

Ejercicio tema 2 (documentación)

Contenido

WXED-WINDOWS 10 Entorno de Desarrollo	2
Configuración inicial	2
Nombre	2
Configuración de red	2
Software	2
Navegador	2
FileZilla.....	3
Filezilla	5
NetBeans	7
USED-UBUNTU SERVER Entorno de Desarrollo.....	15
Configuración inicial	15
Nombre	15
Muestra hostname:	15
Cambiar el hostname:	15
Configuración de red	15
Muestra IP:	15
Cambiar IP:	15
Aplicar la configuración:	16
Servicios.....	16
Apache.....	16
PHP	17
Xdebug.....	17
MySQL	18
Sitios basados en puertos y enjaular usuarios:	20
gestión de errores Apache	22
Alias apache.....	22
Cambiar el orden de prioridad de los index apache	22
DNS	23
GIT	25

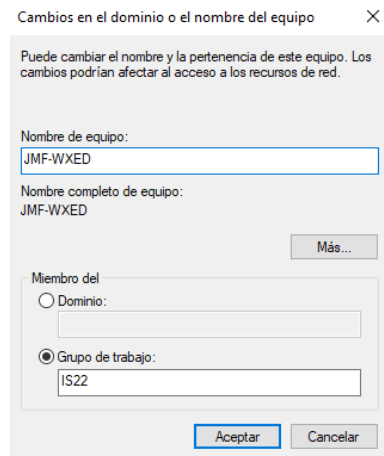
WXED-WINDOWS 10 Entorno de Desarrollo

Configuración inicial

Nombre

Para cambiar el nombre nos dirigimos a Equipo, botón derecho, propiedades.

Clicamos en cambiar configuración, en la pestaña nombre de equipo clicamos en Cambiar.

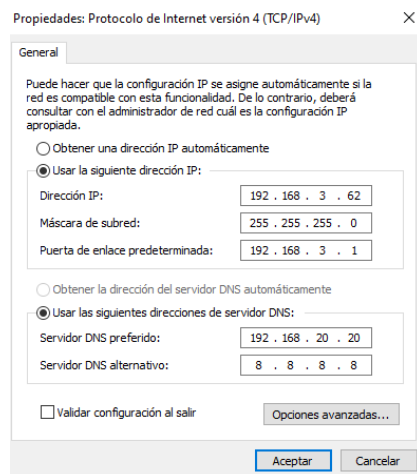


El nombre de la maquina es: XXX_WXED, siendo en este caso XXX las iniciales del alumno.

Configuración de red

Para abrir la configuración de red nos situamos sobre el icono de red (Abajo Derecha) y clicamos en abrir configuración de Red e internet.

Boton derecho sobre la tarjeta de red y clicamos en propiedades, protocolo IPv4.



La IP se genera en base al host 50 de la red + el numero del equipo anfitrión. Gateway: la puerta de enlace es el host 1 de la red 192.168.3/24.

Software

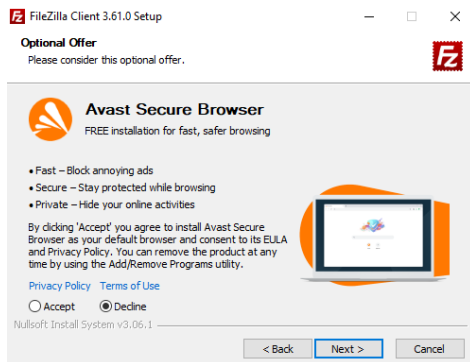
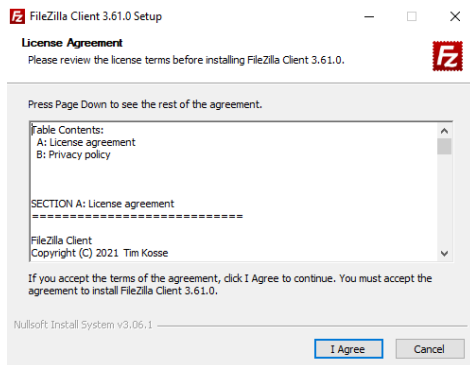
Navegador

Instalaremos nuestro navegador de preferencia(Recomendable tener varios) en este caso he elegido Chrome.

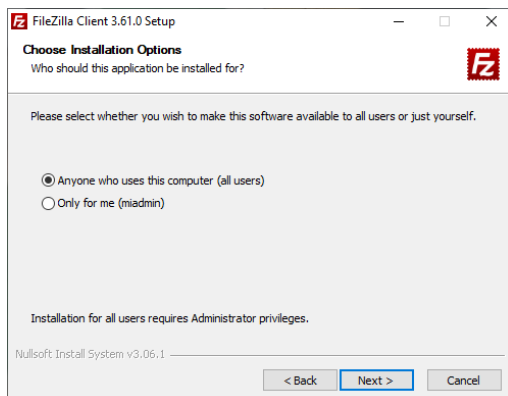


FileZilla

Instalación

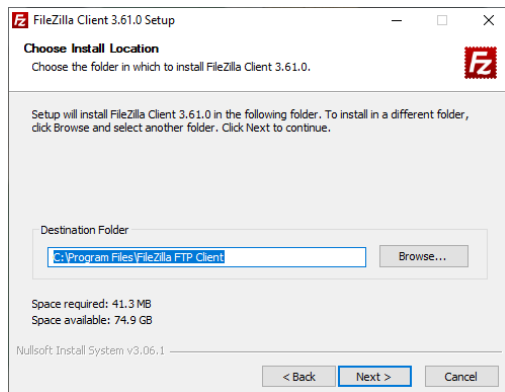
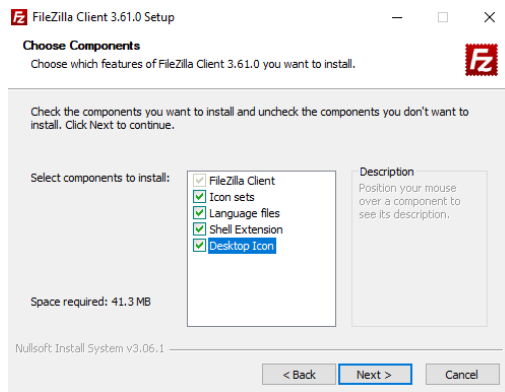


Rechazamos el addware...

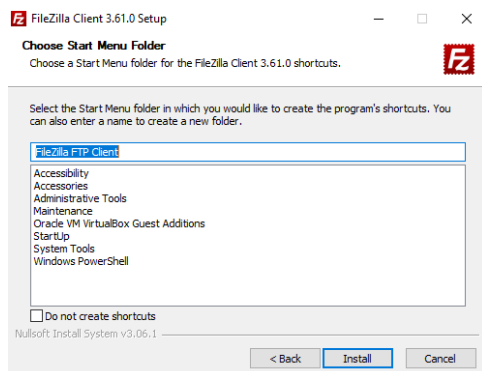


Instalamos en todos los usuarios...

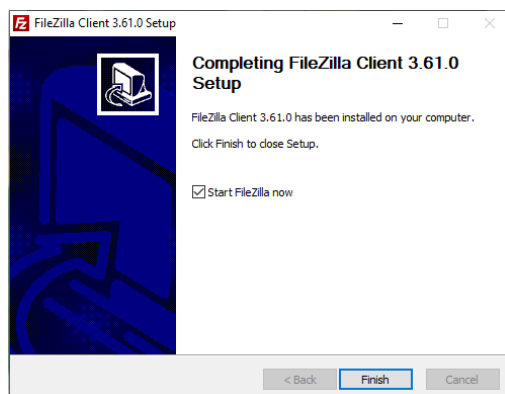
Josué Martínez Fernández



Elegimos la ruta de instalación...



Directorio para los atajos de teclado...



Conexión



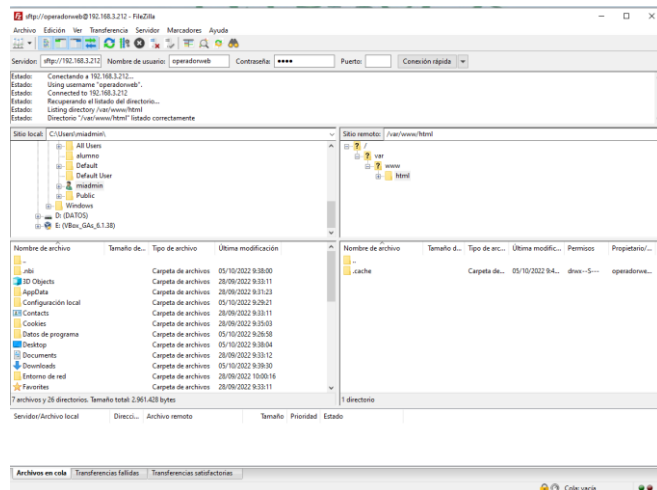
Josué Martínez Fernández

Servidor: IP de la maquina servidor Ubuntu (200+anfitrion).

Nombre de Usuario: Usuario que se conecta (explicada su creación mas adelante).

Contraseña: contraseña del usuario.

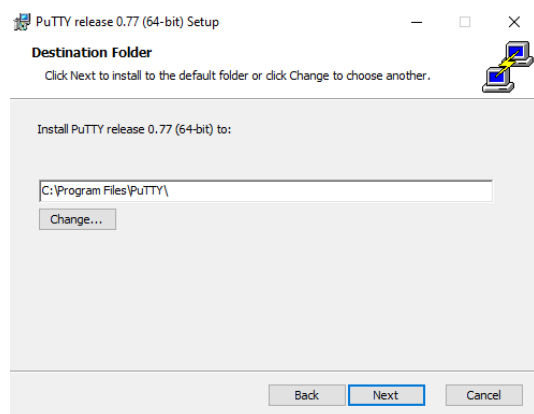
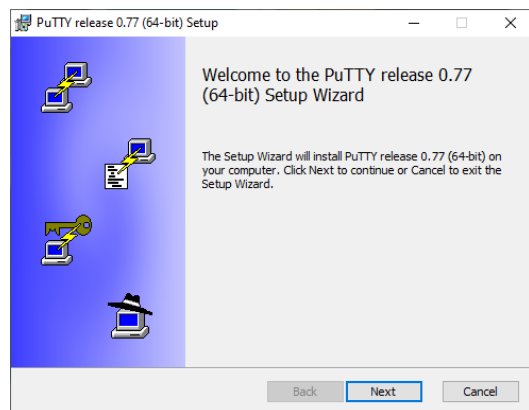
Puerto: Puerto 22 por defecto para SFTP.



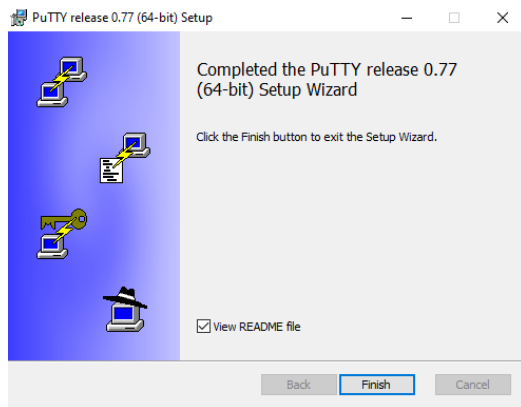
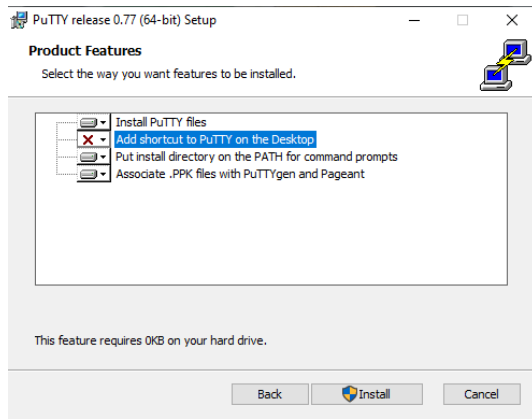
Comprobamos la conexión... En el lado izquierdo se encuentran los archivos y directorios locales, a la derecha los alojados en el servidor.

Filezilla

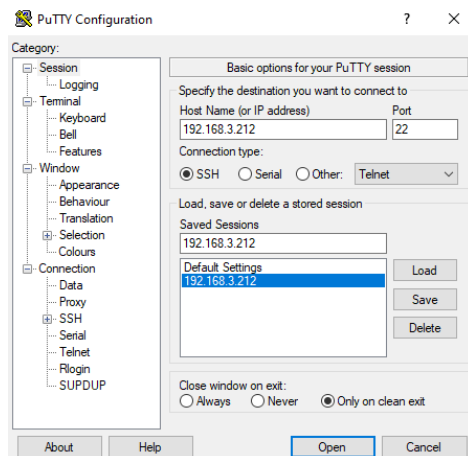
Instalación



Ruta de instalación...

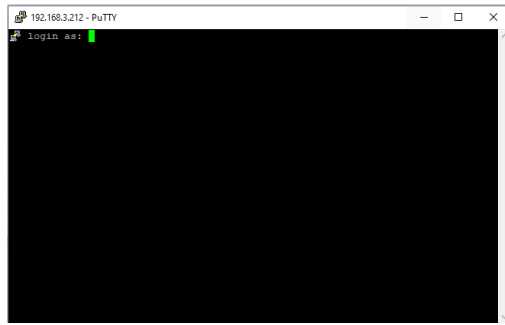


Conexión



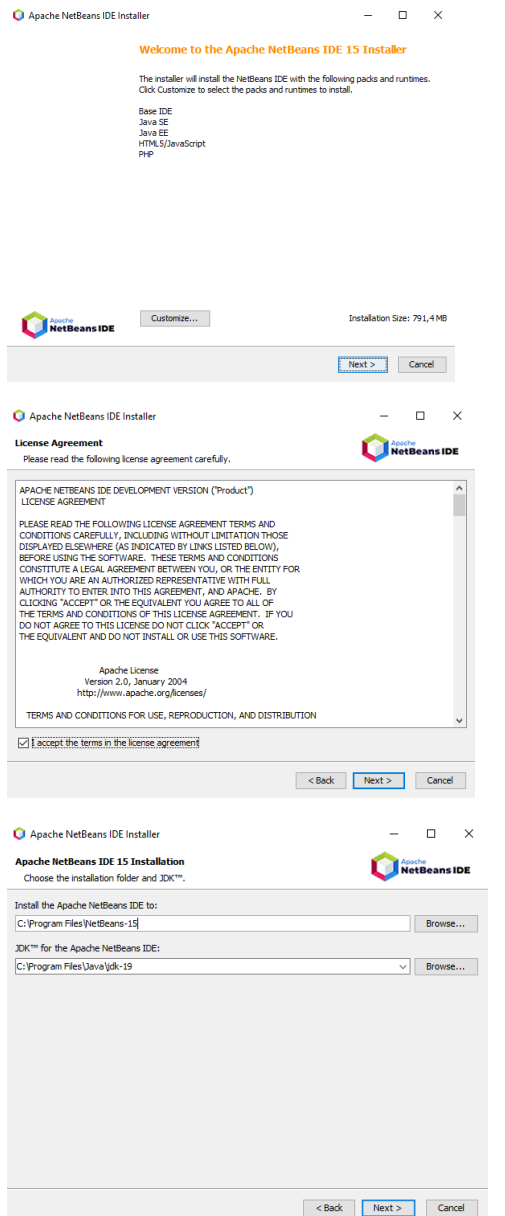
Hostname: IP de la maquina servidor Ubuntu (200+anfitrion).

Port: Puerto 22 por defecto para SSH



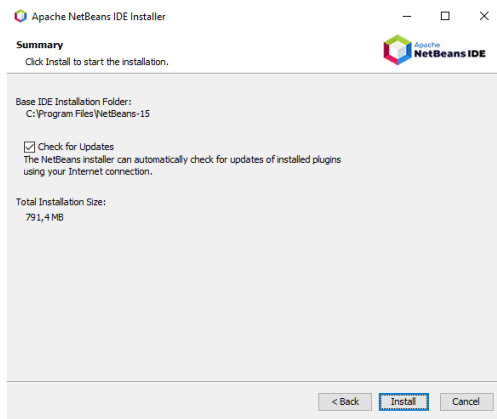
NetBeans

Instalación

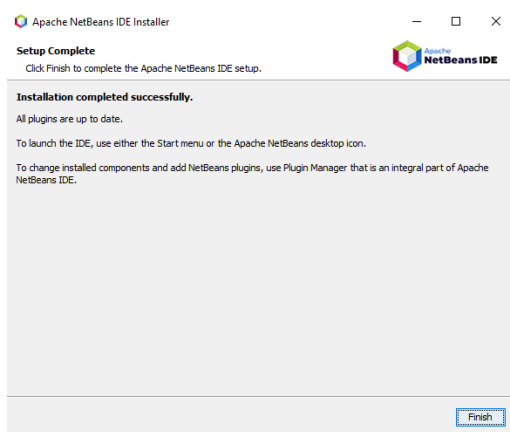


Lugar donde se instalara el IDE.

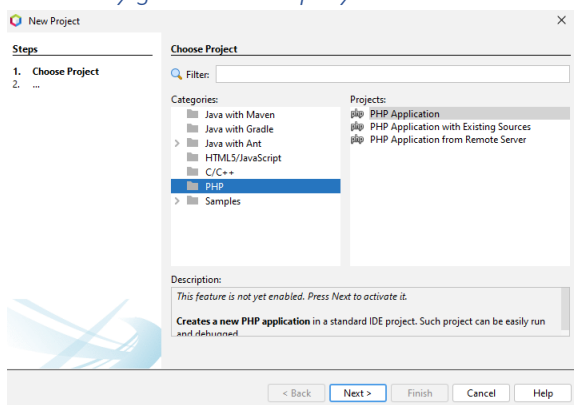
Lugar del que tomar la ruta del JDK.



Búsqueda de actualizaciones



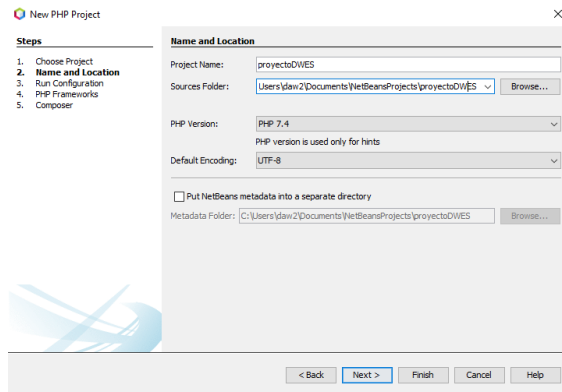
Creación y gestión de un proyecto



Creamos un nuevo proyecto de tipo php y seleccionamos la primera opción.

La segunda seria en caso de que existiesen ya una aplicación, creando el proyecto de forma posterior.

La tercera para descargar la aplicación de un servidor y crear el proyecto.

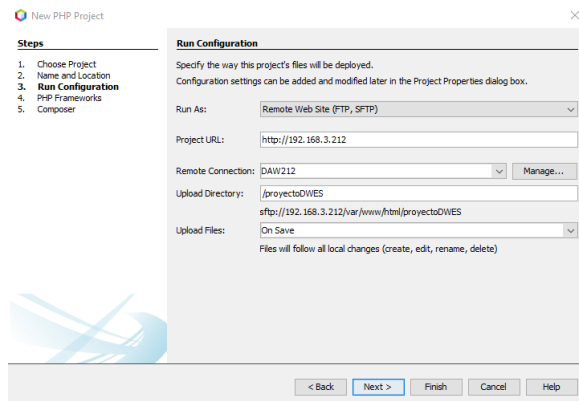


Project name: Nombre del proyecto.

Sources folder: Ubicación local de creación del proyecto.

PHP versión: Versión de php.

Default Encoding: Tipo de codificación de caracteres.



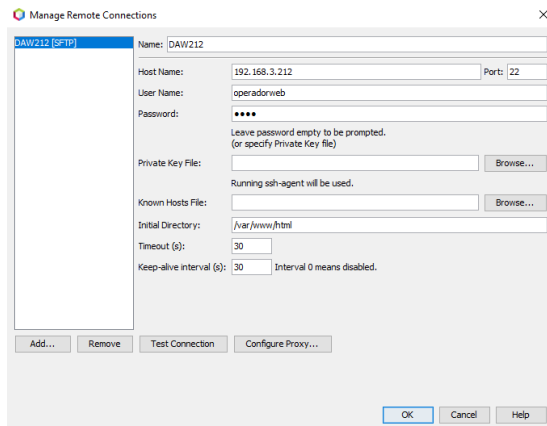
Run As: Seleccionamos un servidor remoto en el que subiremos el proyecto por SFTP.

Project URL: URL remota del proyecto en este caso la IP del servidor.

Remote Connector: Definido en la siguiente captura, es básicamente la conexión.

Upload Directory: Lugar donde se va a subir el proyecto dentro del servidor.

Upload Files: En este caso on save, ya que los archivos se subirán al servidor tras guardarlos de forma local en el IDE.



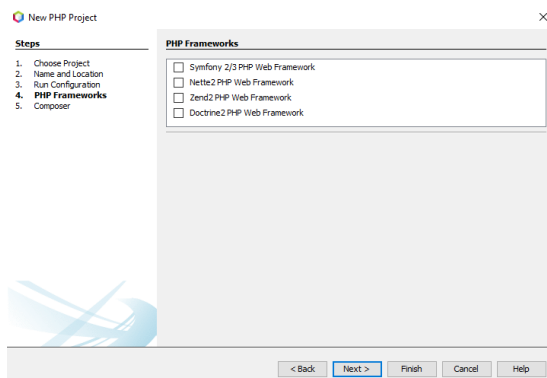
Host Name: IP de la maquina servidor Ubuntu (200+anfitrion).

Port: Puerto 22 por defecto para SFTP.

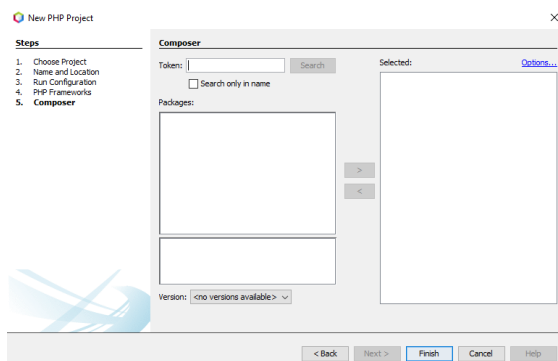
User Name: Usuario que se conecta (explicada su creación mas adelante).

Password: Password del usuario al que se conecta.

Initial Directory: Directorio inicial al que se conecta.



No seleccionamos ningún framework...



Clicamos en finalizar

Depuración



El primer icono ejecuta la depuración

Josué Martínez Fernández

El segundo icono no pertenece al debug

El tercer icono para la ejecución y depuración

El cuarto icono continua con la depuración sin parar salvo que exista un breakpoint

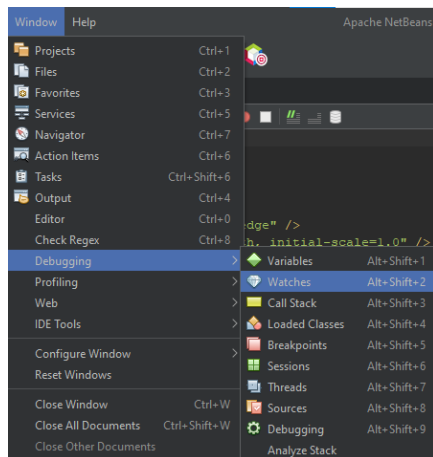
El quinto icono ejecuta una línea de código. Si la instrucción es una llamada a un método, ejecuta el método sin entrar dentro del código del método.

El sexto icono ejecuta una línea de código. Si la instrucción es una llamada a un método, salta al método y continúa la ejecución por la primera línea del método.

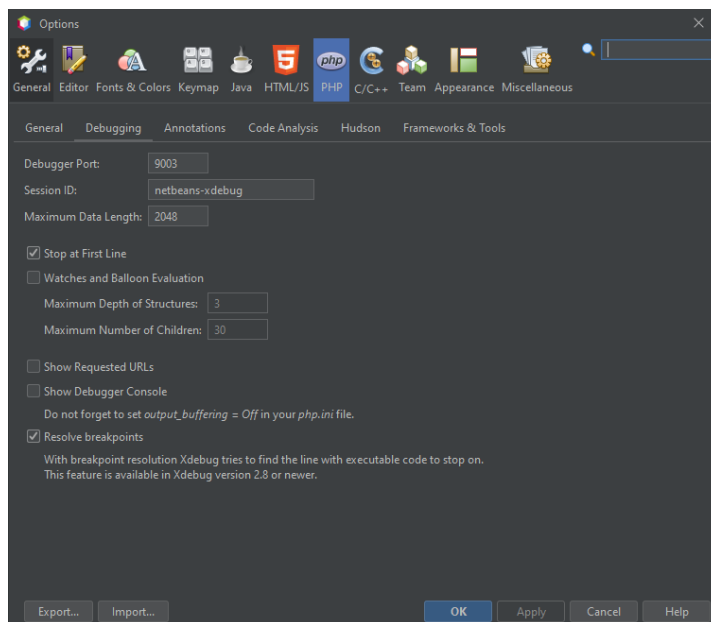
El séptimo icono ejecuta una línea de código. Si la línea de código actual se encuentra dentro de un método, se ejecutarán todas las instrucciones que queden del método y se vuelve a la instrucción desde la que se llamó al método.

El octavo icono ejecuta el programa hasta la instrucción donde se encuentra el cursor.

Es posible mostrar varias ventanas con el debug, por ejemplo la inspección de variables.



Posible solución a errores

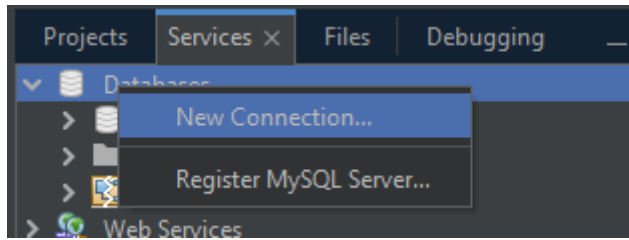


Josué Martínez Fernández

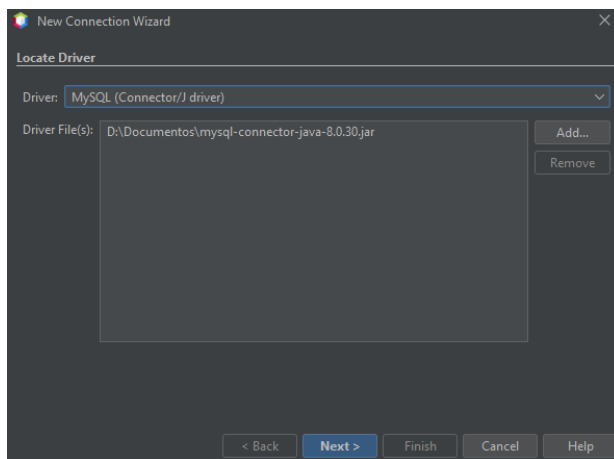
Comprobar que NetBeans usa el puerto 9003 para debuggear.

Es aconsejable el abrir el puerto de entrada 9003 en el cortafuegos.

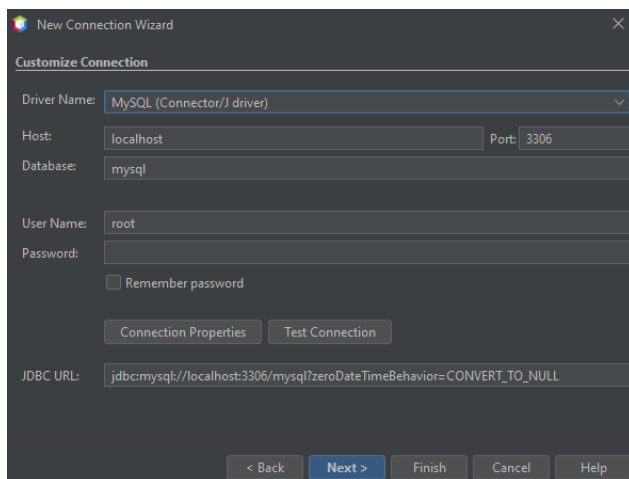
Conexión BBDD



Para realizar una conexión a la BBDD se crea una a través de la pestaña de servicios, botón derecho en Databases, click en nueva conexión.



Seleccionamos el archivo que utilizaremos de conector...



Host: Equipo donde se encuentra la BBDD (192.168.3.2XX).

Port: Puerto 3306 por defecto.

Database: Nombre de la BBDD a la que te desees conectar. (En este caso mysql y a través de ahí creamos la nueva BBDD y su usuario en la ejecución de un script).

Josué Martínez Fernández

UserName: Usuario de MySQL con permisos en la BBDD (adminsqli).

Password: Password del usuario.



Se pueden ejecutar los Scripts al tenerlos abiertos presionando el icono que aparece sobre el código.

También es posible a través de esto ejecutar queries para poder operar con las BBDD

Otra forma de ver las BBDD es a través de la pestaña de servicios y seleccionando la conexión que previamente hemos creado.

GIT & GitHub

En primer lugar es necesario tener creada una cuenta en GitHub para poder utilizar este servicio.


Desde este repositorio remoto podemos acceder tanto al código de nuestras aplicaciones como a datos sobre los commits, merges, tags, cambios realizados...

Para crear un repositorio en GitHub...



Clicamos en new


Owner * Repository name *


 JosueMarFer

 /

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

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.


☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

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

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

 You are creating a public repository in your personal account.

Seleccionamos el propietario y el nombre del repositorio, añadimos una descripción (opcional).

Elegimos si el repositorio queremos que sea publico o privado.

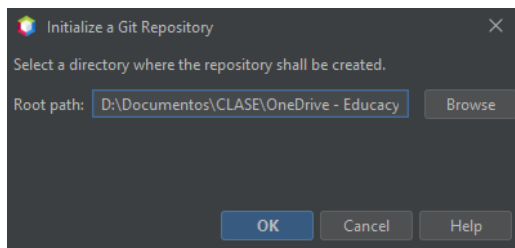
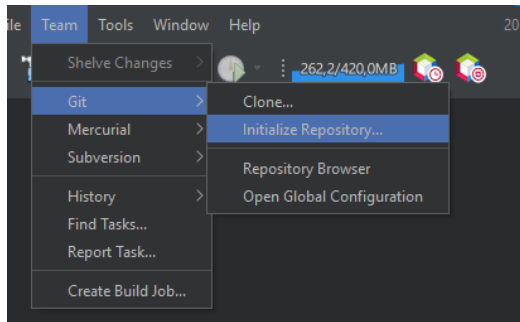
Podemos añadir un archivo readme que contendrá tanto el nombre del repositorio como su descripción en un lenguaje de marcado.

Josué Martínez Fernández

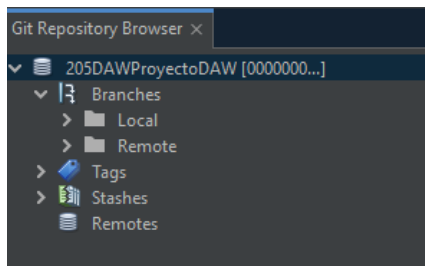
Podemos añadir un gitignore, utilizado para los archivos que no queremos que se suban.

Elegimos el tipo de licencia.

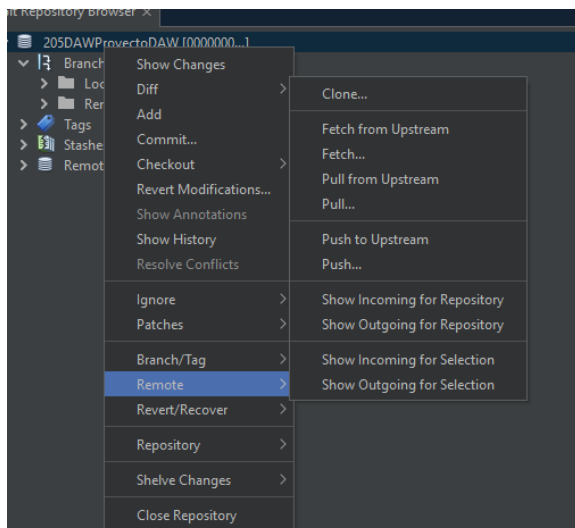
A través de netbeans para inicializar un repositorio seleccionamos un proyecto y desde la siguiente ventana lo inicializamos.



Ubicación de los archivos que queremos en el repositorio (local)



A través de esta ventana podemos comprobar los commits tanto en local como en remoto, las ramas, hacer merges de una rama a otra...



Josué Martínez Fernández

Algunas de las herramientas mas utilizadas son:

Clone: Clona un repositorio remoto.

Pull: Descarga los archivos de un repositorio remoto.

Push: Sube el commit seleccionado al repositorio remoto.

Commit: Crea un nuevo commit.

Add: Añade archivos a el espacio intermedio entre los archivos totales y los que se desea hacer commit.

Merge: Mezcla una rama con otra (Creando o no un nuevo commit).

USED-UBUNTU SERVER Entorno de Desarrollo

Configuración inicial

Nombre

Muestra hostname:

Hostname

Cambiar el hostname:

Editar Fichero:

Sudoedit /etc/hostname

JMF-USED

Editar Fichero:

Sudoedit /etc/hosts

127.0.0.1 localhost

127.0.1.1 JMF-USED

Configuración de red

Muestra IP:

ip a

Cambiar IP:

Editar fichero:

Sudoedit /etc/netplan/00-installer-config.yaml

dhcp: false

addresses: [IP del EQUIPO]

Josué Martínez Fernández

Routes:

To: default

Via: IP DEL GATEWAY

nameservers:

addresses: [IP DNS,IP DNS2]

Aplicar la configuración:

`Sudo netplan apply`

Servicios

Apache

Para instalar el servicio:

`Sudo apt install apache2`

Para ver si esta instalado

`Sudo service apache2 status`

Para la mayoría de servicios del curso es necesario abrir puertos en el servidor, así pues además de el de apache vamos a abrir todos de una vez.

Para ver el estado del cortafuegos:

`Sudo ufw status`

Si esta desactivado:

`Sudo ufw enable`

Para añadir excepciones:

`Sudo ufw allow numeropuerto`

Abrimos:

22 ssh

80 http

443 https

9003 xdebug

8080 tomcat

3306 mysql

53 dns

Creación del usuario apache operadorweb:

Crea un usuario con home en /var/www/html

Pertenece al grupo www-data

El nombre del usuario es operadorweb

```
sudo useradd -d /var/www/html -g www-data operadorweb
```

Define a operadorweb como dueño de /var/www/html

```
sudo chown -R operadorweb:www-data /var/www/html
```

Define permisos para /var/www/html

2 bit de permanencia (todos los archivos de este directorio heredan el grupo)

7 lectura escritura y ejecución para el propietario

7 Lectura escritura y ejecución para el grupo

5 Lectura y ejecución para otros

```
sudo chmod -R 2775 /var/www/html
```

Define una contraseña para operadorweb (necesaria para la autenticación)

```
Sudo passwd operadorweb
```

PHP

Instalar PHP:

```
sudo apt install php
```

Comprobar si el servicio esta instalado:

```
Php -v
```

Modificar los siguientes parámetros del fichero php.ini:

```
Sudo nano /etc/php/8.1/apache2/php.ini
```

	DESARROLLO	PRODUCCIÓN
	file-uploads = On allow-url_fopen = On memory_limit = 256M upload_max_filesize = 100M max_execution_time = 360 date.timezone = Europe/Madrid	file-uploads = On allow-url_fopen = On memory_limit = 256M upload_max_filesize = 100M max_execution_time = 360 date.timezone = Europe/Madrid
ERRORES	display_errors = On display_startup_errors = On	display_errors = Off display_startup_errors = Off

Tras la modificación reiniciar el servidor apache:

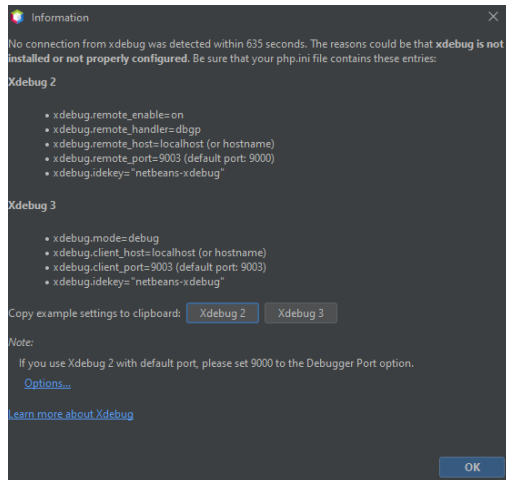
```
Sudo service apache2 restart
```

Xdebug

Instalar Xdebug:

```
Sudo apt install php-xdebug
```

Al intentar debugear un proyecto con netbeans salta el siguiente error:



Para solucionarlo se deben modificar los siguientes parámetros en el fichero php.ini

output_buffering = Off

También se deben modificar el siguiente archivo con las siguientes líneas para solucionarlo:

Sudo nano /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"

MySQL

Instalar el servicio:

Sudo apt-get install mysql-server

Ver si esta instalado:

dpkg -l mysql-server

Entrar en mysql con root y sin password:

sudo mysql -u root

Josué Martínez Fernández

ver si el servicio esta funcionando:

```
sudo service mysql status
```

Ver los puertos y la aplicación a la que están asignados:

```
ss -punta
```

```
puerto udp number tcp all
```

Procedemos a modificar el fichero mysqld.conf:

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

En este fichero comentar las siguientes líneas para permitir la conexión desde cualquier equipo.

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

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

Reiniciamos el servicio:

```
Sudo service mysql restart
```

Instalar el conector de php con la BBDD:

```
sudo apt install php8.1-mysql
```

Entramos como root y dentro de la consola de mysql creamos el usuario:

```
CREATE USER 'adminsql'@'%' IDENTIFIED BY 'paso' ;
```

Crea el usuario

```
GRANT ALL PRIVILEGES ON *.* TO 'adminsql'@'%' WITH GRANT OPTION;
```

Añade todos los privilegios en todas las BBDD, incluso el de crear mas usuarios.

```
FLUSH PRIVILEGES;
```

Recarga los privilegios.

Sitios basados en puertos y enjaular usuarios:

Configurando los sitios virtuales:

Creamos el directorio public_html en el home del usuario:

```
Sudo mkdir /var/www/puerto80/public_html
```

Copiamos y renombramos el fichero 000-default

```
Sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/puerto80.conf
```

Configuramos el fichero con las siguientes líneas:

```
Sudo nano /etc/apache2/sites-available/puerto80.conf
```

```
*:80
```

```
/var/www/puerto80/public_html
```

Habilitamos el puerto en el fichero /etc/apache2/ports.conf

```
Sudo nano /etc/apache2/ports.conf
```

```
Listen 80
```

Creamos una regla en el cortafuegos para el puerto por el que queremos que escuche la petición:

```
Sudo ufw allow 80
```

Habilitamos el sitio:

```
Sudo a2ensite puerto80
```

Tras esto reiniciamos el servicio de apache y nuestro sitio debería ser accesible a través de la IP:puerto en la barra de navegación.

Enjaular usuarios:

Josué Martínez Fernández

El primer paso para enjaular usuarios es crearlos:

```
Sudo useradd -m -d /var/www/puerto81 -g www-data dawPuerto80
```

-m crea el directorio si no existe.

Para enjaular los usuarios en primer lugar hay que cambiar el propietario del home del usuario que vamos a enjaular por root:root:

```
Sudo chown root:root /var/www/puerto80
```

Cambiamos los permisos para que nadie pueda escribir en ese directorio:

```
Sudo chmod 555 /var/www/puerto80
```

Cambiamos los permisos y propietario del directorio public_html por el usuario a enjaular

```
Sudo chown -R dawPuerto80:www-data /var/www/puerto80/public_html
```

```
Sudo chmod -R 2775 /var/www/puerto80/public_html
```

Creamos un grupo para los usuarios enjaulados:

```
Sudo groupadd ftpusers
```

Añadimos el usuario al grupo:

```
Sudo usermod -g ftpusers dawPuerto80
```

Comprobamos que se ha añadido al nuevo grupo:

```
Sudo cat /etc/group
```

```
Sudo cat /etc/passwd
```

Editamos el fichero sshd_config (previo backup por si hay algún error):

```
Sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bk
```

```
Sudo nano /etc/ssh/sshd_config
```

```
Match Group ftpusers
```

```
ChrootDirectory %h
```

```
ForceCommand internal-sftp -u 2
```

Josué Martínez Fernández

AllowTcpForwarding yes

PermitTunnel no

X11Forwarding no

gestión de errores Apache

Para gestionar los errores de apache (recurso inaccesible, recurso no encontrado...) primero debemos modificar una linea en apache2.conf:

```
sudo nano /etc/apache2/apache2.conf
```

AllowOverride All

Tras esto editar el fichero .htaccess Ubicado en la raíz del directorio de la web (/var/www/html):

(De no existir crearlo)

ErrorDocument 404 /pagerror/404.html

Redirect 302 /google "https://www.google.es

ErrorDocument 403 /pagerror/403.html

Modificar las siguientes directivas para que funcionen los alias y no puedas ver recursos sin permisos (apache2.conf, pero también se puede en el .htaccess):

A los directorios cambiar Options +FollowSymLinks -indexed

Alias apache

Para Crear un alias en primer lugar debemos modificar el fichero alis.conf

```
Sudo nano /etc/apache2/mods-available/alias.conf
```

Alias /images/ "/home/miadmin/images"

```
<Directory "/home/miadmin/images">
```

```
Options FollowSymLinks
```

```
AllowOverride None
```

```
Require all granted
```

```
</Directory>
```

Añade el alias images para acceder a los recursos de /home/miadmin/images.

Cambiar el orden de prioridad de los index apache

Modificar el archivo dir.conf ubicado en mods-available:

Josué Martínez Fernández

[DirectoryIndex](#) [index.php](#) [index.html](#) [index.cgi](#) [index.pl](#) [index.xhtml](#) [index.htm](#)

DNS

Instalar el servicio Bind9:

Cambiar dns y sufijo dns:

Editar el fichero named.conf.local

```
Sudo nano /etc/bind/named.conf.local
```

```
//
```

```
// Do any local configuration here
```

```
//
```

```
// Consider adding the 1918 zones here, if they are not used in your
```

```
// organization
```

```
//include "/etc/bind/zones.rfc1918";
```

```
zone "josue.local" {
```

```
    type master;
```

```
    file "/etc/bind/db.josue.local";
```

```
};
```

```
zone "3.168.192.in-addr.arpa" {
```

```
    type master;
```

```
    file "/etc/bind/db.3.168.192.in-addr.arpa";
```

```
};
```

Checkear los ficheros de configuración:

```
Sudo named-checkconf
```

Posteriormente creamos la BBDD que almacena los registros de los recursos:

```
Sudo nano /etc/bind.db.josue.local
```

```
; BIND data for josue.local

;

$TTL 604800

@ IN SOA JMF-USED.josue.local. root.localhost. (
        1      ; Serial
        604800 ; Refresh
        86400  ; Retry
        2419200 ; Expire
        3600 ) ; Negative Cache TTL
;

@ IN NS JMF-USED.josue.local.

; Registros Host

@ IN A 192.168.3.212
JMF-USED IN A 192.168.3.212

; Registros Alias

daw201 IN CNAME JMF-USED.josue.local.
daw202 IN CNAME JMF-USED.josue.local.
daw203 IN CNAME JMF-USED.josue.local.
daw204 IN CNAME JMF-USED.josue.local.

www.daw201 IN CNAME JMF-USED.josue.local.
www.daw202 IN CNAME JMF-USED.josue.local.
www.daw203 IN CNAME JMF-USED.josue.local.
www.daw204 IN CNAME JMF-USED.josue.local.
```

Crear la zona inversa copiando la zona directa:

Josué Martínez Fernández

```
sudo cp /etc/bind/db.josue.local /etc/bind/3.168.192.in-addr.arpa
```

Editamos el fichero, eliminamos los registros CNAME y cambiamos los A por PTR

```
;
; BIND data for 2.168.192.in-addr.arpa
;
$TTL 604800
@ IN SOA JMF-USED.josue.local. root.localhost. (
        1      ; Serial
        604800 ; Refresh
        86400  ; Retry
        2419200 ; Expire
        3600 ) ; Negative Cache TTL
;
@ IN NS JMF-USED.josue.local.

; Registros Punteros
212 IN PTR josue.local.
212 IN PTR JMF-USED.josue.local.
```

Check de zonas:

```
Sudo named-checkzone josue.local /etc/bind/db.josue.local
```

```
Sudo named-checkzone 3.168.192.in-addr.arpa /etc/bind/db.3.168.192.in-addr.arpa
```

Reiniciamos el servicio.

GIT

Ver la versión de git

```
Git --version
```

Información predeterminada de git:

Josué Martínez Fernández

```
Git config --global user.name "JosueMarFer"
```

```
Git config --global user.email "josuebenavente2018@gmail.com"
```

Por defecto viene instalado, vamos a subir algunos archivos al repositorio remoto:

Clonar el repositorio de github:

```
Git clone rutadegithub
```

Te mueves al repositorio y...

Añadir al workspace:

```
Git add nombreadarchivo.extension
```

 (si tiene un . añade todos)

Hacer commit de todo lo añadido al workspace:

```
Git commit -m 'mensaje o comentario'
```

Para ver como se llama el remoto:

```
Git remote
```

Subir al remoto (en este caso origin):

```
Git push origin master
```

Ver el estado del repositorio:

```
Git status
```

Iniciar un nuevo repositorio:

```
Git init
```

Si creas un repositorio vacío y quieres hacer push sin hacer un clone:

```
Git remote add origin direccioncopiadadelrepositorio
```

Moverte a otra rama:

```
Git checkout nombrerama
```

Ver en qué rama estás:

```
Git branch
```

Ver todas las ramas:

```
Git show-branch
```

Crear una nueva rama:

```
Git Branch nombrerama
```

