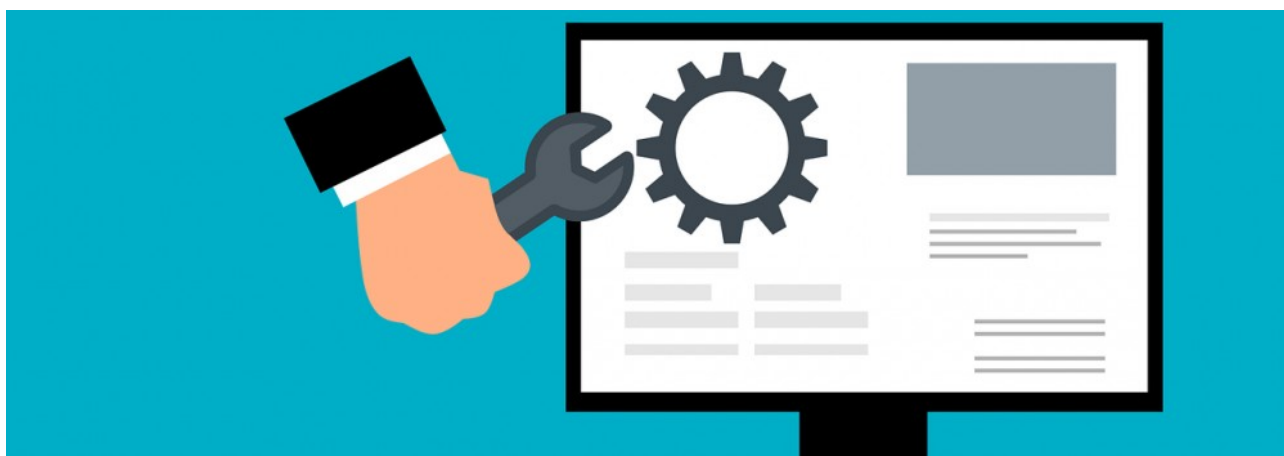


Ejercicios Tema 2



Índice

Ubuntu Server.....	4
Configuración inicial.....	4
Cuentas de administración.....	6
Creación usuario administrador miadmin.....	6
Apache.....	6
Comprobación de estado de Apache.....	7
Apertura de puerto 80 y comprobación de puertos abiertos.....	8
PHP.....	9
MySQL.....	13
Instalación:.....	13
Comprobación de puertos:.....	14
Creación de usuario adminsql.....	17
Gestión conexión base de datos desde NetBeans.....	20
Conexión.....	20
Creación, inserción y borrado Base de Datos.....	24
Comprobación conexión y muestra de datos desde servidor Ubuntu.....	28
Xdebug.....	30
Cuentas de desarrollo y hosting virtual.....	31
Creación usuario de desarrollo “operadorweb”.....	31
Hosting virtual.....	32
WXED – Windows X:.....	34
Instalación y configuración inicial de la máquina.....	34
Datos equipo cliente Windows 10.....	34
Cuentas administradoras y cuenta de desarrollador.....	38
Navegadores.....	39
Google Chrome.....	39
Mozilla Firefox.....	39
Microsoft Edge.....	40
Opera.....	40
Filezilla.....	40
Notepad++.....	41
Netbeans:.....	41
Instalación y configuración inicial (plugings).....	41
Conexión al entorno de desarrollo.....	47
Administración de la base de datos.....	51
Conexión al repositorio – versionado.....	51
Depuración – Configuración de la ejecución para la depuración.....	52
Prueba de depurado de un archivo .php.....	57
Creación de un proyecto PHP.....	59
Proyecto nuevo:.....	60
Creación de proyecto PHP desde un servidor:.....	63
Creación de proyecto PHP con fuentes existentes:.....	67
Paso a explotación.....	69
PHP Doc.....	69

CSS / JS / AJAX / XML / JSON.....	69
-----------------------------------	----

Ubuntu Server

[La distribución de Linux elegida es la Ubuntu f22.04 de 64 bits.](#)

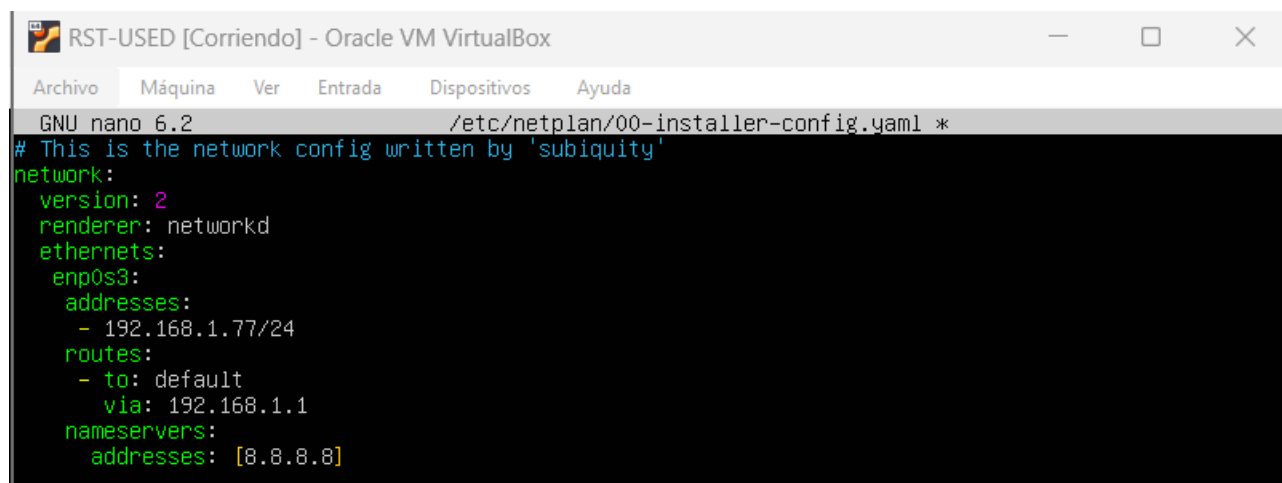
```
miadmin@RST-USED:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.77 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:fed9:e521 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:d9:e5:21 txqueuelen 1000 (Ethernet)
    RX packets 412 bytes 232191 (232.1 KB)
    RX errors 0 dropped 6 overruns 0 frame 0
    TX packets 145 bytes 11207 (11.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 98 bytes 7735 (7.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 98 bytes 7735 (7.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Configuración inicial.

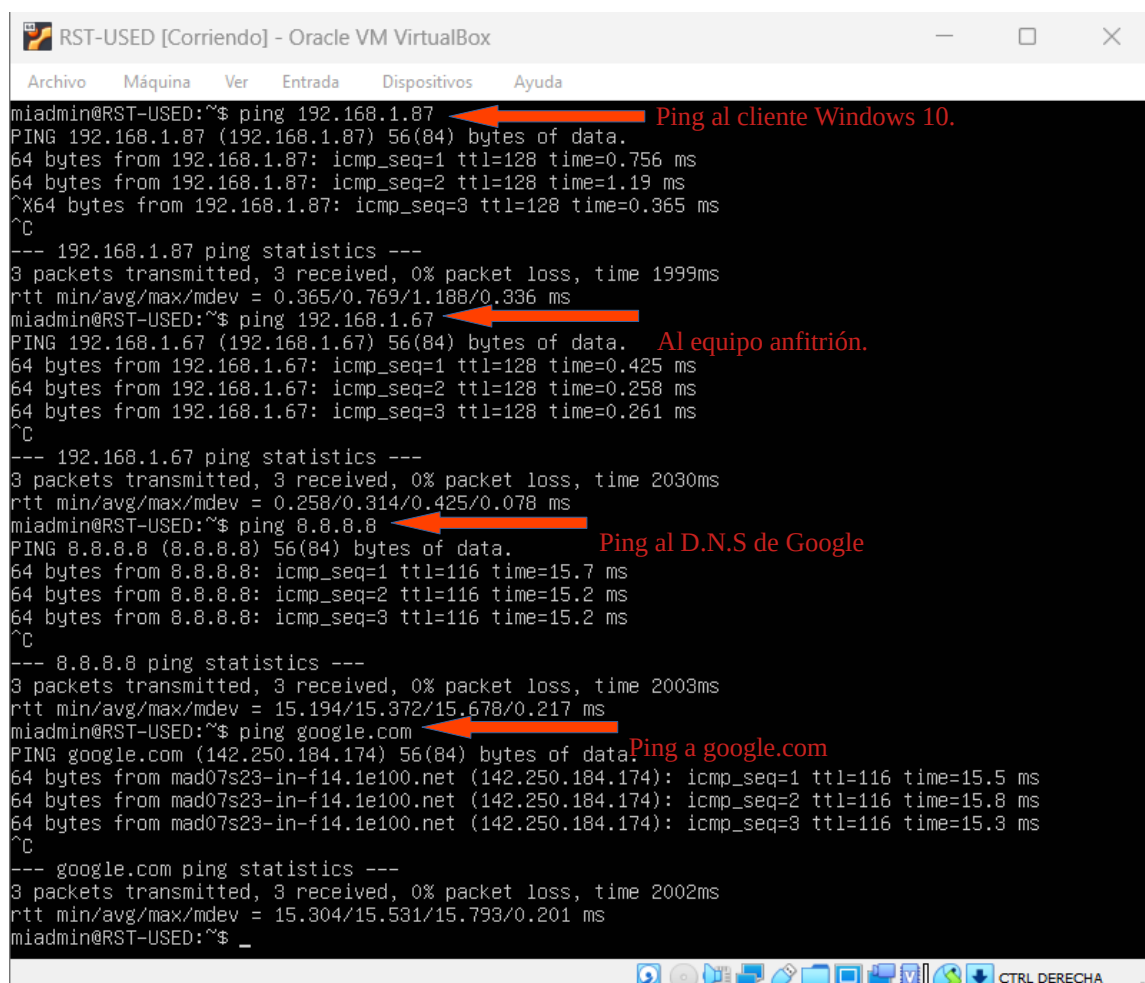
Después de instalar el servidor, hay que cambiar la configuración de red para poder comunicarnos con el cliente, el equipo anfitrión e incluso con el exterior.

Con el siguiente comando `sudo nano /etc/netplan/00-installer-config.yaml` redefinimos las nuevas direcciones ip. Es importante respetar el formato exacto del documento que aparece en la siguiente imagen, siendo especialmente cuidadoso con la sintaxis, los espacios (mejor no usar tabulador) y adaptando las direcciones a las que a nosotros nos interesen.



```
RST-USED [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 6.2 /etc/netplan/00-installer-config.yaml *
# This is the network config written by 'subiquity'
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      addresses:
        - 192.168.1.77/24
      routes:
        - to: default
          via: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8]
```

Para comprobar si hemos tenido éxito, lo siguiente es hacer ping a las diferentes ip´s contra las que trabajará este servidor. A saber, la máquina anfitrión, el cliente Windows X e internet. La siguiente imagen muestra como se ha establecido la conexión con éxito con todas ellas.



```
miadmin@RST-USED:~$ ping 192.168.1.87
PING 192.168.1.87 (192.168.1.87) 56(84) bytes of data.
64 bytes from 192.168.1.87: icmp_seq=1 ttl=128 time=0.756 ms
64 bytes from 192.168.1.87: icmp_seq=2 ttl=128 time=1.19 ms
^X64 bytes from 192.168.1.87: icmp_seq=3 ttl=128 time=0.365 ms
^C
--- 192.168.1.87 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.365/0.769/1.188/0.336 ms
miadmin@RST-USED:~$ ping 192.168.1.67
PING 192.168.1.67 (192.168.1.67) 56(84) bytes of data.
64 bytes from 192.168.1.67: icmp_seq=1 ttl=128 time=0.425 ms
64 bytes from 192.168.1.67: icmp_seq=2 ttl=128 time=0.258 ms
64 bytes from 192.168.1.67: icmp_seq=3 ttl=128 time=0.261 ms
^C
--- 192.168.1.67 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2030ms
rtt min/avg/max/mdev = 0.258/0.314/0.425/0.078 ms
miadmin@RST-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=116 time=15.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=15.2 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=15.2 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 15.194/15.372/15.678/0.217 ms
miadmin@RST-USED:~$ ping google.com
PING google.com (142.250.184.174) 56(84) bytes of data:
64 bytes from mad07s23-in-f14.1e100.net (142.250.184.174): icmp_seq=1 ttl=116 time=15.5 ms
64 bytes from mad07s23-in-f14.1e100.net (142.250.184.174): icmp_seq=2 ttl=116 time=15.8 ms
64 bytes from mad07s23-in-f14.1e100.net (142.250.184.174): icmp_seq=3 ttl=116 time=15.3 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 15.304/15.531/15.793/0.201 ms
miadmin@RST-USED:~$ _
```

Cuentas de administración.

Creación usuario administrador miadmin.

El usuario miadmin se puede configurar durante la instalación del SO.

Tan sencillo como sustituir en cada recuadro el nombre de ubuntu por el que le queramos dar al servidor en primer lugar, a continuación el nombre de la máquina, después el del usuario y en los dos siguientes recuadros se establece un password para este usuario y se confirma.

Comprobación de identidad del usuario.

```
miadmin@RST-USED:~$ whoami
miadmin
```

Apache.

En primer lugar, conviene hacer un update y posterior upgrade para asegurarnos del estado actualizado de todos los paquetes necesarios. A continuación, ejecutamos el comando `sudo apt install apache2`,

```
miadmin@RST-USED:~$ sudo apt install apache2_
```

Comprobación de estado de Apache.

El comando para comprobar el estado de servicio de Apache es **service apache2 status**.

El resultado correcto nos lo indica el “active (running)” mostrado.

```
miadmin@RST-USED:~$ service apache2 status
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-09-29 09:35:24 UTC; 6min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 1637 (apache2)
    Tasks: 55 (limit: 2238)
   Memory: 5.3M
      CPU: 66ms
   CGroup: /system.slice/apache2.service
           └─1637 /usr/sbin/apache2 -k start
             └─1639 /usr/sbin/apache2 -k start
               └─1640 /usr/sbin/apache2 -k start

sep 29 09:35:24 RST-USED systemd[1]: Starting The Apache HTTP Server...
sep 29 09:35:24 RST-USED apachectl[1636]: AH00558: apache2: Could not reliably determine the server
sep 29 09:35:24 RST-USED systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)
```

Por defecto, el único puerto abierto en este punto de la instalación es el 22, de la siguiente forma lo podemos comprobar. Necesitamos abrir también el puerto 80.

```
miadmin@RST-USED:~$ sudo ufw status
[sudo] password for miadmin:
Status: active

To Action From
--
22/tcp ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)
```

Apertura de puerto 80 y comprobación de puertos abiertos.

Con el comando **sudo ufw allow 80** abrimos el puerto 80. Con **sudo ufw status** comprobamos su estado, correcto en este caso, los puertos 22 y 80 han de estar abiertos.

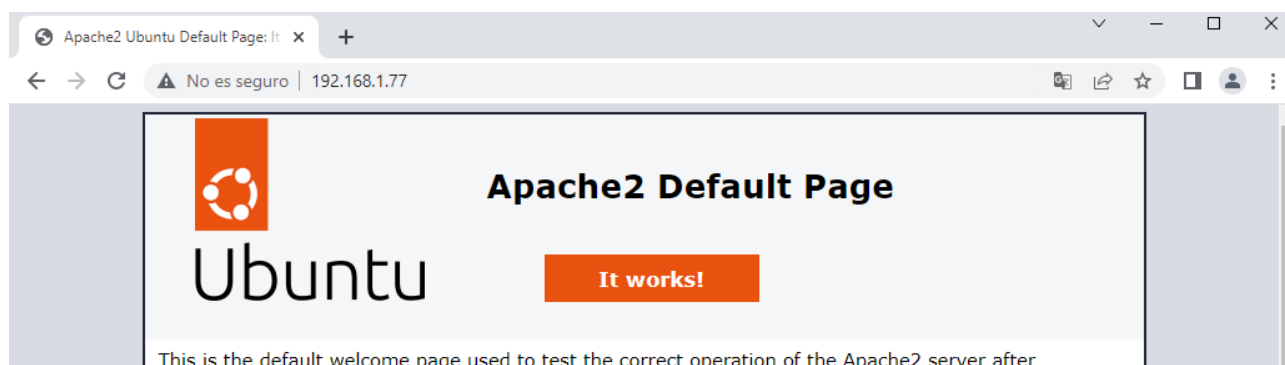
```
miadmin@RST-USED:~$ sudo ufw allow 80
Rule added
Rule added (v6)
miadmin@RST-USED:~$ ufw status
ERROR: You need to be root to run this script
miadmin@RST-USED:~$ sudo ufw status
Status: active

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

miadmin@RST-USED:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:d6:60:22 brd ff:ff:ff:ff:ff:ff
    inet 192.168.3.207/24 brd 192.168.3.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fed6:6022/64 scope link
        valid_lft forever preferred_lft forever
miadmin@RST-USED:~$ _
```

La manera de comprobar que desde el cliente tenemos acceso a la página `index.html` del servidor presente en la ruta `var/www/html/index.html`, es abriendo un navegador desde, por ejemplo, el equipo anfitrión y escribiendo allí la ip de nuestro servidor Ubuntu. Hará una llamada HTTP al servidor y este le responderá enviándole el archivo `index.html` que es el presente en el directorio creado por Apache.

Esta primera página de Apache que certifica el funcionamiento del servidor será la que después nosotros tenemos que sustituir por nuestros propios ficheros.



PHP.

A continuación documento la instalación de PHP aclarando en primer lugar que ha de hacerse casi en su integridad en el servidor Ubuntu, únicamente me referiré a la máquina cliente para comprobar su funcionamiento.

Con este sencillo comando comienza la instalación. `sudo apt install php`

```
miadmin@RST-USED:/$ sudo apt install php_
```

Nos informa de paquetes adicionales que se van a cargar en la primera instalación, nos sugiere paquetes nuevos, nos informa de el peso de los archivos a descargar y del espacio en disco que va a ocupar la instalación. Finalmente nos pide confirmación para proseguir con la instalación.

```
Se instalarán los siguientes paquetes adicionales:
  libapache2-mod-php8.1 php-common php8.1 php8.1-cli php8.1-common php8.1-opcache php8.1-readline
Paquetes sugeridos:
  php-pear
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-php8.1 php php-common php8.1 php8.1-cli php8.1-common php8.1-opcache
  php8.1-readline
0 actualizados, 8 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 5.126 kB de archivos.
Se utilizarán 21,3 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] _
```

Para comprobar si se ha instalado y en tal caso, la versión de PHP, utilizamos el comando `php -v`.

```
miadmin@RST-USED:/$ php -v
PHP 8.1.2 (cli) (built: Aug  8 2022 07:28:23) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2, Copyright (c), by Zend Technologies
miadmin@RST-USED:/$ _
```

Útil se hace en este punto comprobar el árbol de directorios de PHP; para ello ejecutamos el comando “tree -d /etc/php”

```
/etc/php
├── 8.1
│   ├── apache2
│   │   └── conf.d
│   ├── cli
│   │   └── conf.d
│   └── mods-available
6 directories
```

Lo siguiente es abrir el fichero de configuración php.ini con el editor nano. Antes, voy a ir hasta donde se encuentra, a realizar un backup del fichero por seguridad y a mostrar el resto de ficheros presentes en el directorio.

Aquí se ve el fichero de configuración de php (php.ini) que voy a configurar a continuación, también se ve el fichero conf.d al que tendremos que recurrir para configurar el Xdebug, pero eso ya es otra historia.

```
miadmin@RST-USED:/etc/php/8.1/apache2$ sudo cp php.ini phpbackup.ini
[sudo] password for miadmin:
miadmin@RST-USED:/etc/php/8.1/apache2$ dir
conf.d  phpbackup.ini  php.ini
```

El comando para abrir editar el fichero es “sudo nano php.ini”.El archivo tiene valores por defecto que hay que cambiar.

A continuación, tabla con los valores deseados tanto para el entorno de desarrollo que es el que voy a configurar como para el entorno de producción al que ya le llegará el momento.

Desarrollo	Producción
file-uploads = On	file-uploads = On
upload_max_filesize = 256M	upload_max_filesize = 256M
allow_url_fopen = On	allow_url_fopen = On
memory_limit = 256M	memory_limit = 256M
max_executiion_time = 360	max_execution_time = 360
date.timezone = Europe/Madrid	date.timezone = Europe/Madrid
Gestión y muestra de errores	
display_error = On	display_error = Off
display_startup_errors = On	display_startup_errors = Off

Secuencia con todos los valores deseados ya configurados.

```
GNU nano 6.2
; https://php.net/file-uploads
file_uploads = On
```

```
; Maximum allowed size for uploaded files.
; https://php.net/upload-max-filesize
upload_max_filesize = 256M
```

Aconsejo usar la combinación ctrl+w para buscar el texto deseado.

```
; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
; https://php.net/allow-url-fopen
allow_url_fopen = On
```

```
; Maximum amount of memory a script may consume
; https://php.net/memory-limit
memory_limit = 256M
```

```
; Maximum execution time of each script, in seconds
; https://php.net/max-execution-time
; Note: This directive is hardcoded to 0 for the CLI SAPI
max_execution_time = 360
```

```
[Date]
; Defines the default timezone used by the date functions
; https://php.net/date.timezone
date.timezone = Europe/Madrid_
```

Última configuración, salida de errores, aquí es donde tienen distintas configuraciones si se trata de un entorno de producción o de desarrollo.

A la hora de hacer la búsqueda con ctrl+w en esta ocasión hay que sustituir los guiones bajos por guiones normales. Ejemplo: Para buscar “display_errors,” tendremos que escribir “display-errors” en el cuadro de búsqueda.

```
Generating runtime constants...
; Possible Values:
;   Off = Do not display any errors
;   stderr = Display errors to STDERR (affects only CGI/CLI binaries!)
;   On or stdout = Display errors to STDOUT
; Default Value: On
; Development Value: On
; Production Value: Off
; https://php.net/display-errors
display_errors = On

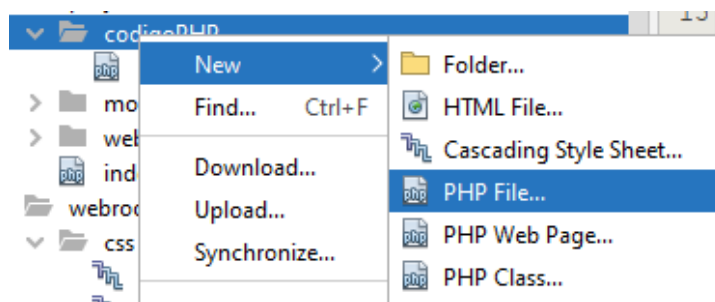
; The display of errors which occur during PHP's startup sequence are handled
; separately from display_errors. We strongly recommend you set this to 'off'
; for production servers to avoid leaking configuration details.
; Default Value: On
; Development Value: On
; Production Value: Off
; https://php.net/display-startup-errors
display_startup_errors = On
```

Ya solo queda reiniciar el servicio de apache2 para confirmar cambios y hacer prueba en la máquina cliente para comprobar que reconoce el lenguaje PHP.

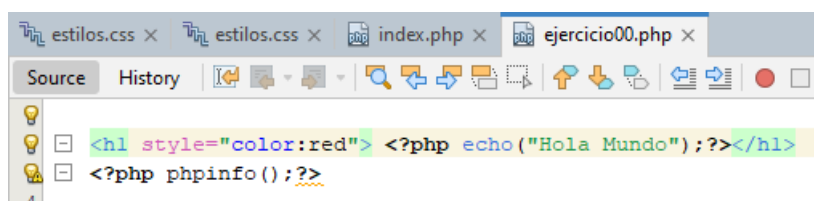
Reiniciar apache2 lo realizamos con el comando “sudo service apache2 restart”.

Para la prueba de ejecución voy a crear un fichero .php en NetBeans dentro de la máquina cliente.

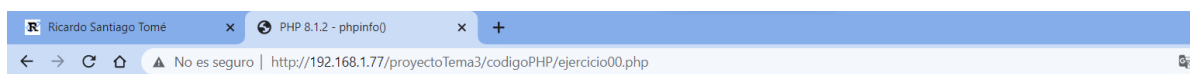
Situándome en la ubicación dentro del árbol de directorios de mi programa y haciendo click derecho muestra el IDE las siguientes ventanas.




La nombramos, abrimos y editamos con el siguiente código.



Si la instalación de PHP en nuestro servidor ha sido exitosa, al ejecutar este archivo nos ha de mostrar una página web con todos los módulos y características del lenguaje PHP. Si no, mostrará una página únicamente con el código mostrado como texto.



Hola Mundo

PHP Version 8.1.2	
	
System	Linux RST-USED 5.15.0-50-generic #56-Ubuntu SMP Tue Sep 20 13:23:26 UTC 2022 x86_64
Build Date	Aug 8 2022 07:28:23
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-ffi.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

¡Trabajo hecho!

MySQL.

Instalación:

Como requisito previo, debemos hacer actualización y mejora del sistema con “sudo apt update” y “sudo apt upgrade”, es siempre conveniente ejecutar estos dos pasos antes de una instalación de importancia para asegurarnos de tener los paquetes más recientes.

```
miadmin@RST-USED:~$ sudo apt update_
```

```
miadmin@RST-USED:~$ sudo apt upgrade
```

Cumplido este requisito previo hay que ejecutar el comando de instalación de mysql “sudo apt install mysql-server.”

```
miadmin@RST-USED:~$ sudo apt install mysql-server_
```

Como requisito de seguridad, nos pide confirmación. Decimos que sí.

```
Se necesita descargar 29,4 MB de archivos.  
Se utilizarán 242 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n] _
```

Si no lanza ningún mensaje de error, podemos comprobar si se ha instalado mysql y en tal caso su versión con el siguiente comando.

```
miadmin@RST-USED:~$ sudo mysql --version  
mysql Ver 8.0.31-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
```

De aquí en adelante, las imágenes que documentan esta instalación serán capturadas desde el terminal del equipo anfitrión previo inicio de sesión con ssh. Hago esto por la facilidad que otorga para el manejo de texto, cortado y copiado del mismo y establecimiento de fondo de pantalla claro.

Realizada esta personalización accedemos a mysql desde la línea de comandos vemos en la imagen siguiente que el prompt ha cambiado, ahora nos muestra mysql>, esto nos termina de confirmar, junto con la impresión de la versión de mysql que nuestra instalación está marchando por el buen camino.

```

miadmin@RST-USED: ~$ sudo mysql
[sudo] password for miadmin:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.31-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2022, 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>

```

Version de MySQL y Ubuntu server.

Prompt de mysql

Comprobada la correcta instalación de MySQL, salgo de su prompt y regreso al símbolo de sistema de Ubuntu para los siguientes pasos.

Comprobación de puertos:

Para saber por que puerto ‘escucha’ MySQL ejecutamos ss -puta. Con estos parámetros nos muestra la dirección y el nombre del programa que escucha.

```

miadmin@RST-USED: ~$ ss -puta
Netid State  Recv-Q Send-Q           Local Address:Port
Peer Address:Port           Process
udp    UNCONN  0      0      0.0.0.0:*
[fe80::a00:27ff:fed9:e521]%enp0s3:dhcpv6-client
[::]:*
tcp    LISTEN  0      70      0.0.0.0:*
127.0.0.1:33060
tcp    LISTEN  0      151     0.0.0.0:*
127.0.0.1:mysql
0.0.0.0:*
tcp    LISTEN  0      4096    0.0.0.0:*
127.0.0.53%lo:domain
0.0.0.0:*
tcp    LISTEN  0      128     0.0.0.0:*
0.0.0.0:ssh
tcp    ESTAB   0      52      192.168.1.67:52130
192.168.1.77:ssh
tcp    LISTEN  0      511     *:http
tcp    LISTEN  0      128     [::]:ssh
[::]:*

```

Para saber el puerto ejecutamos `ss -p` y nos sustituye los nombres por los puertos por los que estos escuchan.

Donde antes estaba mysql, ahora está su puerto, el 3306.

```
miadmin@RST-USED:~$ ss -p
```

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
udp	UNCONN	0	0	127.0.0.1:53		systemd
udp	UNCONN	0	0	0.0.0.0:*		
udp	UNCONN	0	0	[fe80::a00:27ff:fed9:e521]::*		[fe80::a00:27ff:fed9:e521]::*
tcp	LISTEN	0	70	127.0.0.1:3306		mysqld
tcp	LISTEN	0	151	0.0.0.0:*		sshd
tcp	LISTEN	0	151	0.0.0.0:*		sshd

Lo siguiente es abrir ese puerto. Primero comprobamos el estado de los puertos.

```
miadmin@RST-USED:~$ sudo ufw status
```

[sudo] password for miadmin:

Status: active

To	Action	From
22/tcp	ALLOW	Anywhere
80	ALLOW	Anywhere
9003	ALLOW	Anywhere
22/tcp (v6)	ALLOW	Anywhere (v6)
80 (v6)	ALLOW	Anywhere (v6)
9003 (v6)	ALLOW	Anywhere (v6)

Vemos que el 3306 no está. Con el siguiente comando, lo añadimos y adjudicamos regla de apertura.

```
miadmin@RST-USED:~$ sudo ufw allow 3306
```

Rule added

Rule added (v6)

Y volvemos a comprobar nuestros puertos y su estado.

El puerto 3306 ya aparece y su estado es de permitir (conexión) desde cualquier punto.

```
miadmin@RST-USED:~$ sudo ufw status
Status: active

To Action From
--
22/tcp ALLOW Anywhere
80 ALLOW Anywhere
9003 ALLOW Anywhere
3306 ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)
80 (v6) ALLOW Anywhere (v6)
9003 (v6) ALLOW Anywhere (v6)
3306 (v6) ALLOW Anywhere (v6)
```

Para terminar de permitir esta conexión hay que modificar el archivo de configuración `mysqld.cnf`. Lo editaremos con `sudo nano` y dándole la ruta del archivo.

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

El contenido completo del fichero es el siguiente.

```
GNU nano 6.2 /etc/mysql/mysql.conf.d/mysqld.cnf
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# Here is entries for some specific programs
# The following values assume you have at least 32M ram
[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file           = /var/run/mysqld/mysqld.pid
# socket             = /var/run/mysqld/mysqld.sock
# port               = 3306
# datadir            = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir             = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address         = 127.0.0.1
mysqlx-bind-address  = 127.0.0.1
#
# * Fine Tuning
#
[ Read 78 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```


Aquí podemos ver las 2 líneas a ser comentadas.

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 127.0.0.1 ← Estado original
mysqlx-bind-address     = 127.0.0.1 ←
```

Comentadas.

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address            = 127.0.0.1 ← Ya comentadas.
#mysqlx-bind-address     = 127.0.0.1 ←
```

Guardar archivo (ctrl + o), intro y salir (ctrl + x). Ahora no tendrá solo la dirección recién comentada, será accesible desde cualquier dirección.

Reiniciar el servicio para ejecutar y guardar cambios.

Comando **sudo service mysql restart**.

El siguiente paso es instalar el módulo de php que le permita conectarse a bases de datos MySQL. Le decimos que S(si).

```
miadmin@RST-USED:~$ sudo apt install php-mysql
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya
no son necesarios.
  libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  php8.1-mysql
Se instalarán los siguientes paquetes NUEVOS:
  php-mysql php8.1-mysql
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 132 kB de archivos.
Se utilizarán 476 kB de espacio de disco adicional después de esta operación
.
¿Desea continuar? [S/n]
```

Y después de descargar y desempaquetar los paquetes necesarios...

```
Creating config file /etc/php/8.1/mods-available/pdo_mysql.ini with new vers
ion
Configurando php-mysql (2:8.1+92ubuntu1) ...
Procesando disparadores para libapache2-mod-php8.1 (8.1.2-1ubuntu2.8) ...
Procesando disparadores para php8.1-cli (8.1.2-1ubuntu2.8) ...
Scanning processes...
Scanning linux images...
```

...hemos de reiniciar el servicio apache.

```
miadmin@RST-USED:~$ sudo service apache2 restart
```

Y sería aconsejable ver su estado.

```
miadmin@RST-USED:~$ sudo service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor p
   Active: active (running) since Sun 2022-11-13 19:17:59 UTC; 9s ago
     Docs: https://httpd.apache.org/docs/2.4/
```

Activo y corriendo, en plena forma...

Creación de usuario adminsql.

Este usuario se crea para gestionar el sistema gestor de bases de datos en su totalidad sin necesidad de estar logado como root y asumir los riesgos de seguridad que ello puede llegar a provocar.

Desde la línea de comandos, en esta ocasión, de la sesión ssh desde el equipo anfitrión (serían los mismos comandos si lo ejecutamos desde el servidor Ubuntu) accedemos al sistema gestor de bases de datos MySQL que ya hemos instalado, con '**sudo mysql -u root -p**'. Accedemos como usuario root porque por defecto, es el único habilitado para crear nuevos usuarios y dotarlos de privilegios. El comando '**-u**' indica que a continuación se le va a decir con que usuario queremos acceder y '**-p**' hará que nos pida el password para acceder.

```
miadmin@RST-USED:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.31-0ubuntu0.22.04.1 (Ubuntu)
```

Ya en el prompt de MySQL tendremos que escribir nuestras sentencias atendiendo a la sintaxis sql. Cada línea será un query a ejecutar terminado por un ';'.

Entre comillas irá el nombre del usuario a crear, seguido de una arroba y a continuación y nuevamente entre comillas, las direcciones desde las que se podrá conectar al sistema gestor de bases de datos, en esta ocasión será a '%', lo que indica que se podrá conectar desde cualquier dirección ip.

'Query ok' nos informa del éxito de nuestro comando.

NOTA IMPORTANTE: Es aconsejable añadir un parámetro más a la hora de crear el usuario, el comando quedaría como sigue **CREATE USER IF NOT EXISTS**

'adminsql'@'%' IDENTIFIED BY 'paso'. De este modo, si el usuario ya existiera, nos informaría de ello y no nos lanzaría el error de que no se puede crear usuario pero sin más información añadida.

```
mysql> CREATE USER 'adminsql'@'%' IDENTIFIED BY 'paso';
Query OK, 0 rows affected (0,01 sec)
```

Dotamos de todos los privilegios, que incluyen la consulta, actualización, creado y borrado de, datos, tablas y bases de datos y le añadimos 'with grant option' que además dota a este usuario de el privilegio de poder crear otros usuarios y darles o restringirlos permisos.

Ejecutamos también un flush privileges para actualizar los privilegios otorgados.

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

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,00 sec)
```

Ejecutando select user from mysql.user podemos ver los usuarios presentes en nuestro sistema gestor de bases de datos, incluido el recién creado adminsql.

```
mysql> select user from mysql.user;
+-----+-----+
| user          | sec)  |
+-----+-----+
| adminsql      |       |
| debian-sys-maint |       |
| mysql.infoschema |       |
| mysql.session |       |
| mysql.sys     |       |
| root          |       |
+-----+-----+
6 rows in set (0,00 sec)
```

A continuación, salgo del prompt de MySQL y vuelvo a acceder pero ya como el nuevo usuario adminsql.

```
miadmin@RST-USED:~$ sudo mysql -u adminsql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.31-0ubuntu0.22.04.1 (Ubuntu)
```

Por último, con el comando 'show grants', podemos comprobar que el usuario adminsql dispone de todos los privilegios posibles. Incluido grant option.

```
mysql> show grants;
+-----+
| Grants for adminsql@% |
+-----+
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, FILE, REFERENCES, INDEX, ALTER, SHOW DATABASES, SUPER, CREATE TEMPORARY TAB
LES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, CREATE
TABLESPACE, CREATE ROLE, DROP ROLE ON *.* TO 'adminsql'@'%' WITH GRANT OPTION
|
| GRANT APPLICATION_PASSWORD_ADMIN, AUDIT_ABORT_EXEMPT, AUDIT_ADMIN, AUTHENTICATION_POLICY_ADMIN, BACKUP_ADMIN, BINLOG_ADMIN, BINLOG_ENCRYPTION_ADMIN, CLONE_ADMIN,
CONNECTION_ADMIN, ENCRYPTION_KEY_ADMIN, FIREWALL_EXEMPT, FLUSH_OPTIMIZER_COSTS, FLUSH_STATUS, FLUSH_TABLES, FLUSH_USER_RESOURCES, GROUP_REPLICATION_ADMIN, GROUP_REP
PLICATION_STREAM, INNODB_REDO_LOG_ARCHIVE, INNODB_REDO_LOG_ENABLE, PASSWORDLESS_USER_ADMIN, PERSIST_RO_VARIABLES_ADMIN, REPLICATION_APPLIER, REPLICATION_SLAVE_ADMI
N, RESOURCE_GROUP_ADMIN, RESOURCE_GROUP_USER, ROLE_ADMIN, SENSITIVE_VARIABLES_OBSERVER, SERVICE_CONNECTION_ADMIN, SESSION_VARIABLES_ADMIN, SET_USER_ID, SHOW_ROUTINE
, SYSTEM_USER, SYSTEM_VARIABLES_ADMIN, TABLE_ENCRYPTION_ADMIN, XA_RECOVER_ADMIN ON *.* TO 'adminsql'@'%' WITH GRANT OPTION
```

Gestión conexión base de datos desde NetBeans.

Conexión.

Para este punto necesitamos del driver adecuado.

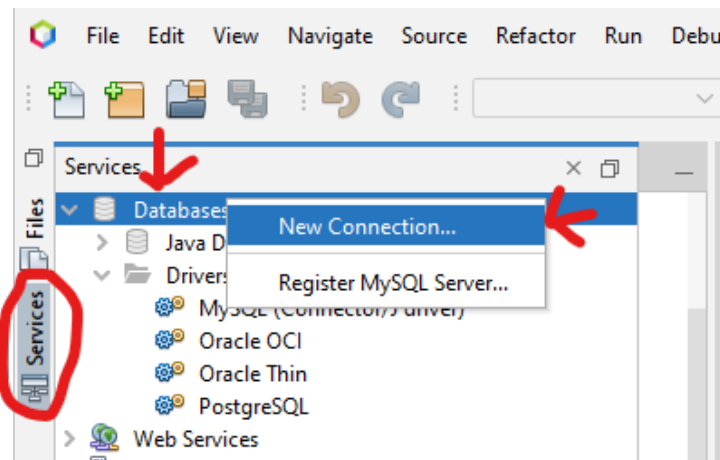
Enlace a descarga → [Driver mysql-connector-java.8.0.30.](http://dev.mysql.com/doc/connector-j/en/)

En concreto, para NetBeans necesitamos el archivo .jar.

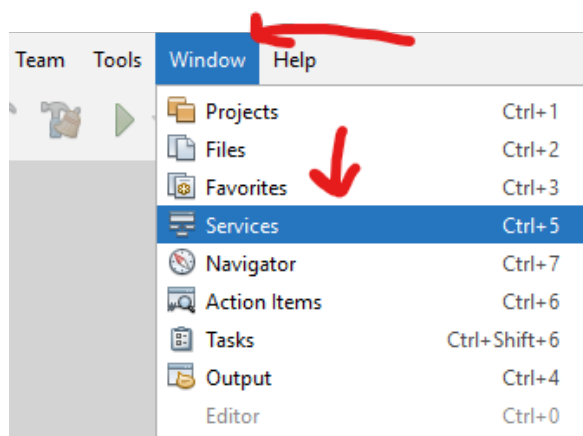
License	GPL 2.0
Categories	JDBC Drivers
Tags	database sql jdbc driver connector mysql
Organization	Oracle Corporation
HomePage	http://dev.mysql.com/doc/connector-j/en/
Date	Jul 25, 2022
Files	pom (2 KB) jar (2.4 MB) View All
Repositories	Central

Lo descargamos y guardamos donde nos convenga. Los siguientes pasos se realizan desde el propio NetBeans.

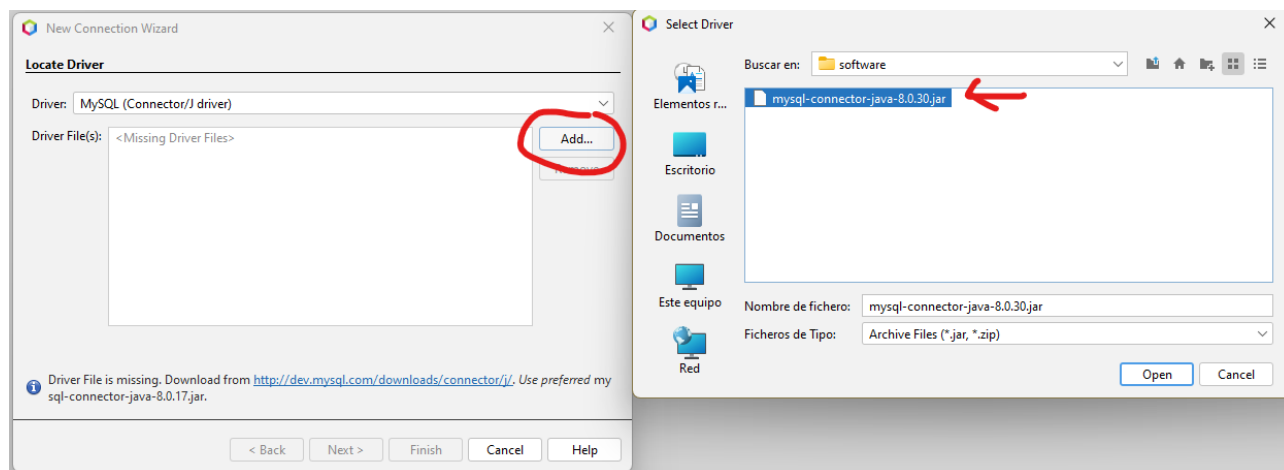
En la esquina superior izquierda seleccionamos la pestaña 'Services'.



Si no la hubiéramos personalizado y ubicado en este lugar, también podemos acceder a ella desde la ventana "Window" en la barra superior del IDE. O utilizando su atajo de teclado 'Ctrl +5'.

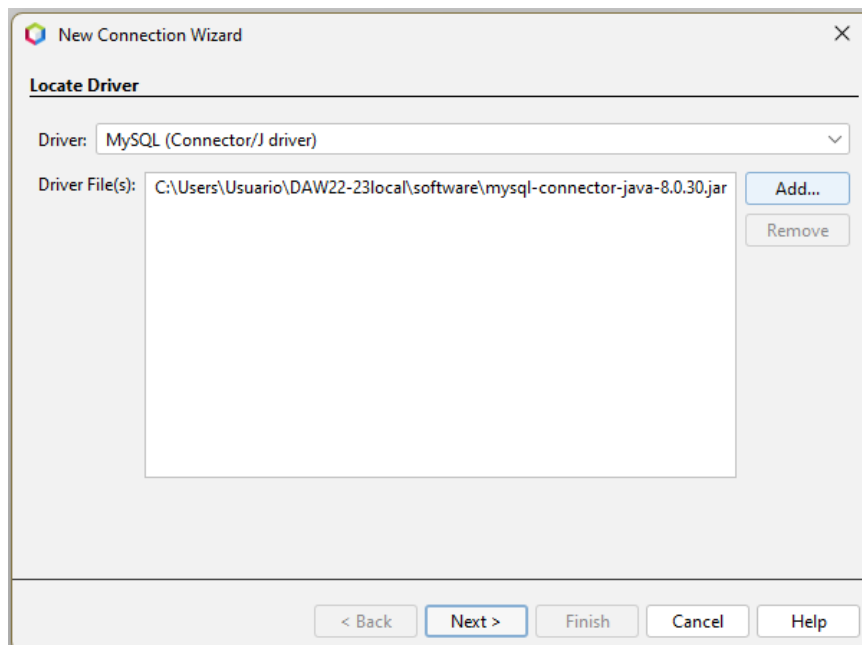


Tras pulsar 'New Connection' veremos este desplegable.

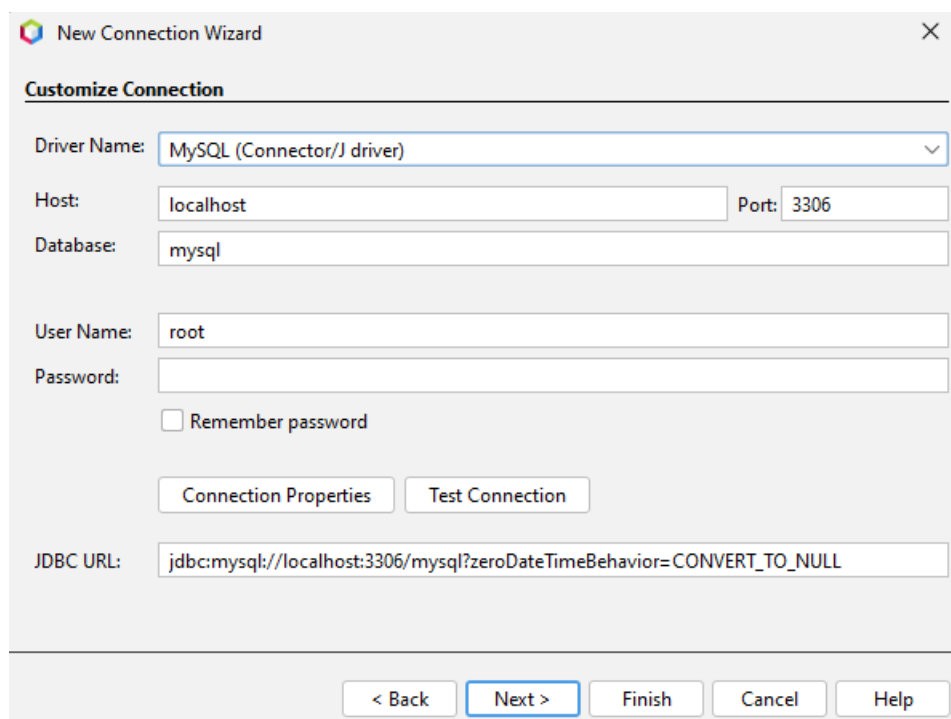


En el hemos de pulsar sobre 'Add'(añadir) y con el selector de ficheros buscar y seleccionar el archivo.jar que previamente hemos descargado.

Nuestra ventana ahora se verá así. Solo hay que dar a 'Next'.

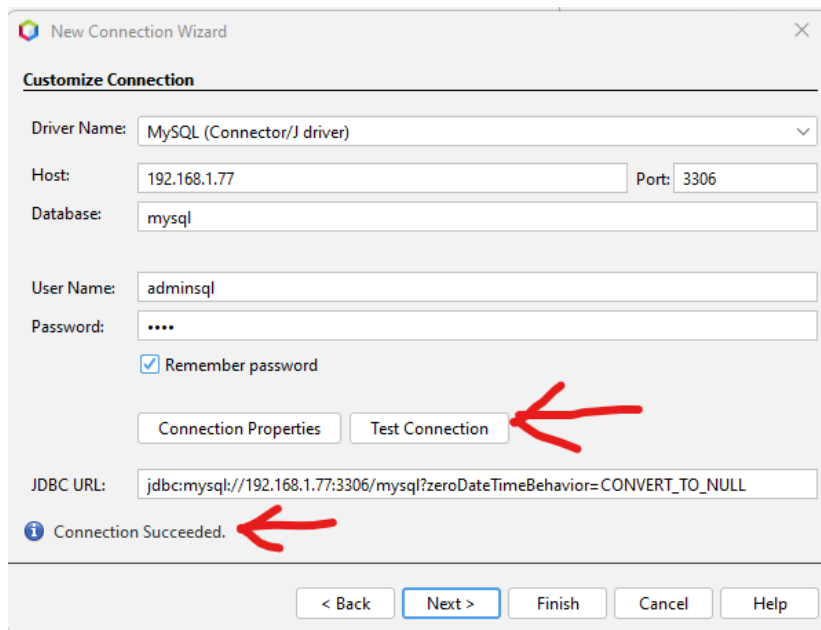


Estos son los valores por defecto que veremos.

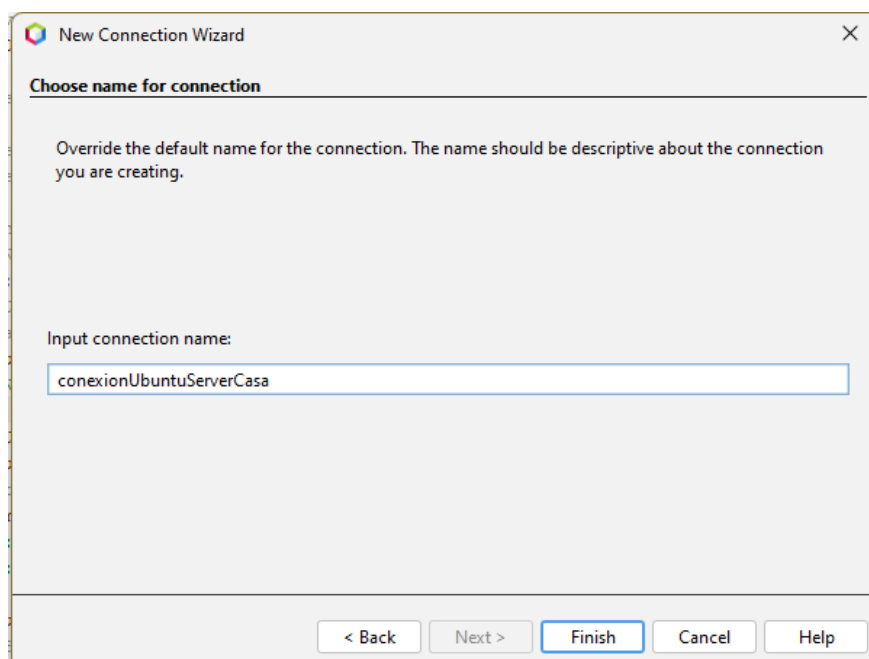


Y estos los valores necesarios para su conexión al entorno de desarrollo en mi máquina virtual Ubuntu server. Para su conexión al entorno de desarrollo sauces.local solo habría que cambiar la ip por la 192.168.20.19.

Antes de guardar esta configuración podemos probar si con estos datos hay comunicación entre el IDE y MySQL en el servidor presionando el botón 'Test Connection'; si hay conexión nos mostrará el mensaje correspondiente.

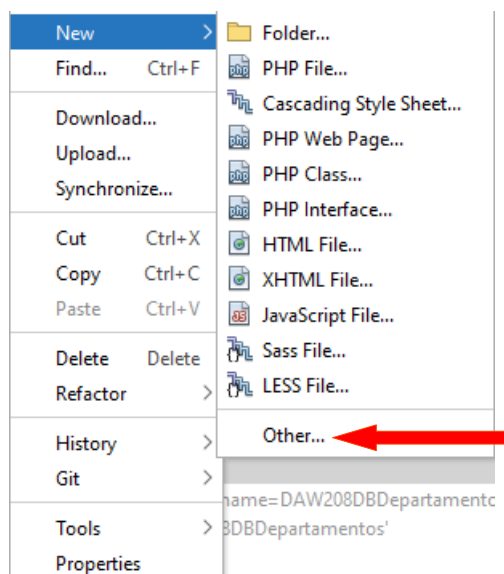


Lo siguiente es un paso a paso. Después de una pantalla en la que no haremos nada, aparece la siguiente. Aquí podemos dar un nombre significativo a la conexión para identificarla mejor con posterioridad.

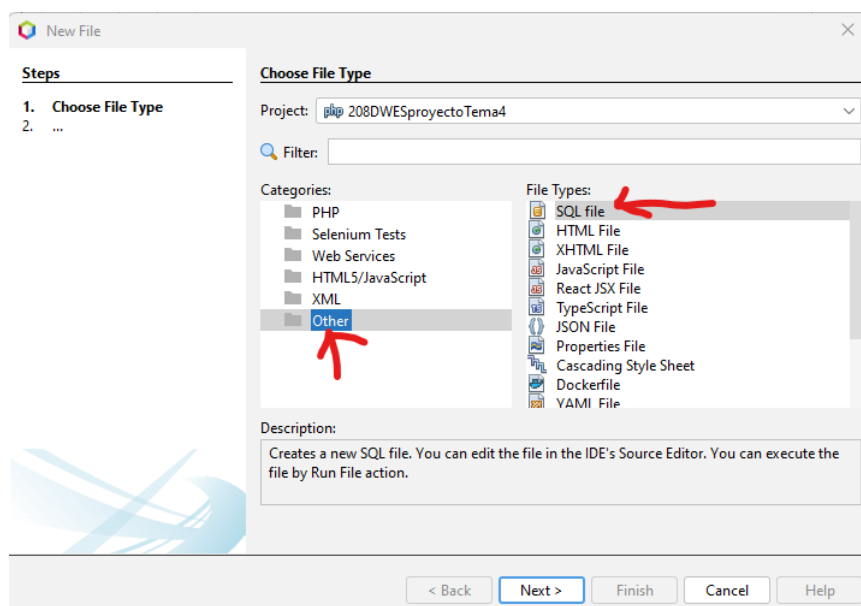


Creación, inserción y borrado Base de Datos.

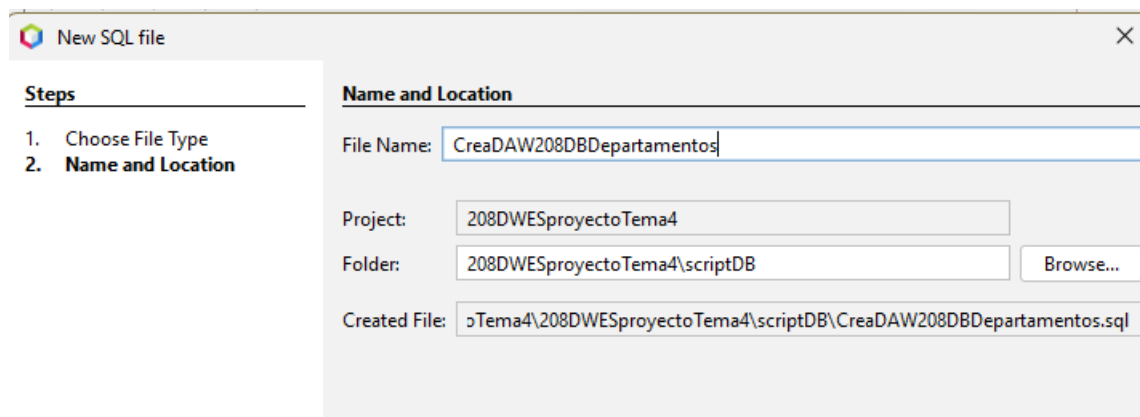
Si todo ha sido correctamente configurado ahora podemos ver en ‘Services’ nuestra nueva conexión y un desplegable en el que nos mostrará, de momento, las bases de datos por defecto y en un futuro las que vayamos creando. Para mostrar este punto voy a crear 3 archivos .sql dentro del propio NetBeans y que desde este IDE podrán ser ejecutados contra MySQL en nuestro servidor. El primer paso es, desde ‘Projects’ y con intención de ubicarlo en la carpeta que elijamos, ‘clickar’ sobre la carpeta, botón derecho del ratón y elegir ‘New’, si no nos aparece el .sql por defecto en el desplegable, hay que elegir ‘Other...’



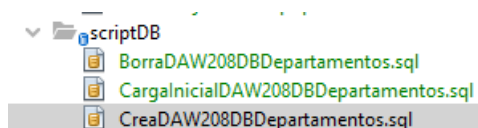
De nuevo ‘Other’ y ya podremos seleccionar el fichero .sql



Le damos nombre y dejamos la ubicación porque ya tiene la adecuada al haber comenzado su creación desde la carpeta en la que queremos que sea alojado.

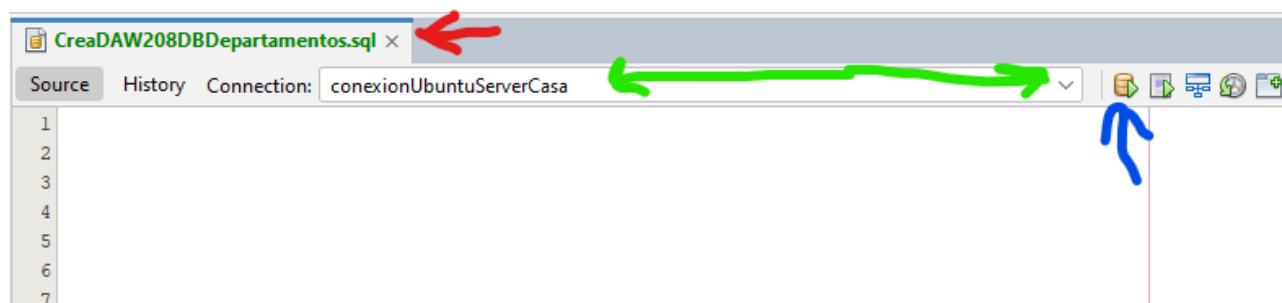


Seleccionamos el botón 'Finish' que aparecerá en la parte inferior de la ventana. Y tendremos el archivo creado y ubicado correctamente. Muestro en la imagen un total de 3 archivos creados de la misma forma. El primero de ellos contendrá el script de creación de una base de datos y un usuario administrador con permisos exclusivamente para esta base de datos. El segundo script contendrá las instrucciones para borrar este usuario y base de datos. Y el tercer script será el que una vez creada la DB, introduzca datos en esta.

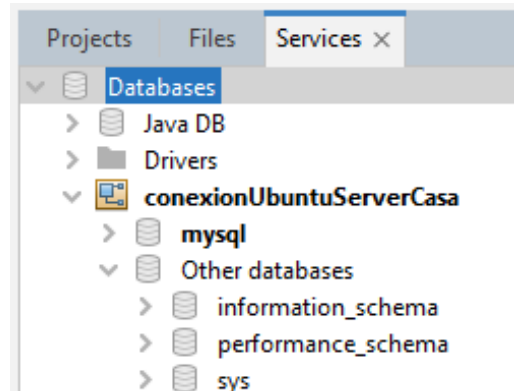


Si abrimos este primer fichero CreaDAW208DBDepartamento.sql veremos lo siguiente.

La flecha roja señala la pestaña abierta de nuestro archivo, la flecha verde señala el desplegable en el que hemos de elegir la conexión a través de la cual queremos ejecutar el archivo y la flecha azul señala el icono que ejecutará el script en nuestro sistema gestor de bases de datos.



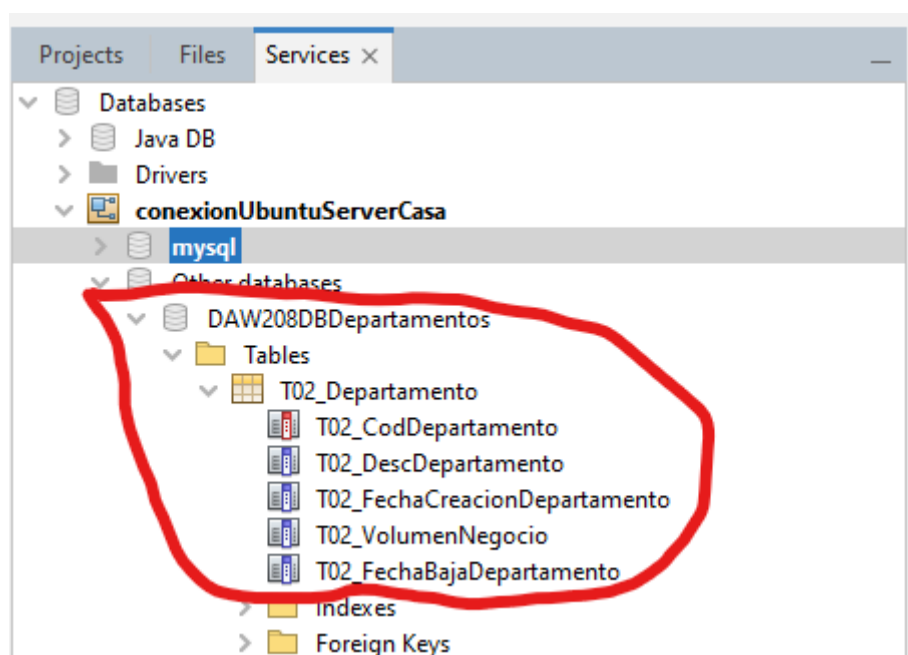
Antes no había ninguna base de datos creada por mí en mi gestor.



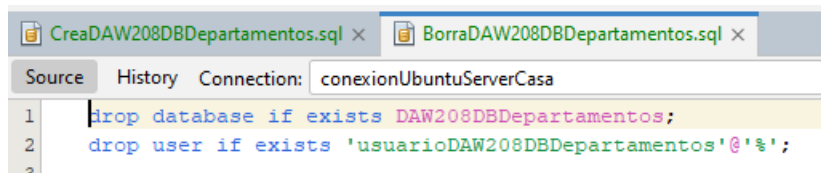
Y ahora, tras pulsar el botón de ejecutar script(flecha azul) y con las siguientes instrucciones en mi script de creación.

```
CreaDAW208DBDepartamentos.sql x
Source History Connection: conexionUbuntuServerCasa
1 create database if not exists DAW208DBDepartamentos;
2 use DAW208DBDepartamentos;
3 create table T02_Departamento(T02_CodDepartamento char(3) primary key,
4 T02_DescDepartamento varchar(255) not null, T02_FechaCreacionDepartamento int not null,
5 T02_VolumenNegocio float not null, T02_FechaBajaDepartamento int null)
6 engine=InnoDB;
7 create user if not exists 'usuarioDAW208DBDepartamentos'@'%' identified by 'paso';
8 grant all privileges on DAW208DBDepartamentos.* to 'usuarioDAW208DBDepartamentos'@'%';
```

Aparece la nueva base de datos con su tabla y campos correspondientes.

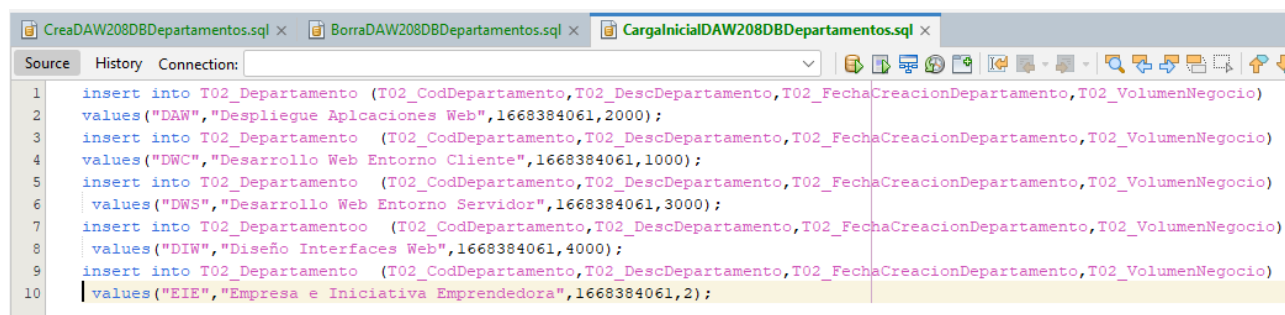


El script de borrado revertiría todo esto para que pudiéramos crear de nuevo la base de datos si hemos cometido algún error.



```
1 drop database if exists DAW208DBDepartamentos;
2 drop user if exists 'usuarioDAW208DBDepartamentos'@'%';
3
```

Y el de carga inicial dotaría de contenido a la tabla y sus campos.



```
1 insert into T02_Departamento (T02_CodDepartamento,T02_DescDepartamento,T02_FechaCreacionDepartamento,T02_VolumenNegocio)
2 values("DAW","Despliegue Aplicaciones Web",1668384061,2000);
3 insert into T02_Departamento (T02_CodDepartamento,T02_DescDepartamento,T02_FechaCreacionDepartamento,T02_VolumenNegocio)
4 values("DWC","Desarrollo Web Entorno Cliente",1668384061,1000);
5 insert into T02_Departamento (T02_CodDepartamento,T02_DescDepartamento,T02_FechaCreacionDepartamento,T02_VolumenNegocio)
6 values("DWS","Desarrollo Web Entorno Servidor",1668384061,3000);
7 insert into T02_Departamento (T02_CodDepartamento,T02_DescDepartamento,T02_FechaCreacionDepartamento,T02_VolumenNegocio)
8 values("DIW","Diseño Interfaces Web",1668384061,4000);
9 insert into T02_Departamento (T02_CodDepartamento,T02_DescDepartamento,T02_FechaCreacionDepartamento,T02_VolumenNegocio)
10 values("EIE","Empresa e Iniciativa Emprendedora",1668384061,2);
```

La sintaxis ha de ser revisada, pues el más mínimo error hará que nuestro script falle, en tal caso, nos avisaría en la ventana ‘Output’ con un mensaje como el que sigue.



```
Output
Remote Log: Remote Synchronization [208DWESproyectoTema4] x CreaDAW208DBDepartamentos.sql execution x CargalIncialDAW208DBDepartamentos.sql execution x

[1:1] Executed successfully in 0,013 s.
1 row affected.

[3:1] Executed successfully in 0,01 s.
1 row affected.

[5:1] Executed successfully in 0,005 s.
1 row affected.

[7:1] Failed in 0 s.
[Exception, Error code 1.146, SQLState 42S02] Table 'DAW208DBDepartamentos.T02_Departamento' doesn't exist
Line 7, column 1

[9:1] Executed successfully in 0,006 s.
1 row affected.

Execution finished after 0,06 s, 1 error occurred.
```

Hay que subsanar el error y sería conveniente borrar la base de datos, volver a ejecutar el script de creación y finalmente ejecutar este último script de inserción de datos debidamente corregido.

Hecho todo lo anterior, si ‘pinchamos’ sobre nuestra tabla en el desplegable y seleccionamos ‘View data’, nos mostrará lo siguiente.

En azul señalada la nueva pestaña que abrirá, en amarillo señalado el query que ha lanzado nuestra acción de 'clickar' sobre 'View Data' y en azul el resultado de la inserción de datos en nuestra tabla.

Connection: conexionUbuntuServerCasa

SQL 1 [conexionUbuntuServerCasa]

SELECT * FROM DAW208DBDepartamentos.T02_Departamento LIMIT 100;

SELECT * FROM DAW208DBDep...

Max. rows: 100 Fetched Rows: 5 Matching Rows:

#	T02_CodDepartamento	T02_DescDepartamento	T02_FechaCreacionDepartamento	T02_VolumenNegocio	T02_FechaBajaDepartamento
1	DAW	Despliegue Aplicaciones Web	1668384061	2.000	<NULL>
2	DIW	Diseño Interfaces Web	1668384061	4.000	<NULL>
3	DWC	Desarrollo Web Entorno Cliente	1668384061	1.000	<NULL>
4	DWS	Desarrollo Web Entorno Servidor	1668384061	3.000	<NULL>
5	EIE	Empresa e Iniciativa Emprendedora	1668384061	2	<NULL>

Comprobación conexión y muestra de datos desde servidor Ubuntu.

Ahora paso a comprobar desde el servidor si todo está allí también registrado y accesible para el usuario que hemos creado específico para esta base de datos. Para ello, pediré acceso a MySQL logándome como este usuario.

```
miadmin@RST-USED:~$ sudo mysql -u usuarioDAW208DBDepartamentos -p
[sudo] password for miadmin:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.31-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2022, 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> |
```

Allí selecciono ver las bases de datos presentes. Y me muestra la recién creada más dos presentes por defecto.

```
mysql> show databases;
+-----+
| Database |
+-----+
| DAW208DBDepartamentos |
| information_schema |
| performance_schema |
+-----+
3 rows in set (0,00 sec)
```

Allí haré un use + nombre de la base de datos para ponerla en uso , y también un describe de la tabla para ver su configuración.

```
mysql> use DAW208DBDepartamentos;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

Database changed

```
mysql> describe T02_Departamento;
```

Field	Type	Null	Key	Default	Extra
T02_CodDepartamento	char(3)	NO	PRI	NULL	
T02_DescDepartamento	varchar(255)	NO		NULL	
T02_FechaCreacionDepartamento	int	NO		NULL	
T02_VolumenNegocio	float	NO		NULL	
T02_FechaBajaDepartamento	int	YES		NULL	

```
5 rows in set (0,00 sec)
```

Con la instrucción select * from T02_Departamento nos mostrará las columnas de la tabla y todos los datos guardados en ellas.

```
mysql> select * from T02_Departamento;
```

T02_CodDepartamento	T02_DescDepartamento	T02_FechaCreacionDepartamento	T02_VolumenNegocio	T02_FechaBajaDepartamento
DAW	Despliegue Aplcaciones Web	1668384061	2000	NULL
DIW	Diseño Interfaces Web	1668384061	4000	NULL
DWC	Desarrollo Web Entorno Cliente	1668384061	1000	NULL
DWS	Desarrollo Web Entorno Servidor	1668384061	3000	NULL
EIE	Empresa e Iniciativa Emprendedora	1668384061	2	NULL

```
5 rows in set (0,00 sec)
```

```
mysql>
```

Continuará...

Xdebug.

A tener en cuenta que al instalar en nuestro servidor habremos de hacerlo como usuario “miadmin”, pues “operadorweb” no figura en el archivo sudoers y no podrá, por tanto, ejecutar comando aún con el comando sudo al principio.

```
$ sudo apt install php-xdebug
[sudo] password for operadorweb:
operadorweb is not in the sudoers file. This incident will be reported.
$
```

Ahora sí.

```
Last login: Sun Oct 16 15:55:02 UTC 2022 on tty1
miadmin@RST-USED:~$ sudo apt install php-xdebug_
```

Pide confirmación. Decimos que sí “S”.

```
[sudo] password for miadmin:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  php8.1-xdebug
Se instalarán los siguientes paquetes NUEVOS:
  php-xdebug php8.1-xdebug
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 625 kB de archivos.
Se utilizarán 1.881 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] ^[S_
```

Reiniciamos apache2 para aplicar cambios y confirmamos estado.

```
miadmin@RST-USED:~$ sudo service apache2 restart
miadmin@RST-USED:~$ sudo service apache2 status
• apache2.service – The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2022-10-16 18:35:18 UTC; 13s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 3073 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 3077 (apache2)
   Tasks: 6 (limit=8888)
```

[Enlace al resto del proceso de instalación](#) realizado desde este punto a caballo entre ,NetBeans y el propio Ubuntu Server.

Cuentas de desarrollo y hosting virtual.

Creación usuario de desarrollo “operadorweb”.

Comando **sudo useradd -d /var/www/html -g www-data operadorweb**

```
miadmin@RST-USED:~$ sudo useradd -d /var/www/html -g www-data operadorweb
```

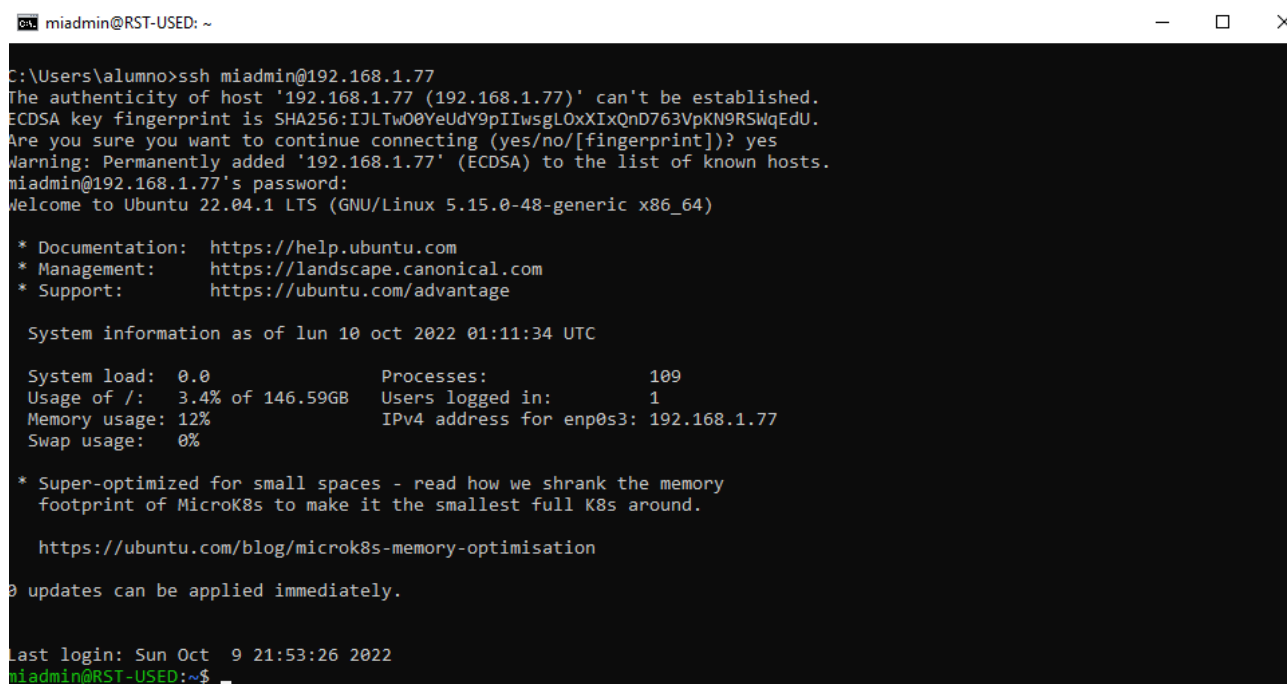
Estableciendo password para operadorweb con comando **sudo passwd operadorweb**.

```
miadmin@RST-USED:~$ sudo passwd operadorweb
New password:
Retype new password:
passwd: password updated successfully
miadmin@RST-USED:~$
```

A continuación, con el comando **cat /etc/passwd | grep operadorweb** el nombre del nuevo usuario, con una ‘x’ oculta su password encriptada, su UID, grupo a los que pertenece, directorio raíz del usuario y shell.

```
miadmin@RST-USED:~$ cat /etc/passwd | grep operadorweb
operadorweb:x:1001:33:/:var/www/html:/bin/sh
miadmin@RST-USED:~$
```

Para conectar desde el cmd de la máquina cliente Windows 10 a través de ssh.



```
miadmin@RST-USED: ~
C:\Users\alumno>ssh miadmin@192.168.1.77
The authenticity of host '192.168.1.77 (192.168.1.77)' can't be established.
ECDSA key fingerprint is SHA256:IJLTW00YeUdY9pIIwsgL0xXIxQnD763VpKN9RSWqEdU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.77' (ECDSA) to the list of known hosts.
miadmin@192.168.1.77's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-48-generic x86_64)

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

System information as of lun 10 oct 2022 01:11:34 UTC

System load:  0.0               Processes:            109
Usage of /:   3.4% of 146.59GB   Users logged in:     1
Memory usage: 12%              IPv4 address for enp0s3: 192.168.1.77
Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.
   https://ubuntu.com/blog/microk8s-memory-optimisation

0 updates can be applied immediately.

Last login: Sun Oct  9 21:53:26 2022
miadmin@RST-USED:~$
```

En Ubuntu pero desde el CMD de la máquina cliente. Esta opción es útil para administrar el servidor si al mismo tiempo se tiene abierta otra sesión de otro usuario(por ejemplo operadorweb) en el servidor, de esta manera podemos administrar el servidor sin necesidad de cerrar la sesión del otro usuario.

Cambiar propietario de carpeta y grupo.

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

A continuación, establecer nuevos permisos, incluido el sticky bit.

```
miadmin@RST-USED:~$ sudo chmod -R 2775 /var/www/html
miadmin@RST-USED:~$ ls -l /var/www
total 4
drwxrwsr-x 8 operadorweb www-data 4096 oct 24 17:42 html
```

Hosting virtual.

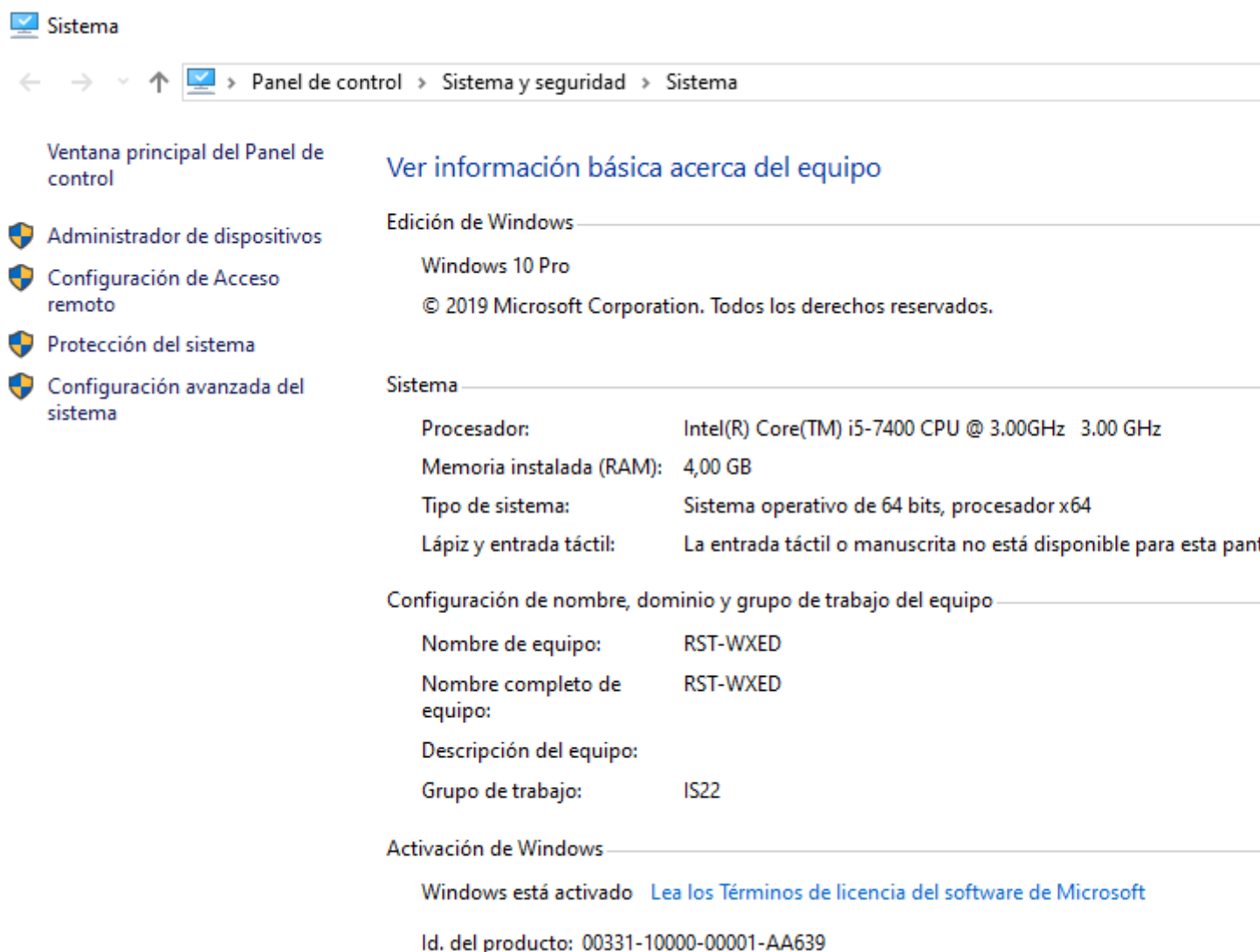
En construcción...

WXED – Windows X:

Instalación y configuración inicial de la máquina.

[La versión instalada es la de 64 bits.](#)

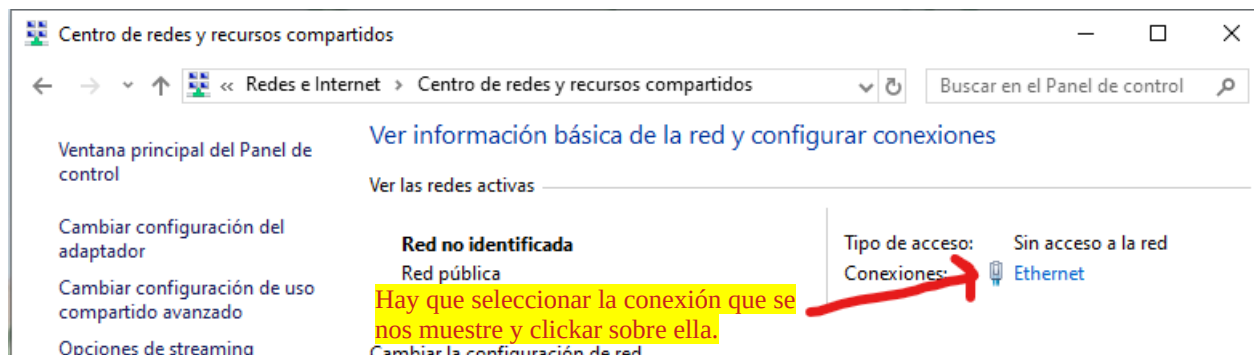
Datos equipo cliente Windows 10.



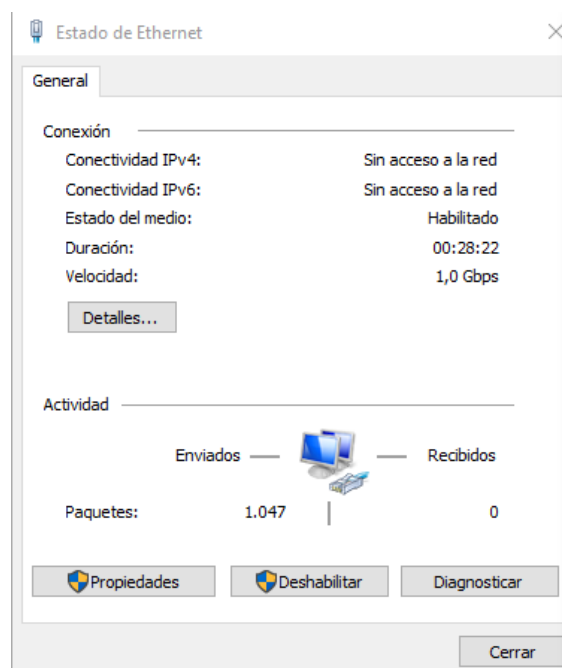
The screenshot shows the Windows 10 'Sistema' (System) window. The left sidebar contains links to 'Administrador de dispositivos', 'Configuración de Acceso remoto', 'Protección del sistema', and 'Configuración avanzada del sistema'. The main content area is titled 'Ver información básica acerca del equipo' and displays the following information:

- Edición de Windows:** Windows 10 Pro, © 2019 Microsoft Corporation. Todos los derechos reservados.
- Sistema:**
 - Procesador: Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz 3.00 GHz
 - Memoria instalada (RAM): 4,00 GB
 - Tipo de sistema: Sistema operativo de 64 bits, procesador x64
 - Lápiz y entrada táctil: La entrada táctil o manuscrita no está disponible para esta pantalla.
- Configuración de nombre, dominio y grupo de trabajo del equipo:**
 - Nombre de equipo: RST-WXED
 - Nombre completo de equipo: RST-WXED
 - Descripción del equipo:
 - Grupo de trabajo: IS22
- Activación de Windows:** Windows está activado. [Lea los Términos de licencia del software de Microsoft](#).
Id. del producto: 00331-10000-00001-AA639

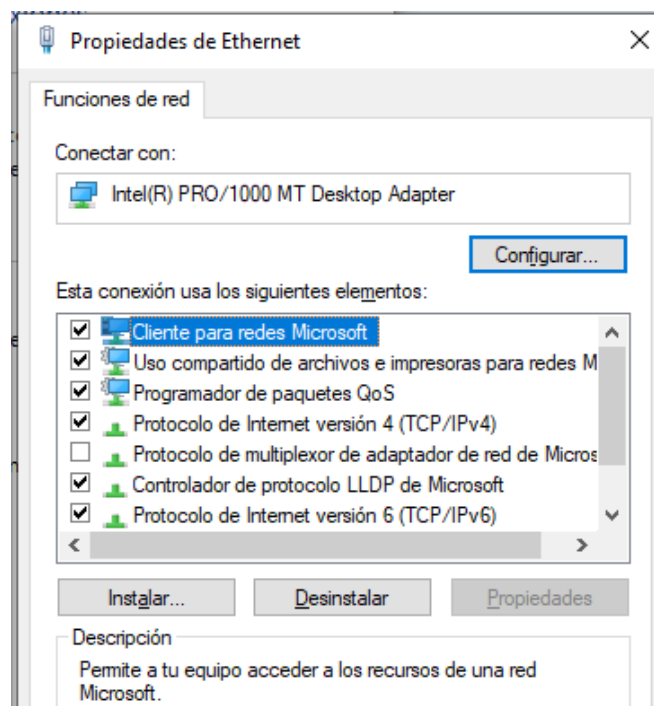
Una de las primeras acciones una vez instalado el SO Windows 10 en nuestra máquina virtual cliente, es configurar sus direcciones ip para poder conectarlo con el servidor e incluso con el exterior para poder descargarnos todas las herramientas que vamos a utilizar en el desarrollo.



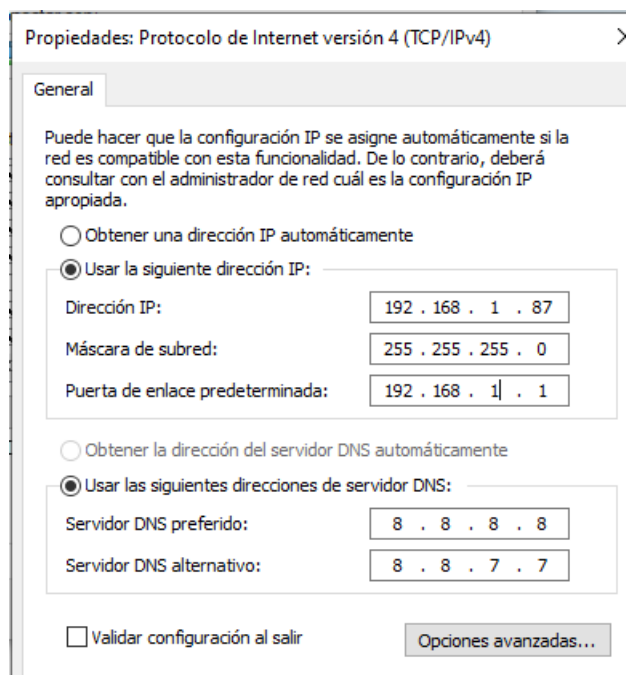
A continuación , en la nueva ventana hay que seleccionar “Propiedades”.



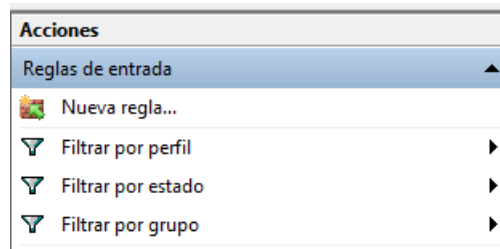
El siguiente paso es elegir el protocolo de internet TCP/IPv4.



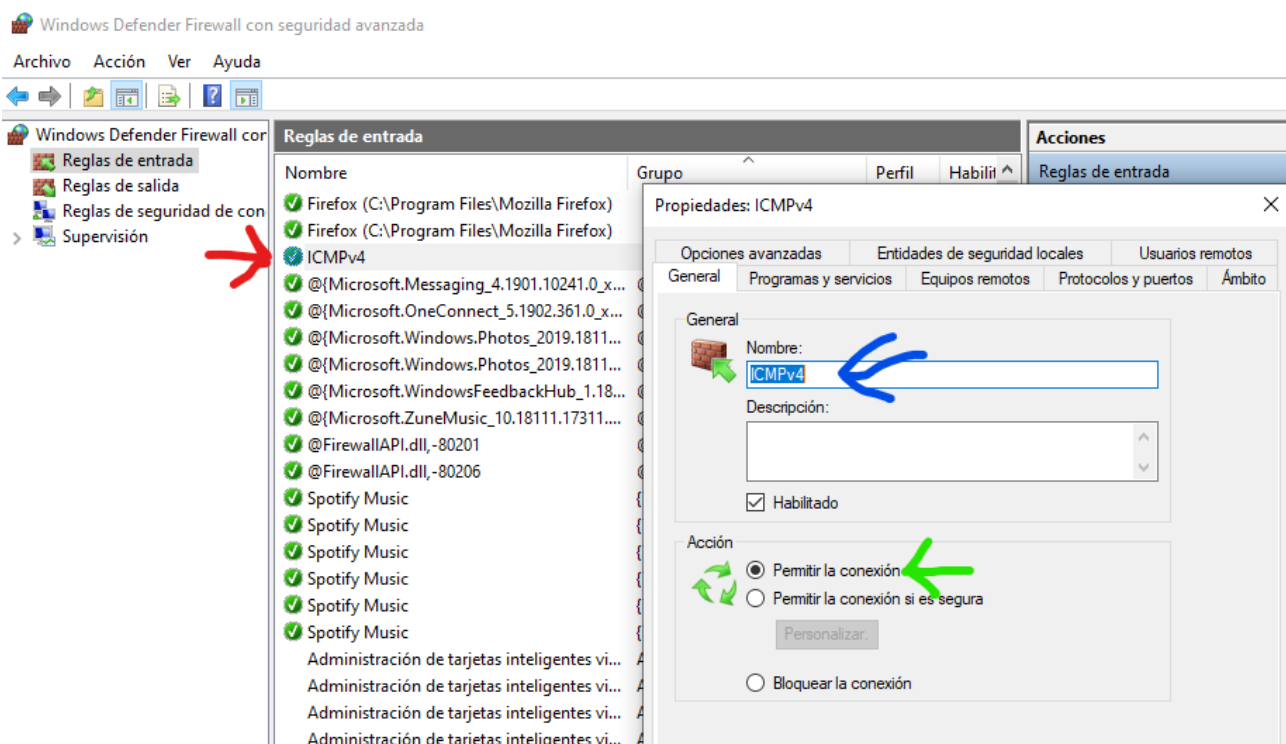
Establecer las direcciones ip.



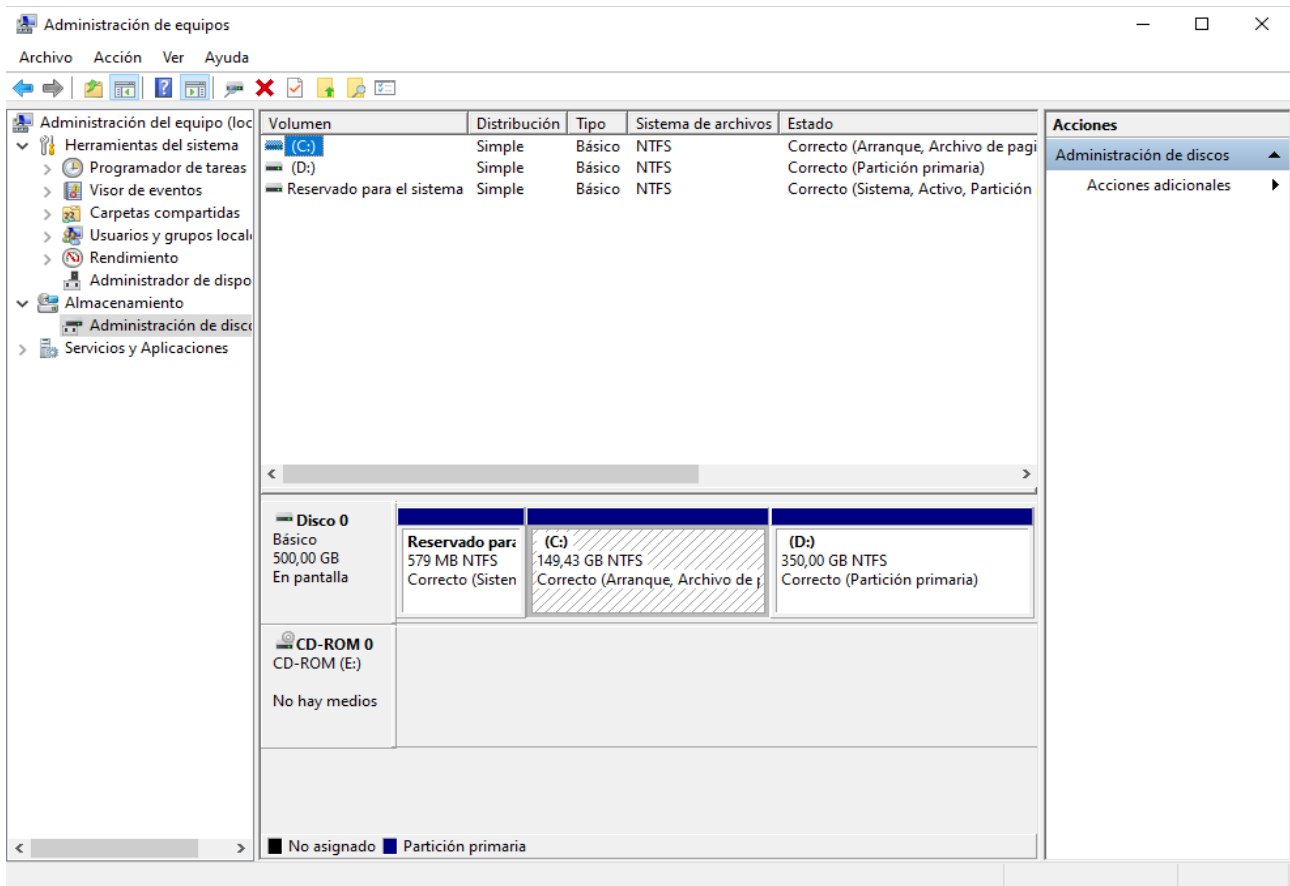
También hay que crear una regla ICMPv4 en la máquina cliente de Windows 10 para permitir el acceso desde otros programas y máquinas externas. Después de seleccionar “Reglas de entrada”, nos aparece a la derecha un desplegable “Acciones” en el que tenemos que seleccionar “Nueva regla...”



Especificaciones configuración regla ICMPv4

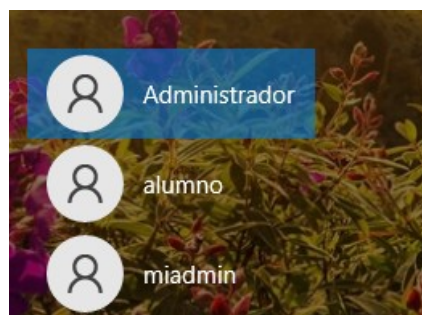


Actuando como usuario administrador muestro a continuación el estado de los discos de la máquina Windows 10.

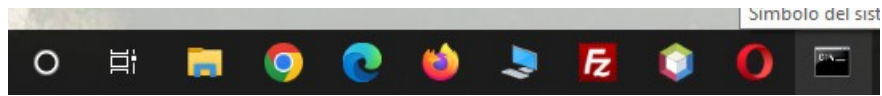


Cuentas administradoras y cuenta de desarrollador.

Tres usuarios, el usuario administrador que solo vamos a utilizar para instalar el SO, todo el software necesario y tareas iniciales de la instalación. Y dos usuarios, el usuario alumno y el usuario miadmin. El usuario alumno será el perfil de desarrollador con el que trabajaremos, por lo tanto, conviene personalizarlo con todas las herramientas que necesitemos. Navegadores, IDE`s, etc.

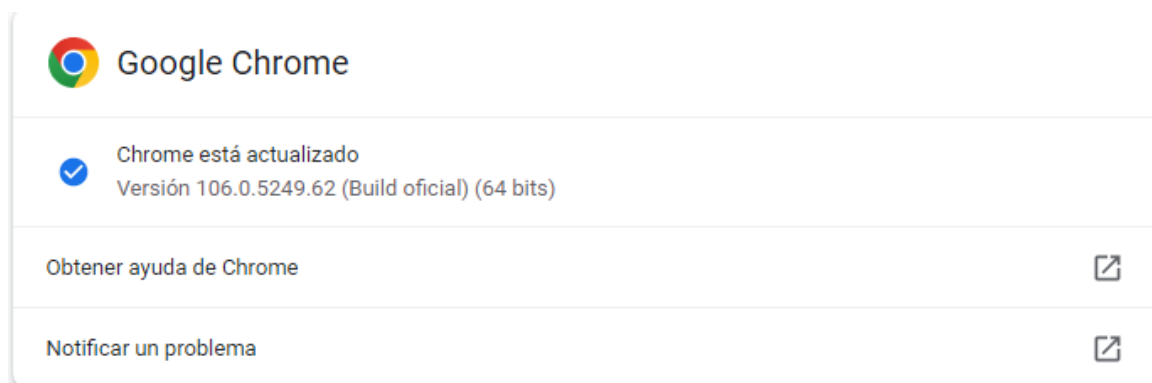


Barra de tareas personalizada para el alumno desarrollador.



Navegadores.

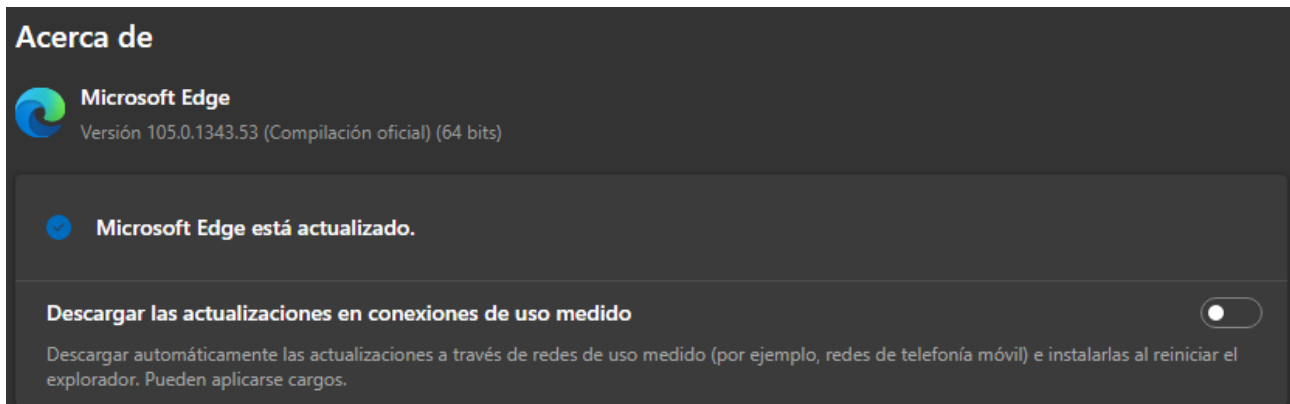
Google Chrome



Mozilla Firefox



Microsoft Edge



Opera.

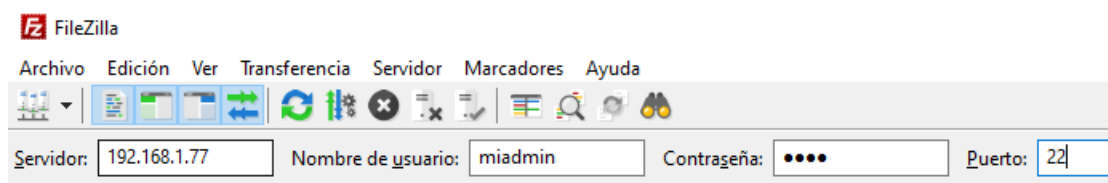


FileZilla.

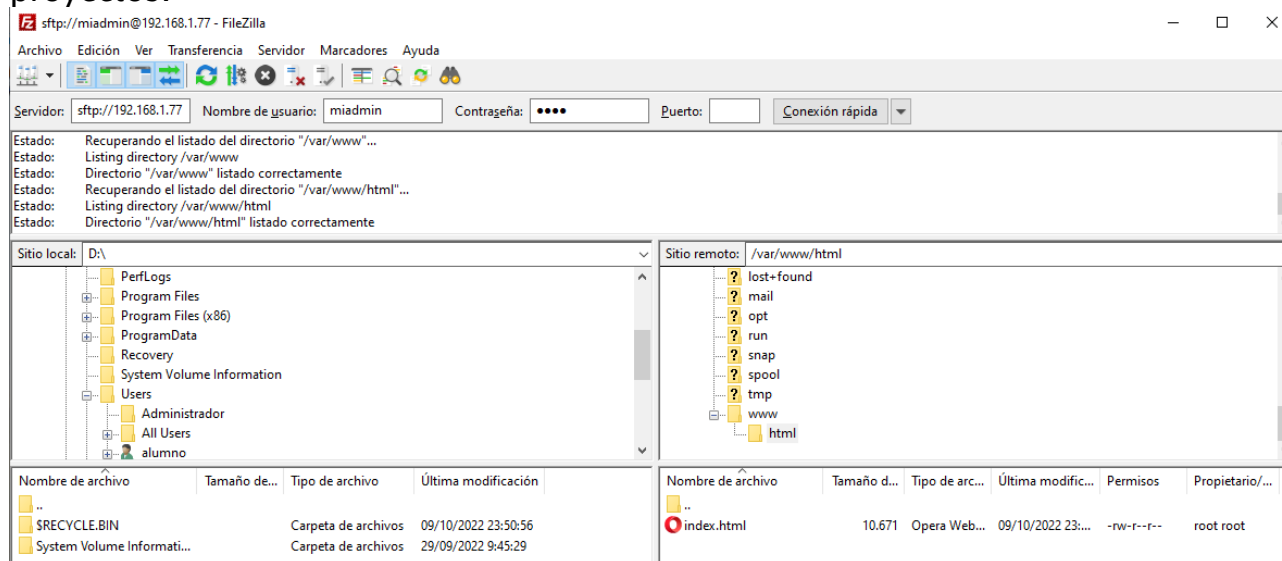
FileZilla vamos a utilizarlo para subir de manera sencilla archivos desde el área de desarrollo del equipo cliente al área de explotación ubicada en el Ubuntu server.



Lo primero es establecer una conexión indicándole ip del servidor, con que usuario nos vamos a conectar, contraseña de este usuario y puerto por el que nos conectaremos.



Conexión establecida, la conexión es sftp y se puede ver la división entre los archivos presentes en local a la izquierda y los remotos a la derecha de la siguiente ventana. En el directorio local a continuación crearemos nuestros proyectos.



Notepad++.

En construcción...

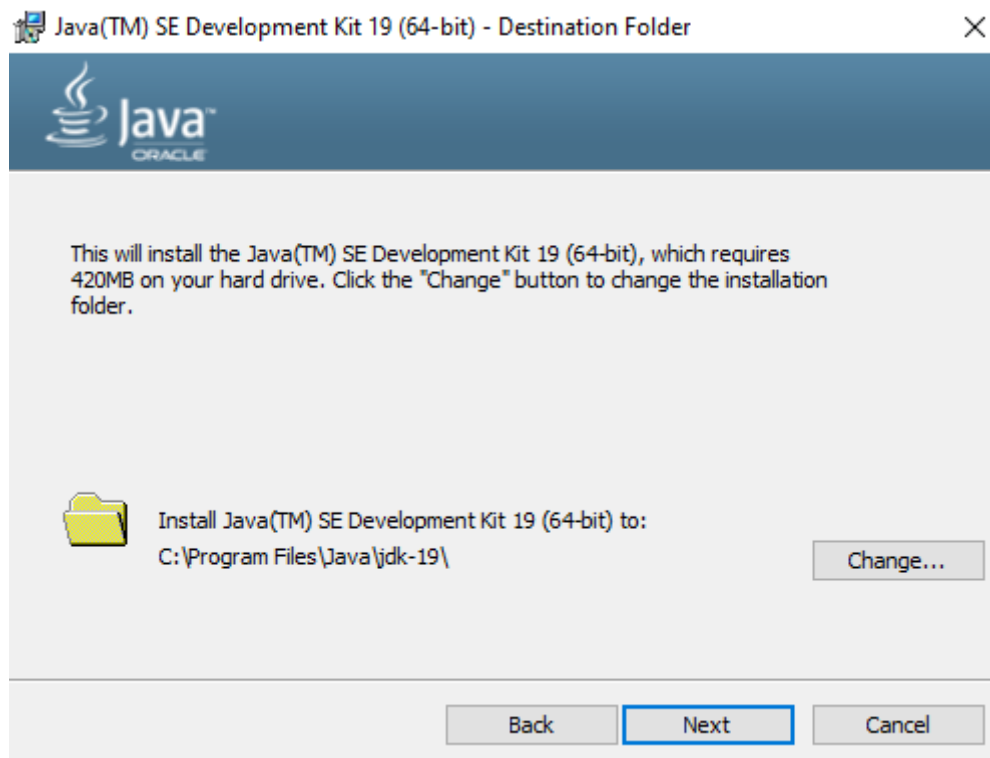
Netbeans:

Instalación y configuración inicial (plugings).

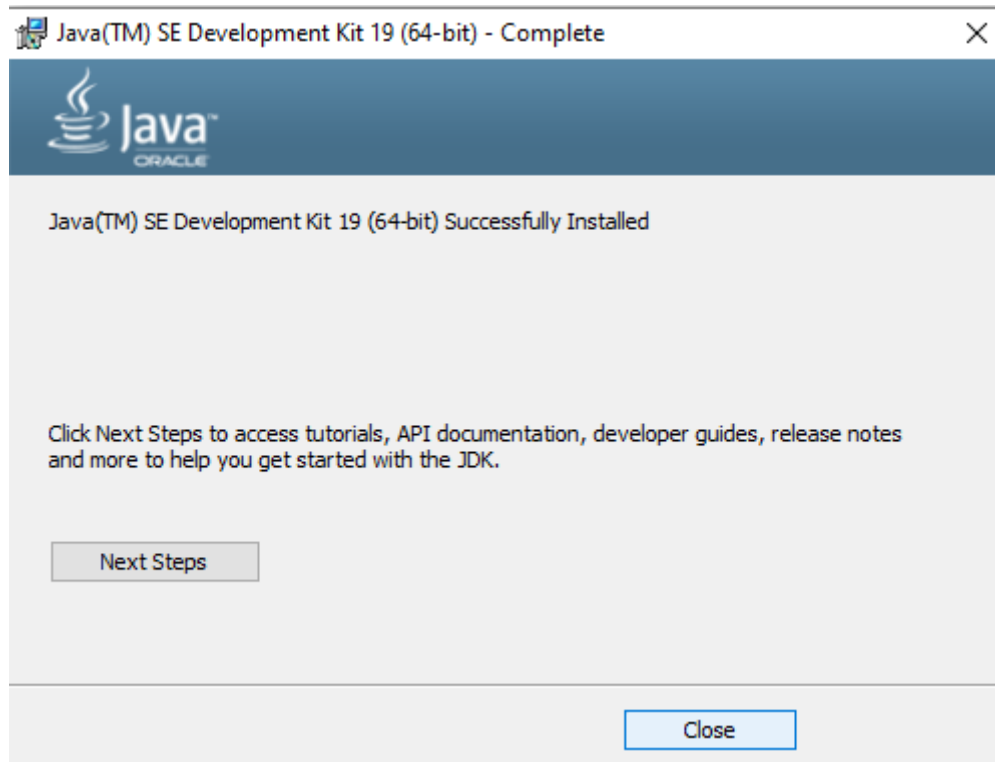
Instalo con cuenta administrador y utilizo con cuenta usuario.

El primer requisito es instalar el **JDK** para Windows de 64 bit, en este caso es la versión 19.

La instalación es un sencillo paso a paso.



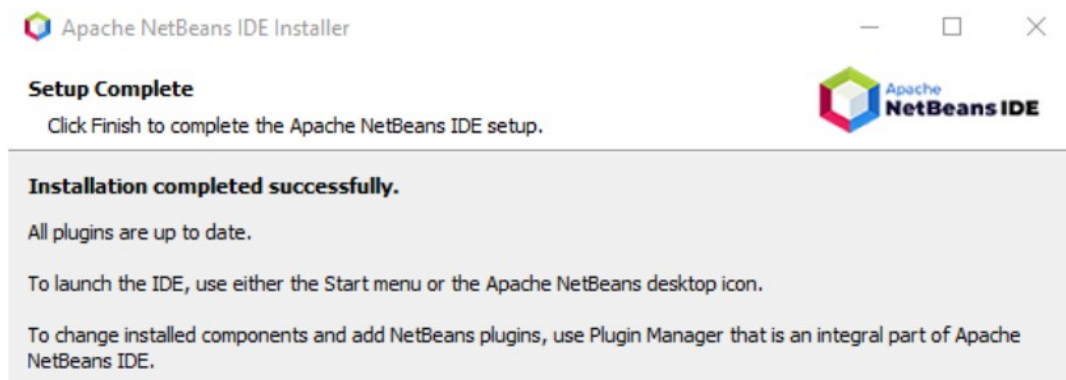
Hecho.



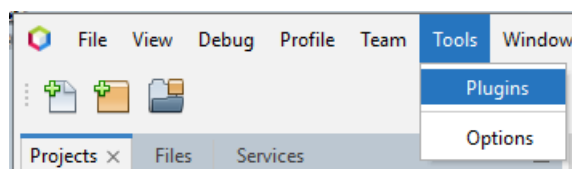
Una vez instalado el JDK pasamos a instalar el propio Netbeans. Instalamos la versión 15 para Windows de 64 bits. Al comenzar la instalación nos muestra los pluggins que vienen instalados por defecto.



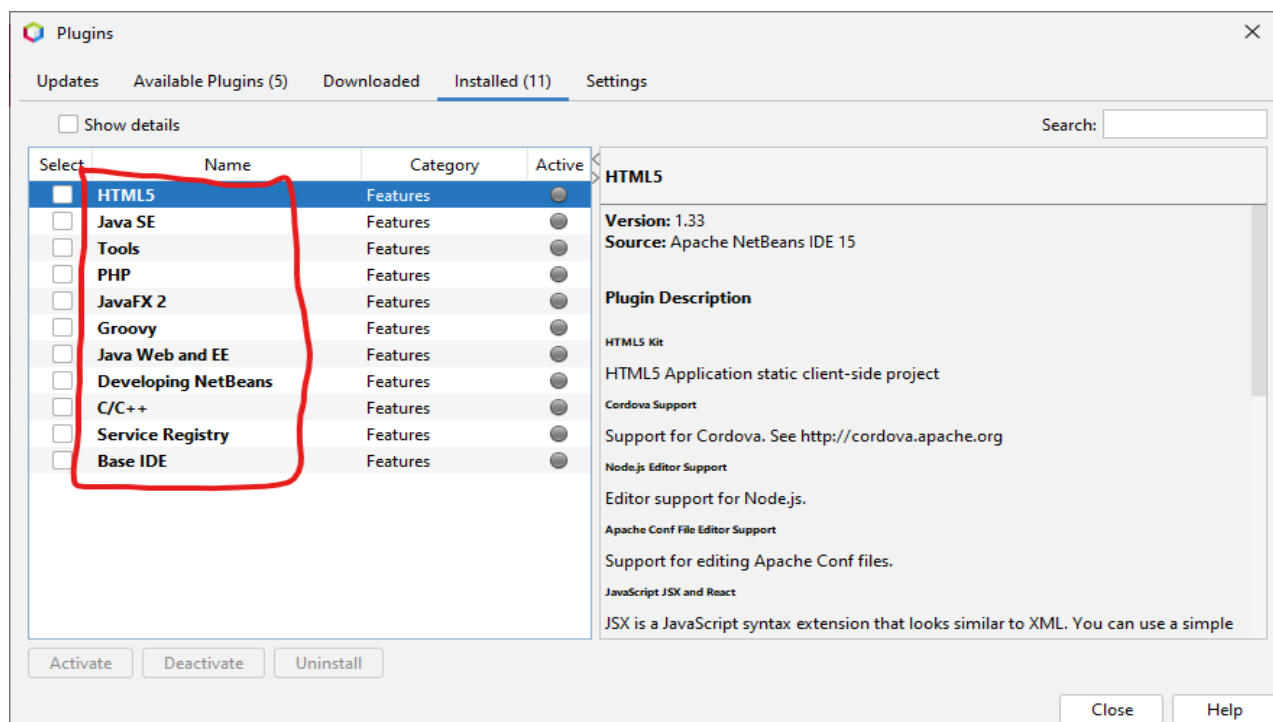
La instalación es un paso a paso eligiendo las opciones por defecto, al terminar la instalación con éxito, nos muestra la siguiente imagen para confirmarlo.



Ubicación de la gestión de pluggins de Neatbeans.



A través de la pestaña “installed” podemos ver los 11 pluggins que NetBeans instaló por defecto así como una breve descripción de cada uno de ellos.



En la pestaña “Available plugins” se nos muestran algunas sugerencias, también podemos realizar nuestra propia búsqueda en internet, en la página oficial de [Oracle](#), descargar el plugin que nos interese y este aparecerá en la citada pestaña de “Available pluggins”.

Hay infinidad de ellos, también se pueden descargar de páginas no oficiales. Poco recomendable.

Most downloaded	
1. Color Codes Preview	143,702
2. NB SpringBoot	143,005
3. Rainbow Braces	107,108
4. HTML Enhancements	96,562
5. GitHub Issues Support	95,560

Elijo [uno que previsualiza](#) en una barra lateral los colores aplicados mediante CSS a las diferentes etiquetas HTML.

Homepage: <https://github.com/junichi11/netbeans-color-codes-preview>
★ 2020-06-18 2022-09-27 143,702

Code Generation Editing

Download matrix

Plugin version	NetBeans compatible
0.13.1	
0.13.2	NB 12.0
0.13.3	NB 12.0
0.13.4	NB 15 - Verification pending NB 14 NB 13 - Verified NB 12.0 - Verified

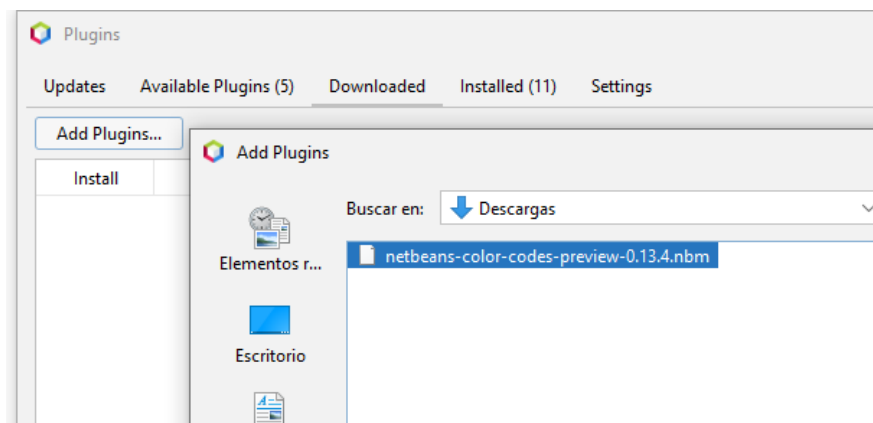
```

1  p.sample {
2      color: #000000;
3      background-color: #99cc00;
4      background: -moz-linear-gradient(#f9f9f9, #e5e5e5);
5      background: -o-linear-gradient(#f9f9f9, #e5e5e5);
6      background: -webkit-gradient(linear, 0% 0%, 0% 100%,
7      background: -webkit-linear-gradient(#f9f9f9, #e5e5e5);
8  }
9
10 p.sample2 {
11     color: #b81212;
12     background-color: #0000ff;
13     background: -moz-linear-gradient(#a00, #0a0);
14 }

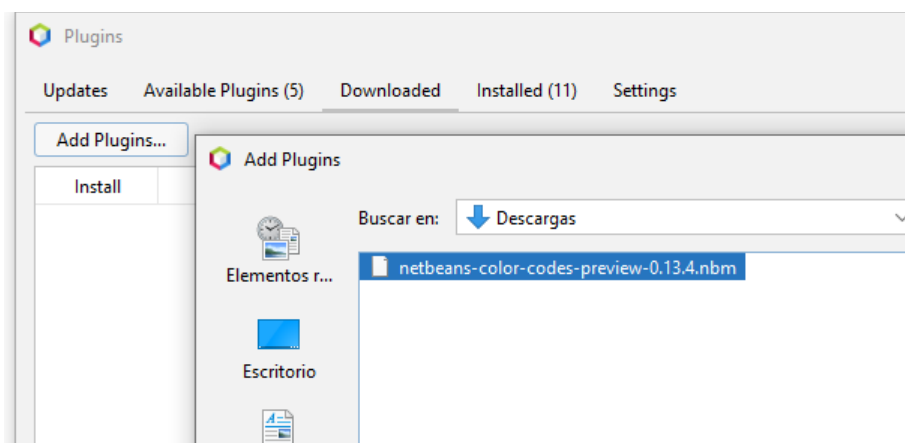
```

#aa0000
#00aa00

Ahora, hay que ir otra vez a la ventana de Tools/Plugins y allí, a la pestaña “Downloaded”, seleccionar “Add plugins...”, seleccionar el plugin descargado buscándolo con el selector de archivos que nos abre Netbeans e instalarlo.



Tras seguir un breve paso a paso, ahora aparece en la pestaña “Installed”.



Aunque estén instalados, no están necesariamente activos, si queremos que pasen a estar activos se lo tenemos que especificar en esta misma ventana seleccionando cada plugin que nos interese y pulsando el botón “Activate”.

Select	Name	Category	Active
<input checked="" type="checkbox"/>	HTML5	Features	<input type="radio"/>
<input type="checkbox"/>	User Installed Plugins	Features	<input checked="" type="radio"/>
<input type="checkbox"/>	Java SE	Features	<input type="radio"/>
<input checked="" type="checkbox"/>	Tools	Features	<input type="radio"/>
<input checked="" type="checkbox"/>	PHP	Features	<input type="radio"/>
<input type="checkbox"/>	JavaFX 2	Features	<input type="radio"/>
<input type="checkbox"/>	Groovy	Features	<input type="radio"/>
<input type="checkbox"/>	Java Web and EE	Features	<input type="radio"/>
<input type="checkbox"/>	Developing NetBeans	Features	<input type="radio"/>
<input type="checkbox"/>	C/C++	Features	<input type="radio"/>
<input type="checkbox"/>	Service Registry	Features	<input type="radio"/>
<input checked="" type="checkbox"/>	Base IDE	Features	<input type="radio"/>

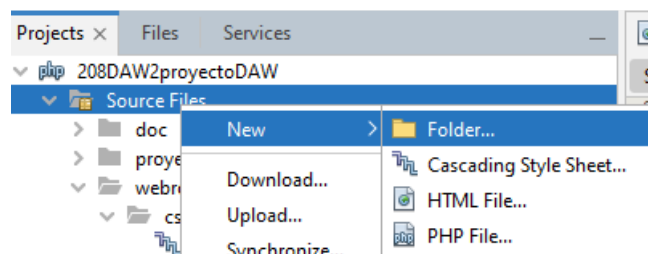
Ahora están activos y listos para facilitarnos el trabajo.

Select	Name	Category	Active
<input checked="" type="checkbox"/>	HTML5	Features	
<input type="checkbox"/>	User Installed Plugins	Features	
<input type="checkbox"/>	Java SE	Features	
<input type="checkbox"/>	Tools	Features	
<input type="checkbox"/>	PHP	Features	
<input type="checkbox"/>	JavaFX 2	Features	
<input type="checkbox"/>	Groovy	Features	
<input type="checkbox"/>	Java Web and EE	Features	
<input type="checkbox"/>	Developing NetBeans	Features	
<input type="checkbox"/>	C/C++	Features	
<input type="checkbox"/>	Service Registry	Features	
<input type="checkbox"/>	Base IDE	Features	

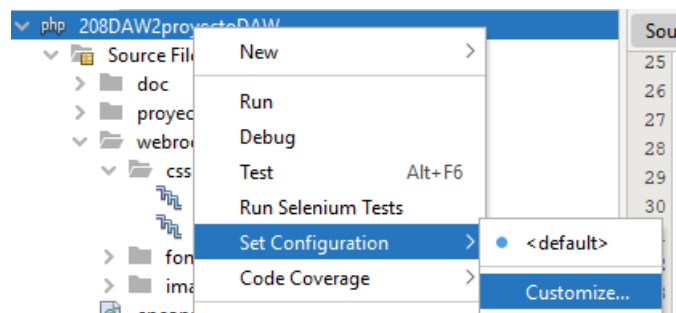
Comprobar y documentar el funcionamiento del plugin descargado e instalado, lo voy a realizar en el apartado “Creación de un proyecto nuevo” aprovechando que ya tendré un proyecto en el que poder hacerlo.

Conexión al entorno de desarrollo.

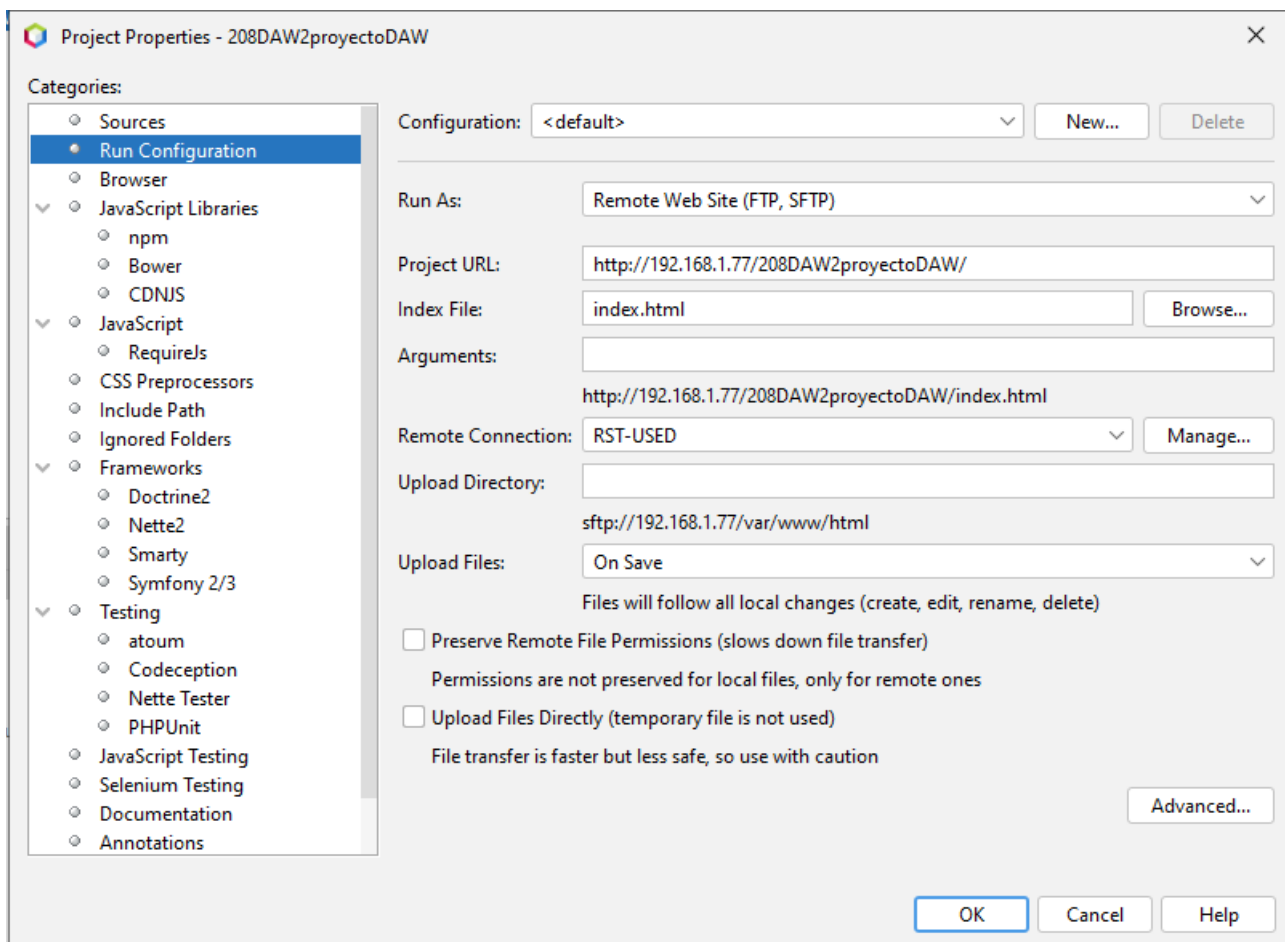
Conviene en este punto crear nuestro primer archivo, index.html, situado en la raíz de 208DAW2proyectoDAW.



Volvemos a “run configuration” seleccionando nuestro proyecto, hacemos click derecho y aparecen estos dos menús desplegables.



En la siguiente imagen se ve una primera pantalla con los datos necesarios para la conexión, aunque será en la pantalla que se nos muestra al pulsar el botón “Manage...” donde estableceremos la conexión propiamente dicha.



A continuación explico detalladamente el contenido y uso de cada una de las pestañas.

1ª pestaña.- **Configuration**: Su valor por defecto es ‘default’, una vez que creamos una configuración, podremos seleccionarla desde aquí.

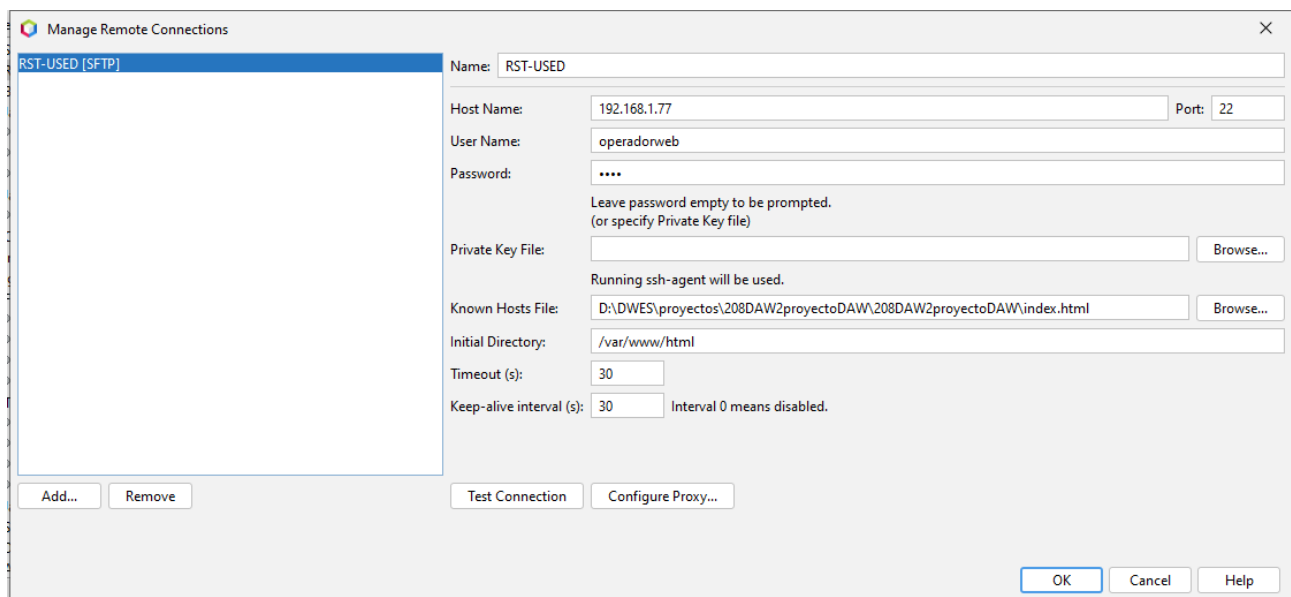
2ª pestaña.- **Run As**: Elegir Remote Web Site (FTP,SFTP) en el desplegable.

3ª pestaña.- **Project URL**: Hemos de especificar la dirección ip del servidor seguida de la carpeta raíz de nuestro proyecto en el cliente.

4ª pestaña.- **Index File**: Podemos escribir el nombre y extensión del archivo que se ejecutará y mostrará cada vez que hagamos una llamada http a nuestro servidor.

5ª pestaña.- **Arguments:** Queda como está, vacía.

6ª pestaña.- **Remote Connection:** Podemos seleccionar en el desplegable alguna conexión ya configurada o pulsar el botón “Manage”. Si optamos por esto último (imprescindible en caso de que no tengamos ninguna conexión previamente configurada) , nos aparece una segunda pantalla “ **Manage Remote Connections**”. En ella tenemos que dar un nombre a la conexión en “**Name**”, la ip en “**Host Name**” , nombre de usuario en “**User Name**” y password de este en “**Password**”. También al seleccionar “**Browse**” de “**Known Hosts File**”, a través de otra nueva pantalla podemos darle la ruta del archivo es el que se cargará al establecer la conexión.

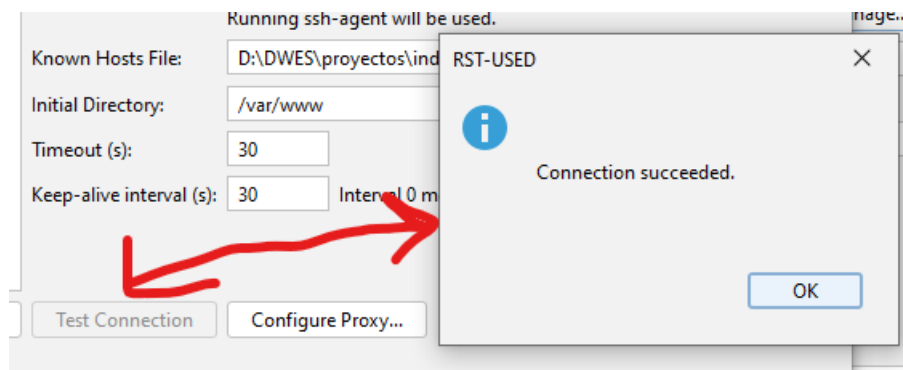


7ª pestaña.(Seguimos con la pantalla “Run Configuration”- “**Upload Directory**” :

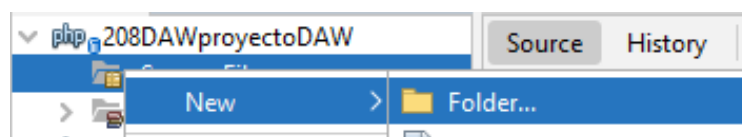
No escribiremos nada en ella, pues se autorrellena el valor con los datos ofrecidos en anteriores pestañas. Aquí nos aparecerá si está bien configurado la dirección del directorio del servidor remoto donde estará alojado nuestro proyecto.

8ª pestaña.- “**Upload Files**”: Aquí le decimos a NetBeans asociado a que evento ha de realizar las subidas de cambios al servidor. Recomendable seleccionar el subir archivos al guardar (On Save), por defecto está al ejecutarse (On Run).

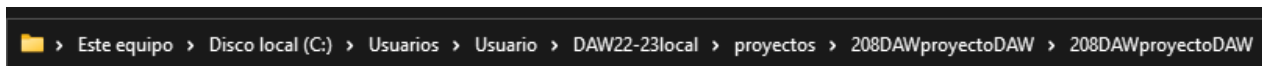
Con todo configurado podemos presionar OK y guardaremos nuestras elecciones, pero antes conviene hacer un “Test connection” y ver si los datos son correctos y se establece la conexión.



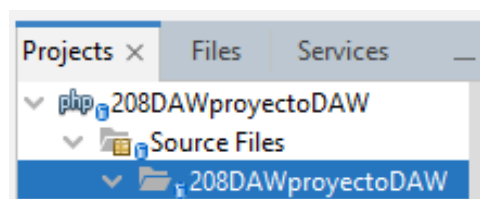
Una vez creado el proyecto y conectado al servidor, hay que crear una carpeta en “Source files” dentro de nuestro proyecto y darle el mismo nombre que al proyecto. De esta forma, esta carpeta que es la que va a contener todos los ficheros y directorios del proyecto, se nos mostrará en el directorio de publicación en el servidor y no hará mucho más fácil identificar el proyecto en remoto.



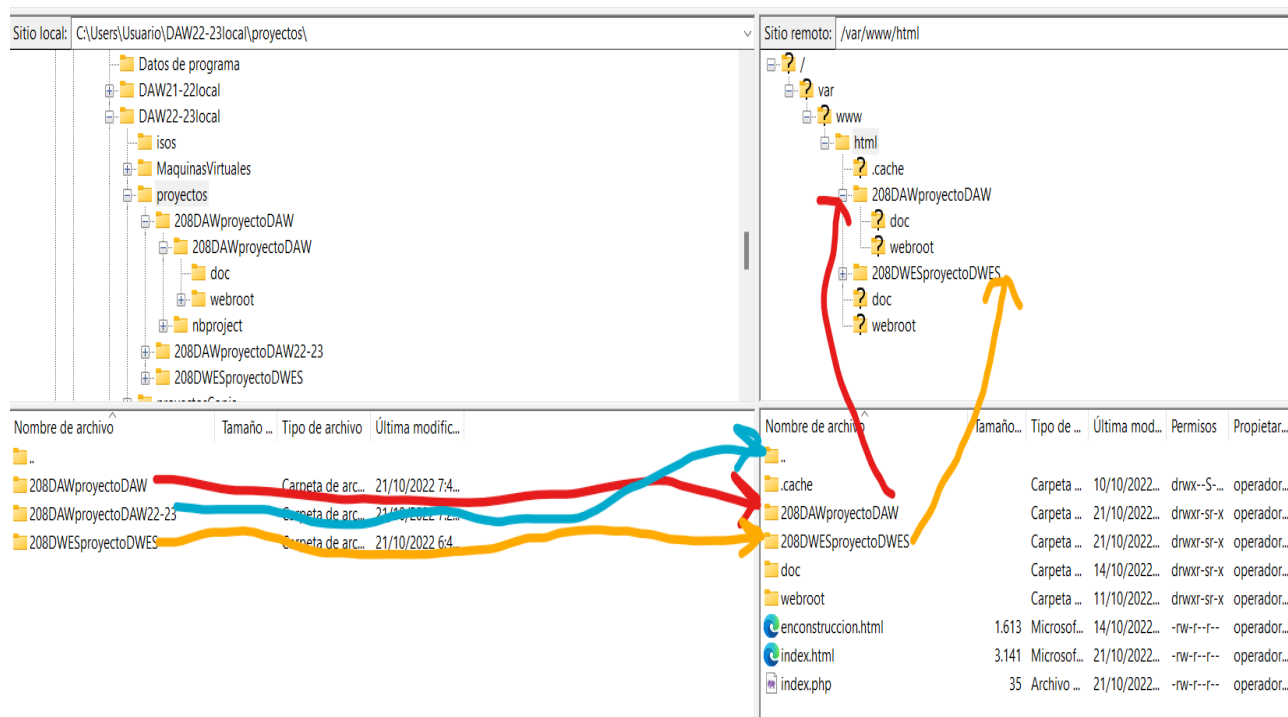
Esta será la estructura creada en local (en la imagen se ve en mi equipo de casa, por lo tanto la ruta difiere respecto a clase).



Esta es la estructura visualizada en NetBeans.



Y a continuación, gracias a Filezilla muestro juntas con el objetivo de poder compararlas, la estructura en local y en remoto.



Administración de la base de datos.

En construcción...

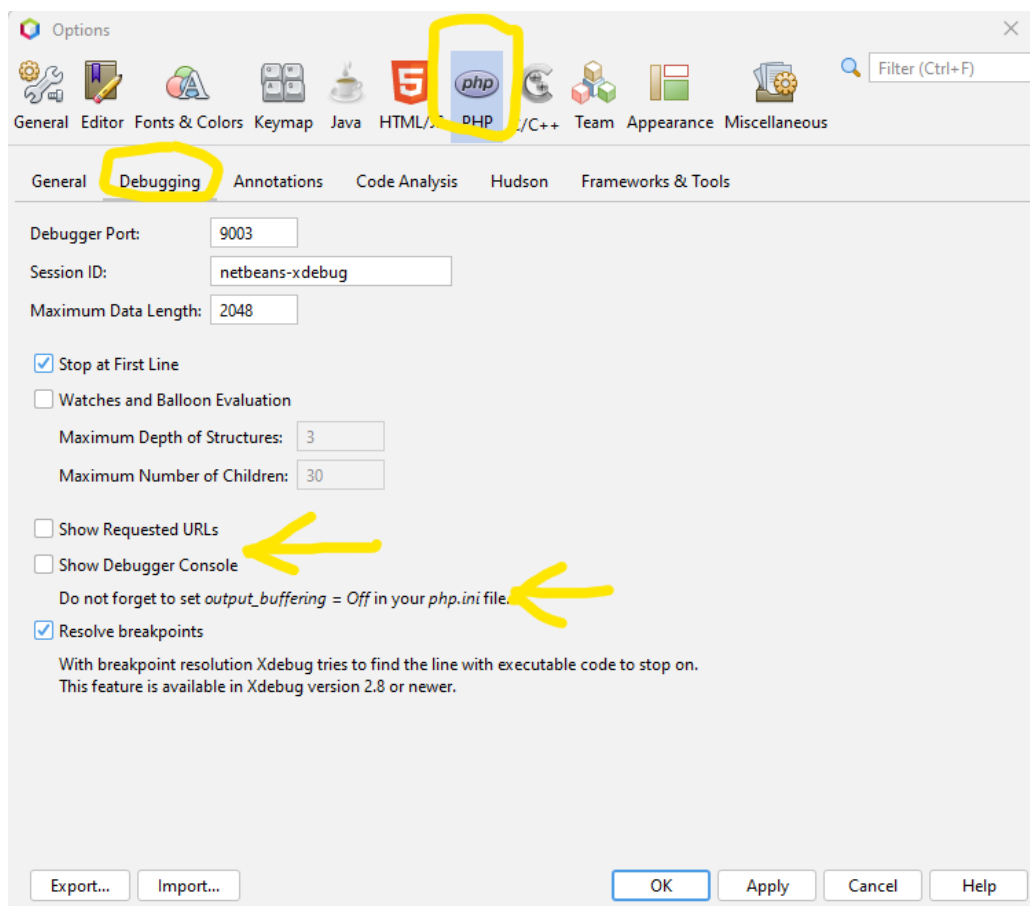
Conexión al repositorio – versionado.

En construcción...

Depuración – Configuración de la ejecución para la depuración.

Instalado ya “Xdebug” en el servidor, toca configurarlo en NetBeans.

En la pestaña “Tools”, presente en la barra superior elegimos “options” en el desplegable y nos presenta la siguiente pantalla.



Dejo marcadas las elecciones que nos conciernen. En primer lugar hay que seleccionar el lenguaje PHP, después la pestaña “Debugging”, hay que dejar seleccionada la casilla “Show Debugger Console” (para que podamos ver gráficamente en nuestro IDE el proceso de depuración del código PHP) y prestamos atención a la advertencia de no olvidar configurar el parámetro indicado en el fichero php.ini en nuestro servidor.

Regreso a continuación en el servidor hasta el directorio raíz para mostrar la ruta absoluta del archivo y desde allí abro para edición el php.ini.

```
miadmin@RST-USED:~$ cd ../../
miadmin@RST-USED:/$ dir
bin  dev  home  lib32  libx32  media  opt  root  sbin  srv  sys  usr
boot  etc  lib  lib64  lost+found  mnt  proc  run  snap  swap.img  tmp  var
miadmin@RST-USED:/$ sudo nano cd /etc/php/8.1/apache2/conf.d/20-xdebug.ini
```

Mejor si escribo bien el comando sudo.

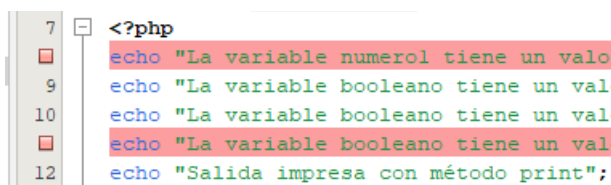
```
miadmin@RST-USED:/$ sudo nano cd /etc/php/8.1/apache2/conf.d/20-xdebug.ini
```

Nos abre el archivo y este está en blanco. Toca ahora escribir los parámetros necesarios.

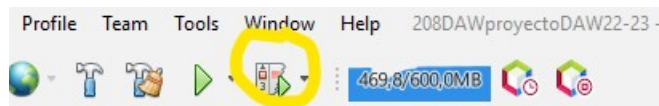
El grueso de parámetros los podemos obtener de iniciar una depuración en el IDE y esperar al mensaje de error en el que nos muestra los que necesita para funcionar.

Para ello hemos de seleccionar un archivo .php de nuestro proyecto, establecer un “Breakpoint” en el código y dar a ejecutar el depurador.

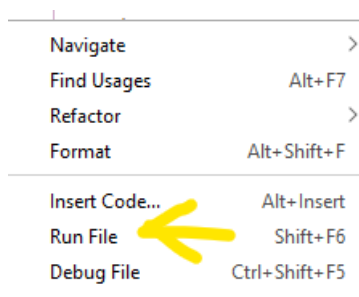
Los “breakpoints” se establecen clickando sobre el número que identifica a cada línea de código, situado a la izquierda del mismo.



Señalado dejo el debuggeador para el proyecto en su conjunto.



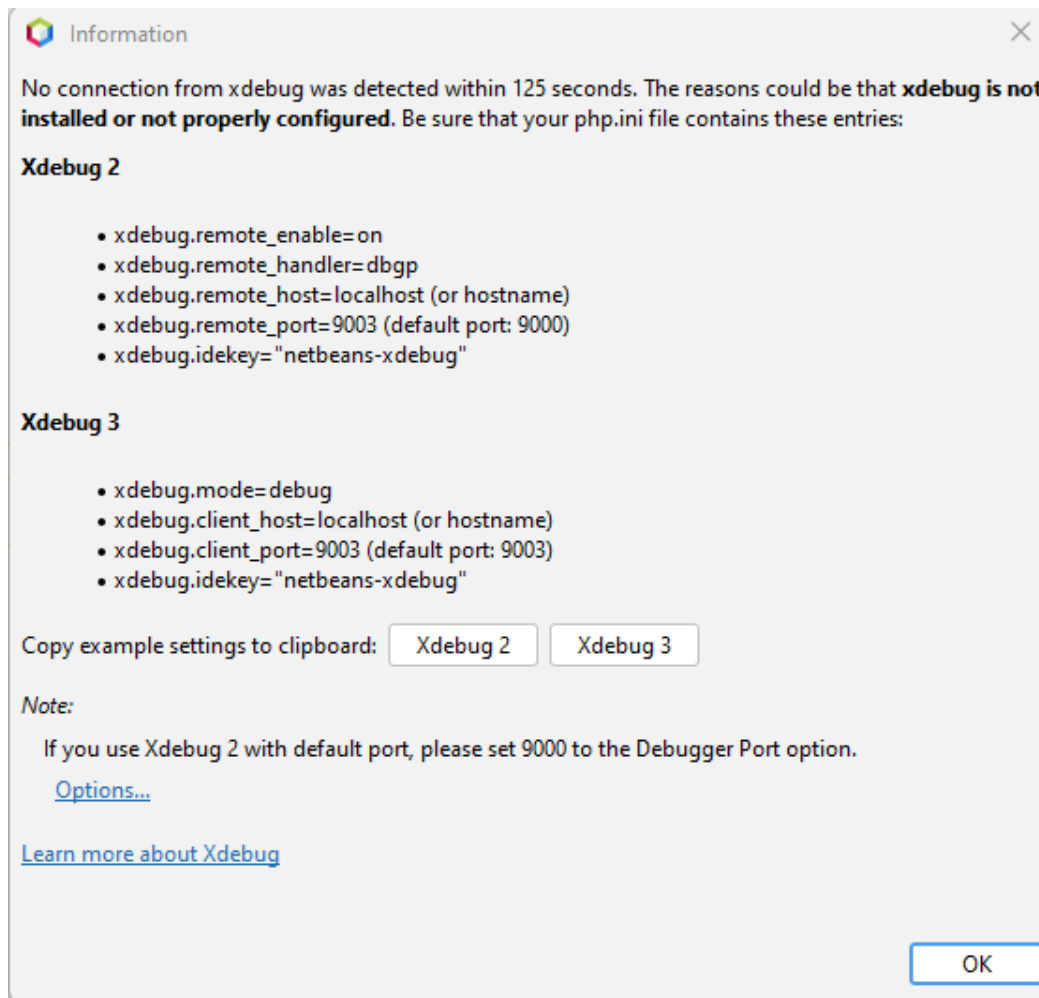
En este caso, al depurar un archivo que no es el principal del proyecto, tenemos que hacer click derecho sobre el código del archivo abierto y en el desplegable que aparece, elegir “Debug file”.



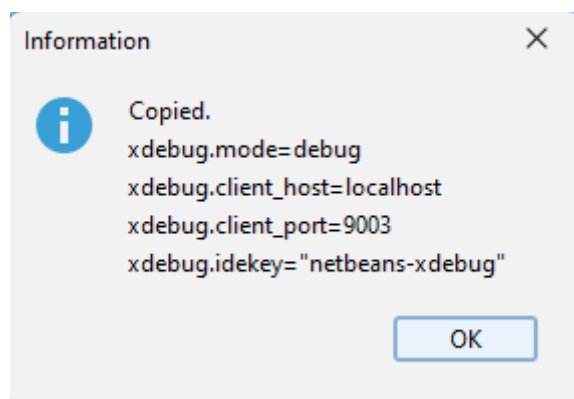
Esto arrancará el depurador y nos mostrará la página html en nuestro navegador. Al no estar aún configurado el archivo php.ini, la depuración se queda en espera.

Waiting For Connection (netbeans-xdebug)

Cuando paramos el depurador nos muestra una pantalla de advertencia que informa de que xdebug no está funcionando y nos indica los parámetros correctos a configurar en el fichero php.ini. Nuestra versión es la Xdebug3.



Al elegir la Xdebug 3 nos aparece una segunda ventana, dar ok y copia los valores al portapapeles.

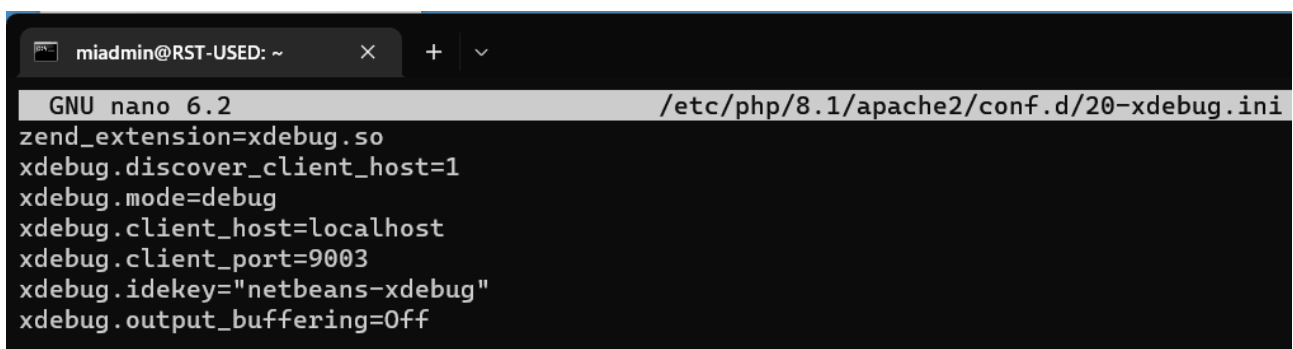


Si tenemos habilitada la comunicación ssh entre el cliente y el servidor podremos hacer uso de la herramienta “pegar” y con ctrl+v pegar en este archivo los valores que copiamos de NetBeans.

```
C:\Windows\System32>ssh miadmin@192.168.1.77
miadmin@192.168.1.77's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-50-generic x86_64)
```

```
miadmin@RST-USED:~$ sudo nano /etc/php/8.1/apache2/conf.d/20-xdebug.i
[sudo] password for miadmin:
miadmin@RST-USED:~$
```

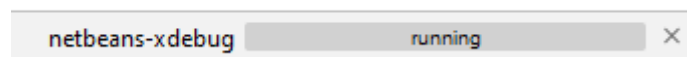
Después de esto, hay que editar el archivo. Esta imagen refleja todos los valores que hemos de escribir para que el depurador funcione.



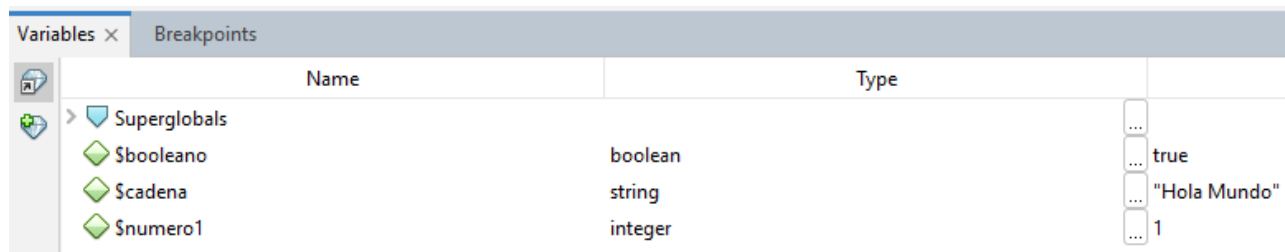
```
GNU nano 6.2 /etc/php/8.1/apache2/conf.d/20-xdebug.ini
zend_extension=xdebug.so
xdebug.discover_client_host=1
xdebug.mode=debug
xdebug.client_host=localhost
xdebug.client_port=9003
xdebug.idekey="netbeans-xdebug"
xdebug.output_buffering=0ff
```

Tiempo ahora de volver a NetBeans y hacer un Debug File de nuevo.

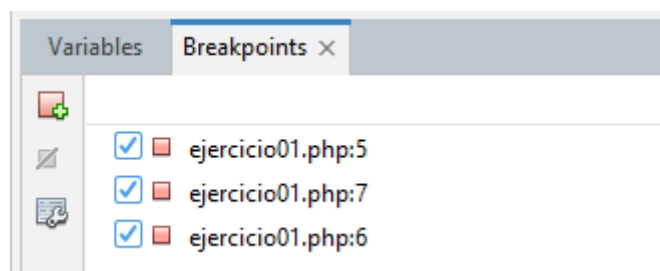
Ahora el netbeans-xdebug está running.



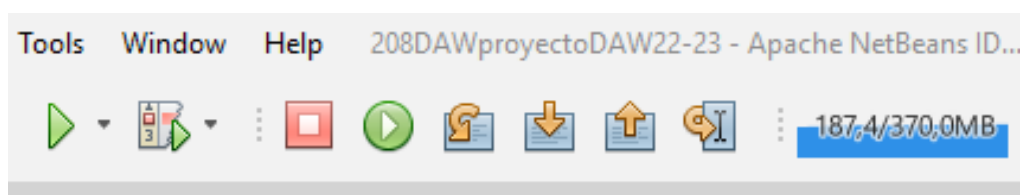
Y podremos ver nuevas ventanas en el entorno de desarrollo que nos muestran las variables por un lado.



Los breakpoints por otro.



Y en la parte superior de IDE estarán ahora habilitadas varias pestañas que sirven para ejecutar diferentes acciones sobre el depurador.



Prueba de depurado de un archivo .php.

Al comienzo del archivo tenemos declaradas e inicializadas las siguientes variables.

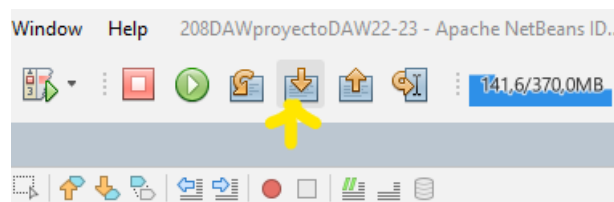
```
<?php
1
2 $numerol = 1;
3 $cadena = "Hola Mundo";
4 $booleano = true;
5 $decimal = 1.5;?>
```

Y gracias al breakpoint establecido en la línea 5, el depurador nos devuelve los valores que hasta ese punto de ejecución conoce, que son:

Variables × Breakpoints	
Name	Type
Superglobals	
\$booleano	boolean
\$cadena	string
\$numero1	integer

Observe se que aún no nos devuelve valor para &decimal , puesto que está parado justo en su línea de ejecución pero aún no lo ha ejecutado. Si hacemos uso de las herramientas del IDE (en este caso “step into”), el depurador retomará su ejecución y dará un siguiente paso para declarar e inicializar \$decimal = 1,5.

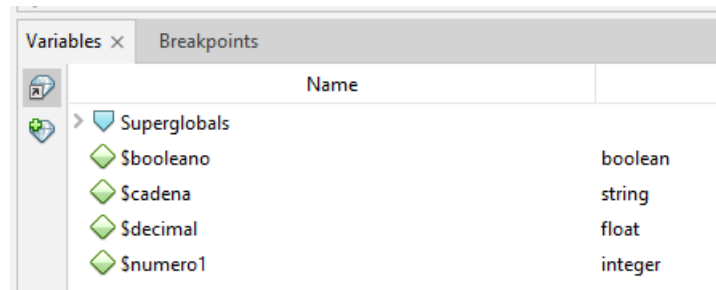
Step into.



El depurador ha avanzado una línea.

```
<?php
1
2 $numerol = 1;
3 $cadena = "Hola Mundo";
4 $booleano = true;
5 $decimal = 1.5;?>
6 <h3 style="color:#003399"><?php echo
```

Y ahora en la ventana de visibilidad de variables nos muestra el valor de \$numero1.



Al mismo tiempo que empieza la depuración, arranca la ejecución en el navegador que tengamos configurado en el IDE. Mientras no detengamos la depuración , este seguirá ejecutando paso a paso el resultado del código.

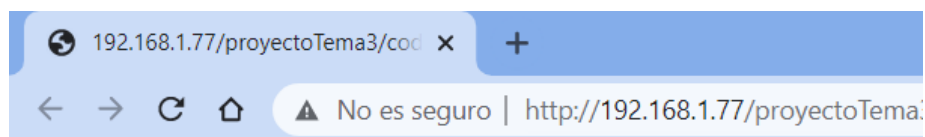
Podemos parar la depuración dando al play del depugeador hasta que haya ejecutado todo el archivo.



O al stop del depugeador e interrumpirá la depuración.



Por último, parará la ejecución en el navegador y mostrará este mensaje.



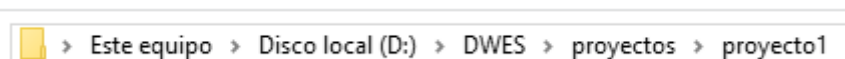
Warning: Header may not contain NUL bytes in **Unknown** on line 0
DEBUG SESSION ENDED

Creación de un proyecto PHP.

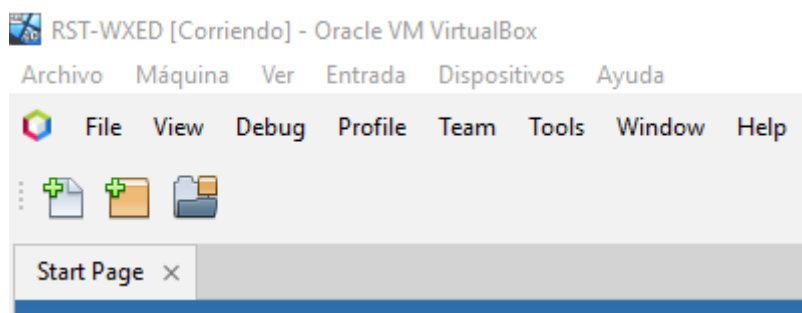
Lo primero es elegir entre crear la carpeta proyectos (aquella en la que estarán alojados todos nuestros proyectos) en la ubicación elegida con el selector de archivos o crearla desde dentro del propio NetBeans al tiempo de crear un primer proyecto. Creo también en este paso la carpeta para el primer proyecto si así lo deseo.

Aquí la creo con el selector de archivos.

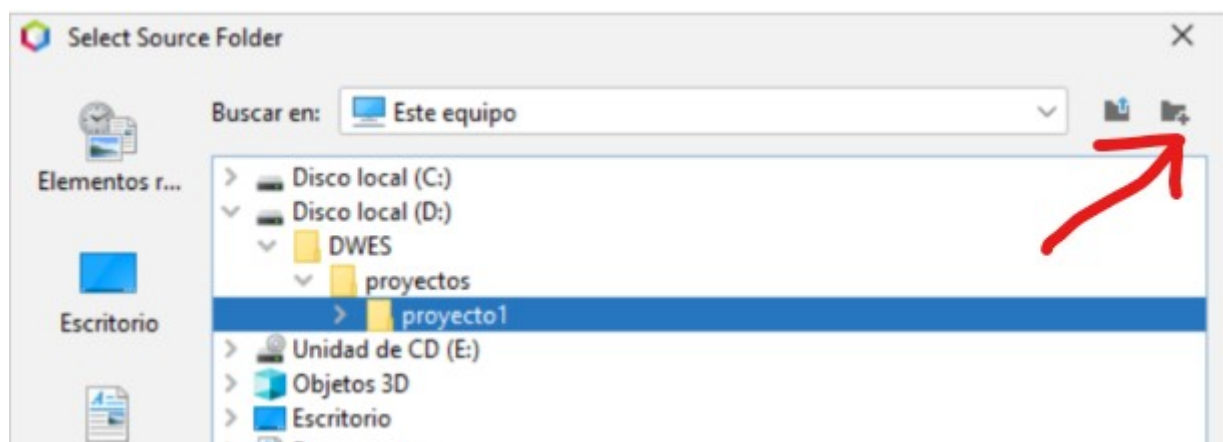
Actualizar las 3 siguientes capturas:



A continuación creamos el proyecto nuevo PHP. Es el icono de la carpeta con el símbolo +.

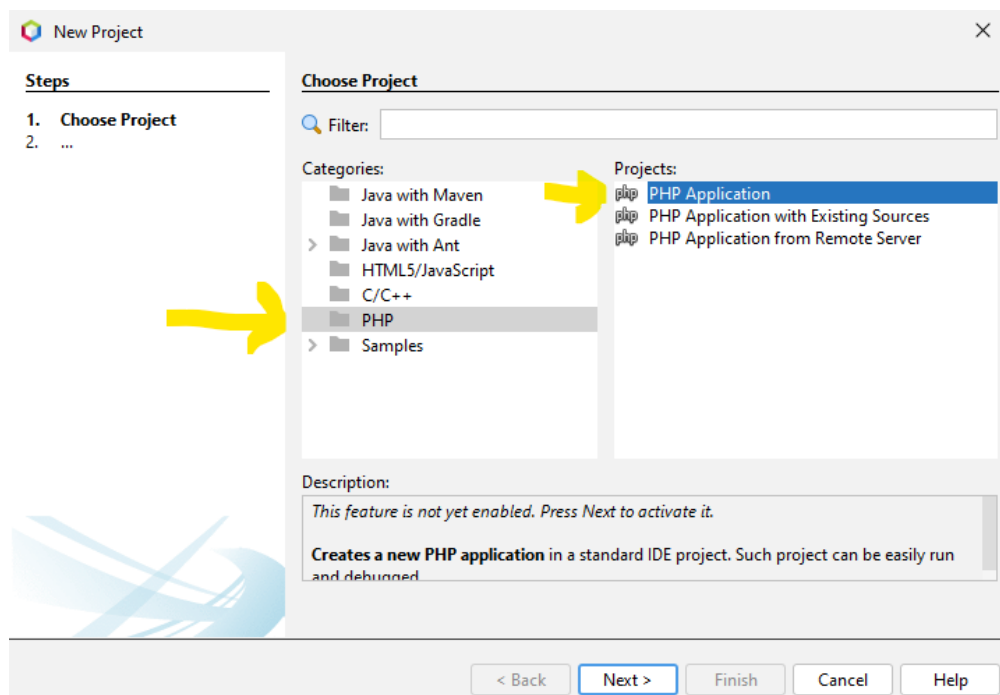


Si no hemos creado la carpeta con anterioridad, ahora podemos hacerlo, el icono señalado es el que crea las carpetas. Tantas como hagan falta.

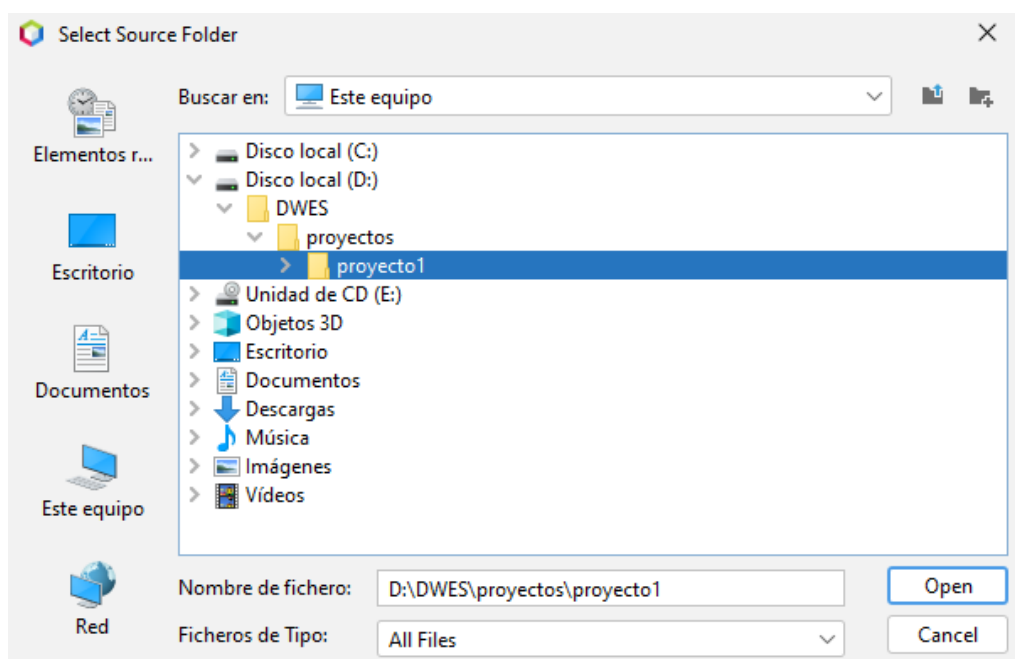


Proyecto nuevo:“PHP Application”.

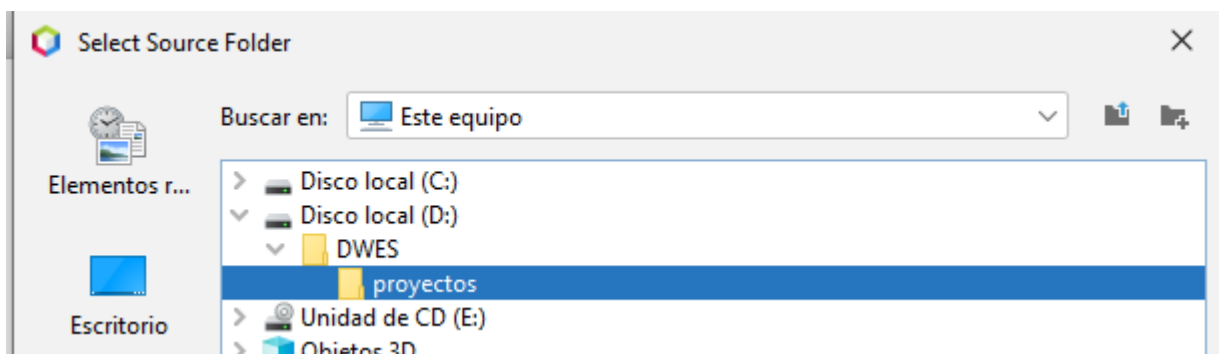
Lo siguiente es elegir el lenguaje en que vamos a programar y de que tipo, si nueva, desde un servidor remoto o con archivos previos existentes, en este apartado elegimos “PHP Application” al tratarse de un proyecto desde 0.



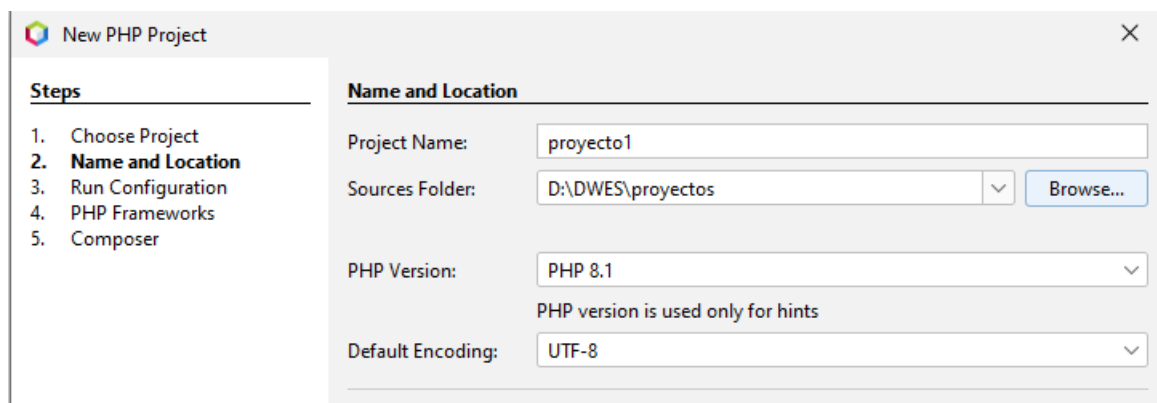
Después elegimos la carpeta fuente; en este punto es donde podemos crear las carpetas que necesitamos.



Hay dos opciones, creando una carpeta con el mismo nombre que el proyecto y alojar este allí o no crearla, con lo cual el nombre de nuestro proyecto no aparecerá en el servidor aunque sí en la estructura de carpetas local. Crearemos el proyecto sin una carpeta del mismo nombre cuando queramos que esta no se refleje en el árbol de directorios remoto, es decir, para que en *var/www/html* aparezca el contenido de los archivos fuente de este proyecto, pero no el proyecto en si mismo en forma de directorio. La segunda forma será elegida para los siguientes proyectos de los que si queremos obtener una visualización de su carpeta contenedora en el servidor.

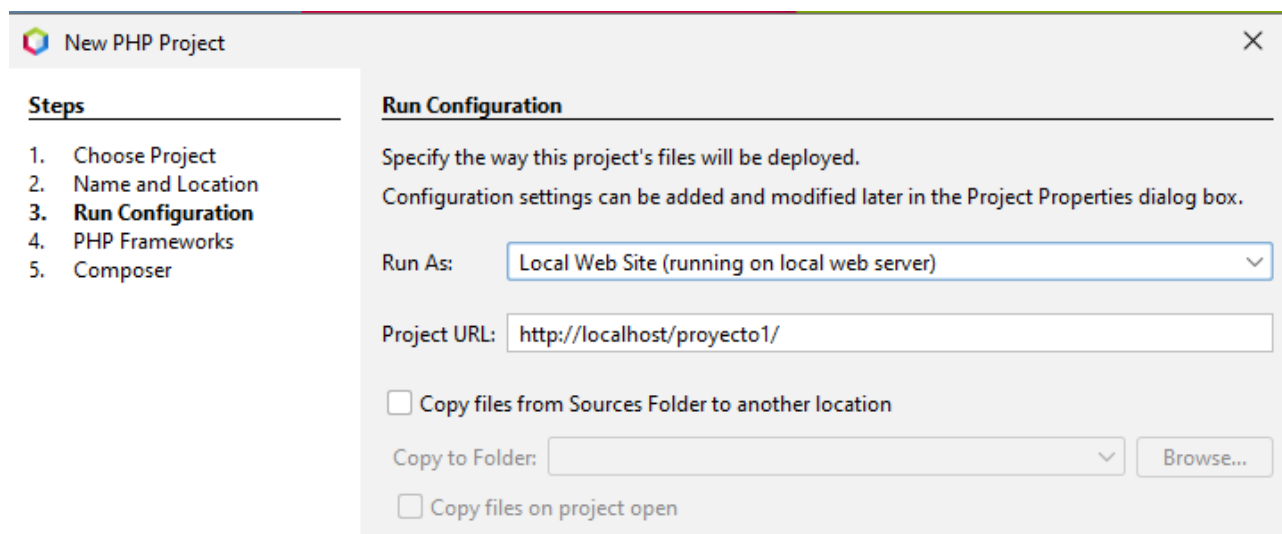


Nombre y carpeta fuente, también nos informa de la versión de PHP y de la codificación por defecto.

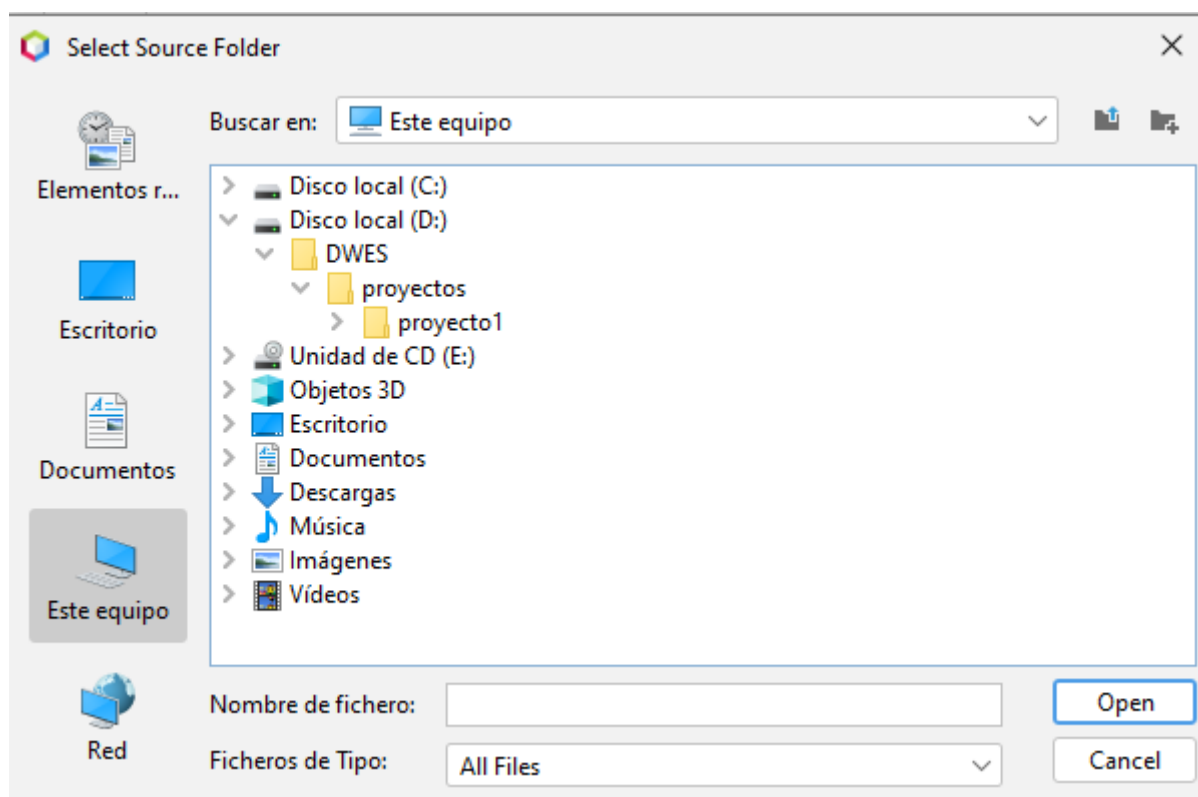


Lo siguiente es el “Run Configuration”. En primer término voy a dejar sus valores por defecto como están, a terminar de crear el proyecto y a renglón seguido a volver al “Run Configuration” habiendo quedado documentados todos estos pasos en el apartado **[“Conexión al entorno de desarrollo”](#)**.

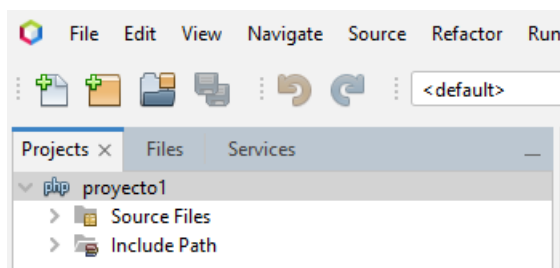
Estos son sus valores por defecto.



Arbol de directorios con la ubicación del nuevo proyecto.



La siguiente pantalla de “PHP Frameworks” la dejamos como está y damos a aceptar. Otra pantalla más “Composer” se llama en esta ocasión, aceptar sin cambiar nada yproyecto creado.

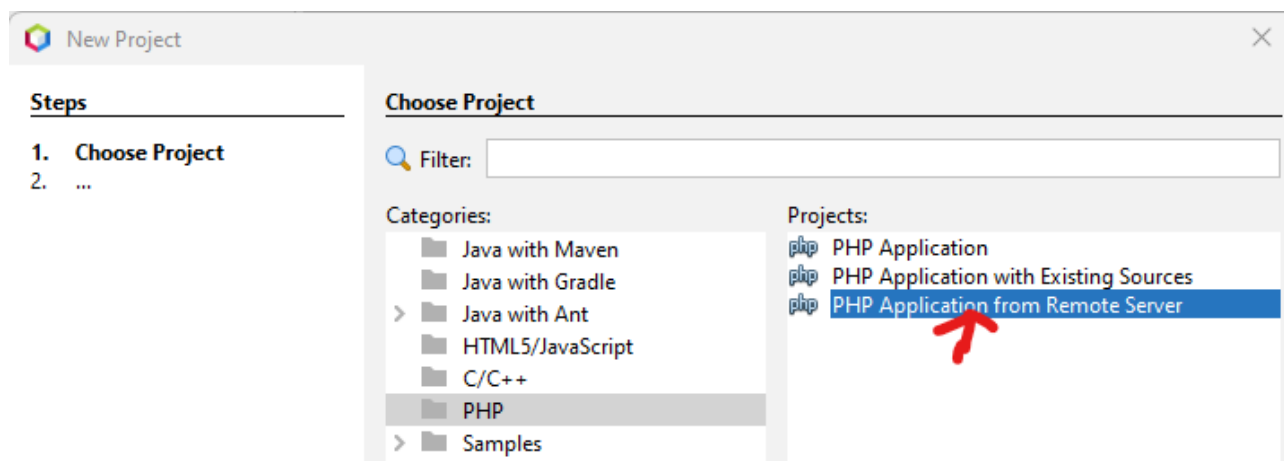


Creación de proyecto PHP desde un servidor:

“PHP Application from Remote Server”.

A continuación documento la creación de un proyecto ya existente en un servidor remoto. Los nombres de proyecto y directorios en local variarán en este ejemplo para acomodarlo a los requisitos de nomenclatura de proyectos del módulo.

Documento únicamente aquellas partes en que difiere de la creación de un proyecto nuevo desde cero.



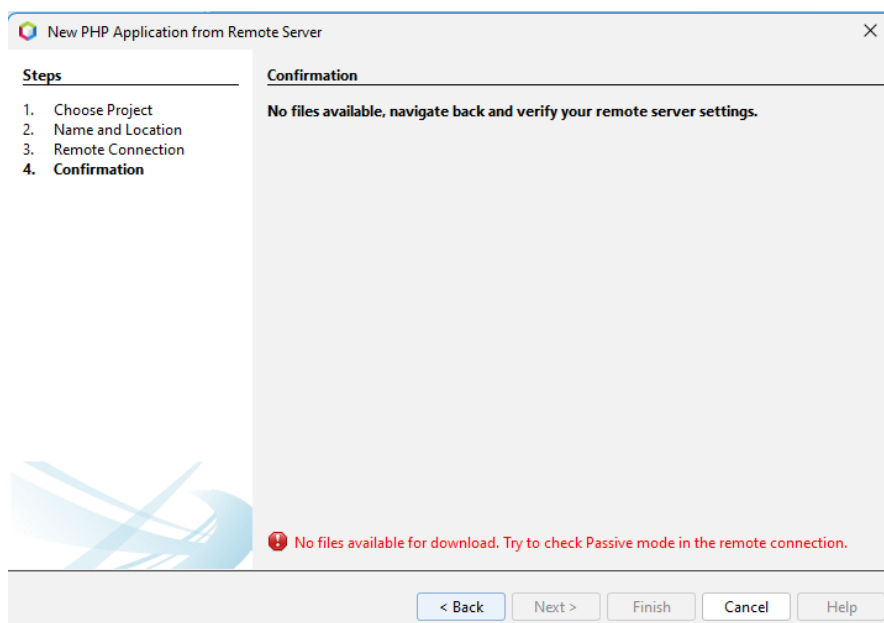
A practicar inglés. Resumiendo, baja desde un servidor una aplicación PHP.

Description:

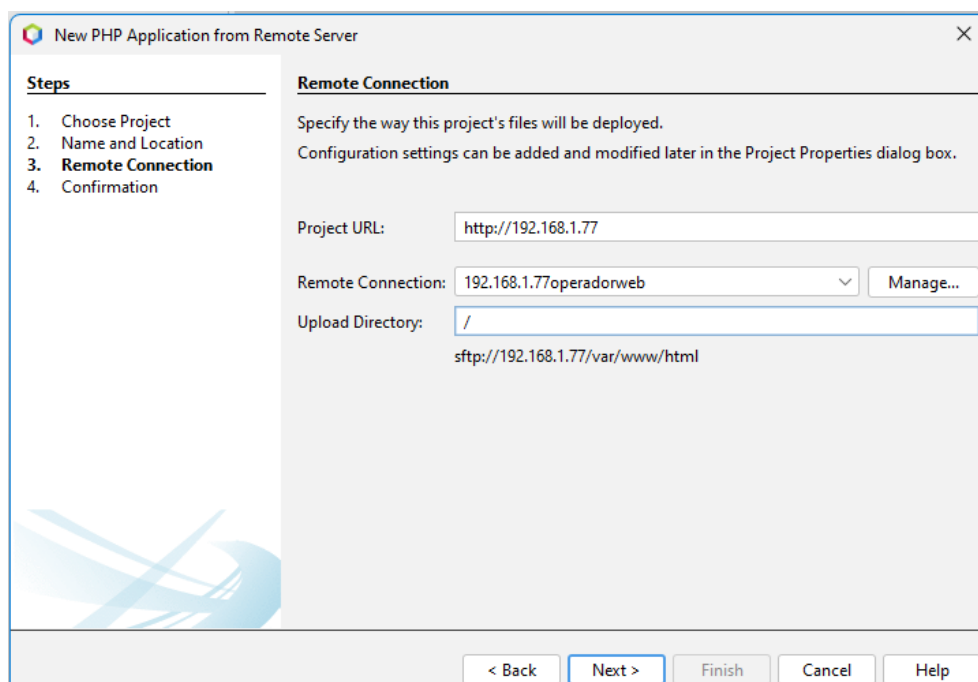
Downloads an existing PHP application, creates a standard IDE project for it and sets up the project properties accordingly (**automatic uploading of local changes to the server**). Such project can be easily run and debugged.

Los pasos omitidos (“Choose Project” y “Name and Location”, son iguales a los de la creación del proyecto nuevo desde 0.

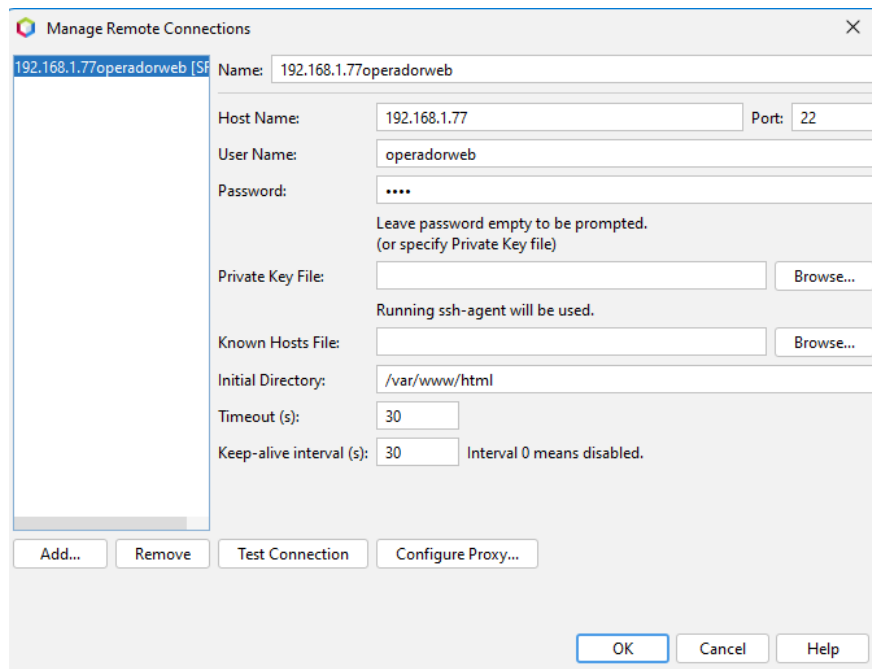
Si se falla en cualquiera de estos pasos, tenemos la posibilidad de volver atrás,



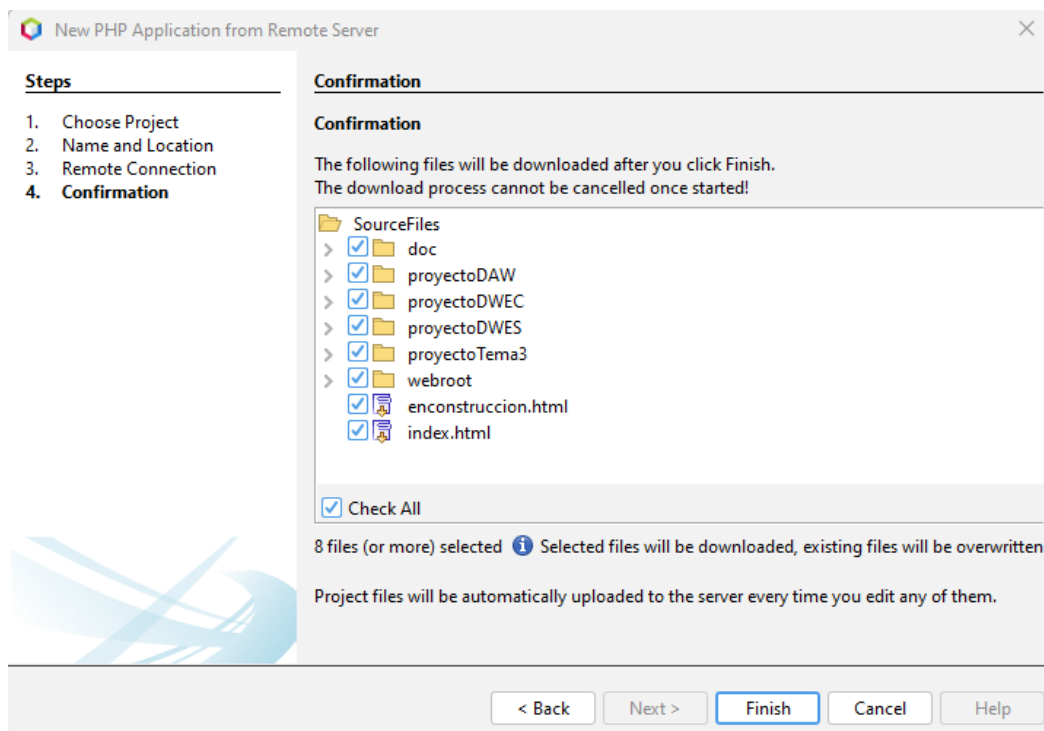
Al tratarse de una conexión al servidor, hemos de indicarle la dirección ip del servidor precedida del protocolo http y el “Upload Directory” será la carpeta raíz que contiene el proyecto que queremos “traer” a NetBeans en local



Si elegimos el botón “Manage” de la “Remote Connection”, muestra esta pantalla. En la que podemos dar nombre y personalizar la conexión. Los valores son los mismos que para el proyecto nuevo desde 0.



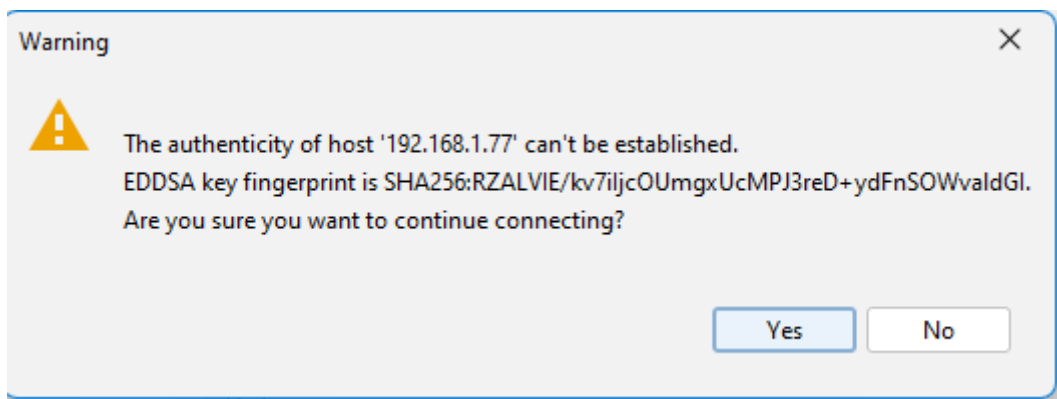
Lo siguiente que se nos muestra es esta pantalla, lo que confirma que hemos configurado todo correctamente y nos muestra las carpetas y ficheros presentes en el directorio del servidor que le hemos indicado.



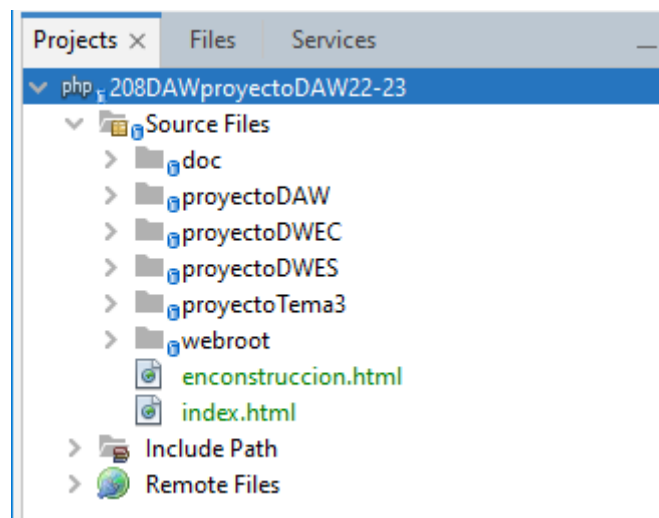
Estos mismos directorios y archivos los muestro ahora en el servidor.

```
$ dir
doc                index.html        proyectoDWEC      proyectoTema3
enconstruccion.html proyectoDAW        proyectoDWES      webroot
$
```

Nos pide confirmación.



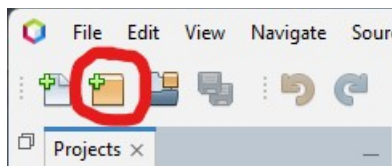
Y ya tenemos el proyecto del servidor “bajado” a local y reconocido como proyecto por NetBeans.



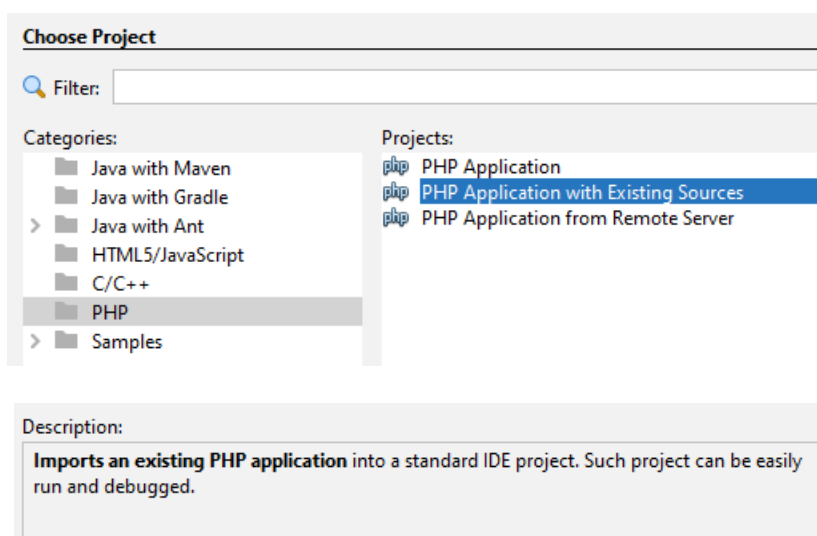
Creación de proyecto PHP con fuentes existentes:

“PHP Application with Existing Sources”.

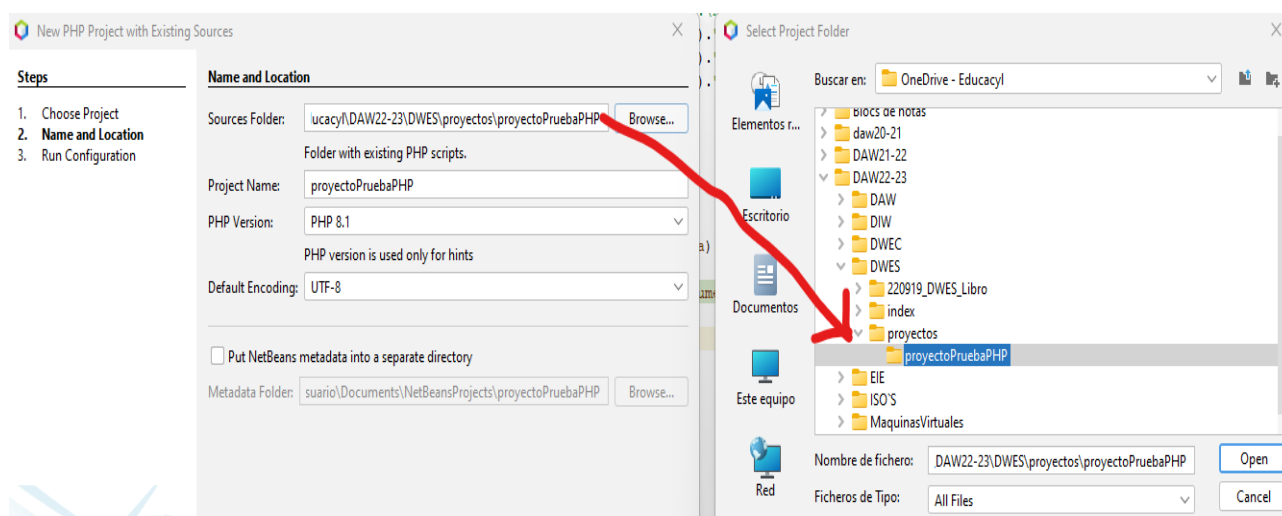
Comenzamos con nuestra vieja amiga la carpeta “New Project”



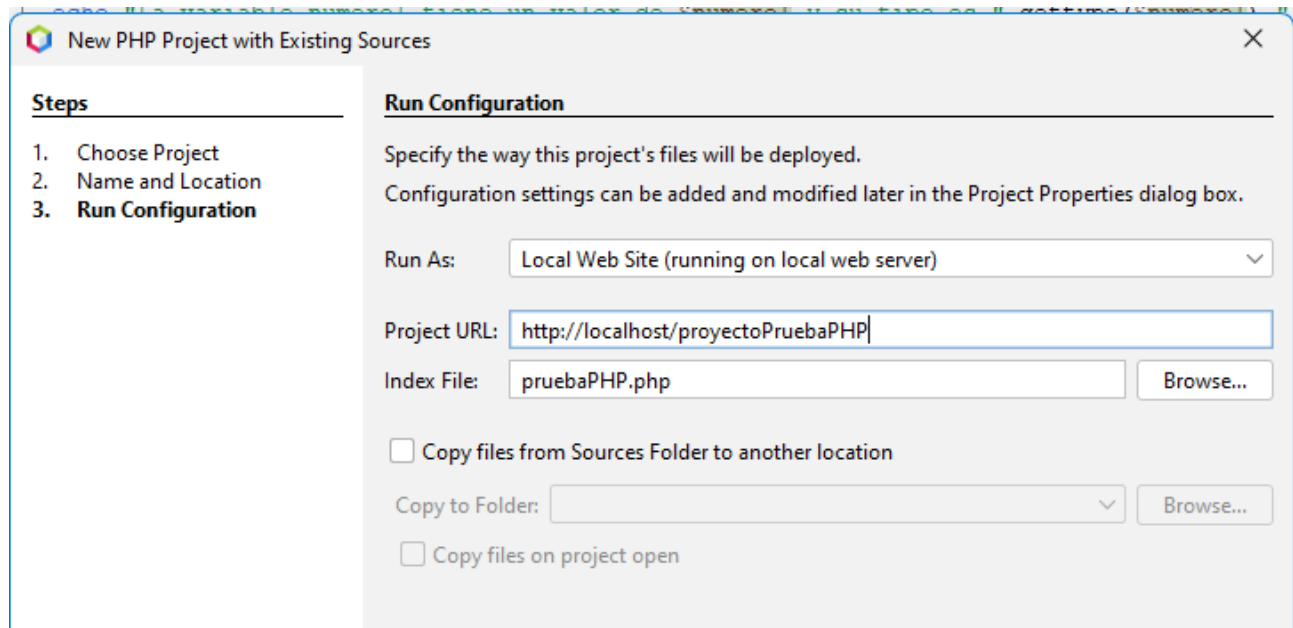
Esta vez el proyecto con el que queremos trabajar en NetBeans es un proyecto existente.



Dentro de la ventana “Name and Location”, en la pestaña “Sources Folder” tenemos que indicarle al IDE donde se ubica el proyecto dentro de nuestro sistema de archivos.



En esta ocasión voy a configurar el proyecto para que corra en un servidor local, es decir, en el propio anfitrión, sin necesidad de alojarlo en un servidor externo.



Paso a explotación.

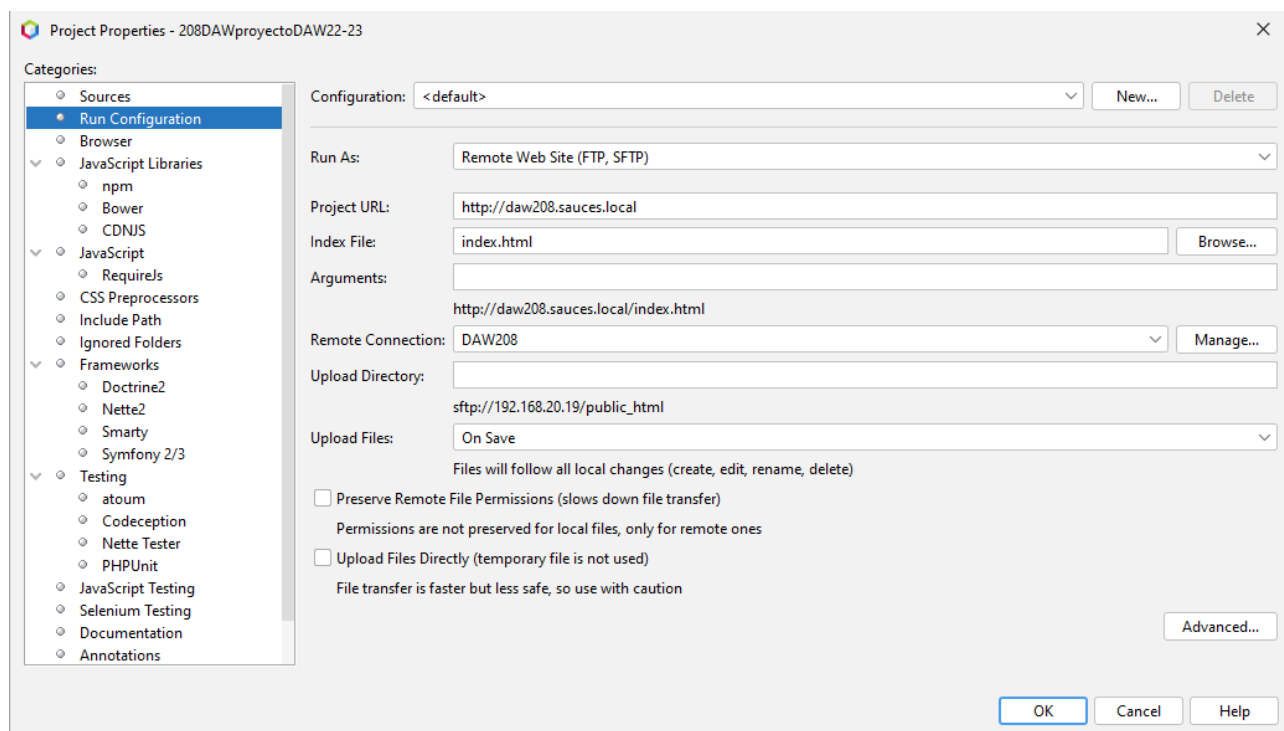
En construcción...

PHP Doc.

En construcción...

CSS / JS / AJAX / XML / JSON.

En construcción...



Manage Remote Connections

192.168.3.207operatorweb [SFTP]
conexionLuis [SFTP]
daw201 [SFTP]
DAW208 [SFTP]
operatorApa [SFTP]

Name: DAW208

Host Name: 192.168.20.19 Port: 22

User Name: DAW208

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

Private Key File: Browse...

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

