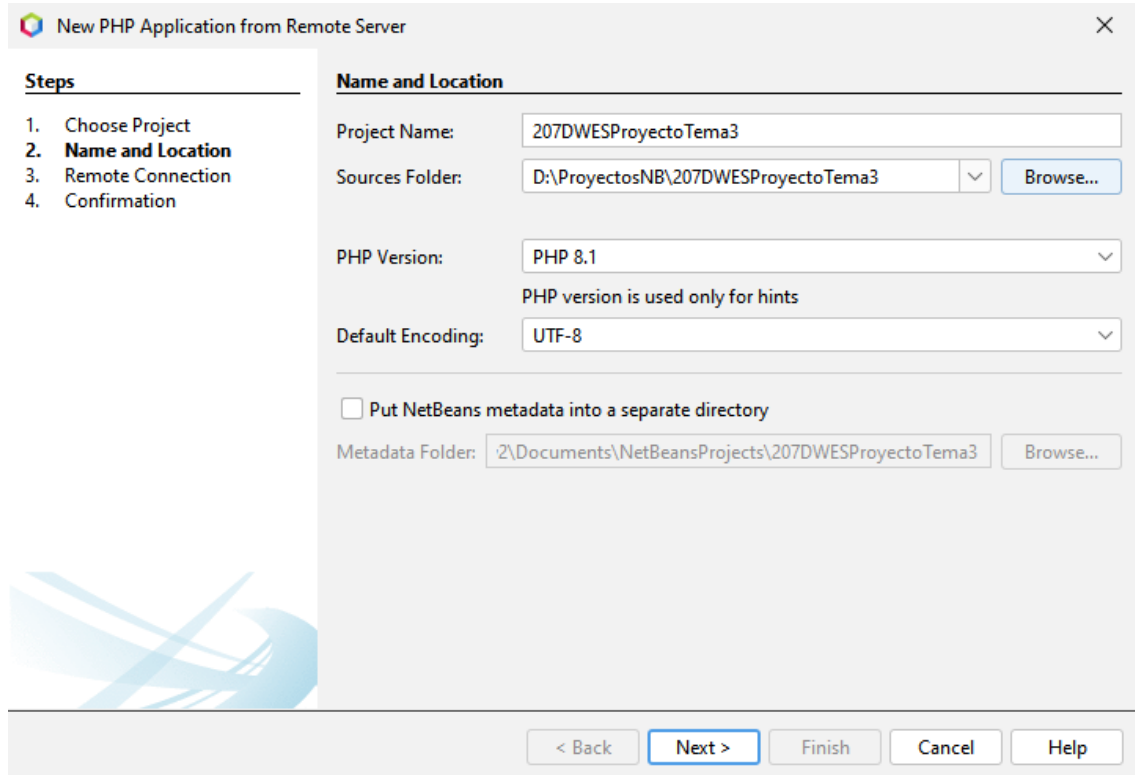
Creamos un nuevo proyecto en NetBeans.



En nuestro caso queremos que el proyecto sea remoto y utilizando el lenguaje php.

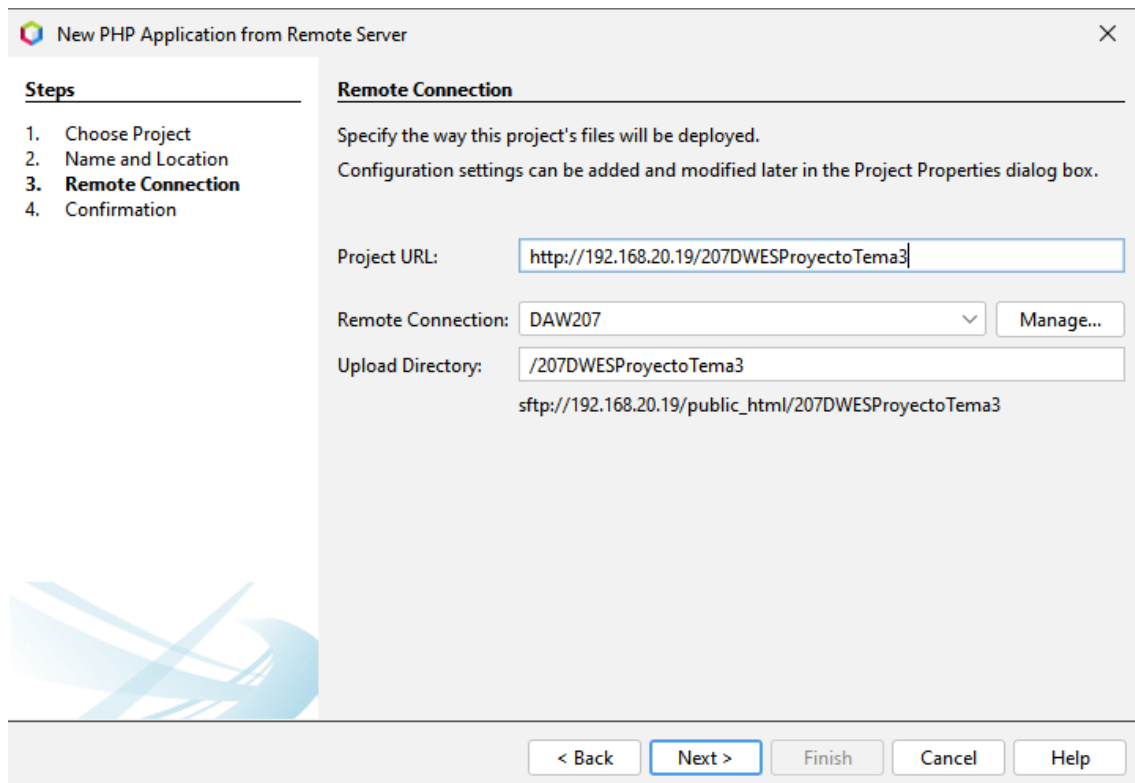


Ponemos el nombre del proyecto, la ubicación del mismo y la versión de php que vamos a utilizar.



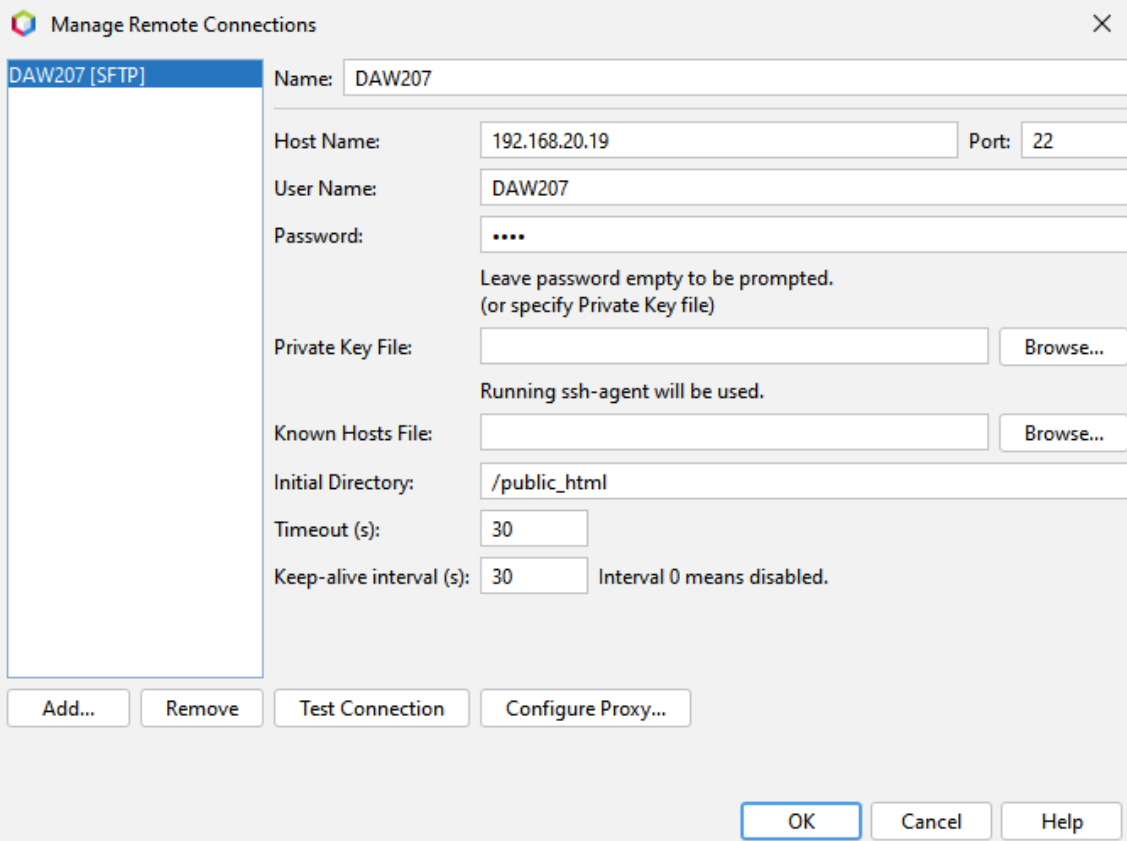
The screenshot shows the 'New PHP Application from Remote Server' dialog box, specifically the 'Name and Location' step. The 'Steps' list on the left indicates the current step is '2. Name and Location'. The 'Project Name' field is filled with '207DWESProyectoTema3'. The 'Sources Folder' is set to 'D:\ProyectosNB\207DWESProyectoTema3' with a 'Browse...' button. The 'PHP Version' is set to 'PHP 8.1' with a dropdown arrow. Below it, a note states 'PHP version is used only for hints'. The 'Default Encoding' is set to 'UTF-8' with a dropdown arrow. There is an unchecked checkbox for 'Put NetBeans metadata into a separate directory'. The 'Metadata Folder' is set to '2\Documents\NetBeansProjects\207DWESProyectoTema3' with a 'Browse...' button. At the bottom, there are navigation buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Configuramos al sitio que queremos que nuestro proyecto remoto se conecte para trabajar en desarrollo.



The screenshot shows the 'New PHP Application from Remote Server' dialog box, specifically the 'Remote Connection' step. The 'Steps' list on the left indicates the current step is '3. Remote Connection'. The 'Project URL' field is filled with 'http://192.168.20.19/207DWESProyectoTema3'. The 'Remote Connection' dropdown is set to 'DAW207' with a 'Manage...' button. The 'Upload Directory' is set to '/207DWESProyectoTema3', and below it, the full path 'sftp://192.168.20.19/public\_html/207DWESProyectoTema3' is displayed. At the bottom, there are navigation buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Aquí se muestran los parámetros que tiene nuestra conexión:



Manage Remote Connections

DAW207 [SFTP]

Name: DAW207

Host Name: 192.168.20.19 Port: 22

User Name: DAW207

Password: ....

Leave password empty to be prompted.  
(or specify Private Key file)

Private Key File: Browse...

Running ssh-agent will be used.

Known Hosts File: Browse...

Initial Directory: /public\_html

Timeout (s): 30

Keep-alive interval (s): 30 Interval 0 means disabled.

Add... Remove Test Connection Configure Proxy...

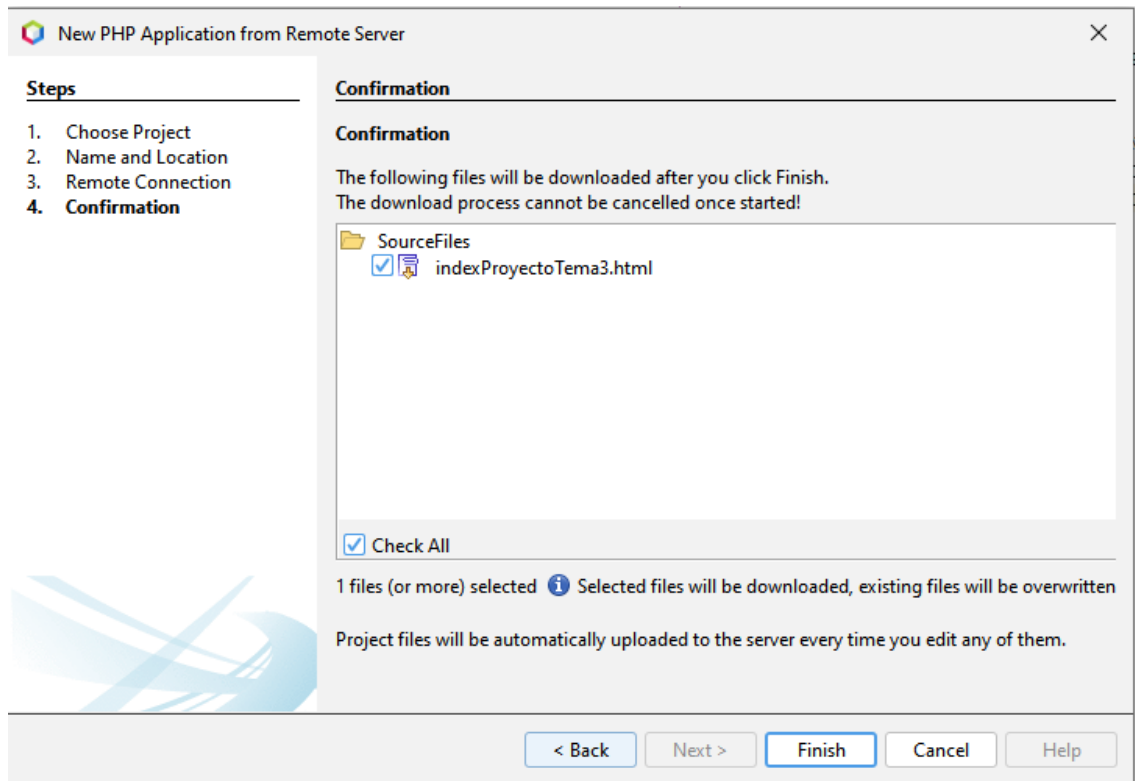
OK Cancel Help

- Name: el nombre personalizado de la conexión
- Host Name: IP del servidor
- Port: 22 (por el que establece la conexión SSH)
- User Name: el nombre del usuario administrador del hosting virtual
- Password: contraseña del usuario
- Initial Directory: directorio padre

Hacemos un “Test Connection” y si todo ha salido bien, seguiremos con la creación del proyecto. En la siguiente ventana se conectará al servidor y solo te podrá dejar crear el proyecto si ese directorio (html/207DAWProyectoDAW) en el servidor contiene algún archivo. Para solucionar esto, podemos crear un fichero auxiliar (index.html o file.txt) desde FileZilla para más tarde eliminarlo

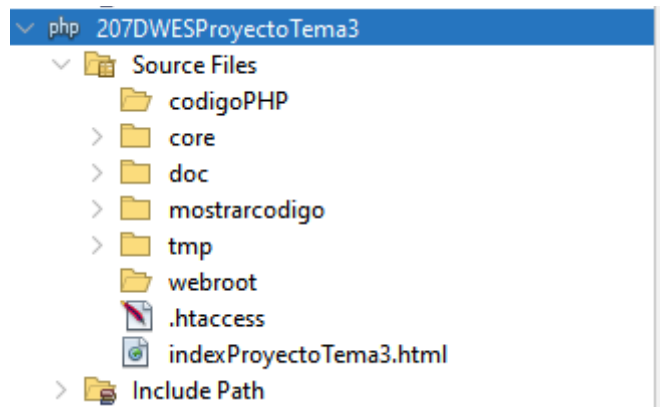


Comprobamos que está bien lo que se encuentra en el archivo que queremos conectarnos y damos en “Finish”.



### 5.2.2. Modificación.

Ahora vamos modificando nuestro proyecto para que se quede igual a los estándares de desarrollo de estructura y almacenamiento de nuestro proyecto.



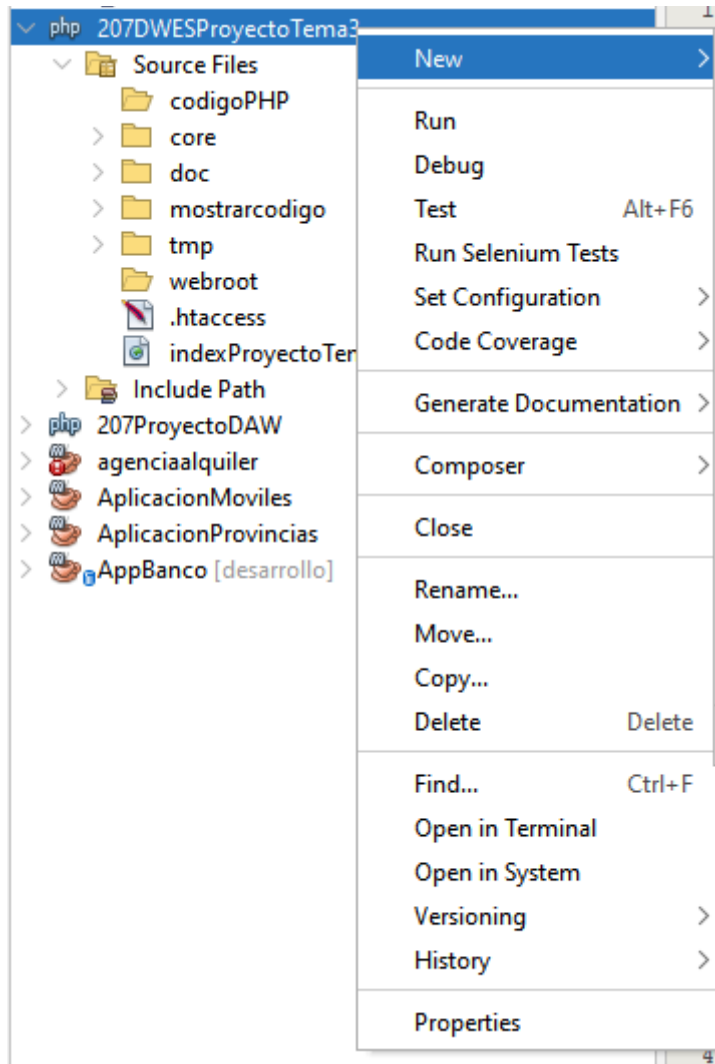
También vamos modificando nuestros proyectos añadiendo archivos y realizando cambios en el código.



```
<td>1</td>
<td>Inicializar variables de los distintos tipos de datos básicos(string, int, float, bool) y mostrar los datos por pantalla (echo, print, printf, print_r,var_dump).</td>
<td><a href="./codigoPHP/ejercicio01.php"></a></td>
<td><a href="./mostrarcodigo/muestraEjercicio01.php"></a></td>
```

### 5.2.3. Borrado

Para borrarlo lo único tendremos que hacer es darle a “delete” en las opciones del proyecto.



### 5.2.4. Prueba

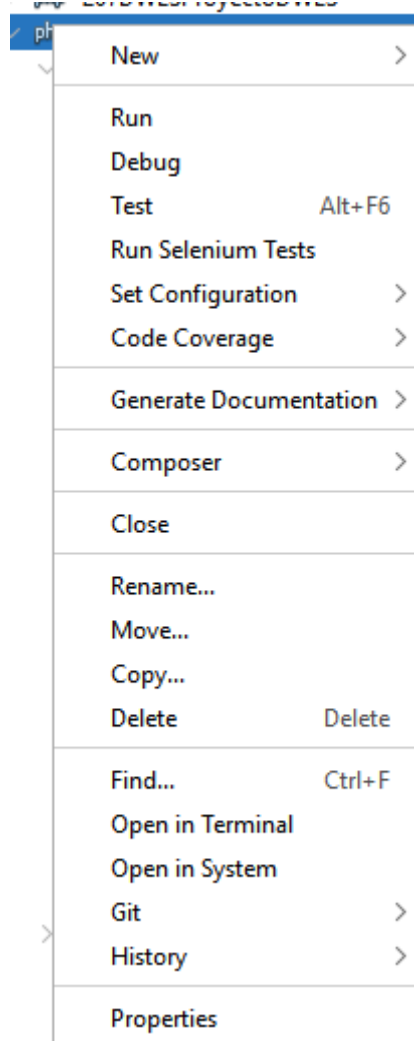
Para probarlo solo nos tenemos que dirigir a nuestro navegador e introducir la dirección de nuestro servidor de desarrollo y ahí podemos probar todos los cambios realizados.

No es seguro daw207.isaues.local/207DWESProyectoTema3/indexProyectoTema3.html

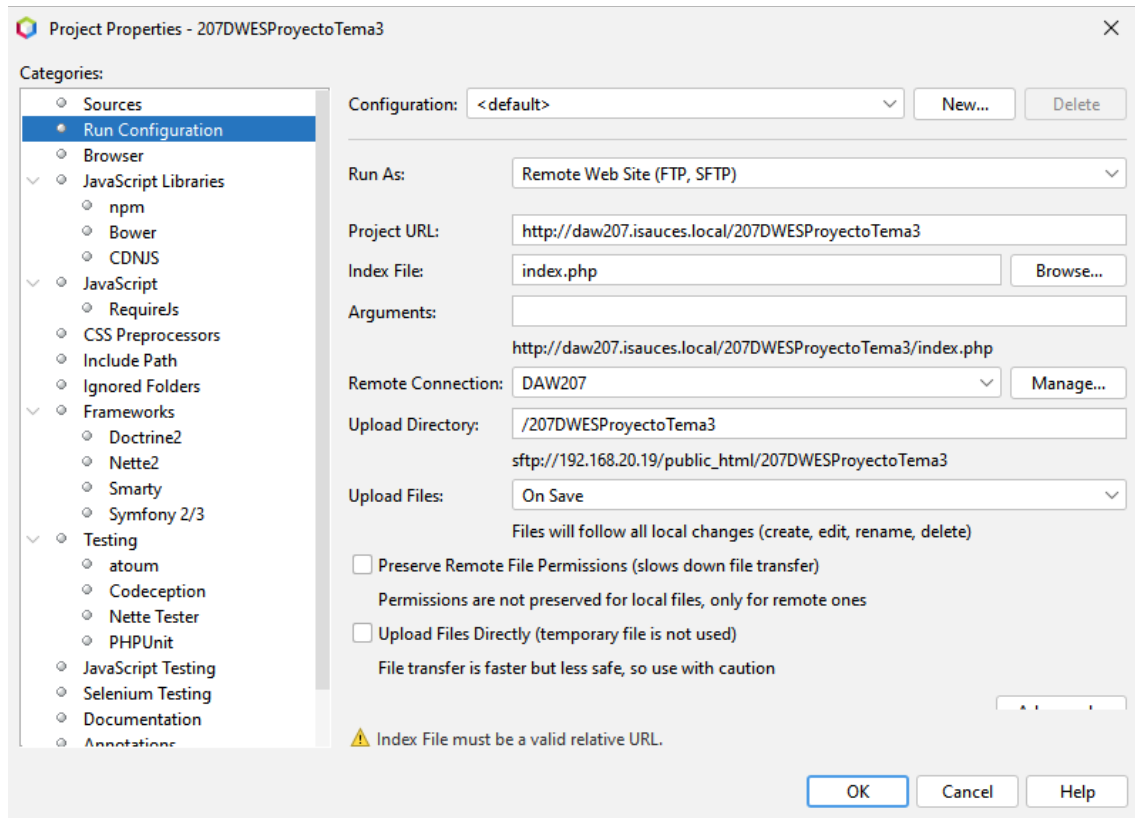
#	Ejercicio	Ejecución	Visualización
0	Hola mundo y phpinfo()		
1	Inicializar variables de los distintos tipos de datos básicos(string, int, float, bool) y mostrar los datos por pantalla (echo, print, printf, print_r, var_dump).		
2	Inicializar y mostrar una variable heredoc.		

### 5.2.5. Configuración.

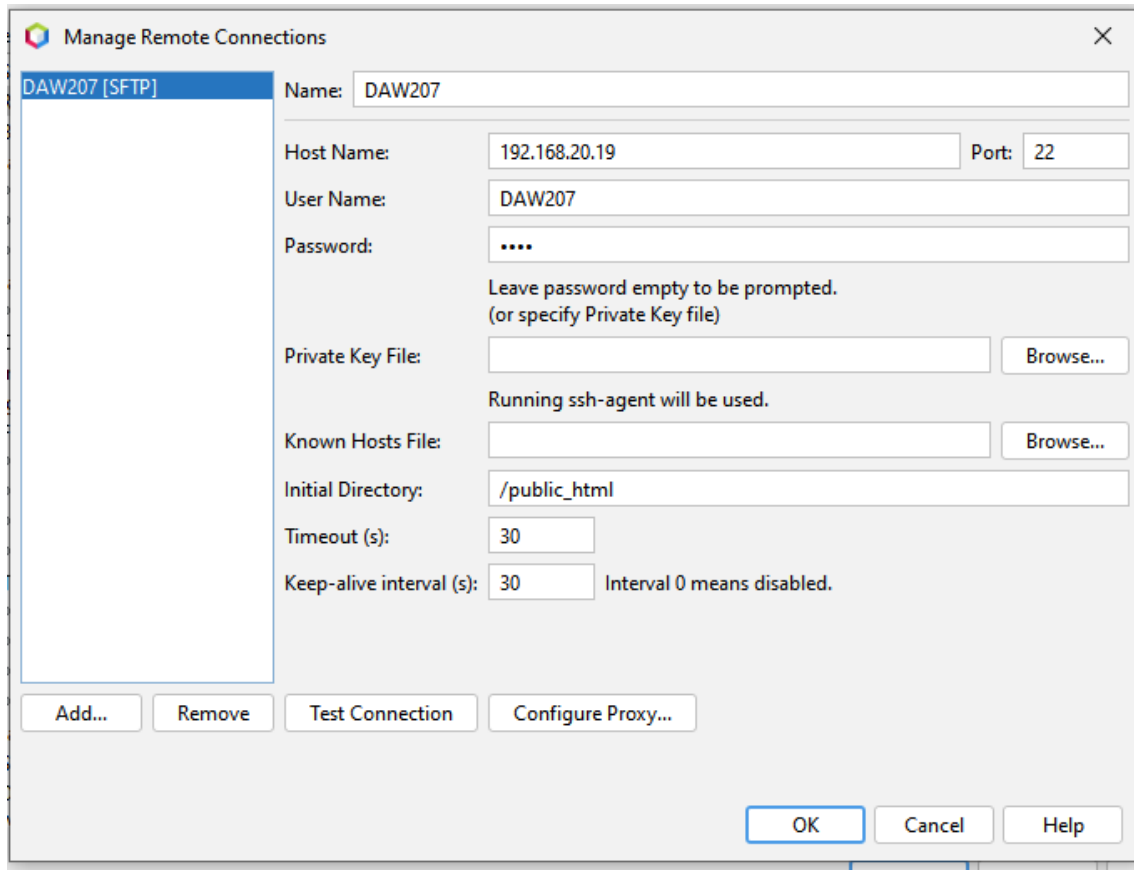
Le damos a “Properties” en el proyecto que queramos configurar



Aquí seleccionamos el enlace donde queremos que se nos abra el archivo que queramos probar y si le damos en “Manage...” iremos a la configuración remota del proyecto



Esta sería una captura de la conexión remota que tiene este proyecto.



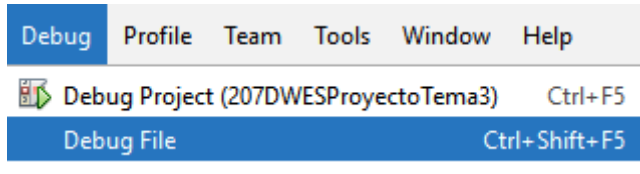
## 5.5. Depuración - Configuración de la ejecución para la depuración.

### 5.5.1. Ejecución de la depuración.

Para empezar pondremos un breakpoint para indicar al debugger donde tiene que parar la ejecución y así a partir de ahí continuar nosotros.

```
26 | echo "<n4>Con echo:</n4>";
    | echo '<p>La variable <span style="color:greenyellow">$cadena</span> es de tipo <span style="color:greenyellow">' . gettype($cadena) .
```

Ahora le daremos a “Debug File” para comenzar a debugar este ejercicio.



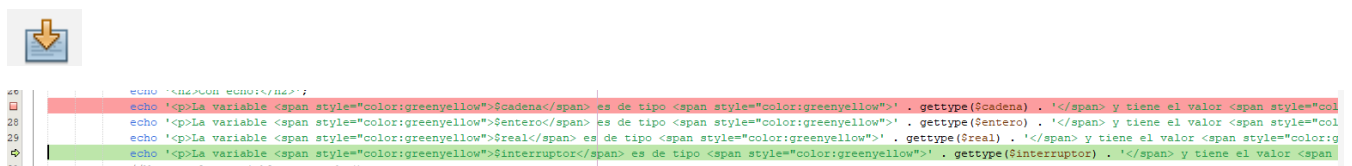
Para comenzar y que llegue al breakpoint le daremos “F5” o en el icono verde que pone continuar.



Una vez haya llegado al breakpoint nos mostrara las variables utilizadas y sus valores correspondientes.



Para una vez aquí ejecutar mas código tenemos varias opciones una de ellas es darle F7 o a “Step into”. Que es un icono con una flecha hacia abajo sobre lo que parece ser un código.



También esta la opción de que la ejecución avance donde esta el cursor del ratón es decir donde clickes con el ratón dentro del código que avance hasta esa línea. Esto se hace dándole F4 o la opción de "Run to cursor".



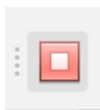
```

echo '<h2>Con echo:</h2>';
echo '<p>La variable <span style="color:greenyellow">$cadena</span> es de tipo <span style="color:greenyellow">' . gettype($cadena) . '</span> y tiene el valor <span style="color:greenyellow">' . $cadena . '</span>';
echo '<p>La variable <span style="color:greenyellow">$entero</span> es de tipo <span style="color:greenyellow">' . gettype($entero) . '</span> y tiene el valor <span style="color:greenyellow">' . $entero . '</span>';
echo '<p>La variable <span style="color:greenyellow">$real</span> es de tipo <span style="color:greenyellow">' . gettype($real) . '</span> y tiene el valor <span style="color:greenyellow">' . $real . '</span>';
echo '<p>La variable <span style="color:greenyellow">$interruptor</span> es de tipo <span style="color:greenyellow">' . gettype($interruptor) . '</span> y tiene el valor <span style="color:greenyellow">' . $interruptor . '</span>';
//Mostrar las variables con echo()
echo ('<h2>Con echo():</h2>');
echo ('<p>La variable <span style="color:greenyellow">$cadena</span> es de tipo <span style="color:greenyellow">' . gettype($cadena) . '</span> y tiene el valor <span style="color:greenyellow">' . $cadena . '</span>');
echo ('<p>La variable <span style="color:greenyellow">$entero</span> es de tipo <span style="color:greenyellow">' . gettype($entero) . '</span> y tiene el valor <span style="color:greenyellow">' . $entero . '</span>');
echo ('<p>La variable <span style="color:greenyellow">$real</span> es de tipo <span style="color:greenyellow">' . gettype($real) . '</span> y tiene el valor <span style="color:greenyellow">' . $real . '</span>');
echo ('<p>La variable <span style="color:greenyellow">$interruptor</span> es de tipo <span style="color:greenyellow">' . gettype($interruptor) . '</span> y tiene el valor <span style="color:greenyellow">' . $interruptor . '</span>');
//Mostrar las variables con print
print '<h2>Con print:</h2>';
print '<p>La variable <span style="color:greenyellow">$cadena</span> es de tipo <span style="color:greenyellow">' . gettype($cadena) . '</span> y tiene el valor <span style="color:greenyellow">' . $cadena . '</span>';
print '<p>La variable <span style="color:greenyellow">$entero</span> es de tipo <span style="color:greenyellow">' . gettype($entero) . '</span> y tiene el valor <span style="color:greenyellow">' . $entero . '</span>';
print '<p>La variable <span style="color:greenyellow">$real</span> es de tipo <span style="color:greenyellow">' . gettype($real) . '</span> y tiene el valor <span style="color:greenyellow">' . $real . '</span>';
print '<p>La variable <span style="color:greenyellow">$interruptor</span> es de tipo <span style="color:greenyellow">' . gettype($interruptor) . '</span> y tiene el valor <span style="color:greenyellow">' . $interruptor . '</span>';
//Mostrar las variables con print()
print('<h2>Con print():</h2>');
print('<p>La variable <span style="color:greenyellow">$cadena</span> es de tipo <span style="color:greenyellow">' . gettype($cadena) . '</span> y tiene el valor <span style="color:greenyellow">' . $cadena . '</span>');
print('<p>La variable <span style="color:greenyellow">$entero</span> es de tipo <span style="color:greenyellow">' . gettype($entero) . '</span> y tiene el valor <span style="color:greenyellow">' . $entero . '</span>');
print('<p>La variable <span style="color:greenyellow">$real</span> es de tipo <span style="color:greenyellow">' . gettype($real) . '</span> y tiene el valor <span style="color:greenyellow">' . $real . '</span>');
print('<p>La variable <span style="color:greenyellow">$interruptor</span> es de tipo <span style="color:greenyellow">' . gettype($interruptor) . '</span> y tiene el valor <span style="color:greenyellow">' . $interruptor . '</span>');

```

Name	Type	Value
Superglobals		
\$cadena	string	"Mi primer String"
\$entero	integer	17
\$interruptor	boolean	true
\$real	float	18.9

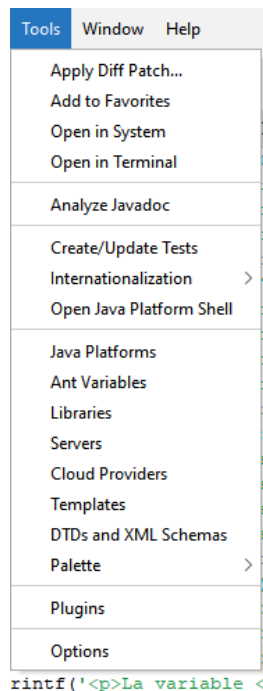
Para detener el debug solo habría que darle al botón que se muestra en la siguiente captura.



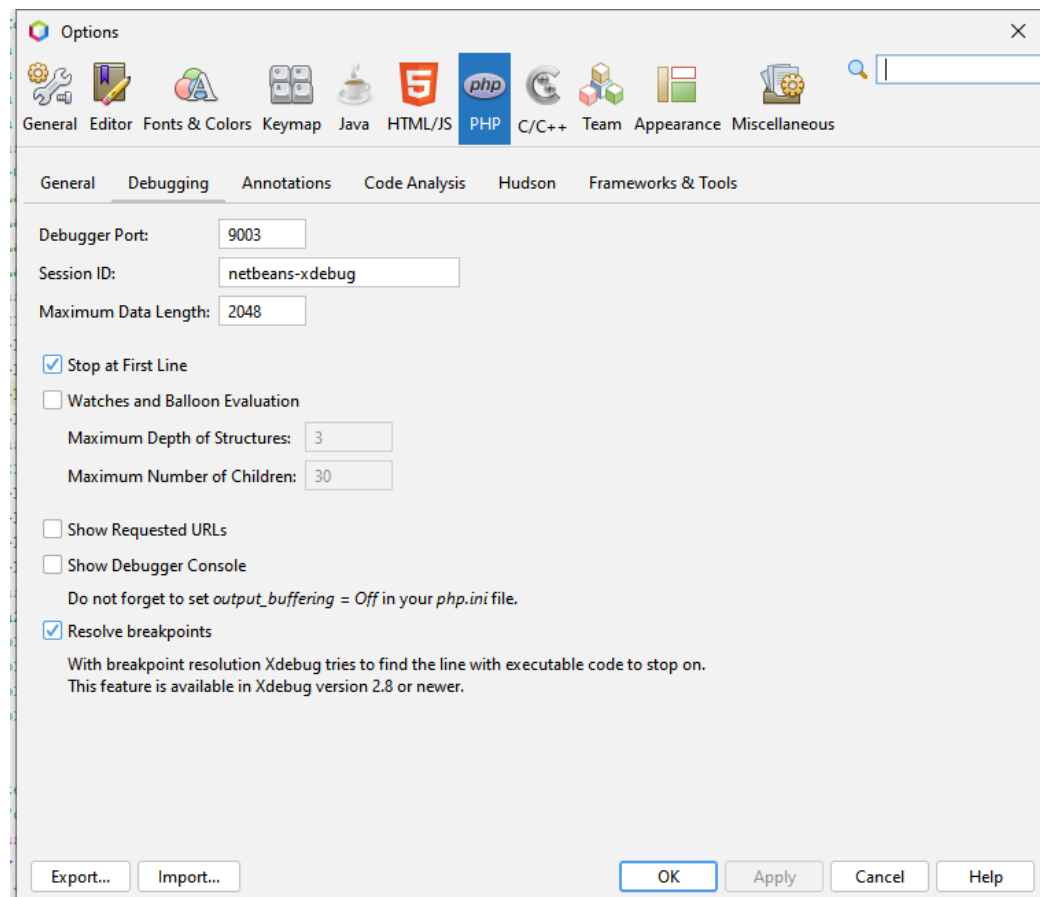


### 5.5.2 Configuración del debug.

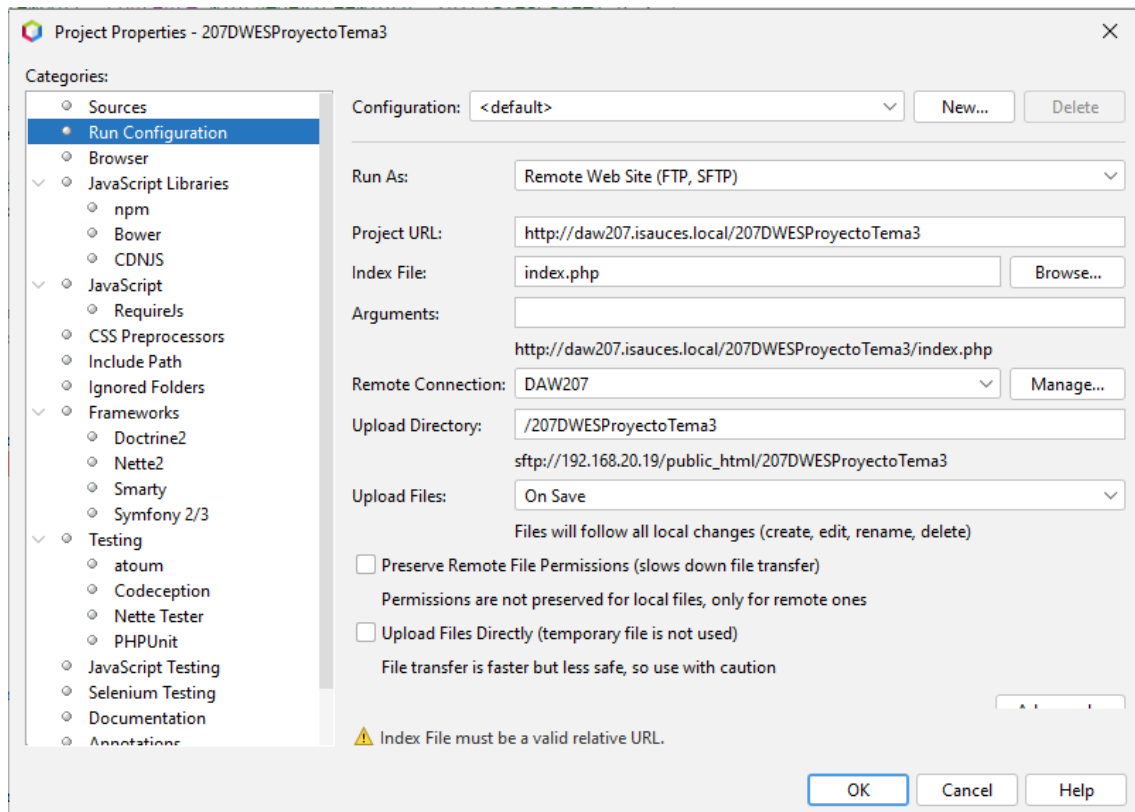
Para la configuración del debugger habría que ir a “Tools” y luego a “Options”.



Y en el apartado debuggins de PHP estaría la configuración.



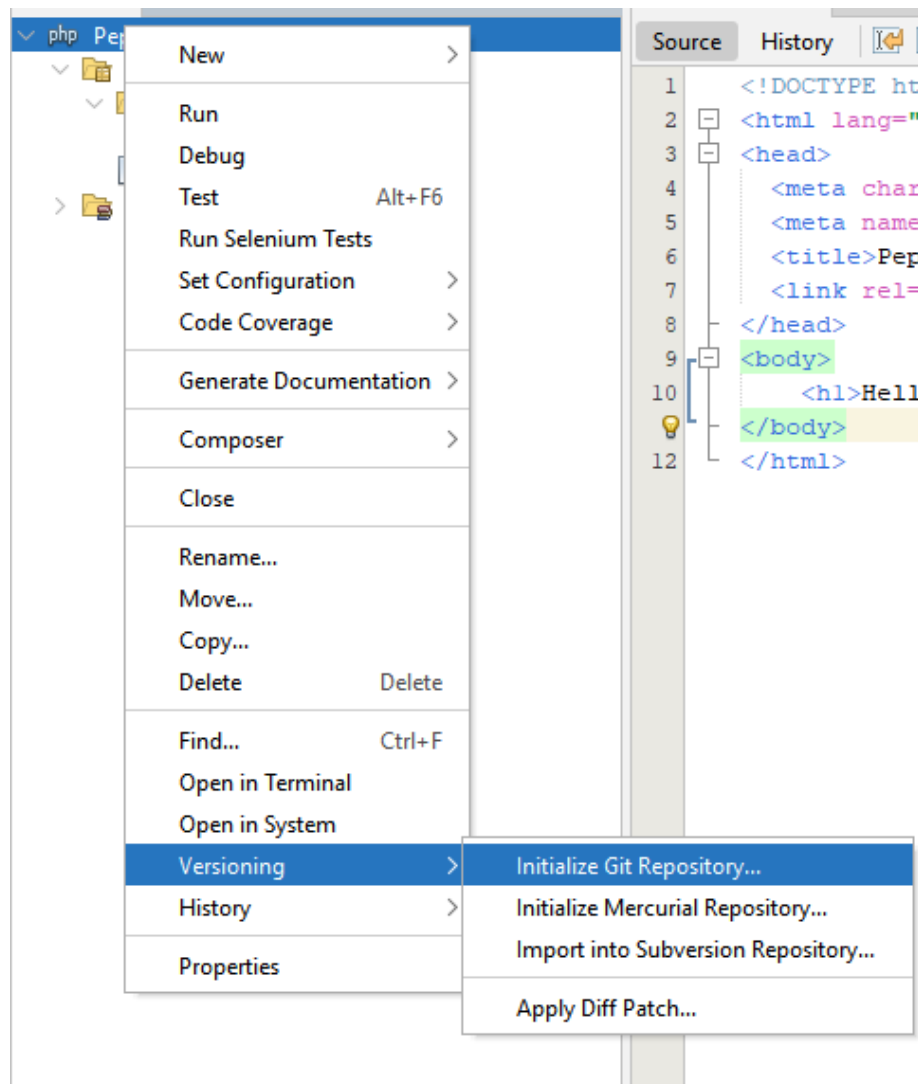
También asegurarnos que la configuración del proyecto esta correctamente.



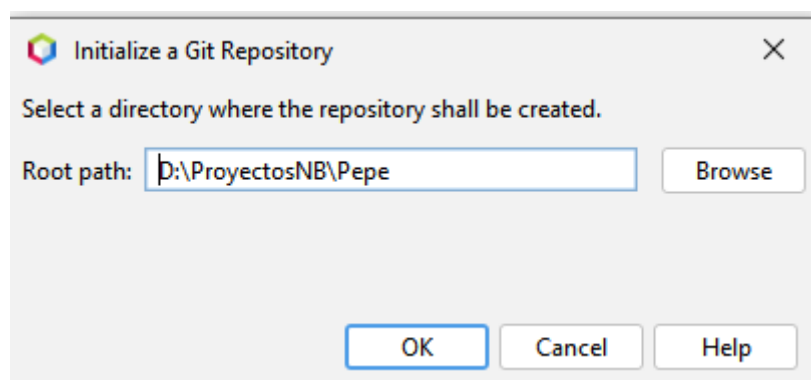
## Git

### Inicializar repositorio

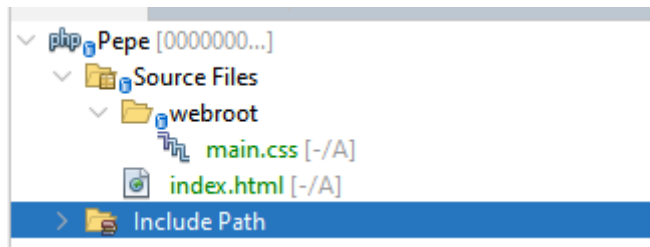
Para inicializar un repositorio en NetBeans solo debemos escoger la opción: "Initialize Git Repository..." como se muestra en la captura.



Aquí nos mostrara la ruta del proyecto que queremos inicializar, comprobamos y si esta correcta le damos a "ok".

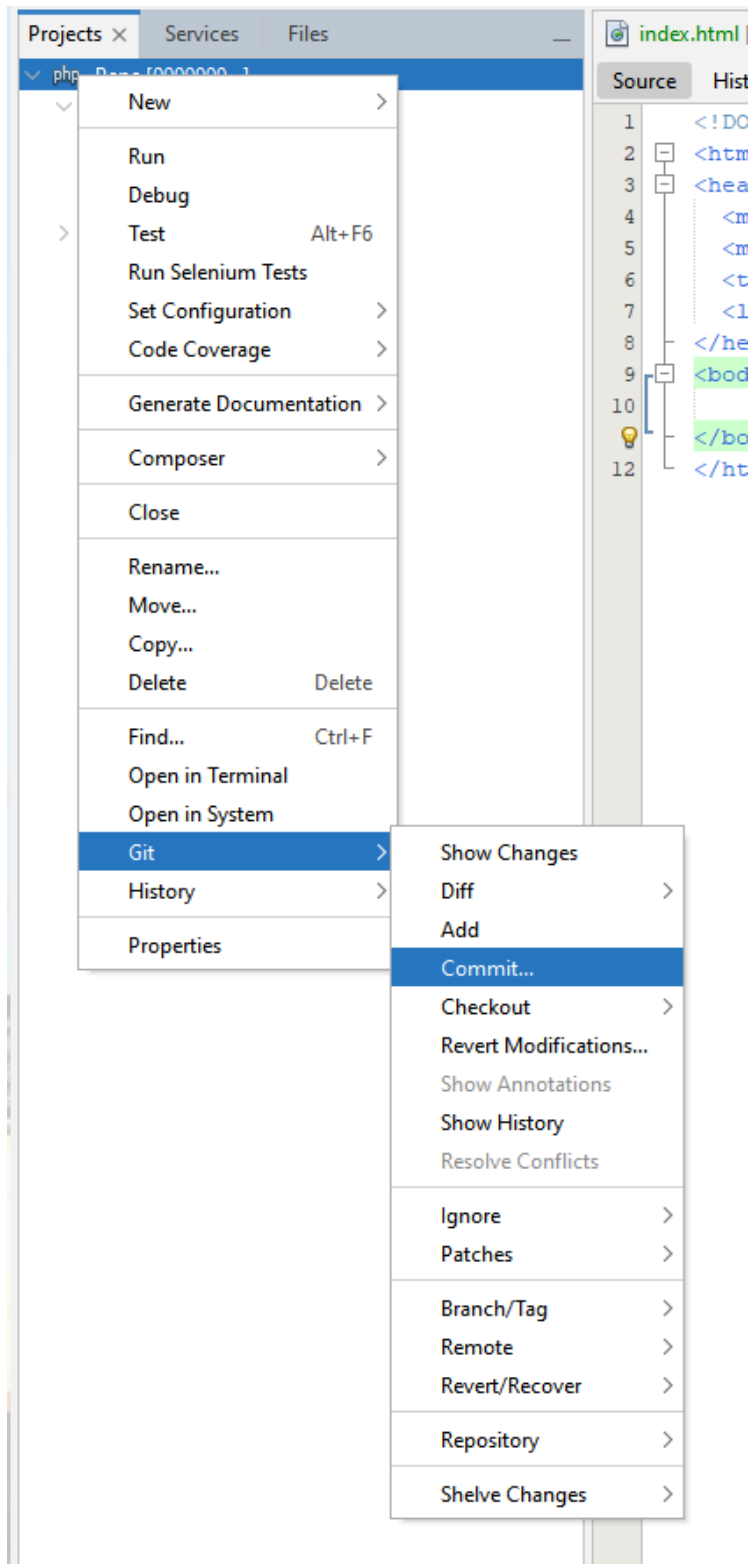


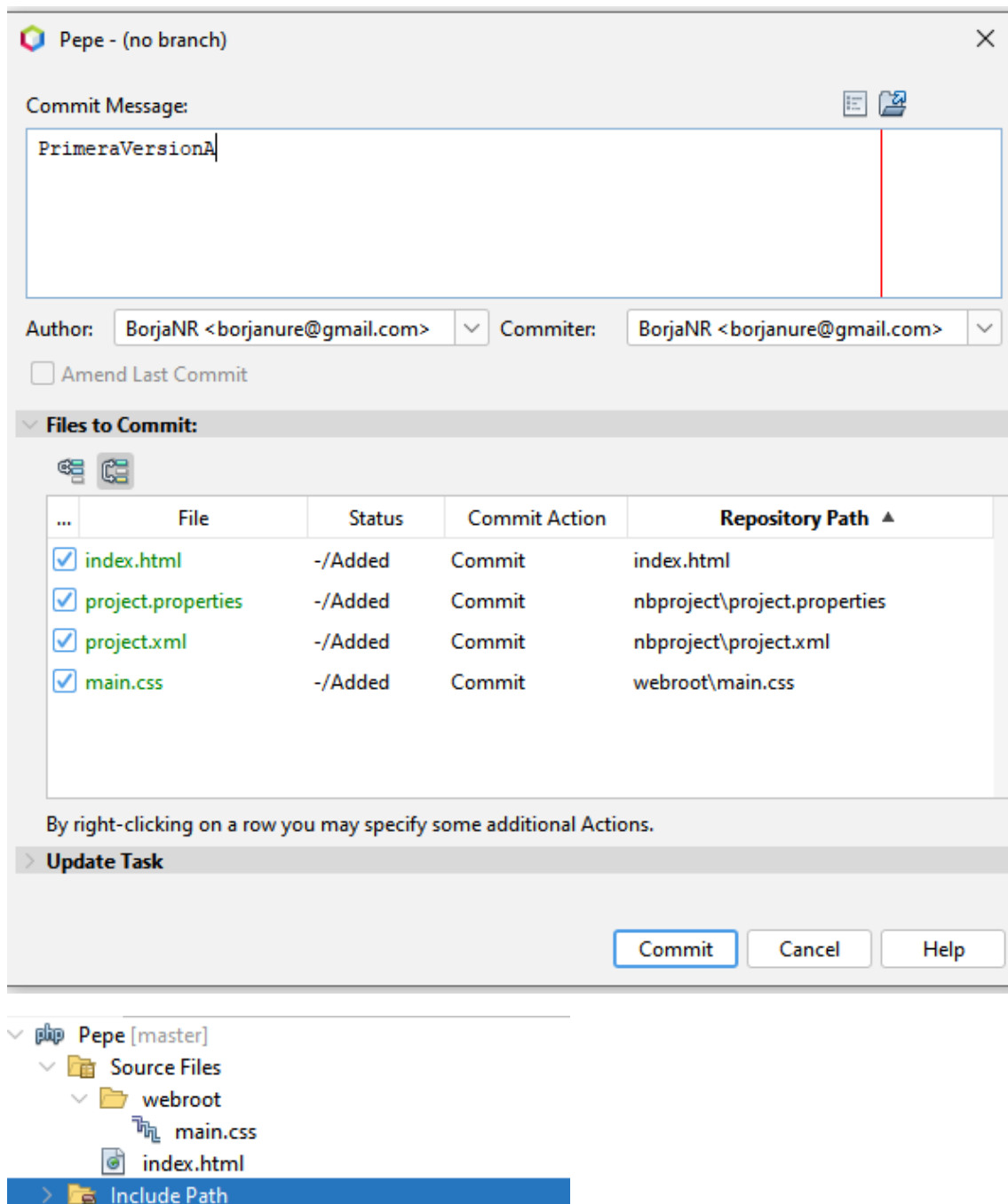
Ya lo tendríamos inicializado y nos aparecería con este aspecto:



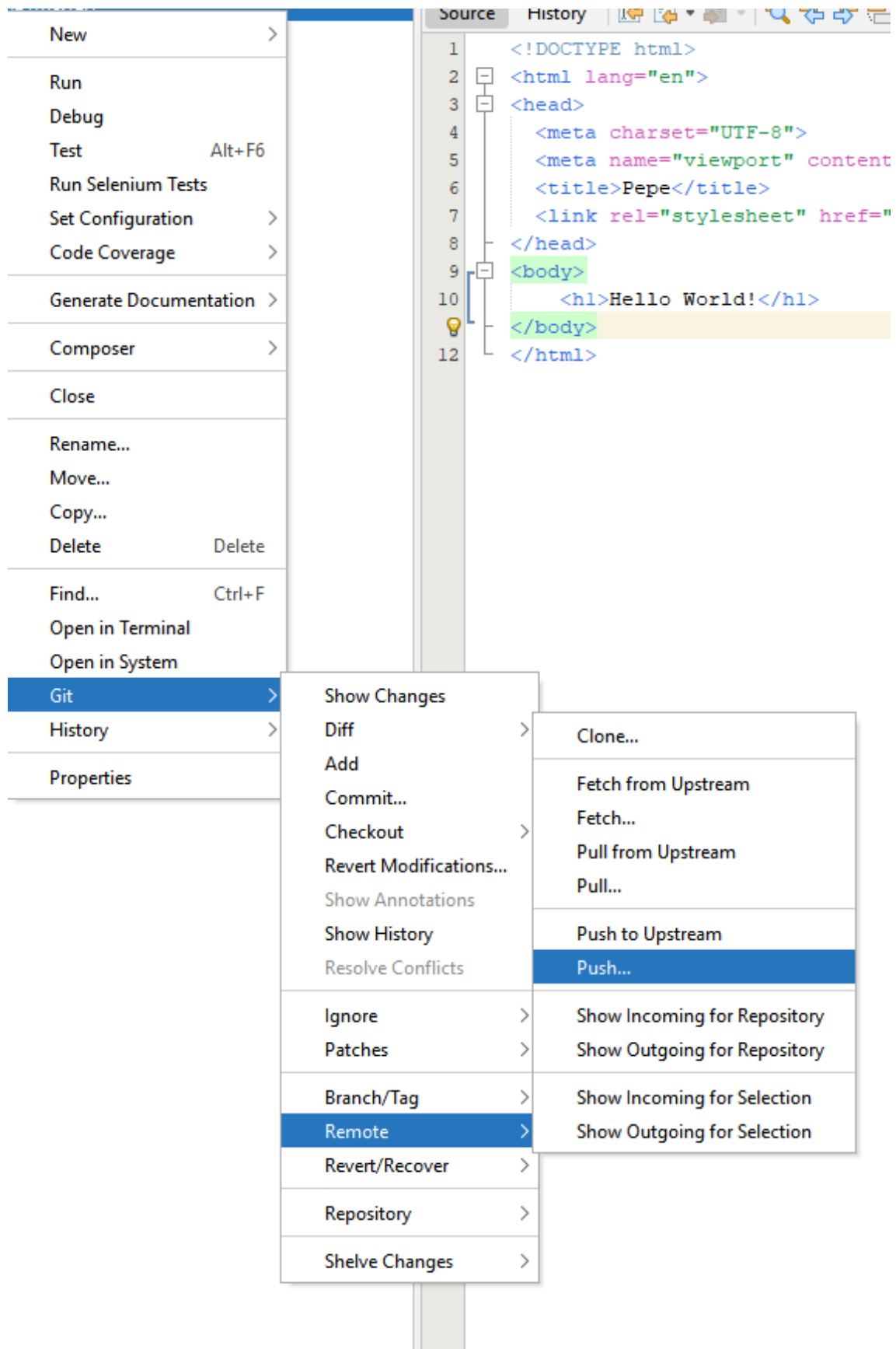
## Commit

Seleccionamos la opción “Commit...” como se muestra en la captura





## Push



Push to Remote Repository

**Steps**

1. Remote Repository
2. Select Local Branches
3. Update Local References

**Remote Repository**

☐ Select Configured Git Repository Location:

☒ Specify Git Repository Location:

Remote Name:  ☒ Persist Remote

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

User:  (leave blank for anonymous access)

Password:  ☒ Save Password

[Proxy Configuration...](#)

< Back Next > Finish Cancel Help

Push to Remote Repository

**Steps**

1. Remote Repository
2. Select Local Branches
3. Update Local References

**Select Local Branches**

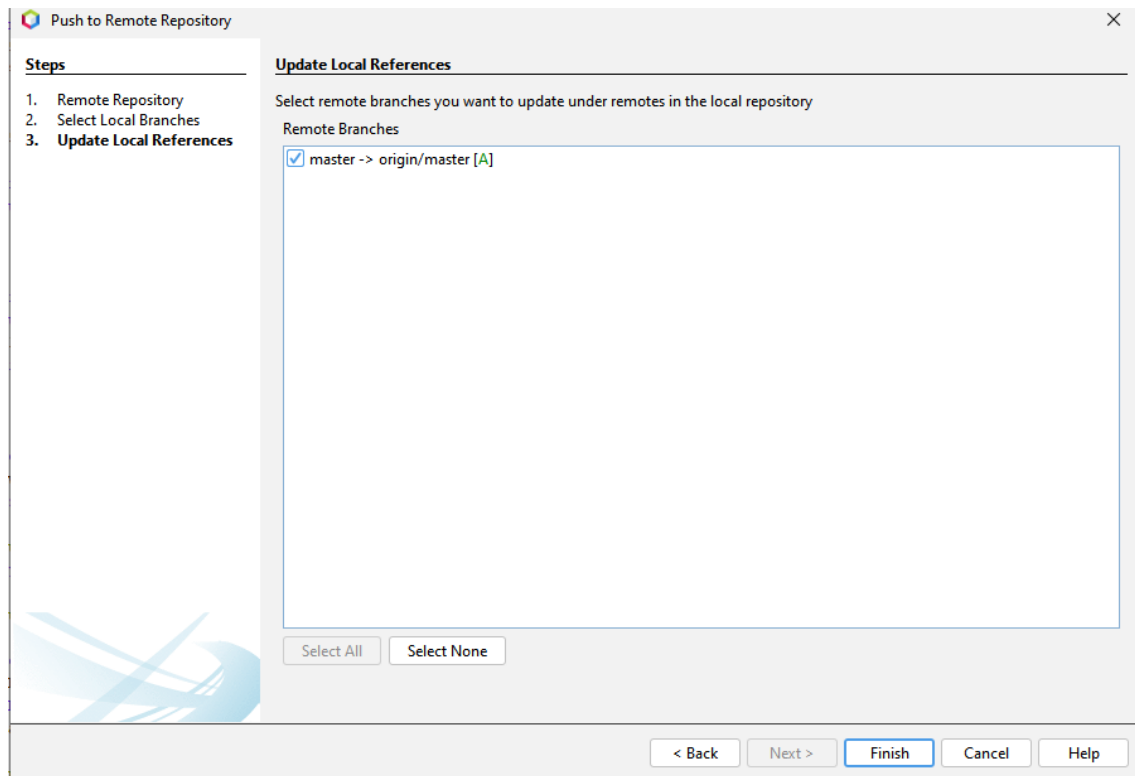
Local Branches

☒ master -> master [A]

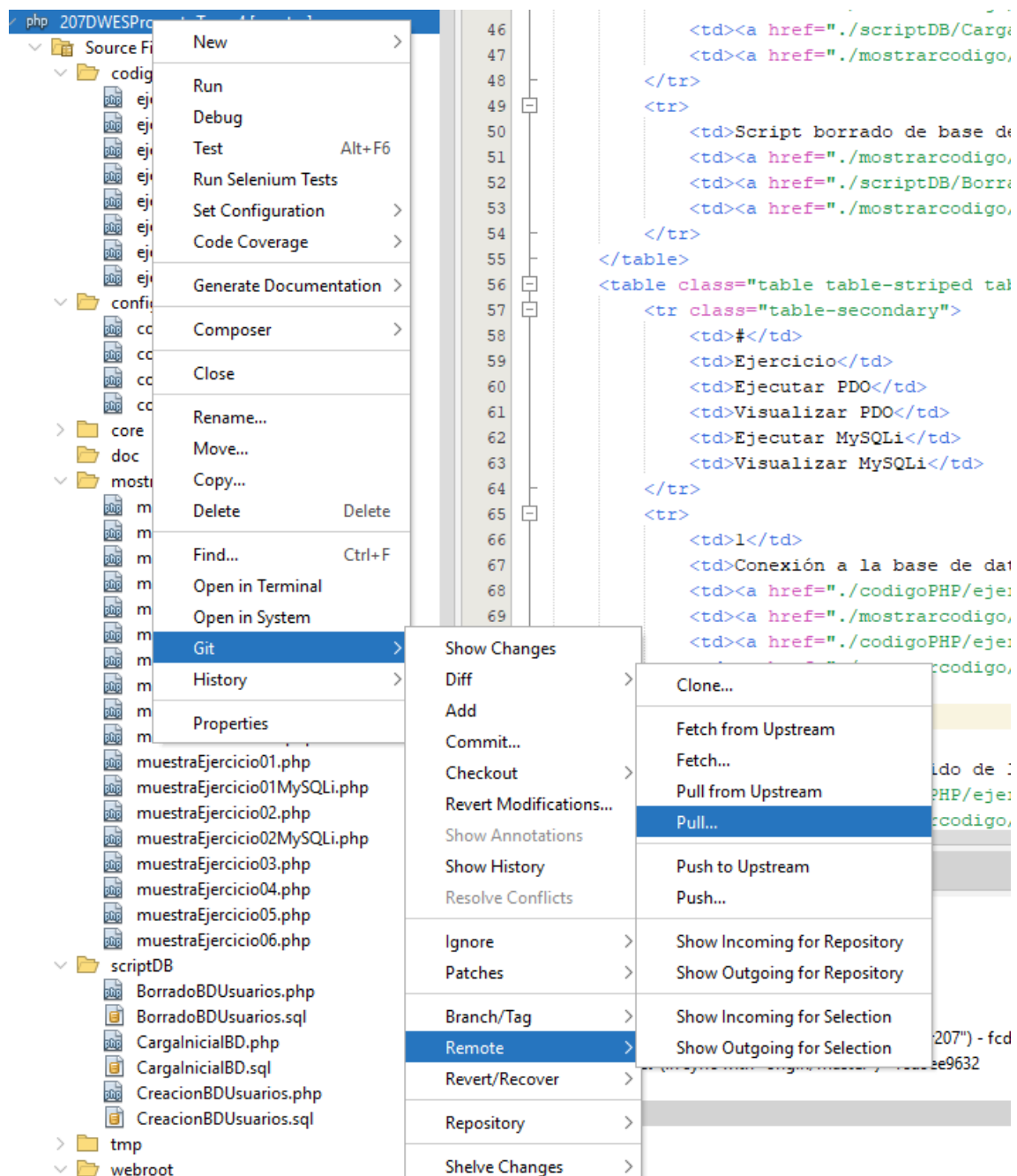
Select All Select None

< Back Next > Finish Cancel Help





## Pull



**Pull from Remote Repository**

**Steps**

1. **Remote Repository**
2. Remote Branches

**Remote Repository**

☒ Select Configured Git Repository Location:

origin:https://BorjaNR@github.com/BorjaNR/207DWESProyectoTema4.git

☐ Specify Git Repository Location:

Remote Name:  ☒ Persist Remote

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

User:  (leave blank for anonymous access)

Password:  ☒ Save Password

< Back **Next >** Finish Cancel Help

**Pull from Remote Repository**

**Steps**

1. Remote Repository
2. **Remote Branches**

**Remote Branches**

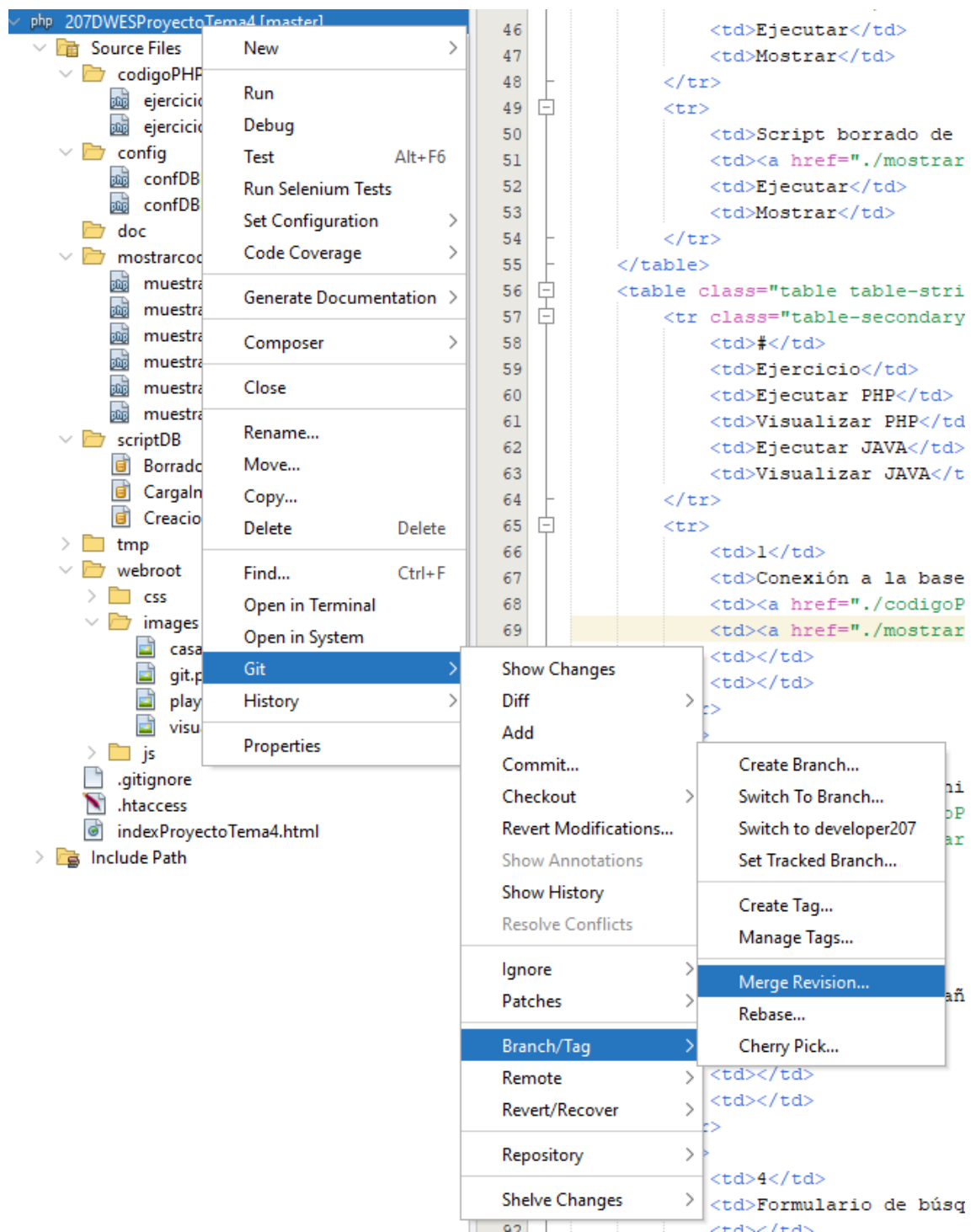
Remote Branches

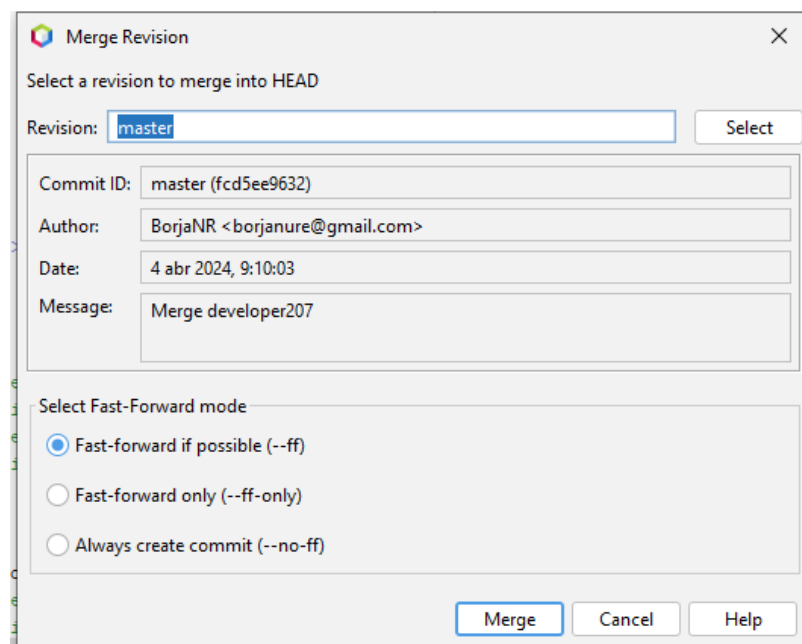
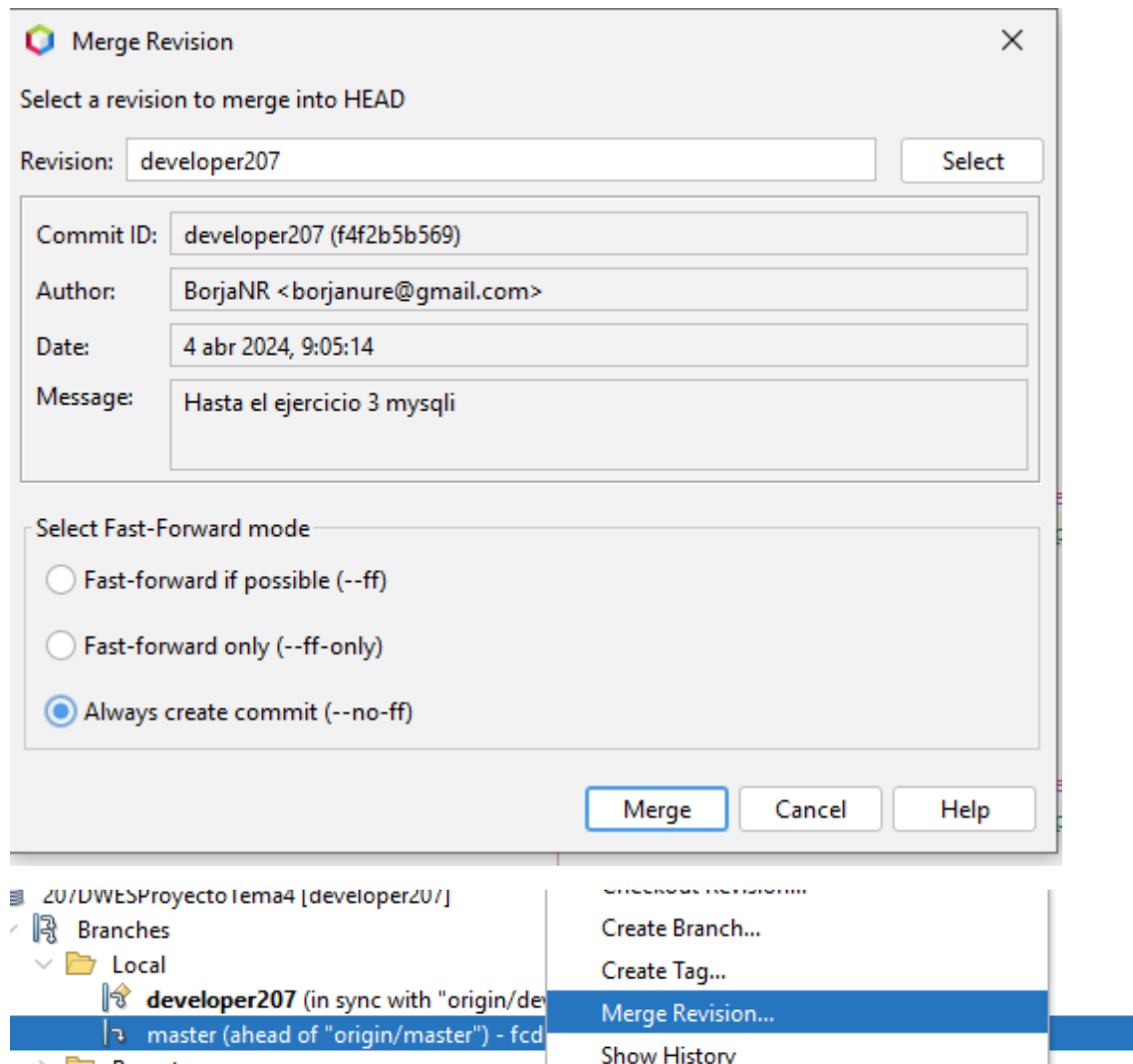
<input checked="" type="checkbox"/>	developer207 -> origin/developer207
<input checked="" type="checkbox"/>	master -> origin/master

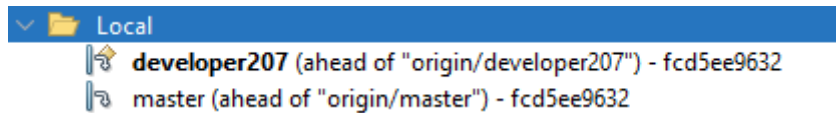
**i** Branch **origin/master** will be merged into the current branch **master**.

< Back **Next >** **Finish** Cancel Help

## Merge







# GitHub



## Crear nuevo repositorio

### 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 \*  BorjaNR / Repository name \*

✓ Pepe is available.

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

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

.gitignore template: **None** ▼

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

Choose a license



License: **None** ▼

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

 You are creating a public repository in your personal account.

Create repository

Quick setup — if you've done this kind of thing before

 Set up in Desktop or ☐ HTTPS ☐ SSH  

# Crear release

## Releases

No releases published

[Create a new release](#)

Releases

Tags

v2.0.0

Target: master

Previous tag: auto

Generate release notes

Excellent! This tag will be created from the target when you publish this release.

v2.0.0

WritePreview

H B I

- Ejercicios del 1 al 6 hechos mediante PDO

- Ejercicios del 1 al 2 hechos mediante MySQLi

- Scripts de creación, borrado y carga inicial

- Archivos de configuración

Markdown is supported Paste, drop, or click to add files

Attach binaries by dropping them here or selecting them.

☐ Set as a pre-release

This release will be labeled as non-production ready

Publish release

Save draft

v2.0.0

Latest

Compare

BorjaNR released this now v2.0.0 fcd5ee9

- Ejercicios del 1 al 6 hechos mediante PDO
- Ejercicios del 1 al 2 hechos mediante MySQLi
- Scripts de creación, borrado y carga inicial
- Archivos de configuración

▼ Assets 2

Source code (zip)

16 minutes ago

Source code (tar.gz)

16 minutes ago