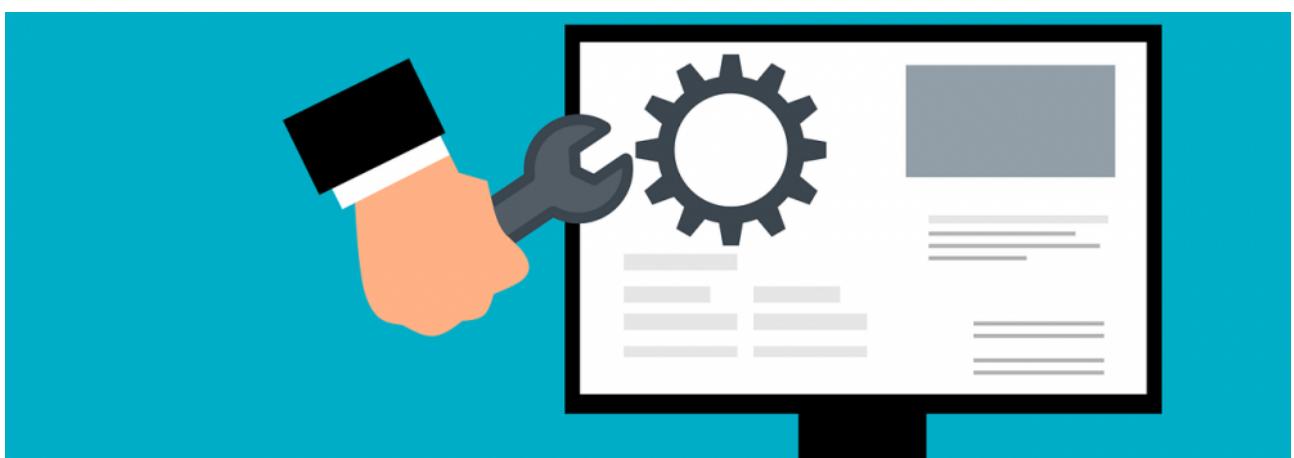


Ejercicios tema 2



Índice

Ubuntu Server.....	3
Configuración inicial.....	3
Cuentas de administración.....	5
Creación usuario administrador miadmin.....	5
Apache.....	5
Comprobación de estado de Apache.....	6
Apertura de puerto 80 y comprobación de puertos abiertos.....	7
PHP.....	8
MySQL.....	12
Xdebug.....	13
Cuentas de desarrollo y hosting virtual.....	14
Creación usuario de desarrollo “operadorweb”.....	14
Hosting virtual.....	15
WXED – Windows X:.....	17
Instalación y configuración inicial de la máquina.....	17
Datos equipo cliente Windows 10.....	17
Cuentas administradoras y cuenta de desarrollador.....	21
Navegadores.....	22
Google Chrome.....	22
Mozilla Firefox.....	22
Microsoft Edge.....	23
Opera.....	23
Filezilla.....	24
Notepad++.....	25
Netbeans:.....	25
Instalación y configuración inicial (plugings).....	25
Conexión al entorno de desarrollo.....	30
Administración de la base de datos.....	34
Conexión al repositorio – versionado.....	34
Depuración - Configuración de la ejecución para la depuración.....	35
Prueba de depurado de un archivo .php.....	40
Creación de un proyecto PHP.....	42
Proyecto nuevo:.....	43
Creación de proyecto PHP desde un servidor:.....	46
Creación de proyecto PHP con fuentes existentes:.....	50
Paso a explotación.....	52
PHP Doc.....	52
CSS / JS / AJAX / XML / JSON.....	52

Ubuntu Server

La distribución de Linux elegida es la Ubuntu f22.04 de 64 bits.

```
miadmin@RST-USDED:~$ 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
```

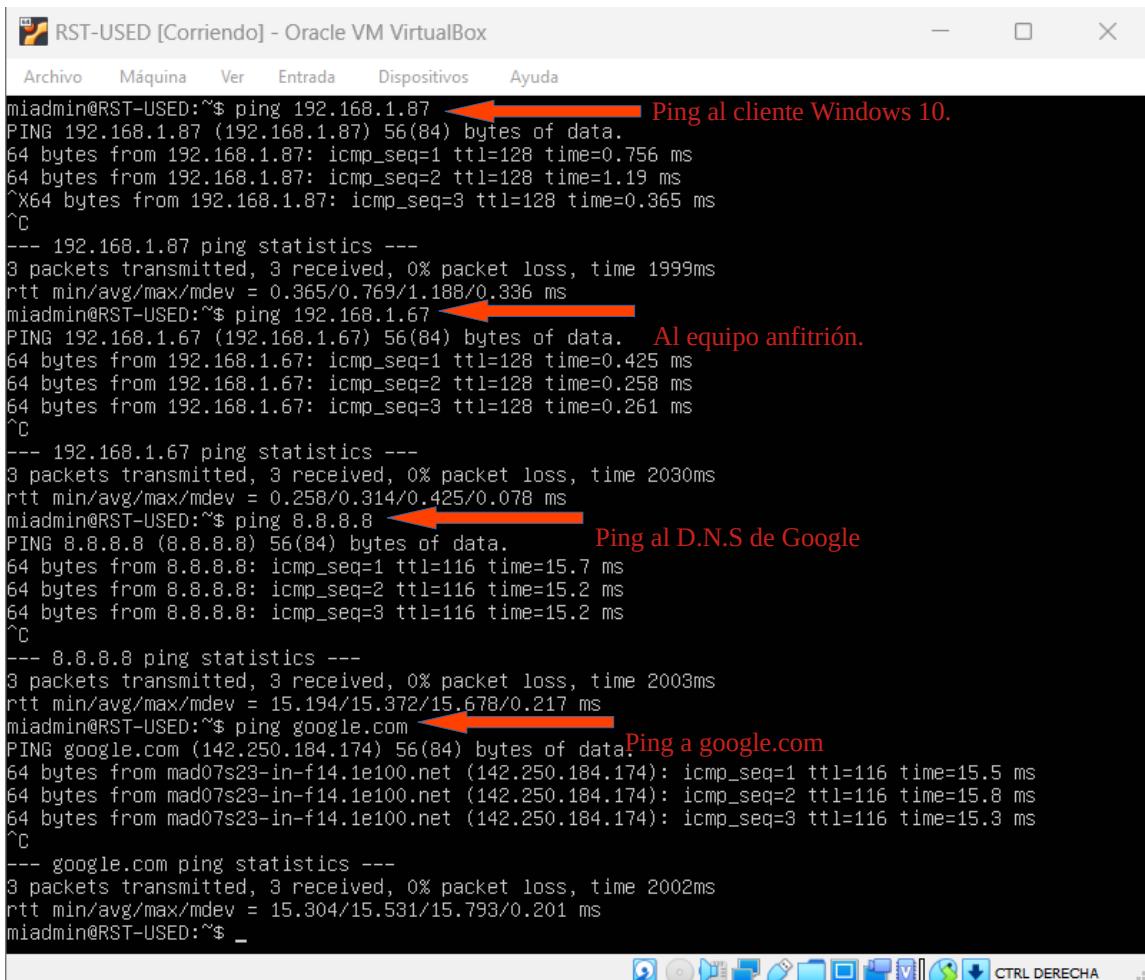
Configuración inicial.

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-USDED [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.



RST-USDED [Corriendo] - Oracle VM VirtualBox

```
miadmin@RST-USDED:~$ ping 192.168.1.87 Ping al cliente Windows 10.
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-USDED:~$ ping 192.168.1.67 Al equipo anfitrión.
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-USDED:~$ ping 8.8.8.8 Ping al D.N.S de Google
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-USDED:~$ ping google.com Ping a 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-USDED:~$ _
```

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-USDED:~$ 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-USDED:~$ 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's
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.

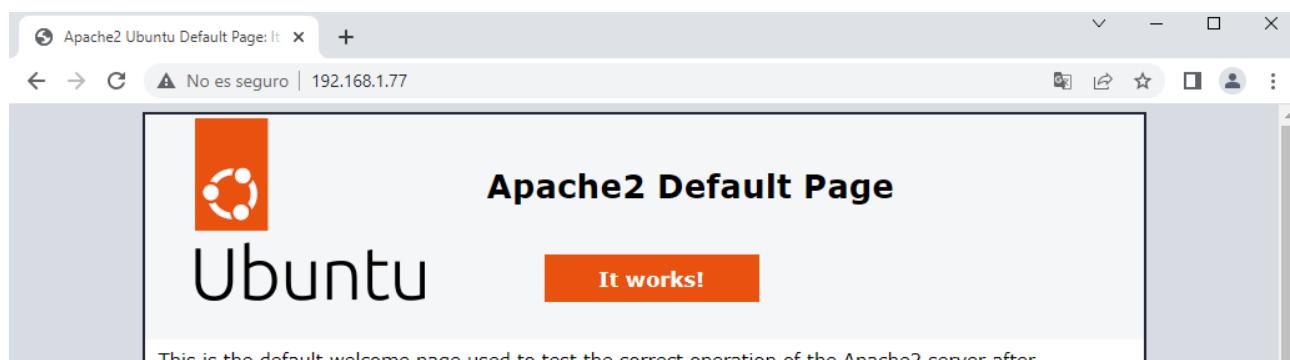
```
miadmin@RST-USDED:~$ sudo ufw allow 80
Rule added
Rule added (v6)
miadmin@RST-USDED:~$ ufw status
ERROR: You need to be root to run this script
miadmin@RST-USDED:~$ sudo ufw status
Status: active

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

miadmin@RST-USDED:~$ 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-USDED:~$ _
```

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-US ED:/$ 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-US ED:/$ 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-US ED:/$ _
```

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-USDED:/etc/php/8.1/apache2$ sudo cp php.ini phpbackup.ini
[sudo] password for miadmin:
miadmin@RST-USDED:/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_executiiion_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.

```
; 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
```

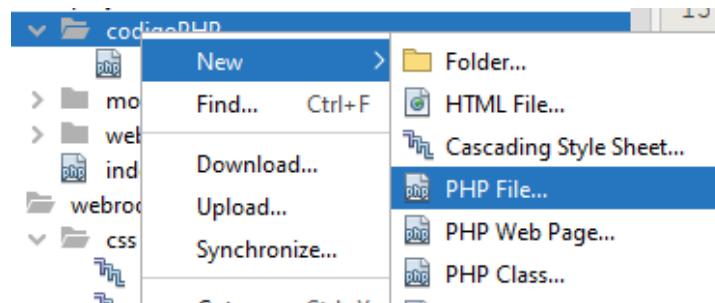
```
; The display of errors which occur during PHP's startup sequence are handled
; separately from display_errors. We strongly recommend you set this to 'off'
; for production servers to avoid leaking configuration details.
; Default Value: On
; Development Value: On
; Production Value: Off
; https://php.net/display-startup-errors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

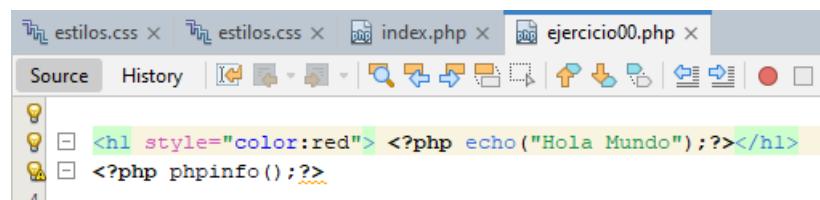
Reinic平 apache2 lo realizamos con el comando “sudo service apache2 restart”.

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

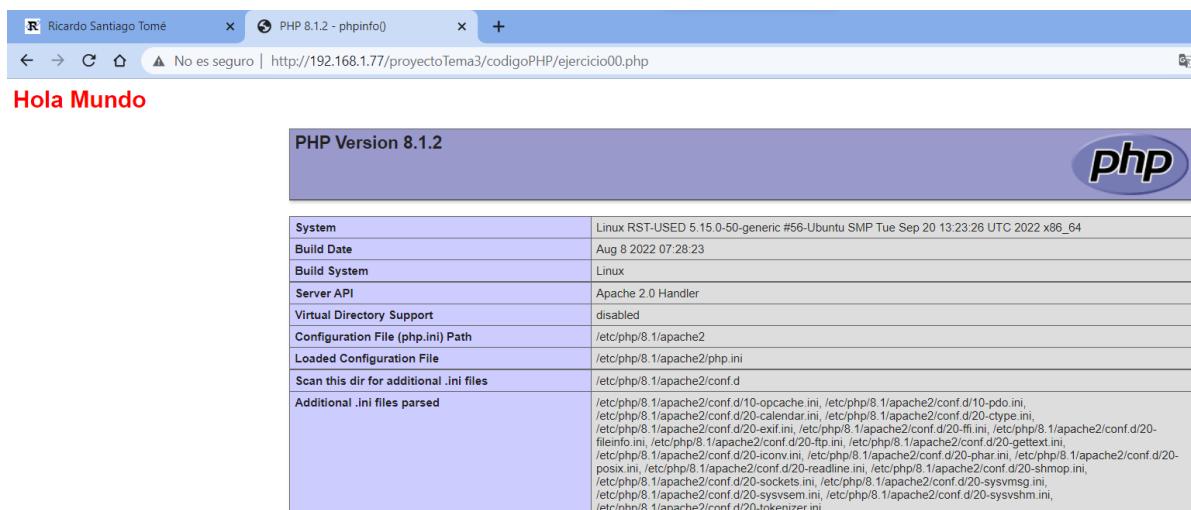
Situándome en la ubicación dentro del árbol de directorios de mi programa y haciendo click derecho muestra el IDE las siguientes ventanas.



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.



¡Trabajo hecho!

MySQL.

En construcción...

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-USDED:~$ 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-USDED:~$ sudo service apache2 restart  
miadmin@RST-USDED:~$ 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: 2000)
```

[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-USDED:~$ sudo useradd -d /var/www/html -g www-data operadorweb
```

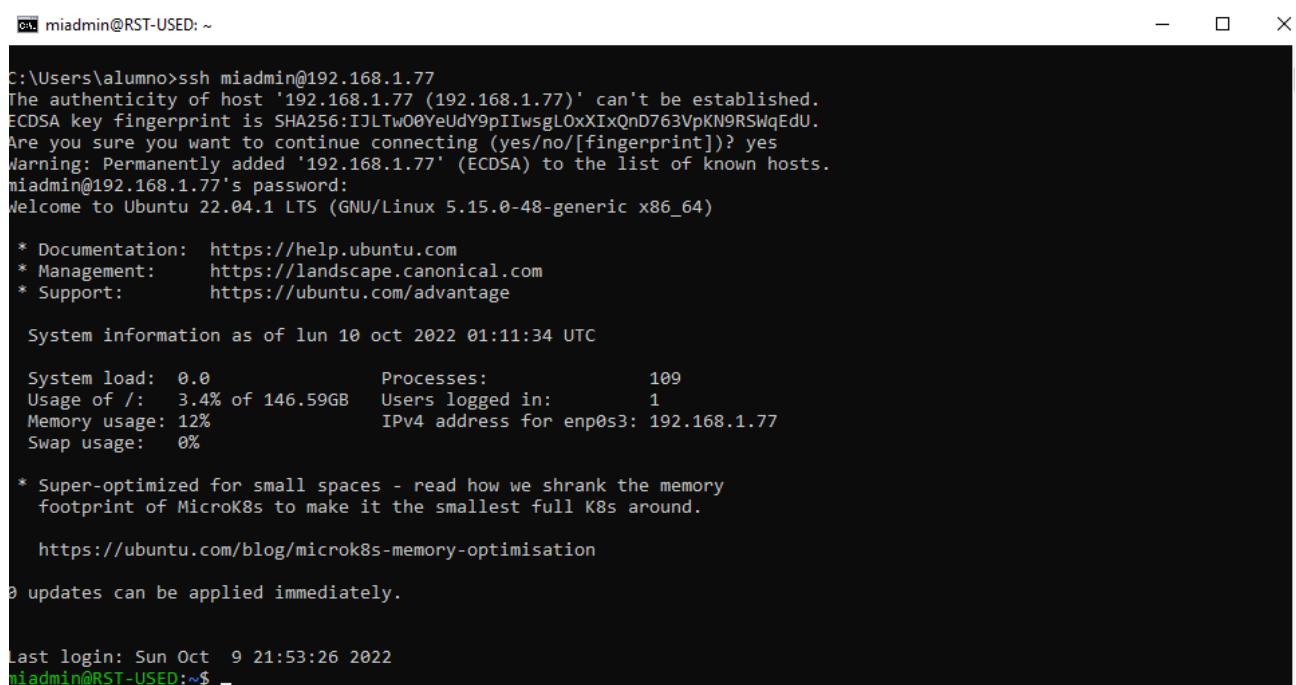
Estableciendo password para operadorweb con comando **sudo passwd operadorweb**.

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

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 donde raíz del usuario y shell.

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

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



```
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:IJLTw00YeUdY9pIIwsgLOxXIxQnD763VpKN9RSWqEdU.
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%
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-USDED:~$
```

En Ubuntu. Cambiar propietario de carpeta y grupo.

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

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

```
miadmin@RST-US ED:~$ sudo chown -R operadorweb:www-data /var/www/html  
miadmin@RST-US ED:~$ sudo chmod -R 2775 /var/www/html  
miadmin@RST-US ED:~$ ls -l /var/www  
total 4  
drwxrwsr-x 2 operadorweb www-data 4096 sep 29 09:35 html  
miadmin@RST-US ED:~$ _
```

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 Control Panel System window. At the top left is a blue checkmark icon next to the word "Sistema". Below the title bar, there's a navigation bar with back, forward, up, and search buttons, followed by the text "Panel de control > Sistema y seguridad > Sistema".

Ventana principal del Panel de control

- Administrador de dispositivos
- Configuración de Acceso remoto
- Protección del sistema
- Configuración avanzada del sistema

Ver información básica acerca del equipo

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 pant

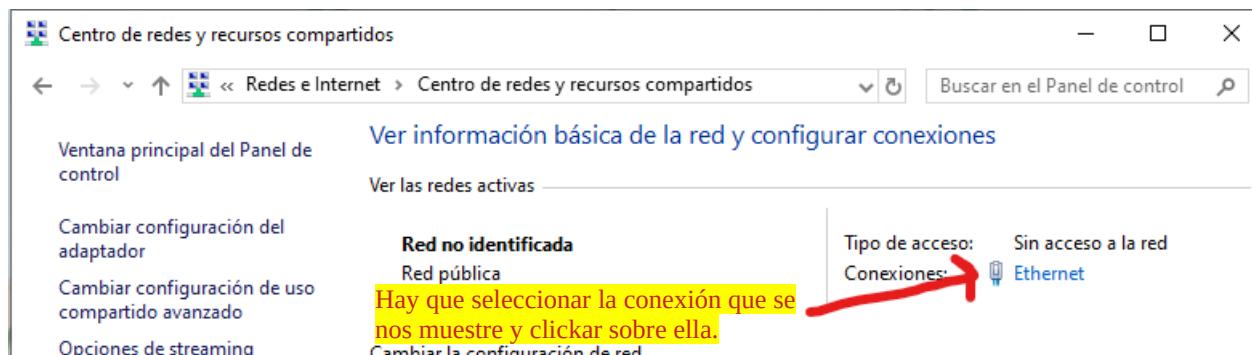
Configuración de nombre, dominio y grupo de trabajo del equipo

Nombre de equipo:	RST-WXED
Nombre completo de equipo:	RST-WXED
Descripción del equipo:	
Grupo de trabajo:	IS22

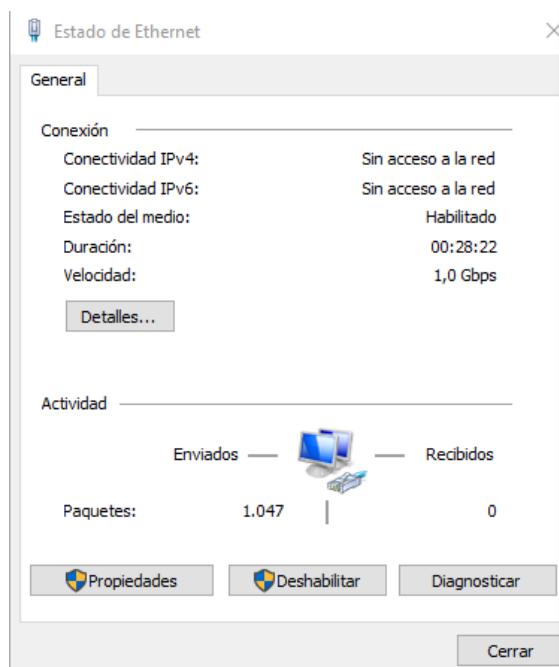
Activación de Windows

Windows está activado [Lea los Términos de licencia del software de Microsoft](#)
Id. del producto: 00331-10000-00001-AA639

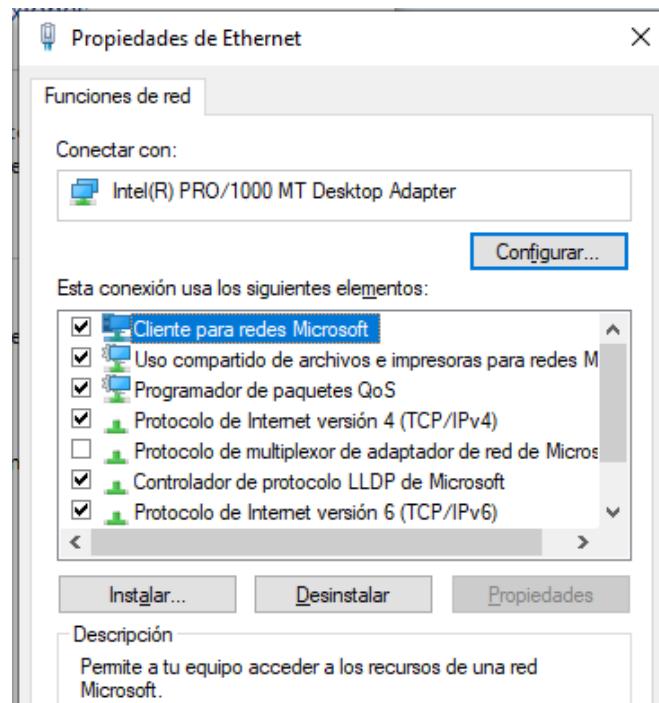
Una de las primeras acciones una vez instalado el SO Windows 10 en nuestra máquina virtual cliente, es configurar sus direcciones ip para poder conectarlo con el servidor e incluso con el exterior para poder descargarnos todas las herramientas que vamos a utilizar en el desarrollo.



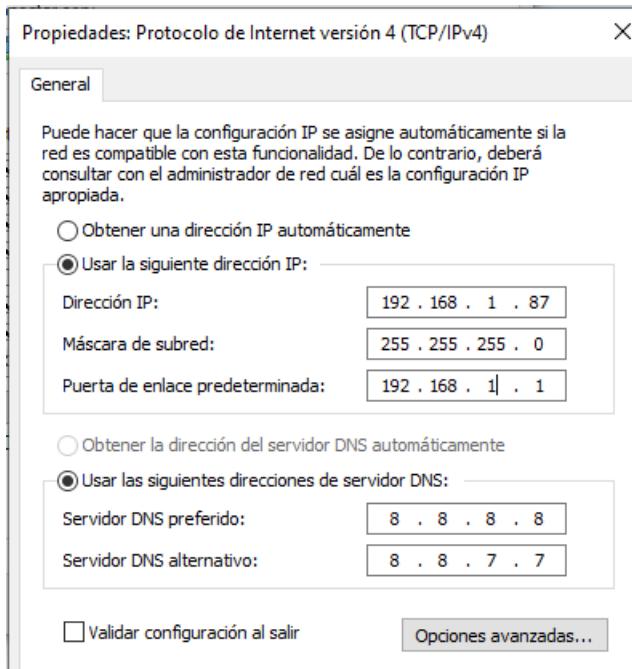
A continuación , en la nueva ventana hay que seleccionar “Propiedades”.



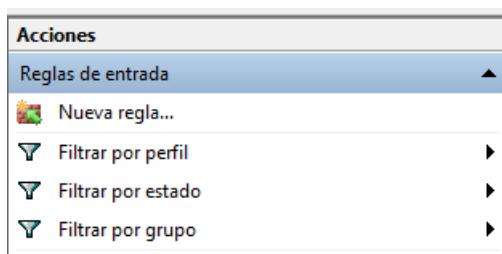
El siguiente paso es elegir el protocolo de internet TCP/IPv4.



Establecer las direcciones ip.



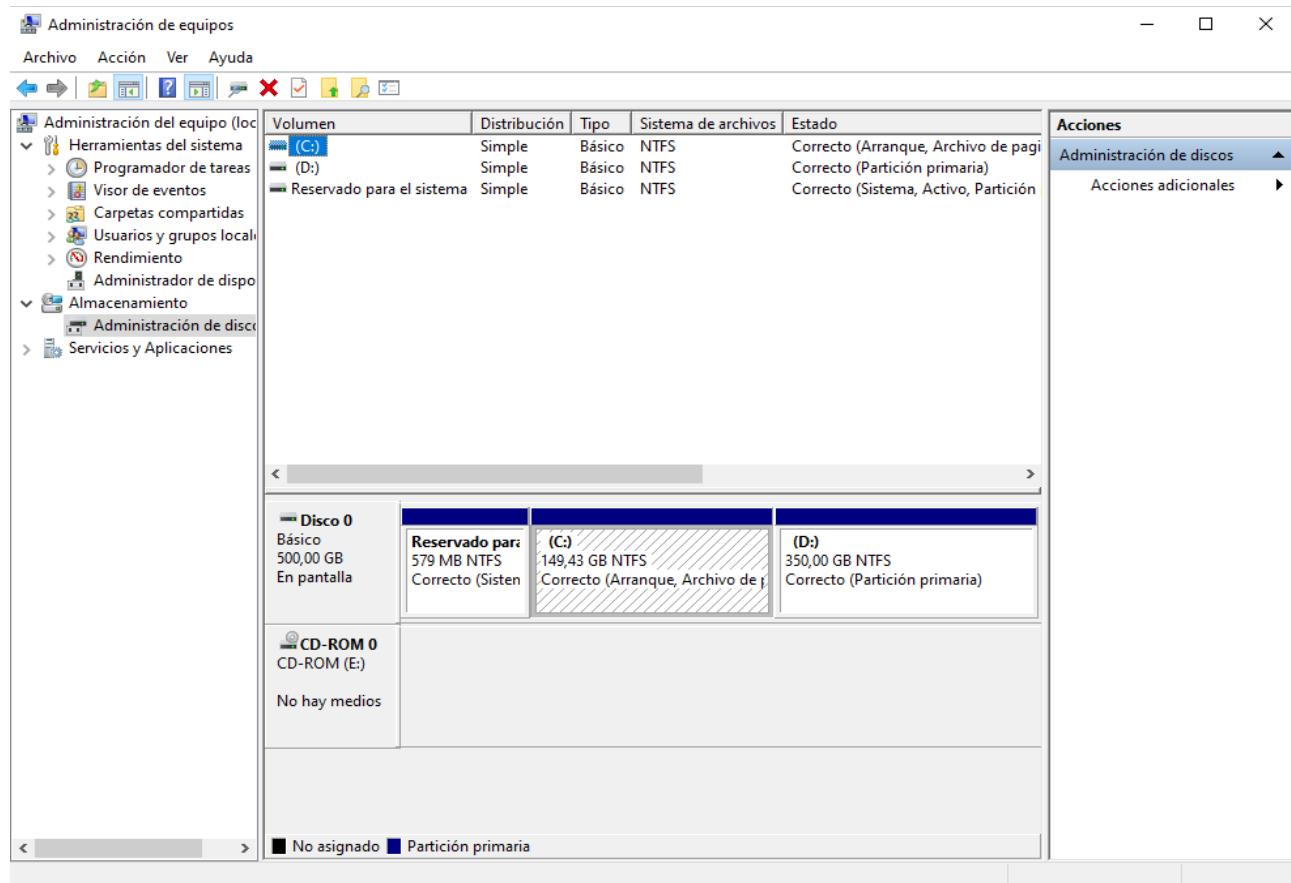
También hay que crear una regla ICMPv4 en la máquina cliente de Windows 10 para permitir el acceso desde otros programas y máquinas externos. Después de seleccionar “Reglas de entrada”, nos aparece a la derecha un desplegable “Acciones” en el que tenemos que seleccionar “Nueva regla...”



Especificaciones configuración regla ICMPv4

General	Acción
<input type="text" value="ICMPv4"/> <input checked="" type="checkbox"/> Habilitado	<input checked="" type="radio"/> Permitir la conexión

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.

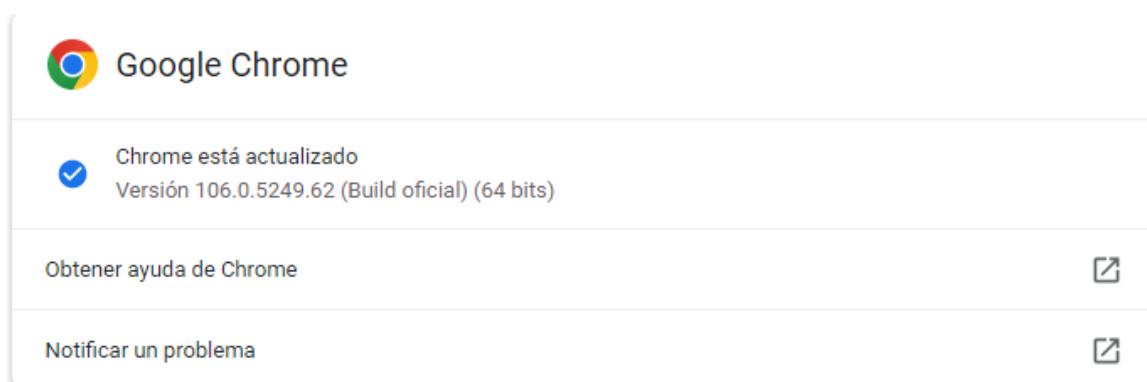


Barra de tareas personalizada para el alumno desarrollador.



Navegadores.

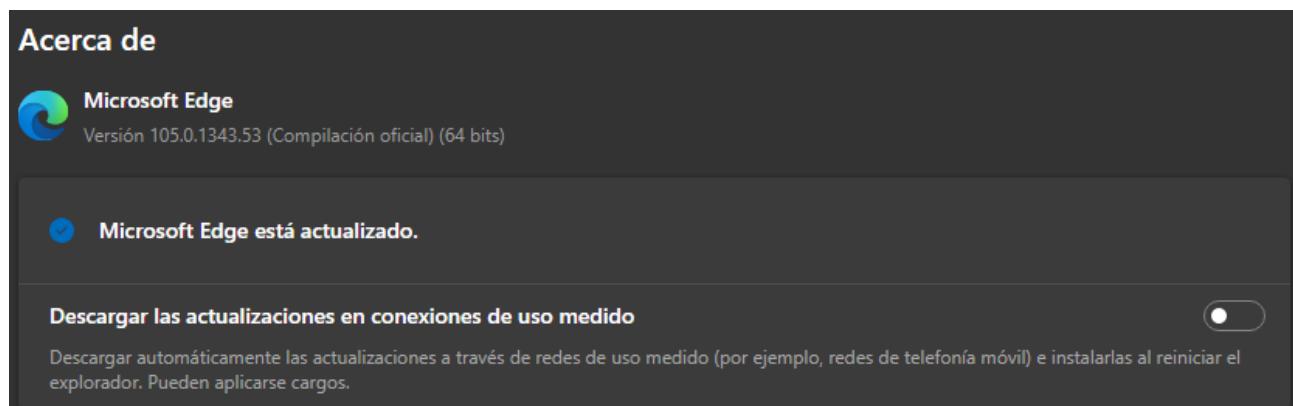
Google Chrome



Mozilla Firefox



Microsoft Edge

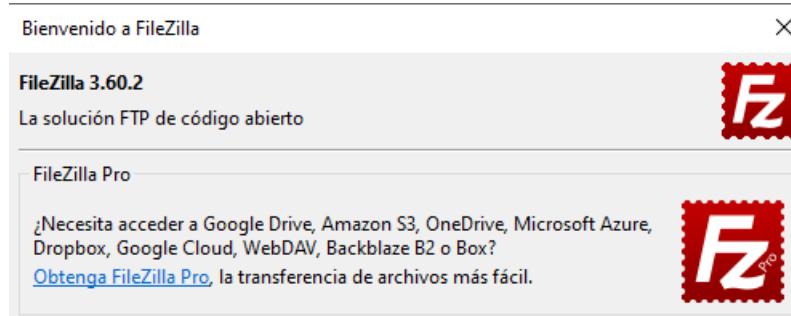


Opera.



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.

The screenshot shows the FileZilla interface with a successful connection to 'sftp://miadmin@192.168.1.77'. The left panel shows the local directory structure on drive D:\, which includes 'PerfLogs', 'Program Files', 'Program Files (x86)', 'ProgramData', 'Recovery', 'System Volume Information', 'Users', 'Administrador', 'All Users', and 'alumno'. The right panel shows the remote directory structure at '/var/www/html', which includes 'lost+found', 'mail', 'opt', 'run', 'snap', 'spool', 'tmp', 'www', and 'html'. Below the panels are two tables showing file lists:

Nombre de archivo	Tamaño de...	Tipo de archivo	Última modificación
..			
SRECYCLE.BIN		Carpeta de archivos	09/10/2022 23:50:56
System Volume Informati...		Carpeta de archivos	29/09/2022 9:45:29

Nombre de archivo	Tamaño d...	Tipo de arc...	Última modific...	Permisos	Propietario/...
index.html	10.671	Opera Web...	09/10/2022 23:...	-rw-r--r--	root root

NotePad++.

En construcción...

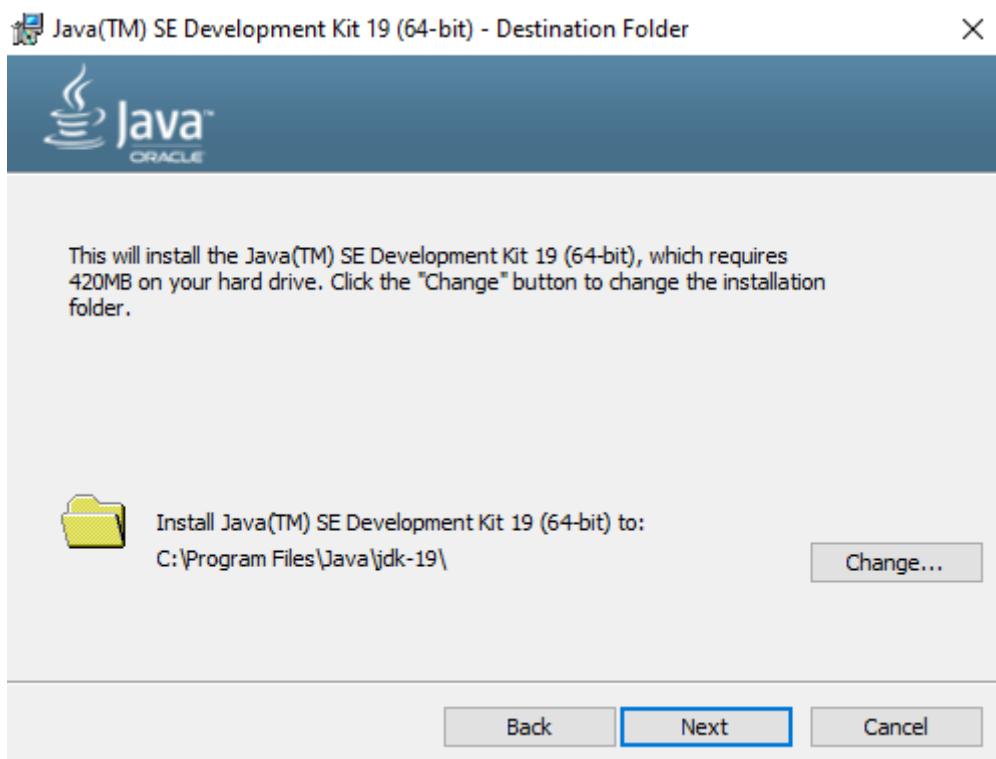
Netbeans:

Instalación y configuración inicial (plugings).

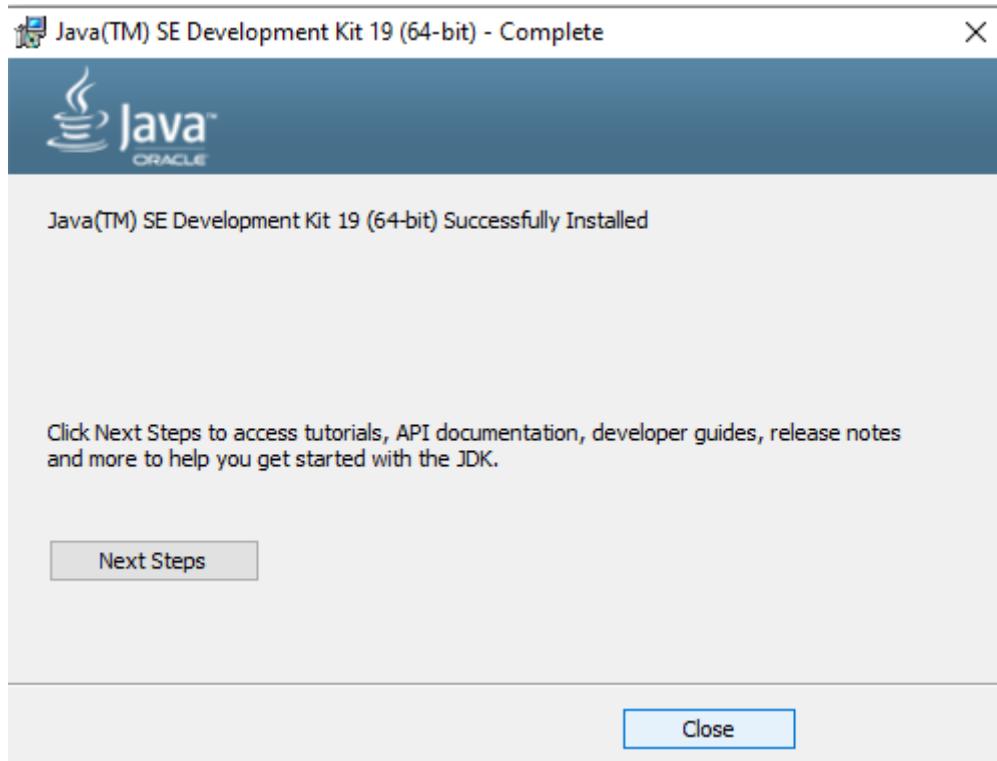
Instalo con cuenta administrador y utilizo con cuenta usuario.

El primer requisito es instalar el [JDK](#) para Windows de 64 bit, en este caso es la versión 19.

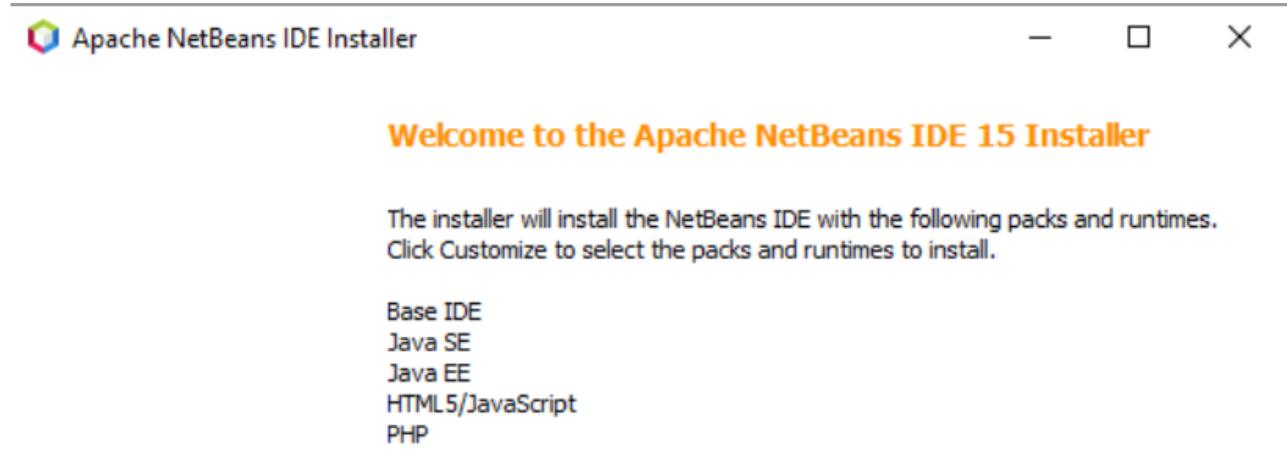
La instalación es un sencillo paso a paso.



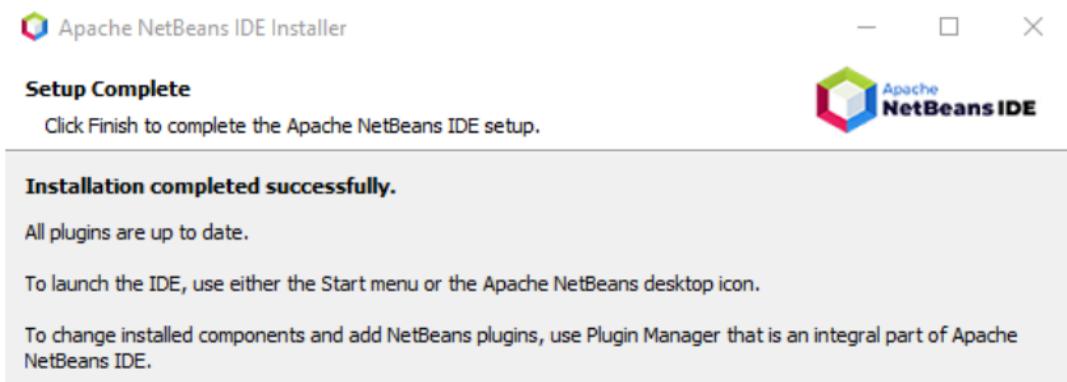
Hecho.



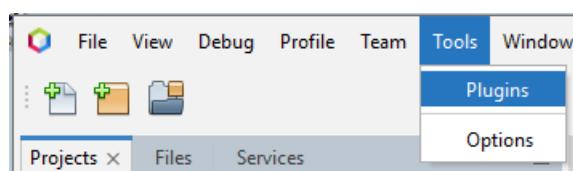
Una vez instalado el JDK pasamos a instalar el propio Netbeans. Instalamos la versión 15 para Windows de 64 bits. Al comenzar la instalación nos muestra los pluggins que vienen instalados por defecto.



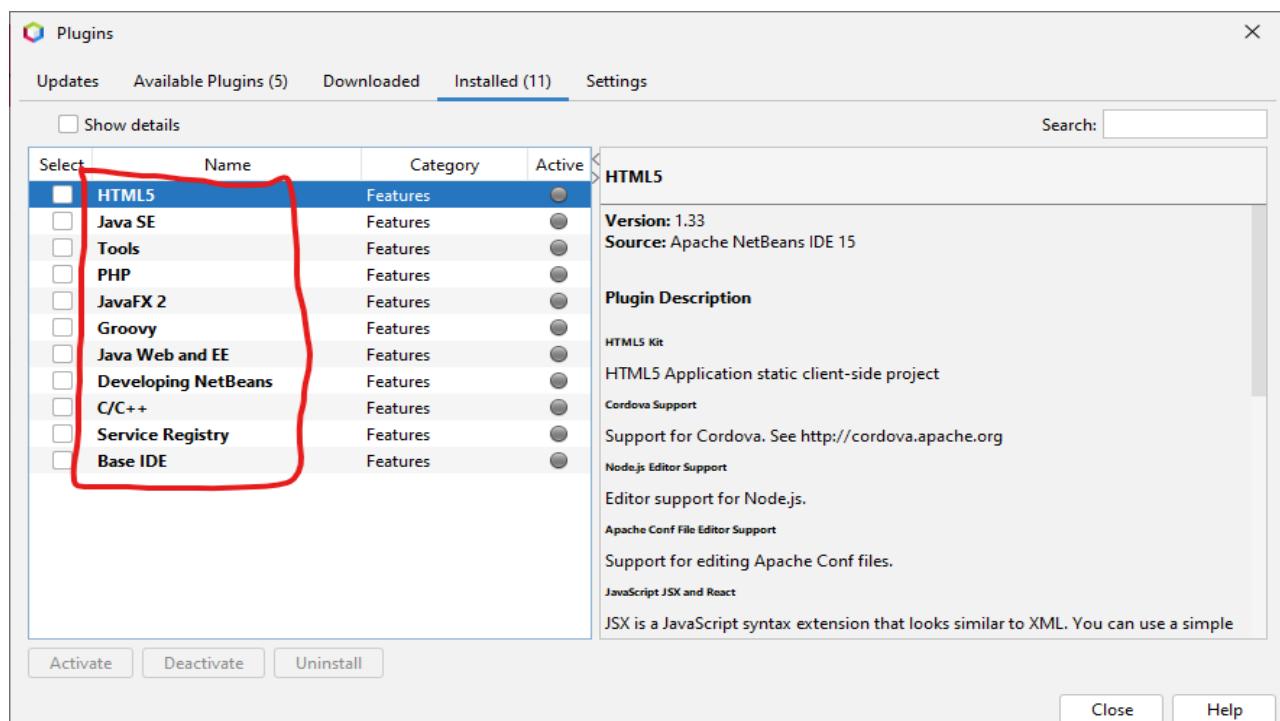
La instalación es un paso a paso eligiendo las opciones por defecto, al terminar la instalación con éxito, nos muestra la siguiente imagen para confirmarlo.



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.

The screenshot shows a list of top-downloaded plugins:

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.

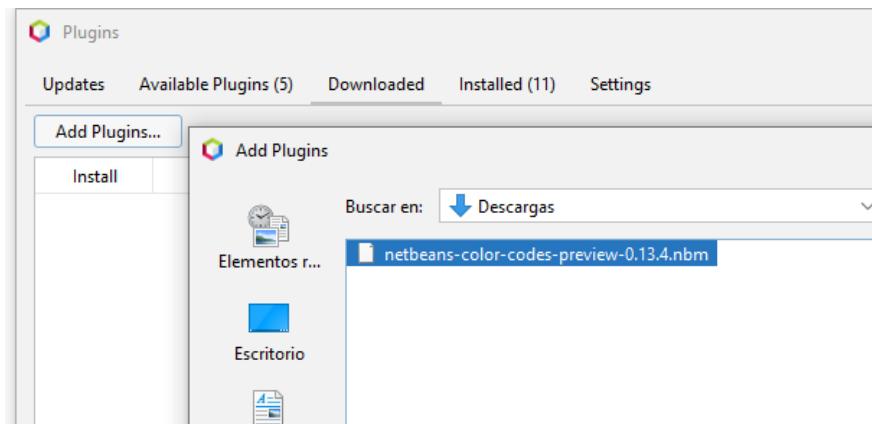
The screenshot shows the 'Color Codes Preview' plugin interface. It includes:

- Homepage: <https://github.com/junichi11/netbeans-color-codes-preview>
- Date: 2020-06-18
- Last Update: 2022-09-27
- Downloads: 143,702
- Version Buttons: 0.13.1, 0.13.2, 0.13.3, 0.13.4
- Download Matrix Buttons: NB 12.0, NB 14, NB 13 - Verified, NB 12.0 - Verified
- Code Editor Area:

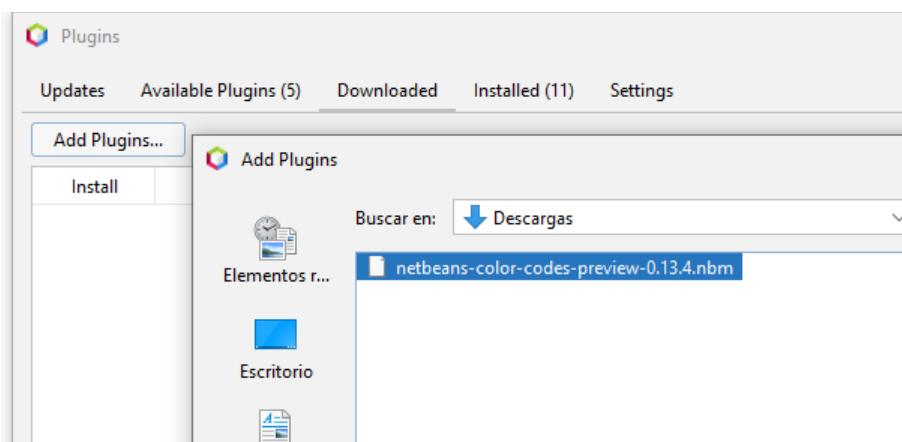

```

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
15     #aa0000
16     #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"/>

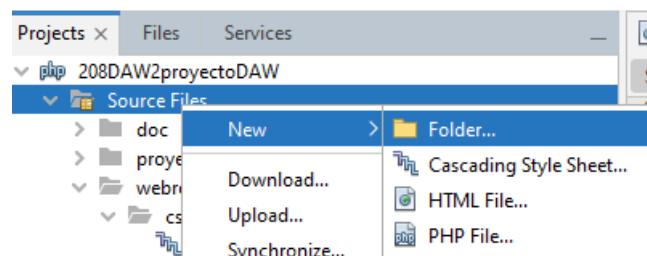
Ahora están activos y listos para facilitarnos el trabajo.

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	●

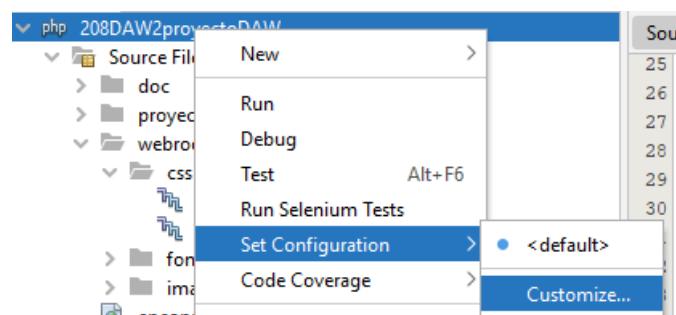
Comprobar y documentar el funcionamiento del plugin descargado e instalado, lo voy a realizar en el apartado “Creación de un proyecto nuevo” aprovechando que ya tendré un proyecto en el que poder hacerlo.

Conexión al entorno de desarrollo.

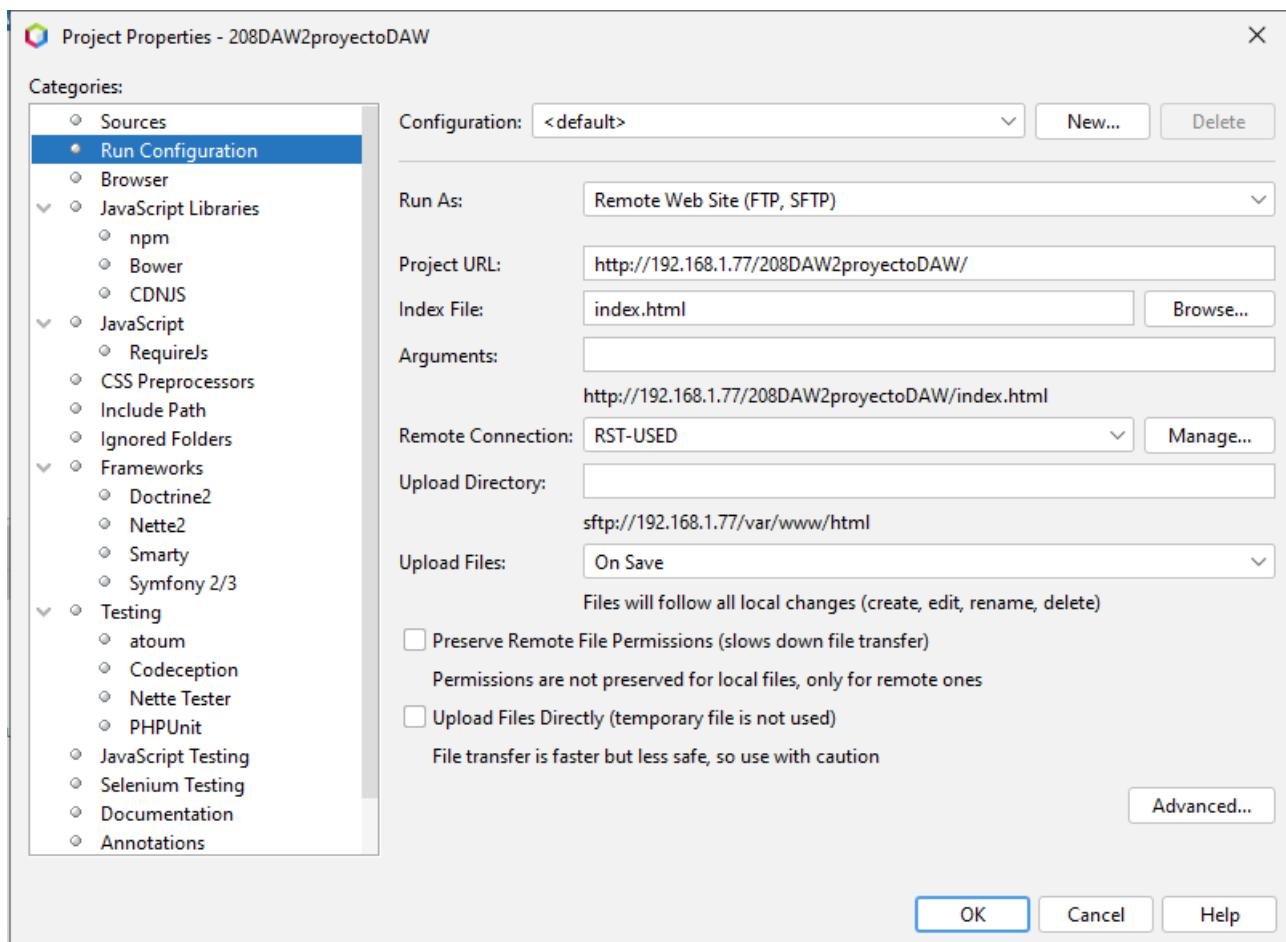
Conviene en este punto crear nuestro primer archivo, index.html, situado en la raíz de 208DAW2proyectoDAW.



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.

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

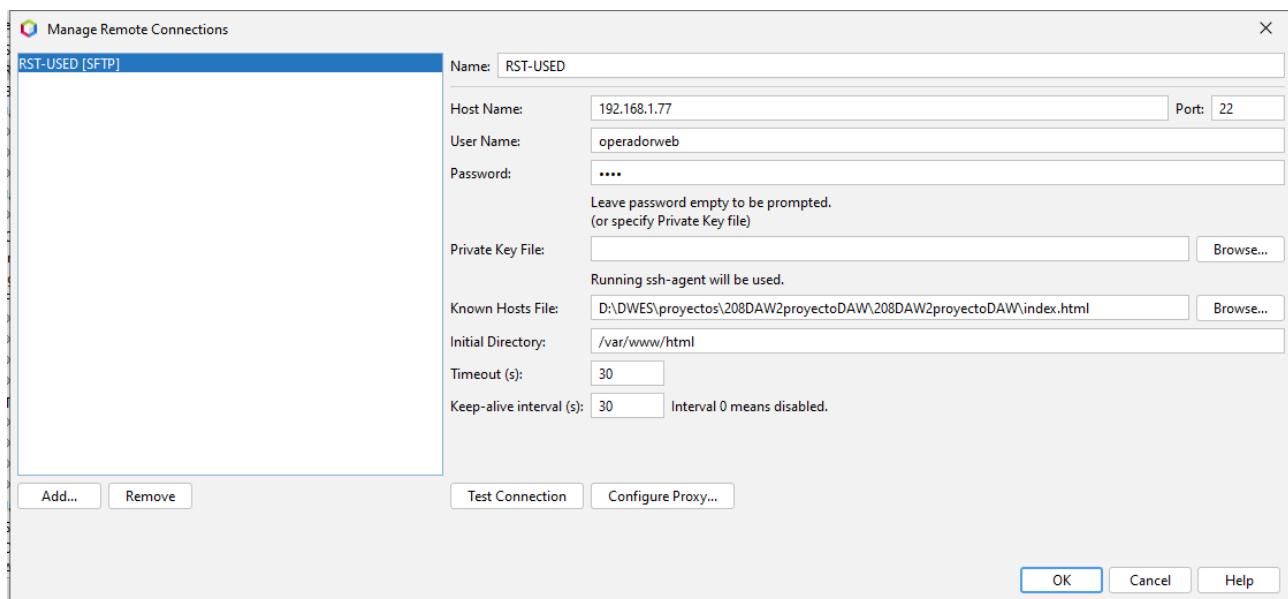
2^a pestaña.- **Run As:** Elegir Remote Web Site (FTP,SFTP) en el desplegable.

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

4^a 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.

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

6^a 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.

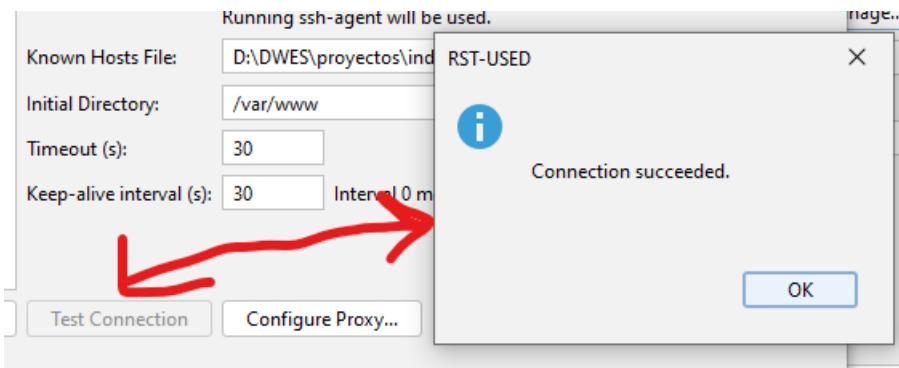


7^a pestaña.(Seguimos con la pantalla “Run Configuration”- “**Upload Directory**” :

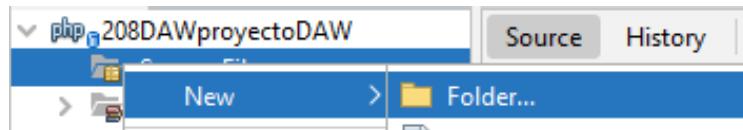
No escribiremos nada en ella, pues se autorrellena el valor con los datos ofrecidos en anteriores pestañas. Aquí nos aparecerá si está bien configurado la dirección del directorio del servidor remoto donde estará alojado nuestro proyecto.

8^a 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).

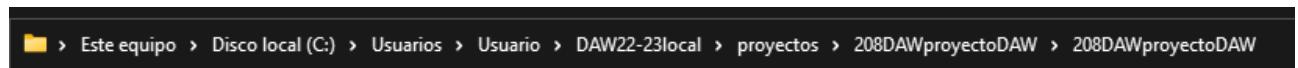
Con todo configurado podemos presionar OK y guardaremos nuestras elecciones, pero antes conviene hacer un “Test connection” y ver si los datos son correctos y se establece la conexión.



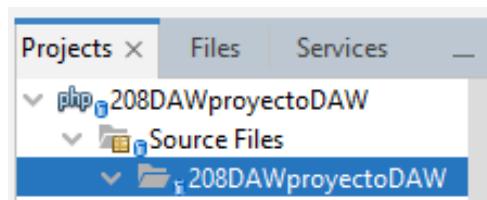
Una vez creado el proyecto y conectado al servidor, hay que crear una carpeta en “Source files” dentro de nuestro proyecto y darle el mismo nombre que al proyecto. De esta forma, esta carpeta que es la que va a contener todos los ficheros y directorios del proyecto, se nos mostrará en el directorio de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



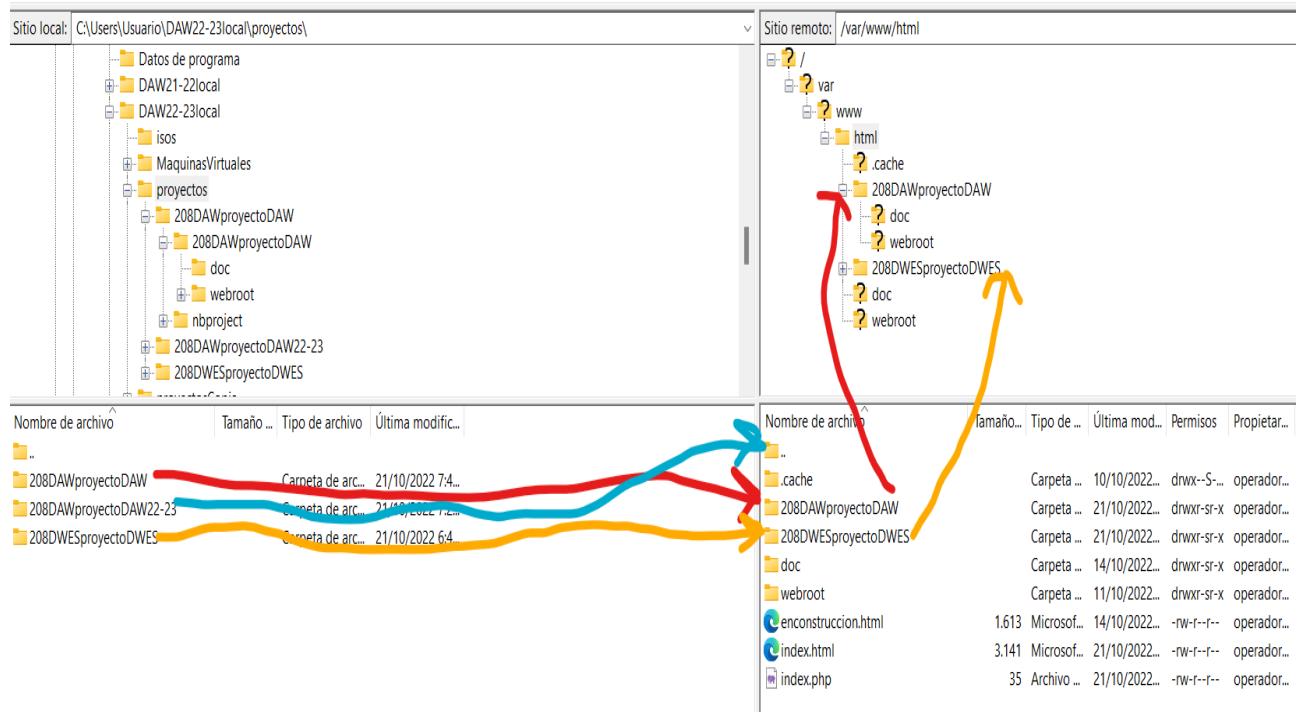
Esta será la estructura creada en local (en la imagen se ve en mi equipo de casa, por lo tanto la ruta difiere respecto a clase).



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

Mejor si escribo bien el comando sudo.

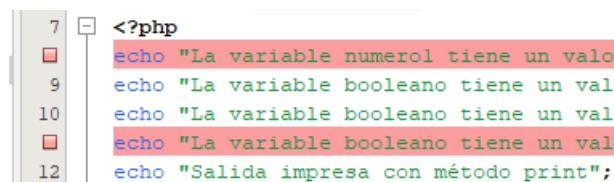
```
miadmin@RST-USDED:~$ sudo nano cd /etc/php/8.1/apache2/conf.d/20-xdebug.ini
```

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.

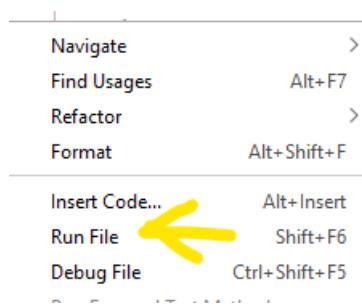
Para ello hemos de seleccionar un archivo .php de nuestro proyecto, establecer un “Breakpoint” en el código y dar a ejecutar el depurador.

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

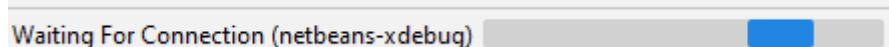


Señalado dejo el depurador para el proyecto en su conjunto.

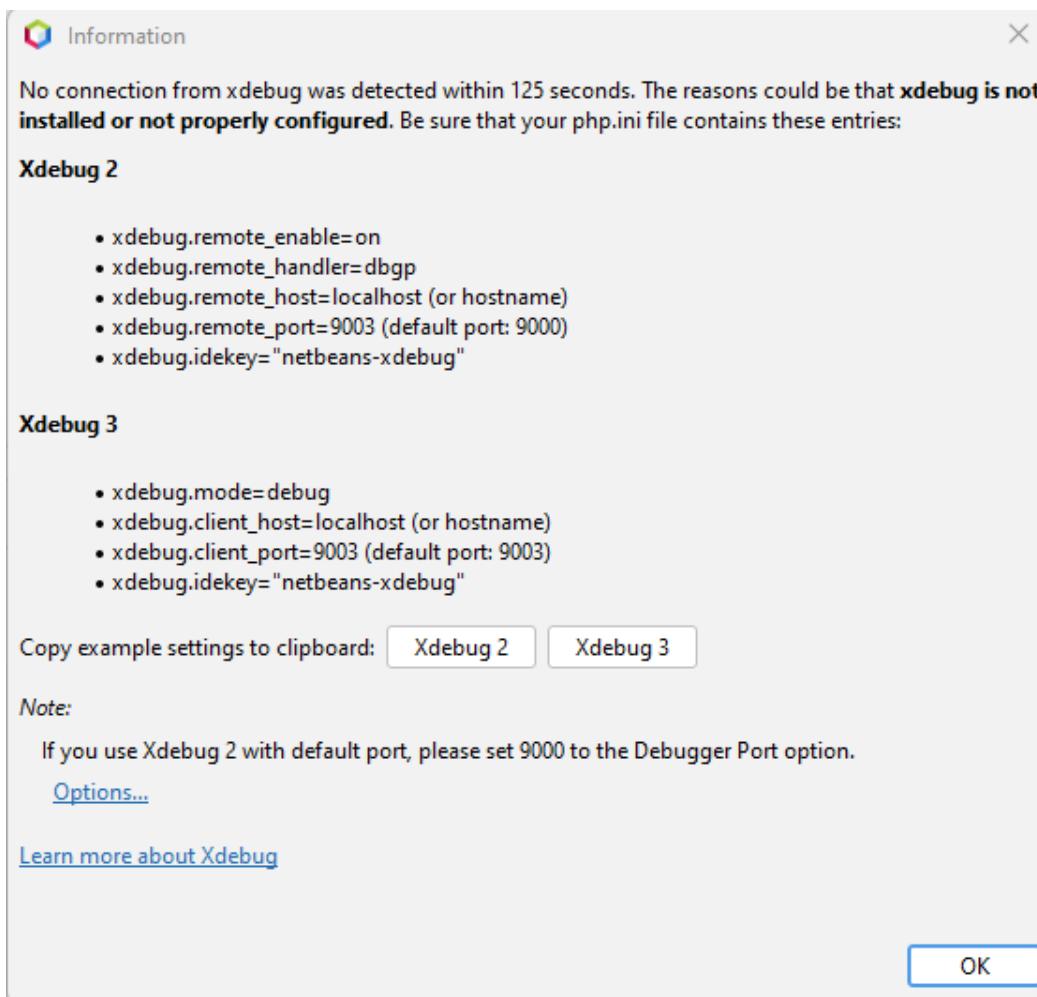
En este caso, al depurar un archivo que no es el principal del proyecto, tenemos que hacer click derecho sobre el código del archivo abierto y en el desplegable que aparece, elegir “Debug file”.



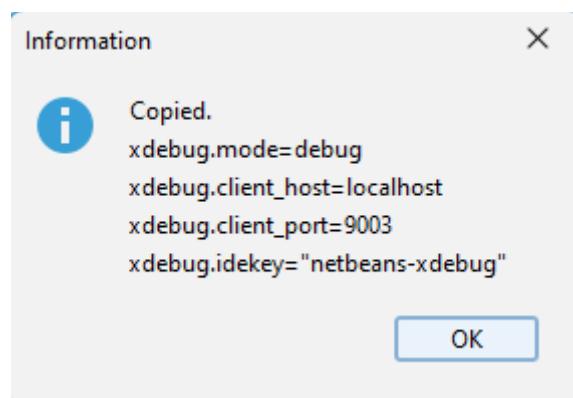
Esto arrancará el depurador y nos mostrará la página html en nuestro navegador. Al no estar aún configurado el archivo php.ini, la depuración se queda en espera.



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.

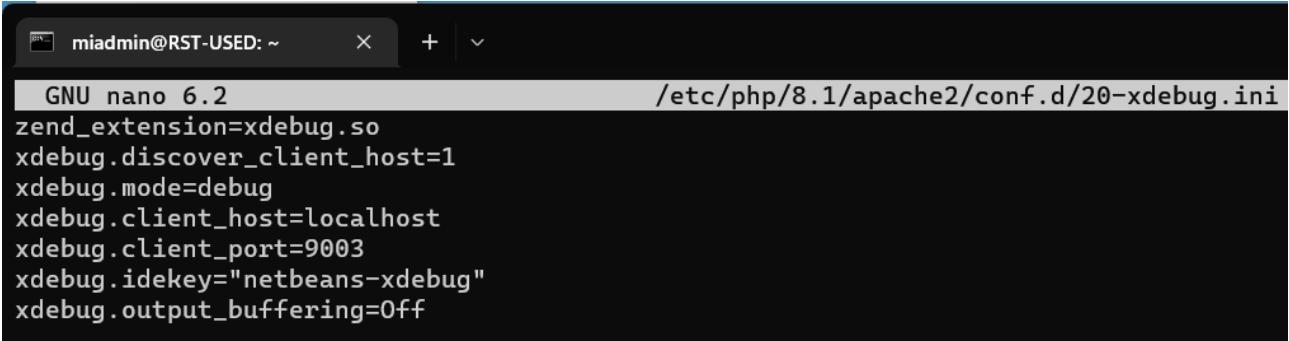


Si tenemos habilitada la comunicación ssh entre el cliente y el servidor podremos hacer uso de la herramienta “pegar” y con ctrl+v pegar en este archivo los valores que copiamos de NetBeans.

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

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



The screenshot shows a terminal window titled "miadmin@RST-USDED: ~". The command entered is "sudo nano /etc/php/8.1/apache2/conf.d/20-xdebug.ini". The nano editor is displaying the contents of the file, which include configuration settings for Xdebug:

```
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=Off
```

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

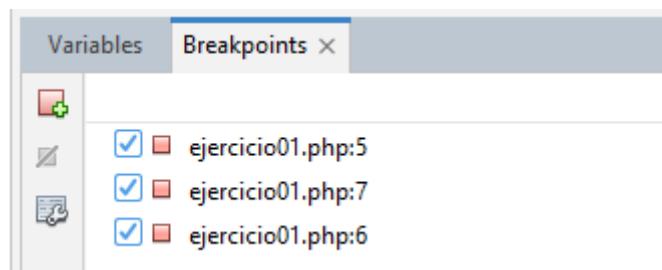
Ahora el netbeans-xdebug está running.



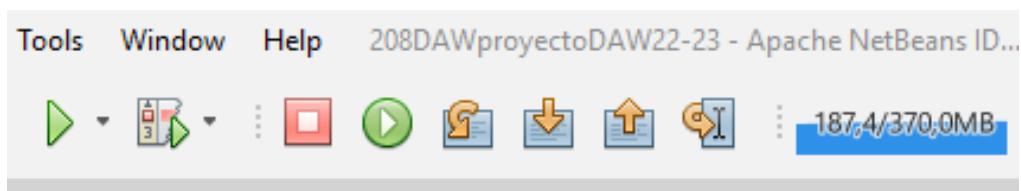
Y podremos ver nuevas ventanas en el entorno de desarrollo que nos muestran las variables por un lado.

Variables		Breakpoints
	Name	Type
...	\$booleano	boolean
...	\$cadena	string
...	\$numero1	integer
...
...	true	
...	"Hola Mundo"	
...	1	

Los breakpoints por otro.

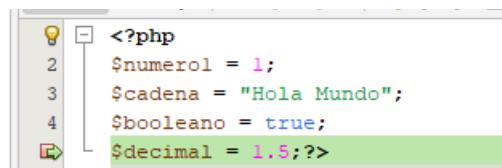


Y en la parte superior de IDE estarán ahora habilitadas varias pestañas que sirven para ejecutar diferentes acciones sobre el depurador.



Prueba de depurado de un archivo .php.

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



```
<?php
$numero1 = 1;
$cadena = "Hola Mundo";
$booleano = true;
$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

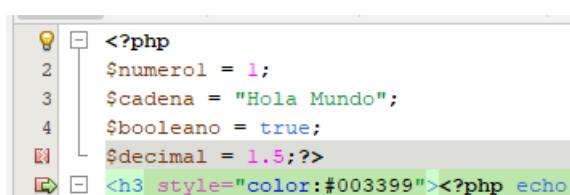
	true	true
	"Hola Mundo"	"Hola Mundo"
	1	1

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
$numero1 = 1;
$cadena = "Hola Mundo";
$booleano = true;
$decimal = 1.5;?>
<h3 style="color:#003399"><?php echo
```

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

	Name
>	Superglobals
↳	\$booleano
↳	\$cadena
↳	\$decimal
↳	\$numero1
	boolean
	string
	float
	integer

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.

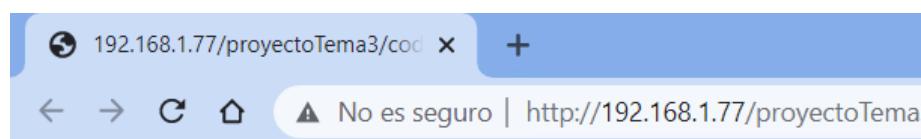
Podemos parar la depuración dando al play del depurador hasta que haya ejecutado todo el archivo.



O al stop del depurador 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.

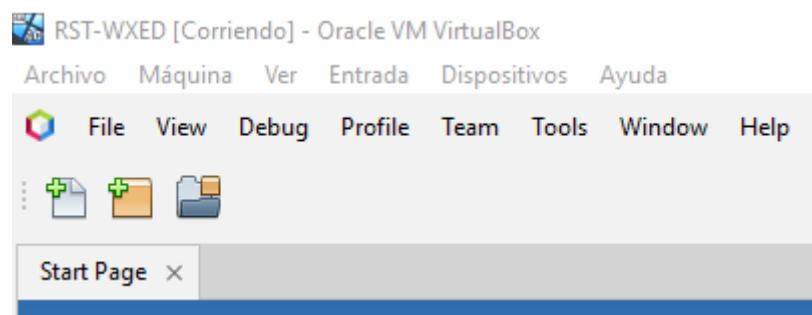
Lo primero es elegir entre crear la carpeta proyectos (aquella en la que estarán alojados todos nuestros proyectos) en la ubicación elegida con el selector de archivos o crearla desde dentro del propio NetBeans al tiempo de crear un primer proyecto. Creo también en este paso la carpeta para el primer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

Aquí la creo con el selector de archivos.

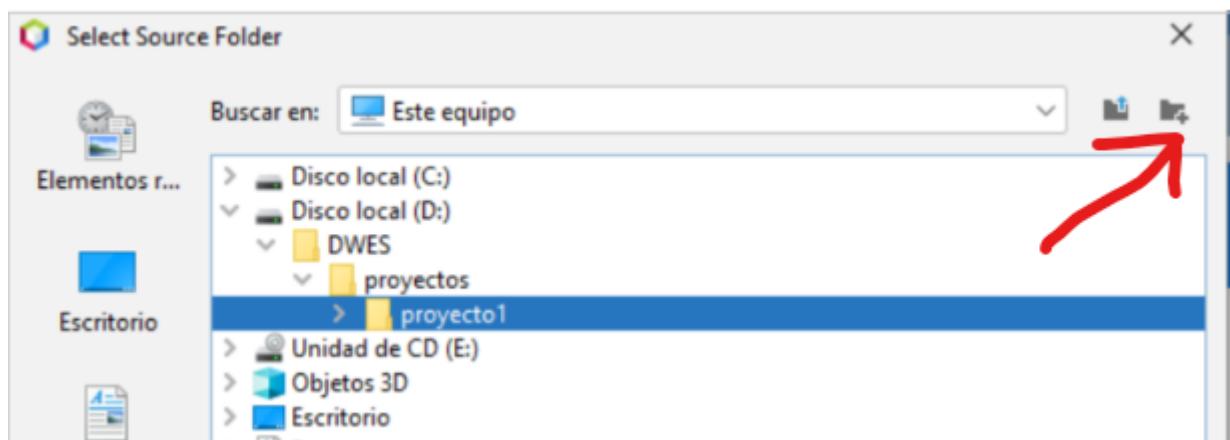
Actualizar las 3 siguientes capturas:



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



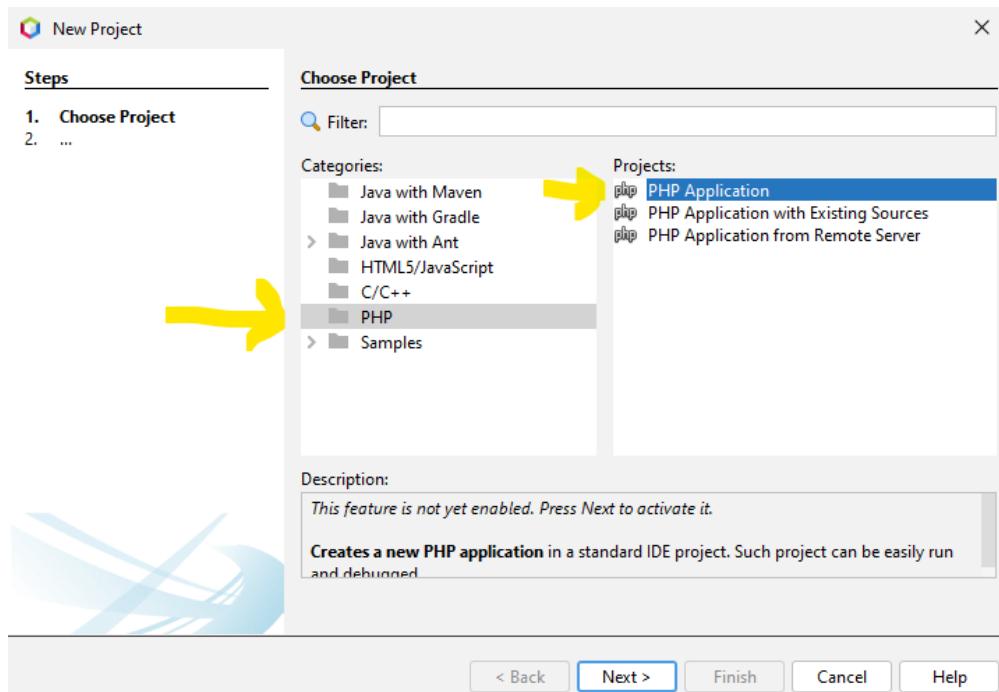
Si no hemos creado la carpeta con anterioridad, ahora podemos hacerlo, el icono señalado es el que crea las carpetas. Tantas como hagan falta.



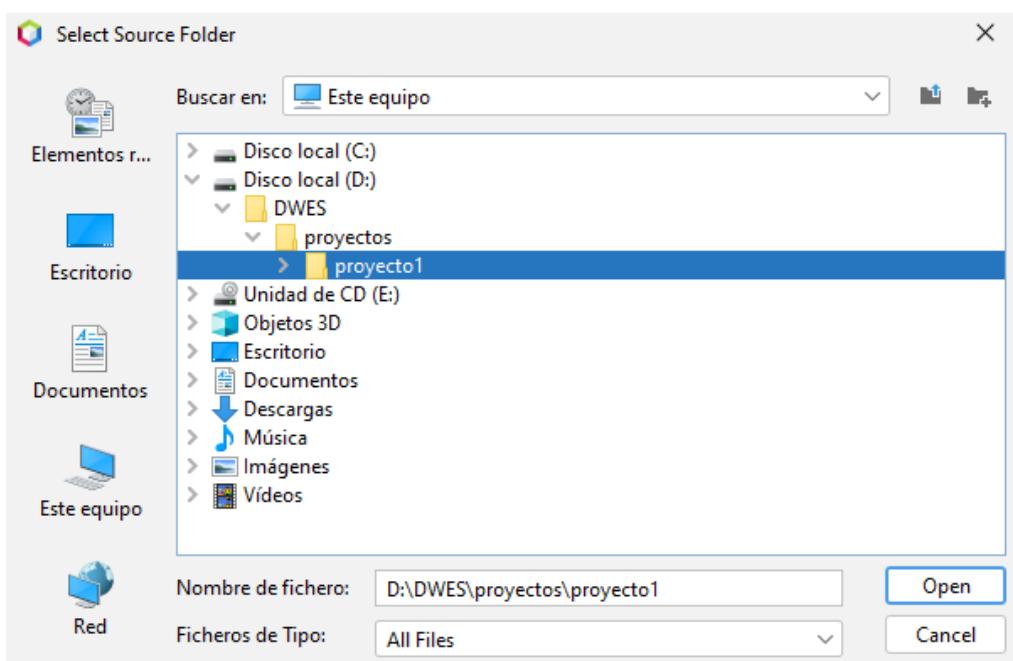
Proyecto nuevo:

“PHP Application”.

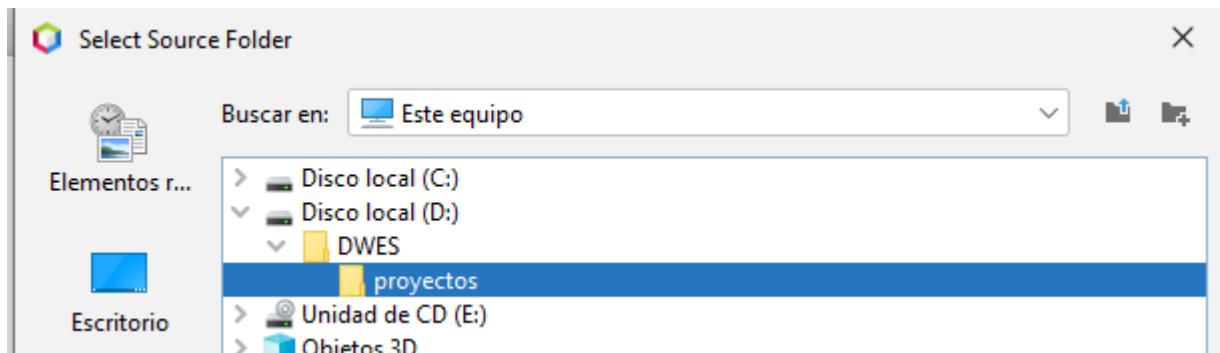
Lo siguiente es elegir el lenguaje en que vamos a programar y de que tipo, si nueva, desde un servidor remoto o con archivos previos existentes, en este apartado elegimos “PHP Application” al tratarse de un proyecto desde 0.



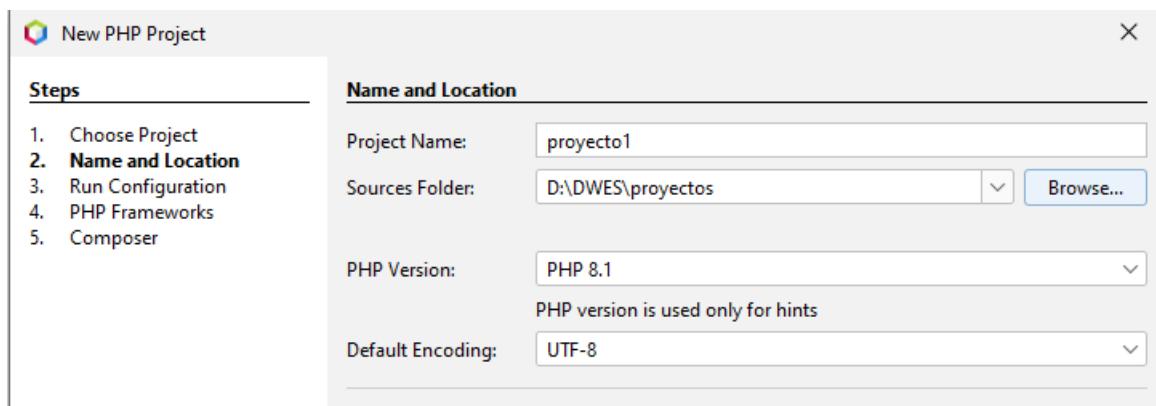
Después elegimos la carpeta fuente; en este punto es donde podemos crear las carpetas que necesitemos.



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 querremos obtener una visualización de su carpeta contenedora en el servidor.

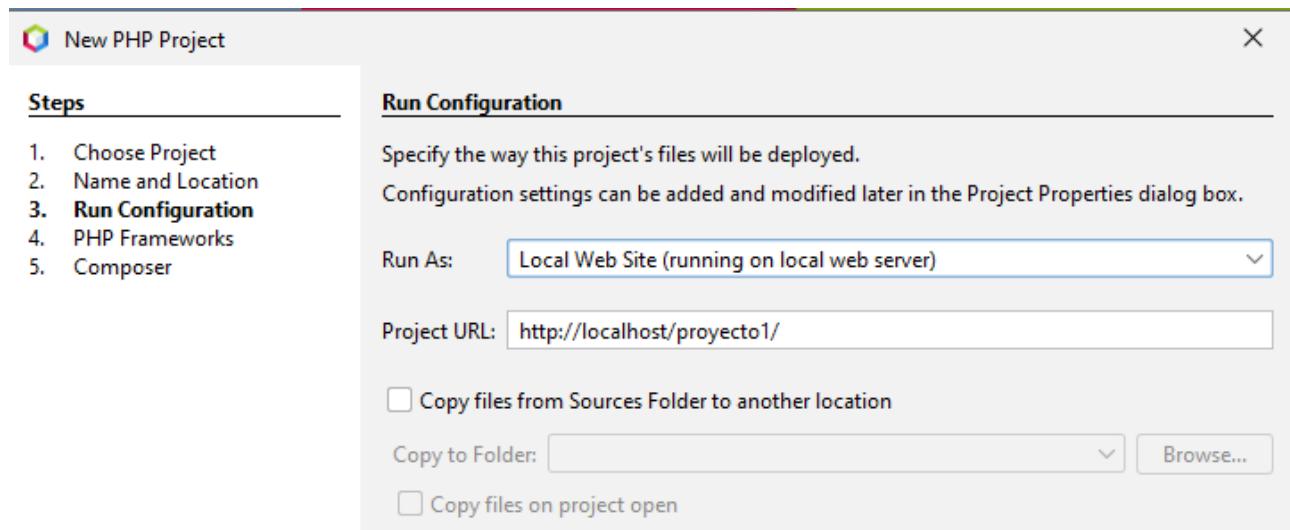


Nombre y carpeta fuente, también nos informa de la versión de PHP y de la codificación por defecto.

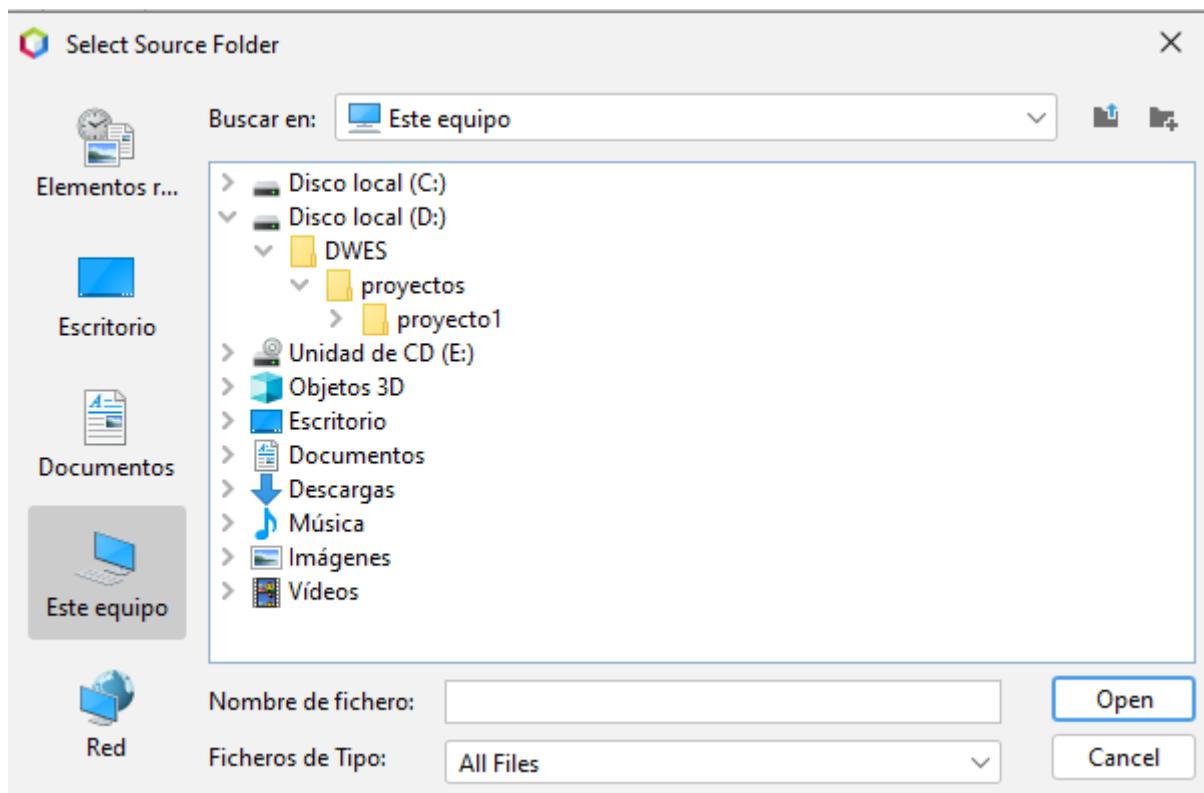


Lo siguiente es el “Run Configuration”. En primer término voy a dejar sus valores por defecto como están, a terminar de crear el proyecto y a renglón seguido a volver al “Run Configuration” habiendo quedado documentados todos estos pasos en el apartado [**“Conexión al entorno de desarrollo”**](#).

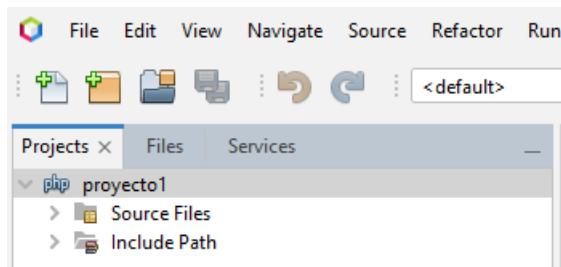
Estos son sus valores por defecto.



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 y proyecto creado.

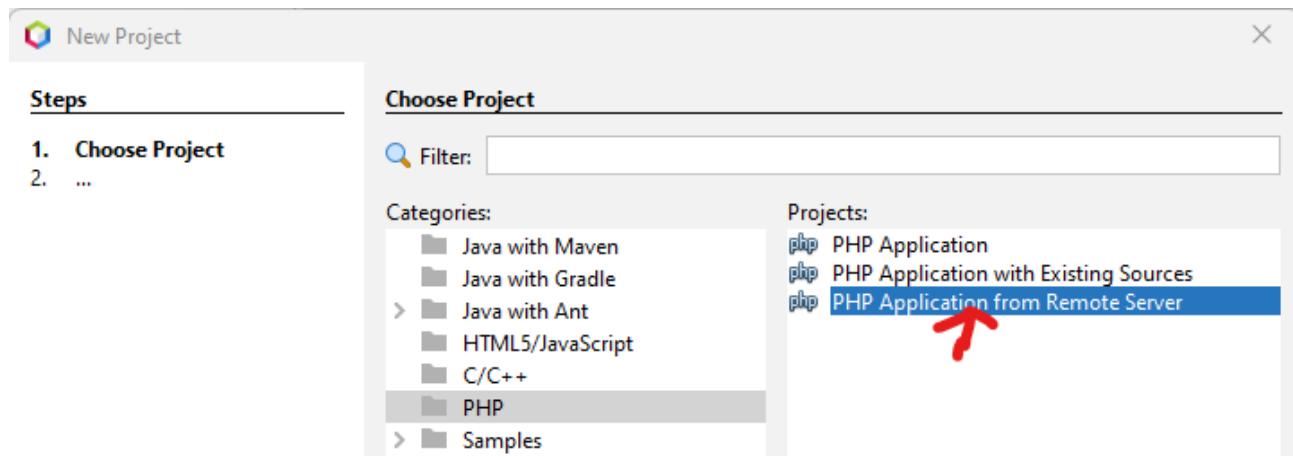


Creación de proyecto PHP desde un servidor:

“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.



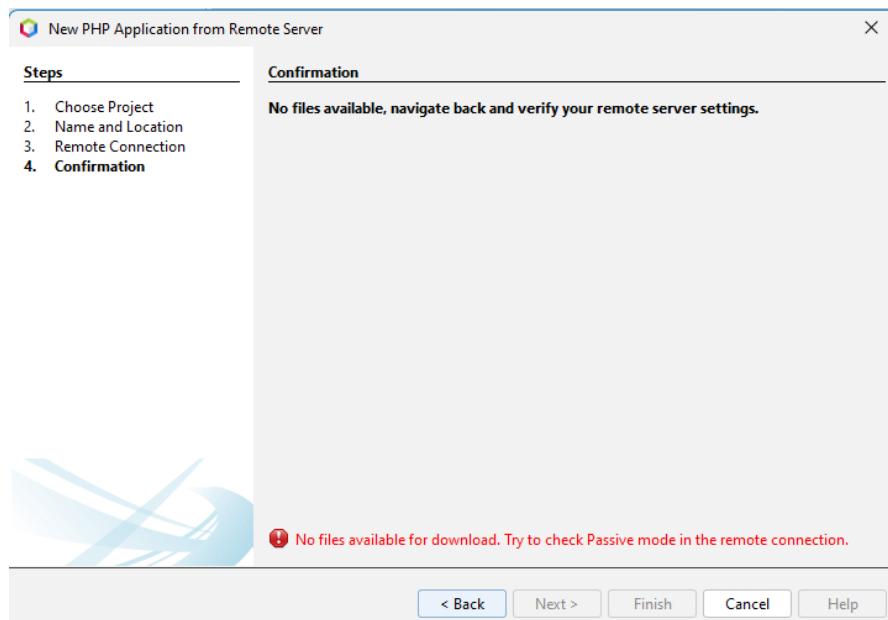
A practicar inglés. Resumiendo, baja desde un servidor una aplicación PHP.

Description:

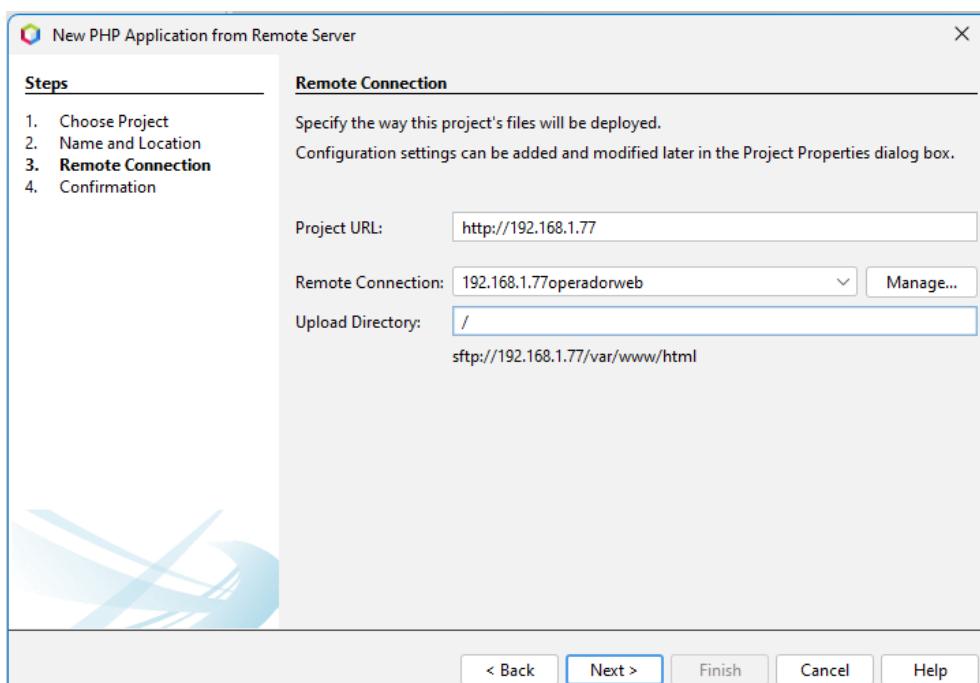
Downloads an existing PHP application, creates a standard IDE project for it and sets up the project properties accordingly (automatic uploading of local changes to the server). Such project can be easily run and debugged.

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,

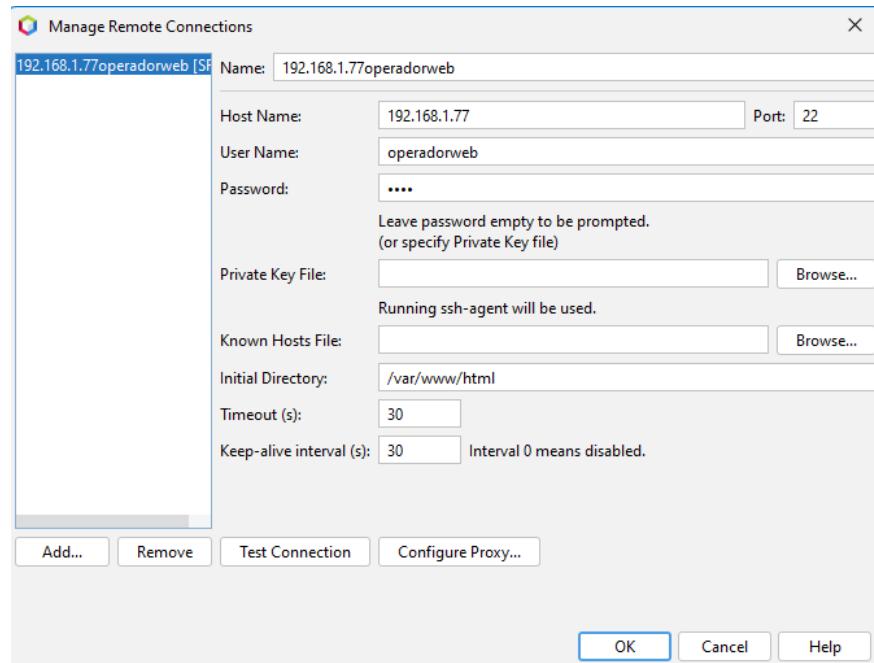


Al tratarse de una conexión al servidor, hemos de indicarle la dirección ip del servidor precedida del protocolo http y el “Upload Directory” será la carpeta raíz que contiene el proyecto que queremos “traer” a NetBeans en local

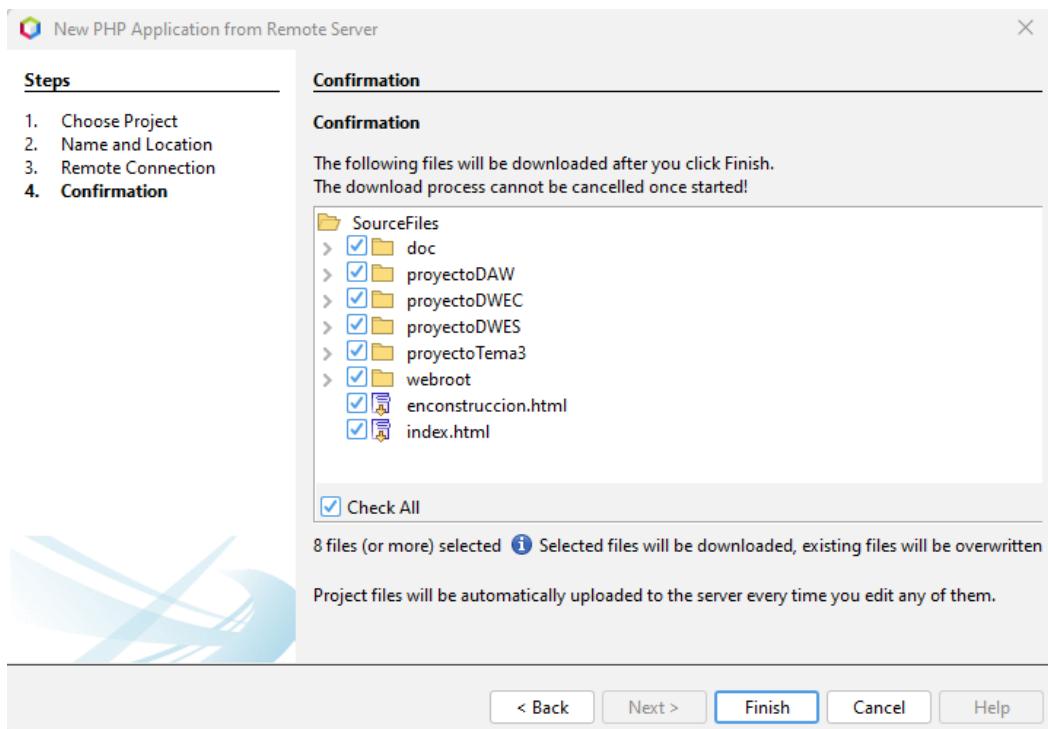


Si elegimos el botón “Manage” de la “Remote Connection”, muestra esta pantalla.

En la que podemos dar nombre y personalizar la conexión. Los valores son los mismos que para el proyecto nuevo desde 0.



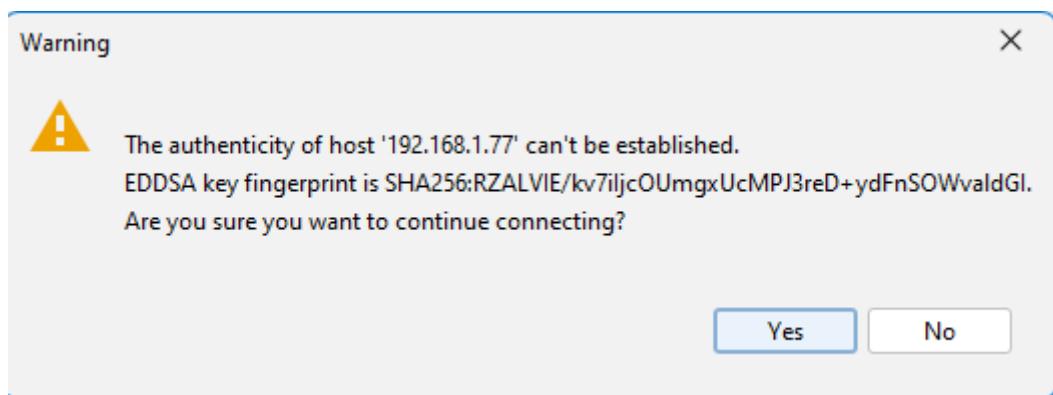
Lo siguiente que se nos muestra es esta pantalla, lo que confirma que hemos configurado todo correctamente y nos muestra las carpetas y ficheros presentes en el directorio del servidor que le hemos indicado.



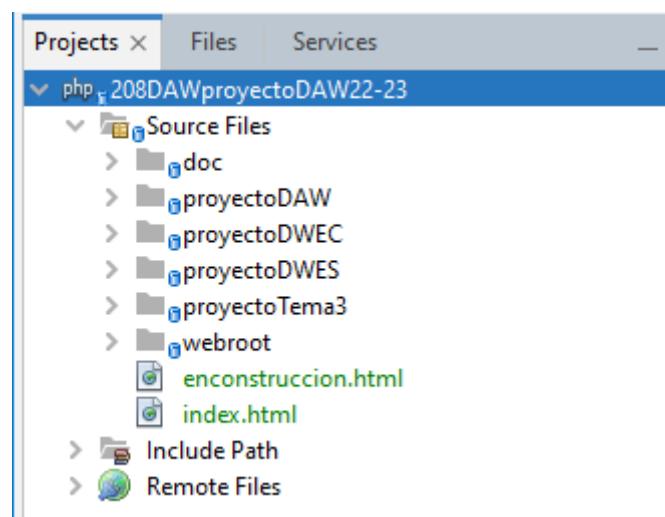
Estos mismos directorios y archivos los muestro ahora en el servidor.

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

Nos pide confirmación.

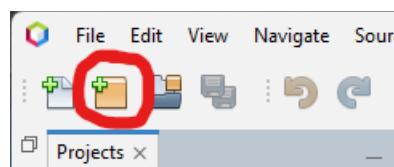


Y ya tenemos el proyecto del servidor “bajado” a local y reconocido como proyecto por NetBeans.

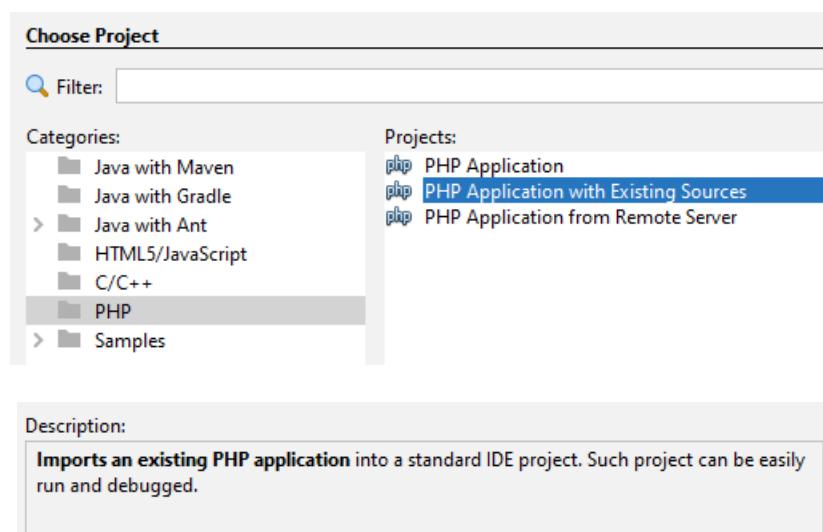


Creación de proyecto PHP con fuentes existentes:**“PHP Application with Existing Sources”.**

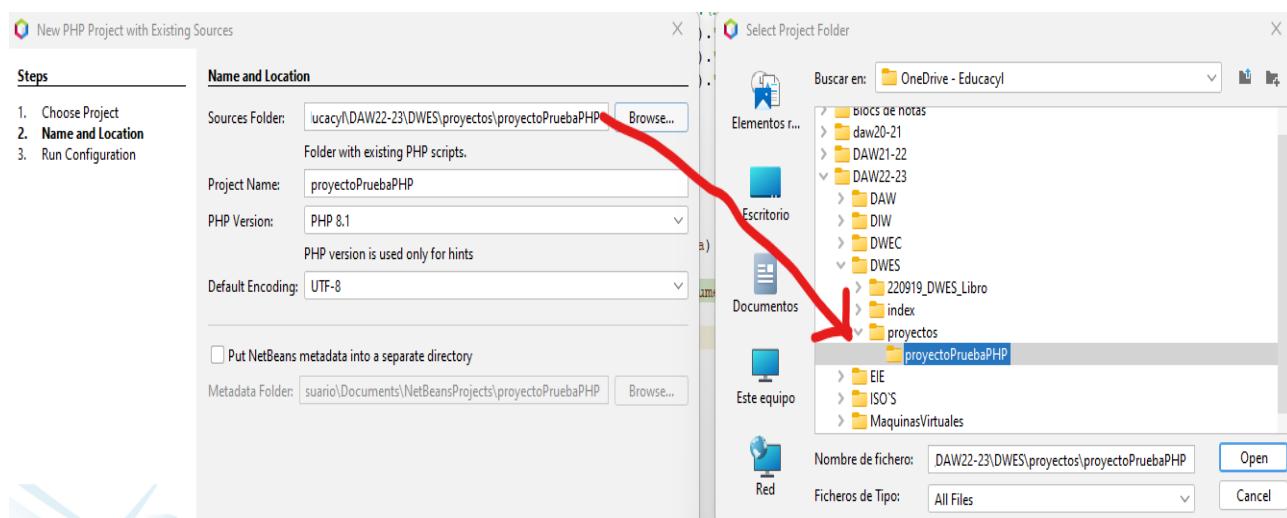
Comenzamos con nuestra vieja amiga la carpeta “New Project”



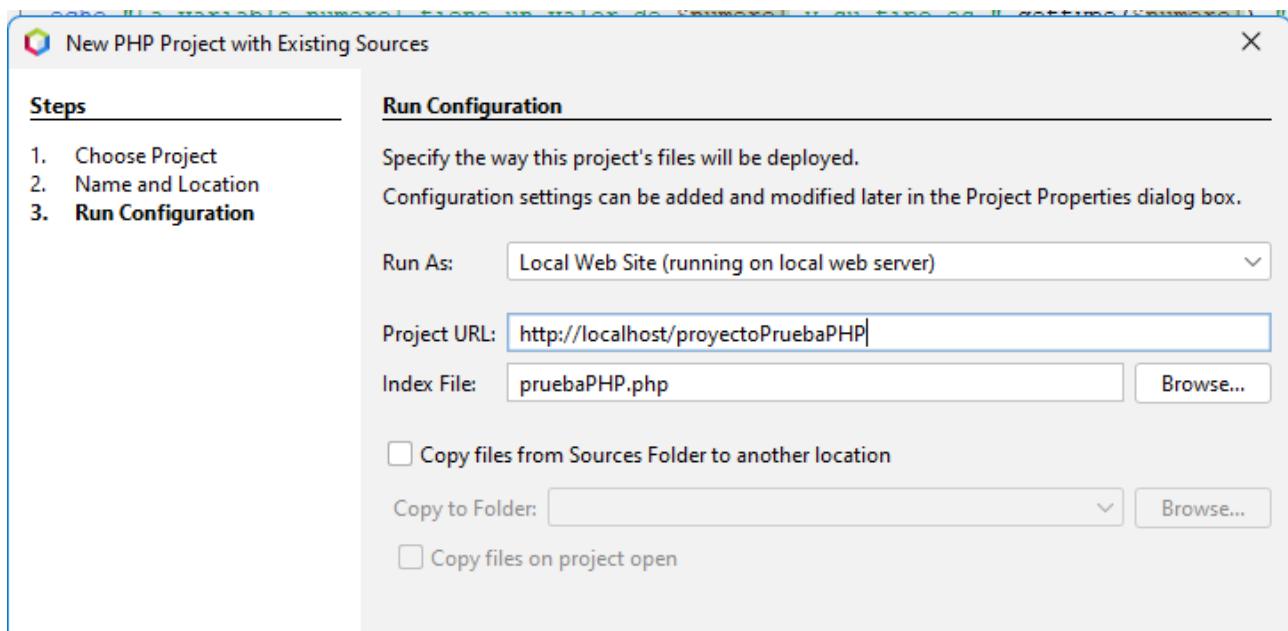
Esta vez el proyecto con el que queremos trabajar en NetBeans es un proyecto existente.



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...