

DOCUMENTACION DE LAS MAQUINAS

Laura Fernández

DESARROLLO DE APLICACIONES WEB

DOCUMENTACION LFF-USED

○ Creación de la maquina limpia	4
▪ Sistema operativo	4
▪ Particiones	4
▪ Nombre del equipo	5
▪ Cuentas	5
▪ Configuración de red	6
▪ Servicio ssh.....	7
○ Instalación de software	8
▪ Descripción del programa Apache	8
▪ Utilidad en esta maquina	8
▪ Describir el proceso de instalación.....	8
▪ Ficheros de configuración	9
▪ ARCHIVOS DE CONFIGURACIÓN APACHE	10
▪ Descripción del programa PHP	10
▪ Utilidad en esta maquina	10
▪ Describir el proceso de instalación.....	10
▪ ARCHIVOS DE CONFIGURACIÓN PHP.INI	11
▪ Descripción del programa MySql.....	12
▪ Utilidad en esta maquina	12
▪ Describir el proceso de instalación.....	12
○ Mantenimiento.....	14
○ Log / Monitorización	14
○ Copia de seguridad	15

DOCUMENTACION LFF-WXED

○ Creación de la maquina limpia	19
▪ Sistema operativo	19
▪ Particiones	19
▪ Nombre del equipo	19
▪ Cuentas	20
▪ Configuración de red	20
▪ Servicio ssh.....	21
○ Instalación de software	22
▪ Descripción del programa Filezilla.....	22
▪ Utilidad en esta maquina	22
▪ Describir el proceso de instalación.....	22
▪ Conectarnos al servidor mediante Filezilla	22
▪ Descripción del programa Notepad++	23
▪ Utilidad en esta maquina	23
▪ Conectarnos al servidor mediante Notepad++	23
▪ Descripción del programa NetBeans	24
▪ Utilidad en esta maquina	24
▪ Conectarnos al servidor mediante Netbeans.....	24
▪ Descripción del programa WorkBench	27
▪ Utilidad en esta maquina	27
▪ Conectarnos al servidor mediante workBench	27
○ Crear estructura de directorios.....	29
○ Repositorio.....	31
▪ GITLAB.....	31
▪ GITHUB	27
▪ Trabajar varios en el mismo proyecto(Ramas).....	30
○ Subir ficheros a explotación	35
▪ Filezilla	35
▪ Mediante Repositorio	36
○ Subir al 1 and 1 las aplicaciones.....	37
○ Documentación PHPDoc	38
▪ Documentar una clase.....	41

DOCUMENTACION LFF-USEE

○ Creación de la maquina limpia	43
▪ Sistema operativo	43
▪ Particiones	43
▪ Nombre del equipo	43
▪ Cuentas	44
▪ Configuración de red	44
▪ Servicio ssh.....	44
○ Instalación de software	45
▪ Descripción del programa Apache	45
▪ Utilidad en esta maquina	45
▪ Comprobar la instalación del servicio	45
▪ Descripción del programaPHP	46
▪ Utilidad en esta maquina	46
▪ Comprobar la instalación del servicios.....	46
▪ ARCHIVOS DE CONFIGURACIÓN PHP.INI	47
▪ Descripción del programa MySql.....	48
▪ Utilidad en esta maquina	48
▪ Describir el proceso de instalación.....	48
○ Mantenimiento.....	50
○ Log / Monitorizacion	50
○ Copia de seguridad	51
○ Transferir archivos de desarrollo a explotación.	52

DOCUMENTACION DE LA MAQUINALFFUSED

Creación de la maquina limpia

Sistema operativo

Utilizaremos el Ubuntu Server 18.04 como servidor.

Le pondremos 500GB de memoria

2 GB de memoria Ram



Particiones

Realizaremos 3 particiones una para la swap con 4GB otra para datos con 100GB y por ultimo el resto para el sistema.

```

MOUNT POINT      SIZE      TYPE      DEVICE TYPE
[ /                395.997G  ext4      partition of local disk ► ]
[ /var            100.000G  ext4      partition of local disk ► ]
[ SWAP             4.000G    swap      partition of local disk ► ]

```

Nombre del equipo

El nombre del equipo que utilizaremos será LFFUSED

Para cambiar en Ubuntu el nombre deberemos meter este comando al final añadiendo el nombre que le queremos dar.

```
miadmin@lffused:~$ sudo hostnamectl set-hostname LFFUSED
miadmin@lffused:~$
```

Luego nos meteremos en este fichero para cambiar a esta variable a true /etc/cloud/cloud.cfg

```
# This will cause the set+update hostname module to not operate (if true)
preserve_hostname: true
# Example datasource config
```

Y por último comprobaremos que lo hemos cambiado correctamente con el siguiente comando:

```
miadmin@lffused:~$ hostnamectl
  Static hostname: LFFUSED
        Icon name: computer-vm
        Chassis: vm
        Machine ID: 0c2c60f6887647b398f2091c9439f80e
        Boot ID: f22c693329394628bf2042e9ec6152ec
        Virtualization: oracle
        Operating System: Ubuntu 18.04.1 LTS
        Kernel: Linux 4.15.0-29-generic
        Architecture: x86-64
```

Cuentas

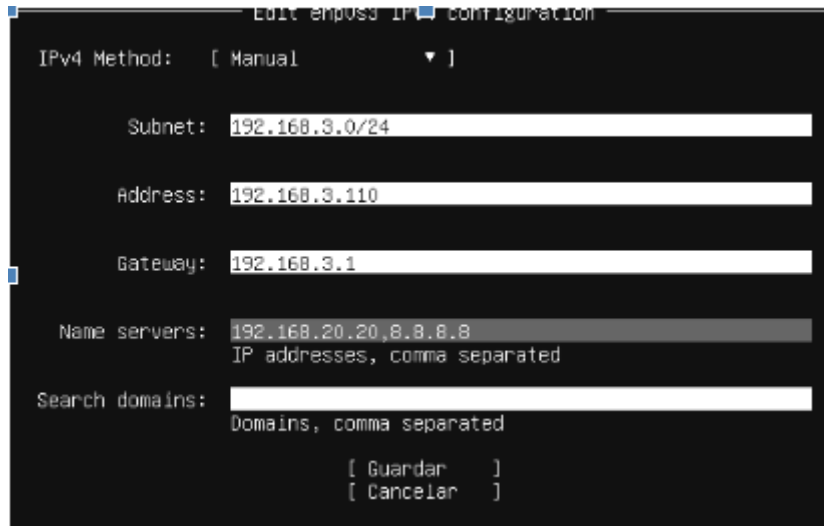
Crearemos una cuenta que es miadmin para administrar el servidor que es con la cuenta que crearemos la máquina.

Otra cuenta que será operadorweb para organizar los servicios Web

```
miadmin@LFFUSED:~$ sudo adduser --ingroup www-data --home /var/www/html --no-create-home operadorweb
```

Configuracion de red

Configuraremos la red en la instalacion pero tambien veremos donde se encuentran los ficheros de configuracion dentro de Ubuntu.



Para ver la configuración tenemos que abrir el fichero de configuración de red que se encuentra en `etc/netplan/50-cloud-init.yaml`

```
# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      addresses:
        - 192.168.3.110/24
      dhcp4: false
      gateway4: 192.168.3.1
      nameservers:
        addresses:
          - 192.168.20.20
          - 8.8.8.8
        search: []

version: 2
```

Servicio ssh

El servicio ssh viene instalado en este servidor, mostraremos los ficheros y como saber si está activo para poder conectarnos desde fuera.

Vemos que está activo, nos podremos conectar desde el Putty desde el cliente.

```
miadmin@LFFUSED:/$ sudo service ssh status
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2018-09-26 19:36:44 UTC; 9min ago
  Process: 1124 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 1128 (sshd)
  Tasks: 1 (limit: 1113)
  CGroup: /system.slice/ssh.service
          └─1128 /usr/sbin/sshd -D
```


Instalación de software

Descripción del programa Apache

Apache es el Servidor Web más utilizado, líder con el mayor número de instalaciones a nivel mundial muy por delante de otras soluciones como el IIS (Internet Information Server) de Microsoft. Apache es un proyecto de código abierto y uso gratuito, multiplataforma (hay versiones para todos los sistemas operativos más importantes), muy robusto y que destaca por su seguridad y rendimiento.

Utilidad en esta maquina

Su utilidad en esta máquina será para subir nuestros contenidos a la web, le enviaremos una petición al navegador que nos lo devolverá con un índice en HTML que es donde tendremos las asignaturas y sus ejercicios.

Describir el proceso de instalación

Antes de instalar haremos un Update y un Upgrade para tener el servidor actualizado.

Luego instalaremos apache2 con el siguiente comando: `sudo apt-get install apache2`

```
miadmin@LFFUSED:/$ sudo apt-get install apache2_
```

Comprobaremos si el servicio esta activo con el siguiente comando:

```
miadmin@LFFUSED:/$ sudo service apache2 status
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Wed 2018-09-26 20:08:53 UTC; 2min 2s ago
   Main PID: 2342 (apache2)
     Tasks: 55 (limit: 1113)
    CGroup: /system.slice/apache2.service
            └─2342 /usr/sbin/apache2 -k start
               2345 /usr/sbin/apache2 -k start
               2346 /usr/sbin/apache2 -k start
```

Con este comando veremos las aplicaciones que tiene Apache2

```
miadmin@LFFUSED:/$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  OpenSSH
```

Ficheros de configuración

Estos son los directorios y los ficheros que tiene Apache 2:

El fichero Apache2 contiene la configuración de Apache con muchas opciones de configuración:

```
miadmin@LFFUSED:/etc/apache2$ ls
apache2.conf  conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars      mods-available  ports.conf    sites-enabled
miadmin@LFFUSED:/etc/apache2$
```

```
GNU nano 2.9.3          apache2.conf

    AllowOverride None
    Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

#<Directory /srv/>
#     Options Indexes FollowSymLinks
#     AllowOverride None
#     Require all granted
#</Directory>
```

El fichero ports.conf contiene los puertos por los que escucha este servicio:

```
GNU nano 2.9.3          ports.conf

# If you just change the port or add more ports here, you will
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80
```

Donde subiremos los index sera en `var/www/html`

```
miadmin@LFFUSED:/var/www$ ls
html
```

ARCHIVOS DE CONFIGURACIÓN APACHE

Entre los archivos para la configuración, están los siguientes:

1. **ports.conf**, almacena las directivas que determinan los puertos TCP por los que Apache escucha. Esta es la configuración por defecto. La sección en el archivo ports.conf se ejecuta si se incluye el módulo llamado ssl_module.
2. **apache2.conf**, archivo de configuración principal. Casi toda la configuración se puede hacer desde este archivo, aunque es recomendable usar archivos designados por separado para hacerlo más simple. Este archivo se divide en tres secciones principales: configuración para el proceso del servidor Apache global, configuración para el servidor predeterminado y configuración de los Hosts virtuales.
3. **sites-available**, contiene todos los archivos de hosts virtuales que definen diferentes sitios web. Estos establecerán qué contenido se sirve para qué solicitudes. Estas son configuraciones disponibles, no configuraciones activas.
4. **sites-enabled**, este directorio establece qué definiciones de host virtual se están utilizando realmente. Generalmente, este directorio consta de enlaces simbólicos a archivos definidos en el directorio "sitios disponibles".

Descripción del programa PHP

PHP, acrónimo recursivo en inglés de PHP Hypertext Preprocessor (preprocesador de hipertexto), es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el HTML resultante.

PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en muchos sistemas operativos y plataformas sin ningún costo.

Utilidad en esta maquina

PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa CGI, como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies.

Describir el proceso de instalación

Instalaremos PHP con el comando `sudo apt-get install php`

```
miadmin@LFFUSED:~$ sudo apt-get install php
Levando lista de paquetes... Hecho
```

Veremos la versión que nos ha instalado con el siguiente comando

```
miadmin@LFFUSED:/$ sudo php -v
PHP 7.2.10-0ubuntu0.18.04.1 (cli)
```

Para llegar al directorio `apache2` tenemos que seguir la ruta siguiente donde se encuentra `php.ini` que es el fichero de configuración del php

```
miadmin@LFFUSED:/etc/php/7.2$ ls
apache2  cli  mods-available
```

```
miadmin@LFFUSED:/etc/php/7.2/apache2$ ls
conf.d  php.ini
```

ARCHIVOS DE CONFIGURACIÓN PHP.INI

Parámetros de php.ini

1. **safe mode**, modo seguro, si está desactivado se habilitan todas las funciones de PHP, para un uso educativo es conveniente ser funcional y no activarlo (off). Sin embargo, para servicios de hosting se recomienda activarlo.
2. **display errors**, muestra los errores en la misma página, es útil para encontrar cualquier fallo con facilidad.
3. **max_execution_time**, tiempo máximo de ejecución de una acción PHP, si dejamos que se ejecute indefinidamente podría colapsar el sistema. □
post_max_size, tamaño máximo de datos que se pueden enviar al servidor mediante POST.
4. **upload_max_filesize**, tamaño máximo de archivo que se puede subir al servidor. Si se va a trabajar con archivos grandes es conveniente cambiarlo.
5. **max_file_uploads**, número máximo de archivos que pueden ser subidos al servidor en una petición sencilla.

```
; display_errors
;   Default Value: On
;   Development Value: On
;   Production Value: Off

; display_startup_errors
;   Default Value: Off
;   Development Value: On
;   Production Value: Off

; error_reporting
;   Default Value: E_ALL & ~E_NOTICE & ~E_STRICT & ~E_DEPRECATED
;   Development Value: E_ALL
;   Production Value: E_ALL & ~E_DEPRECATED & ~E_STRICT

;
; html_errors
;   Default Value: On
;   Development Value: On
;   Production value: On

; log_errors
;   Default Value: Off
;   Development Value: On
;   Production Value: On

; max_input_time
;   Default Value: -1 (Unlimited)
;   Development Value: 60 (60 seconds)
;   Production Value: 60 (60 seconds)
```

Descripción del programa MySql

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL AB fue adquirida por Sun Microsystems en 2008, y ésta a su vez fue comprada por Oracle Corporation en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL.

Utilidad en esta maquina

Para administrar bases de datos desde el cliente con un software adicional como Workbench

Describir el proceso de instalación

Instalaremos el servicio con el comando apt-get install mysql-server

```
miadmin@LFFUSED:/$ sudo apt-get install mysql-server_
```

Abrimos la consola de mysql, entrando con el comando mysql y la contraseña anteriormente establecida en la instalación.

Entramos como sudo, para administrar los usuarios y contraseñas pertinentes:

```
sudo mysql
```

Asignamos a un usuario una nueva contraseña

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'paso';
```

Creación de nuevos usuarios

Podemos crear usuarios con

```
CREATE USER 'operadorBBDD'@'localhost' IDENTIFIED BY 'contraseña';
```

Y darle privilegios sobre todas las tablas

```
GRANT ALL PRIVILEGES ON *.* TO 'operadorBBDD'@'localhost' WITH GRANT OPTION;
```

O sobre alguna en concreto

```
GRANT ALL PRIVILEGES ON 'nombreBD'.'nombreTabla' TO 'operadorBBDD'@'localhost' WITH GRANT OPTION;
```

Mantenimiento

SSH:

Podemos parar, iniciar o reiniciar el servicio con el comando `sudo service ssh stop/start/reboot/restart` respectivamente.

Apache:

En Apache, para reiniciar el servicio debemos ejecutar el comando `sudo service apache2 restart/reload`, también podemos pararlo o iniciarlo con `stop/start`, para cambiar la carpeta desde la que sirve Apache lo podemos modificar con `sudo nano /etc/apache2/sites-enabled/000-default.conf` y modificando en `DocumentRoot` la ruta predeterminada de Apache `/var/www/html` por la que queramos nosotros.

MySQL:

Podemos reiniciar, parar e iniciar el servicio MySQL con el comando `sudo service MySQL reload/restart/stop/start`, respectivamente.

PHP:

PHP no es un servicio, no podemos pararlo ni iniciarlo, ni reiniciarlo, y en principio no requiere de mantenimiento.

Log / Monitorizacion

SSH:

Podemos comprobar el estado del servicio SSH con el comando `sudo service ssh status`. Podemos ver los log en la siguiente dirección `/var/log/auth.log`

Apache:

Podemos ver el estado de Apache con el comando `service apache2 status`:
Podemos ver los puertos que están abiertos con el comando `netstat -ltn`. También podemos ver la lista de app instaladas con `sudo ufw app list`.
Podemos ver los log en la siguiente dirección `/var/log/apache2/access.log`

MySQL:

En MySQL podemos ver el estado del servicio con el comando `sudo service mysql status`. Podemos ver los log en la siguiente dirección `/var/log/mysql/mysql_error.log`

PHP:

Para comprobar que se ha instalado el PHP correctamente lo veremos con el comando `php -v`, para ver los módulos que tiene PHP se usa el comando `apt-cache search php`. Para ver los logs de php `/var/log/apache2/error.log`

Copia de seguridad

SSH:

Para guardar la configuración de SSH en caso de querer cargarla en otra máquina o por razones de seguridad deberemos copiar el archivo de configuración `/etc/ssh/sshd_config` en nuestra máquina anfitriona.

Apache:

Si queremos quedarnos con la configuración de Apache debemos guardar en el anfitrión los archivos ubicados en `/etc/apache2/sites-enabled/000-default.conf` y `/etc/apache2/ports.conf`.

MySQL:

Si queremos guardar la configuración de MySQL debemos guardar los archivos ubicados en `/etc/mysql/my.cnf`, `/etc/mysql/mysql.cnf` y `/etc/mysql/mysql.conf.d/mysqld.cnf`.

PHP:

Si queremos guardar la configuración de PHP debemos hacer una copia del archivo ubicado en el path `/etc/php/7.2/apache2/php.ini`.

DOCUMENTACION DE LA MAQUINALFFW10ED

Creación de la maquina limpia **Sistema operativo**

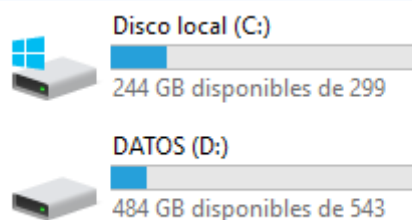
Hemos elegido para el entorno de desarrollo en entorno cliente un Windows 10.



Particiones

Las particiones que tendra este sistema operativo seran:

Una particion para el sistema y un particion para los datos con los siguientes volúmenes:



Nombre del equipo

El nombre del equipo :LFFW10ED

Cuentas

Tenemos dos usuarios uno que sera miadmin para administracion del sistema:

Escriba un nombre de usuario (por ejemplo, Juan):

 Escriba un nombre de equipo:

Y otra cuenta que sera operador web para administrar los servicios web que sera operadorweb



Configuración de red

La configuracion de red que le pondremos a este equipo sera la siguiente:

Propiedades: Protocolo de Internet versión 4 (TCP/IPv4) X

General

Puede hacer que la configuración IP se asigne automáticamente si la red es compatible con esta funcionalidad. De lo contrario, deberá consultar con el administrador de red cuál es la configuración IP apropiada.

☐ Obtener una dirección IP automáticamente

☒ Usar la siguiente dirección IP:

Dirección IP:

Máscara de subred:

Puerta de enlace predeterminada:

☐ Obtener la dirección del servidor DNS automáticamente

☒ Usar las siguientes direcciones de servidor DNS:

Servidor DNS preferido:

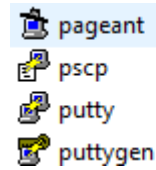
Servidor DNS alternativo:

☐ Validar configuración al salir

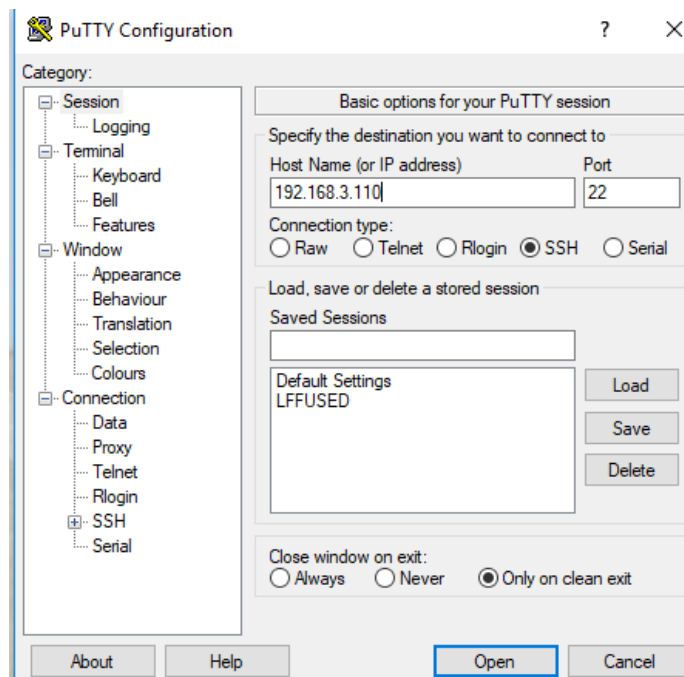
Servicio ssh

Para el servicio de la conexión remota utilizaremos el putty

El cual descargaremos y instalaremos en nuestro equipo



Se nos abrirá una ventana como esta al iniciar la aplicación y escribiremos la ip del servidor para conectarnos remotamente desde el cliente.



Despues nos saldra la consola para trabajar desde el cliente.

Introduciremos usuario y contraseña y ya estariamos dentro del servidor.

Instalación de software

Descripción del programa Filezilla

FileZilla, en sus versiones cliente o servidor, ambiente multiplataforma,3 de código abierto y software libre, licenciado bajo la Licencia Pública General de GNU. Soporta los protocolos FTP, SFTP y FTP sobre SSL/TLS (FTPS).

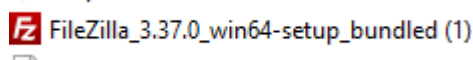
Utilidad en esta maquina

La utilidad del filezilla en esta maquina sera para transferir o descargar archivos desde el cliente al servidor.

Describir el proceso de instalación

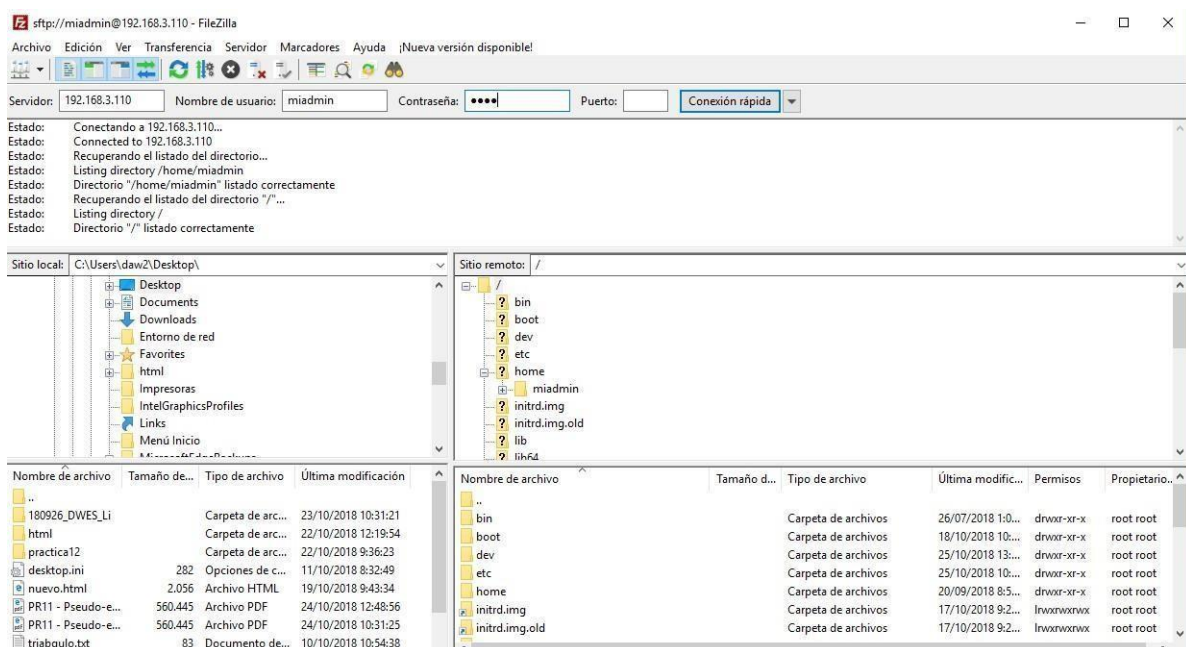
Descargaremos el Filezilla desde la pagina.

Se nos descargara un .exe el cual ejecutaremos



Conectarnos al servidor mediante Filezilla

Nos conectaremos al servidor mediante el Filezilla escribiendo la direccion ip, usuario y contraseña de nuestro servidor para ver los archivos que tenemos en el servidor:



Descripción del programa Notepad++

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

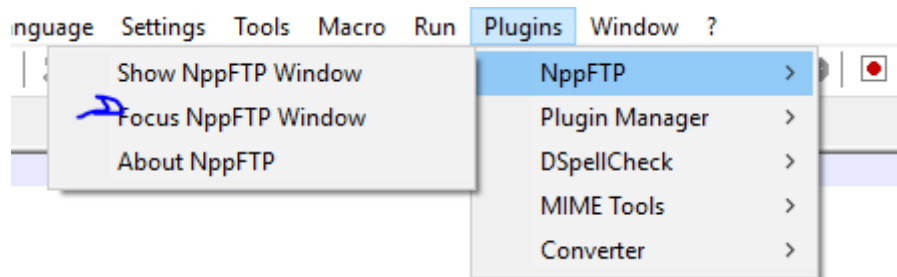
Utilidad en esta maquina

La utilidad en esta maquina sera para conectarnos al servidor y editar los ficheros, crear carpetas, crear ficheros... Y se cambiara directamente desde el cliente los archivos del servidor

Conectarnos al servidor mediante Notepad++

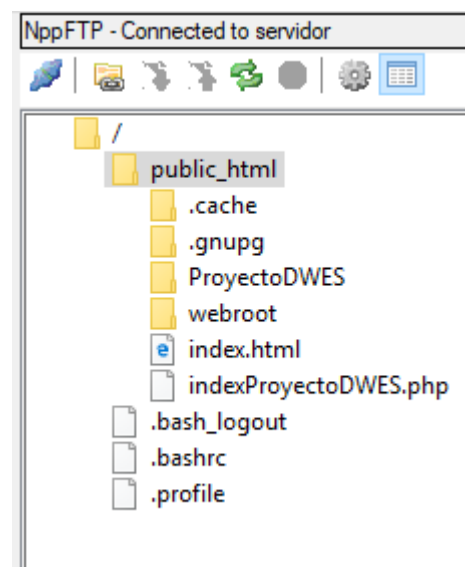
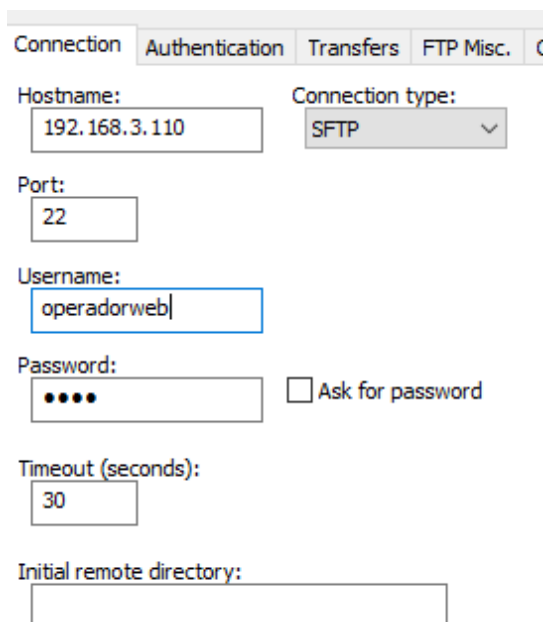
Nos conectaremos por Notepad++ al servidor:

Haremos la conexión mediante un plugin que hay que instalar:



Despues crearemos una nueva conexión con la ip del servidor, nombre de usuario y contraseña y entraremos al servidor.

Despues nos mostrara todos los ficheros a los que puede acceder ese usuario y ya podremos trabajar si tenemos los permisos necesarios.



Descripción del programa NetBeans

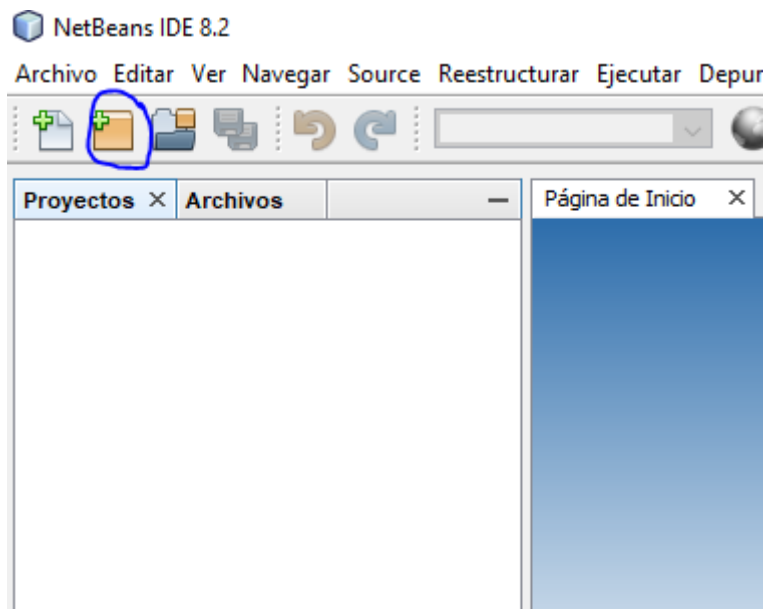
NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

Utilidad en esta maquina

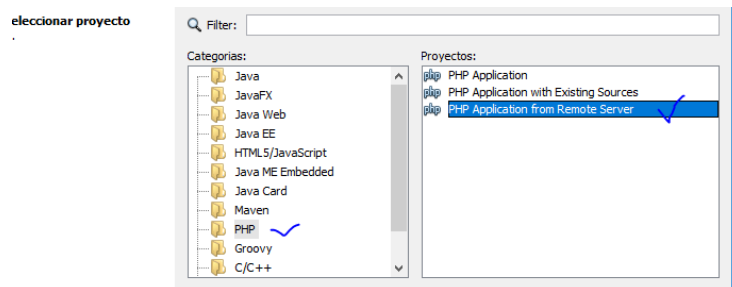
La utilidad del NetBeans en esta maquina sera la de conectarnos al servidor para editar los ficheros del servidor desde el cliente.

Conectarnos al servidor mediante Netbeans

1. Lo primero para conectar en NetBeans con el servidor es iniciar un nuevo proyecto



2. En este punto elegiremos un proyecto PHP y le daremos a remote server.



3. Le daremos un nombre al proyecto y seleccionaremos el PHP 7.0

Pasos

1. Seleccionar proyecto
2. **Name and Location**
3. Remote Connection
4. Confirmation

Name and Location

Project Name: ✓

Sources Folder: Browse...

PHP Version: ✓
PHP version is used only for hints

Default Encoding:

☐ Put NetBeans metadata into a separate directory

Metadata Folder: Browse...

4. En este paso es MUY IMPORTANTE poner bien la IP del servidor y a continuación describiremos los parámetros de configuración dándole a Manage

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

Project URL: ✓

Remote Connection: Manage...

Upload Directory:

5. En este punto pondremos el nombre de la conexión el nombre del usuario la contraseña y le indicaremos a que carpeta tiene que ir en este caso a public_html

Name:

Host Name: Port:

User Name:

Password:

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

Private Key File: Browse...

Running ssh-agent will be used.

Known Hosts File: Browse...

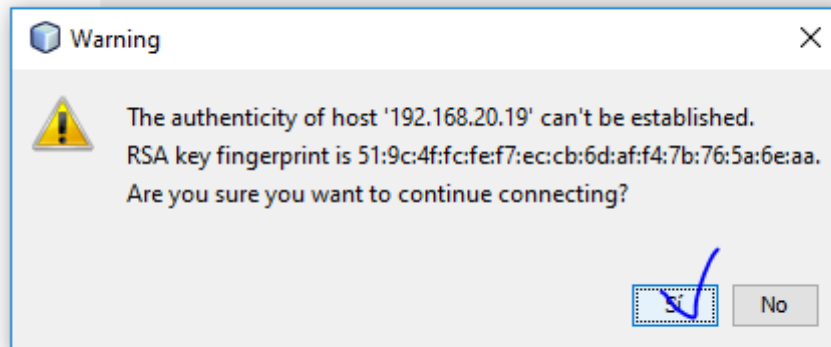
Initial Directory:

Timeout (s):

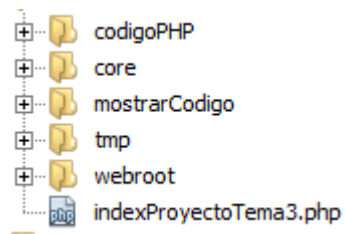
Keep-alive interval (s): Interval 0 means disabled.

Manage...

6. Nos saldrá este mensaje y le daremos que si autorizamos la conexión con el servidor



7. Una vez conectados al servidor creamos la estructura de directorios dentro de el proyecto de su tema en este caso la crearemos para el Tema3



Descripción del programa WorkBench

MySQLWorkbenches una herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, gestion y mantenimiento para el sistema de base de datos MySQL. Es el sucesor de DBDesigner 4 de fabFORCE.net, y reemplaza el anterior conjunto de software, MySQL GUIToolsBundle.

Utilidad en esta maquina

La utilidad en esta maquina sera la de manejar las bases de datos del servidor.

Conectarnos al servidor mediante workBench

Crearemos una nueva conexión con nuestro servidor:

Connection Name: LFFUSEDmysql

Connection Remote Management System Profile

Connection Method: Standard TCP/IP over SSH

Parameters SSL Advanced

SSH Hostname: 192.168.3.110:22 SSH se

SSH Username: miadmin Name c

SSH Password: Store in Vault ... Clear SSH us

SSH Key File: ... Path to

MySQL Hostname: 127.0.0.1 MySQL

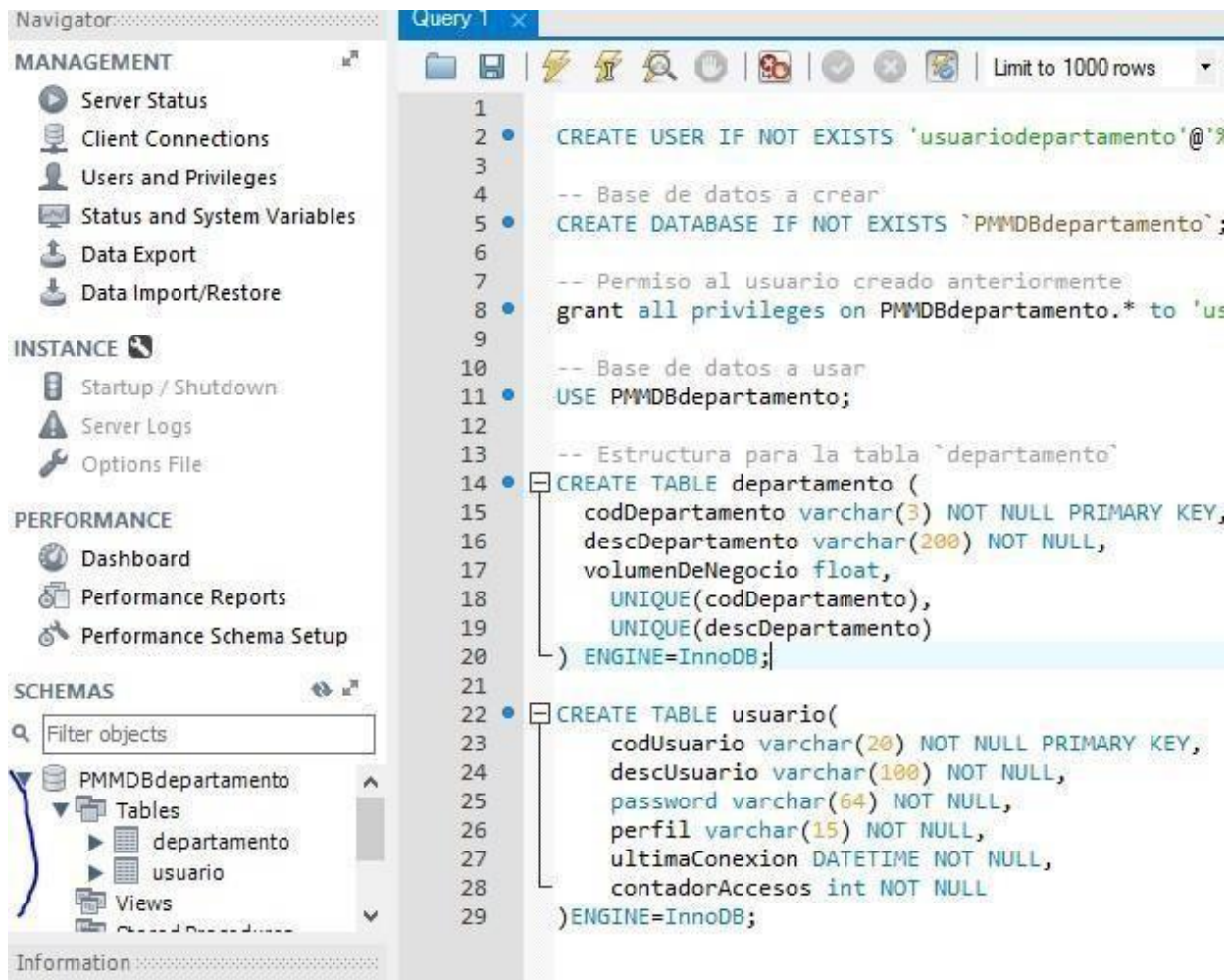
MySQL Server Port: 3306 TCP/IP

Username: root Name c

Password: Store in Vault ... Clear The My later if

Default Schema: The sch blank t

Después entraremos a ver las bases de datos del servidor en la cual meteremos una base de datos ya creada que se llamara departamento con dos tablas:



The screenshot shows the MySQL Workbench interface. On the left, the 'Navigator' pane displays the 'SCHEMAS' section with a search filter 'Filter objects'. Under 'PMMDBdepartamento', the 'Tables' folder is expanded, showing 'departamento' and 'usuario'. The main 'Query 1' window contains the following SQL script:

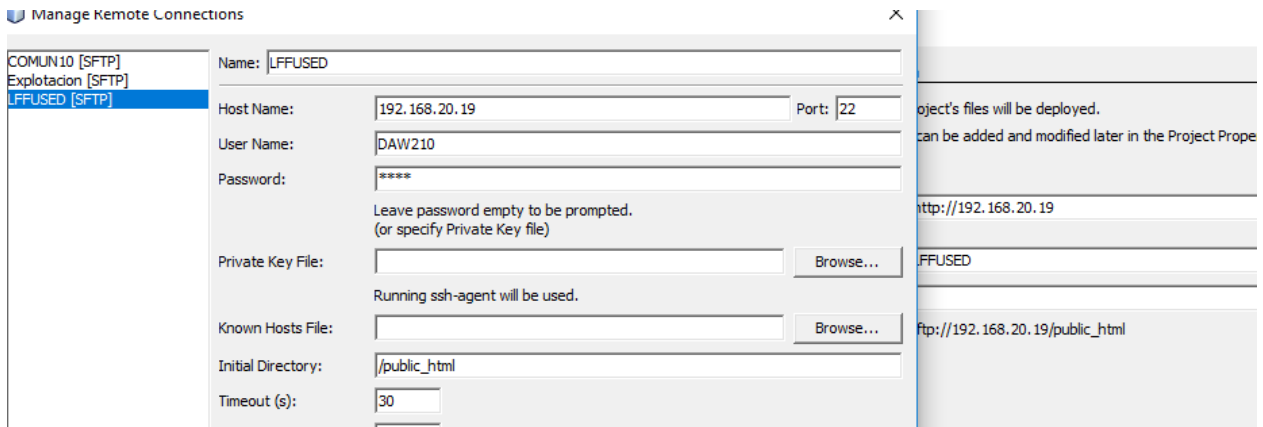
```

1
2 CREATE USER IF NOT EXISTS 'usuariodepartamento'@'%'
3
4 -- Base de datos a crear
5 CREATE DATABASE IF NOT EXISTS `PMMDBdepartamento`;
6
7 -- Permiso al usuario creado anteriormente
8 grant all privileges on PMMDBdepartamento.* to 'us
9
10 -- Base de datos a usar
11 USE PMMDBdepartamento;
12
13 -- Estructura para la tabla `departamento`
14 CREATE TABLE departamento (
15     codDepartamento varchar(3) NOT NULL PRIMARY KEY,
16     descDepartamento varchar(200) NOT NULL,
17     volumenDeNegocio float,
18     UNIQUE(codDepartamento),
19     UNIQUE(descDepartamento)
20 ) ENGINE=InnoDB;
21
22 CREATE TABLE usuario(
23     codUsuario varchar(20) NOT NULL PRIMARY KEY,
24     descUsuario varchar(100) NOT NULL,
25     password varchar(64) NOT NULL,
26     perfil varchar(15) NOT NULL,
27     ultimaConexion DATETIME NOT NULL,
28     contadorAccesos int NOT NULL
29 )ENGINE=InnoDB;
  
```

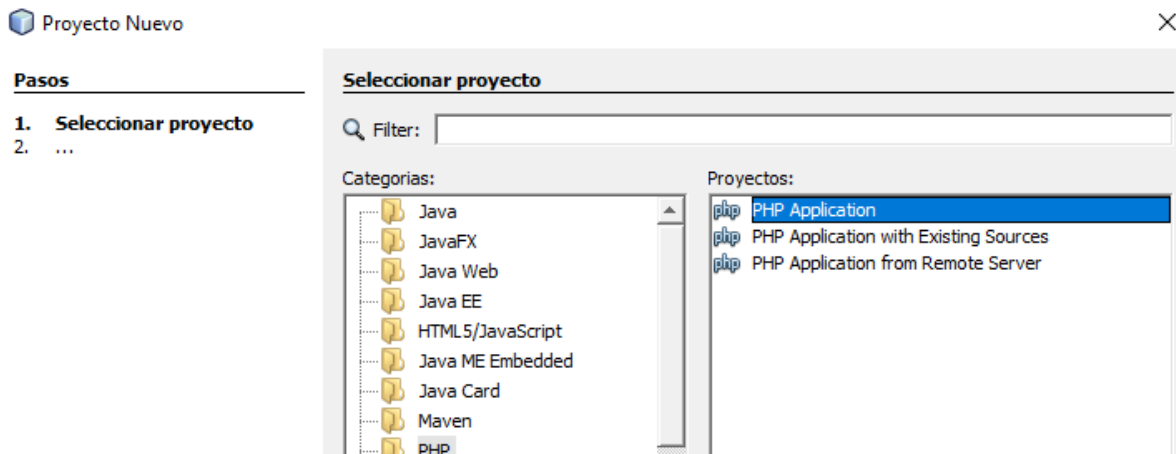
Crear estructura de directorios

- Crearemos un proyecto que englobe a todas las asignaturas y dentro de la carpeta DWES crearemos un proyecto por cada Tema.

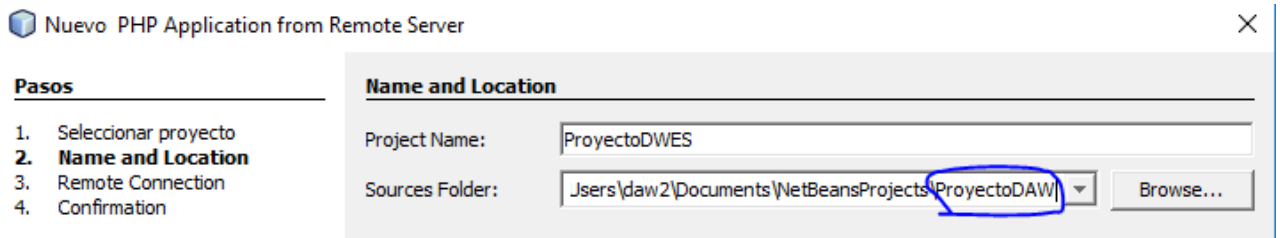
1. Crearemos el proyecto DAW y lo conectaremos con el servidor.



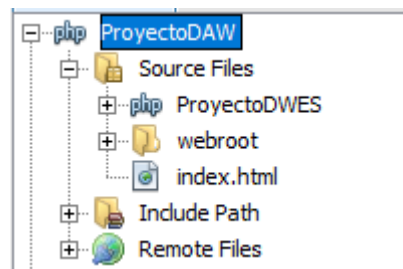
2. Crearemos otro proyecto para la asignatura DWES este lo crearemos en local ya que al estar dentro de el proyecto DAW principal que ya esta en el servidor automáticamente este se subirá al servidor aunque este creado en local.



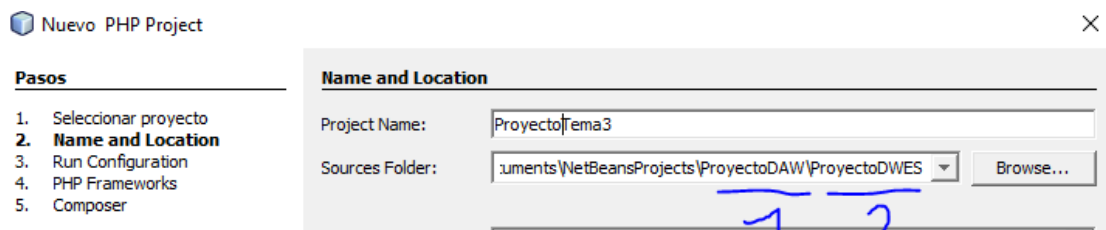
3. Seleccionaremos que el proyecto se cree dentro de ProyectoDAW como anteriormente hemos indicado



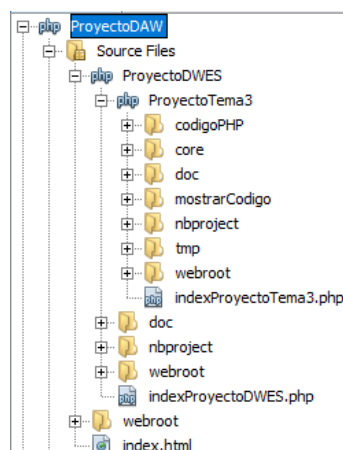
4. Ya tendremos el proyecto DAW con su proyectoDwes dentro.



5. Lo mismo para el Tema3 del ProyectoDWES que lo crearemos dentro de la carpeta ProyectoDWES.



6. Y así iremos construyendo la estructura de directorios completándola con los proyectos de los temas siguientes.



Repositorio

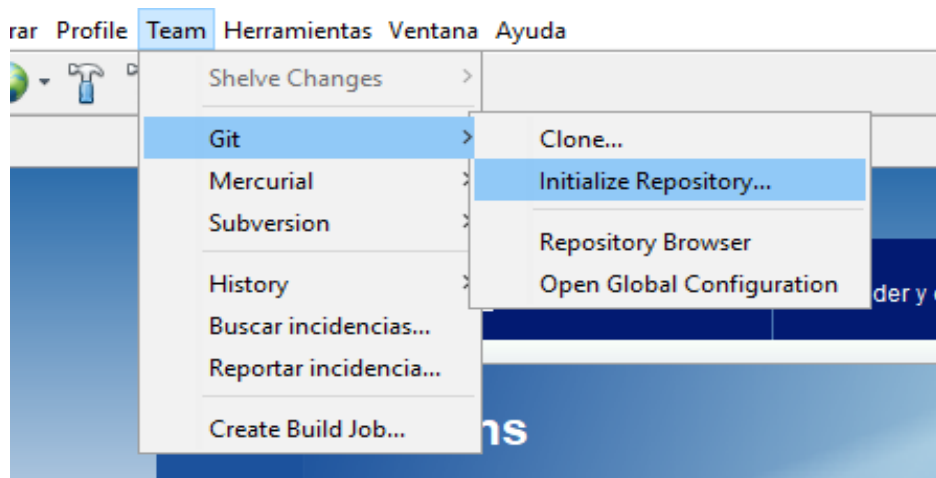
GITLAB

Un repositorio es un espacio centralizado donde se almacena, organiza, mantiene y difunde información digital, habitualmente archivos informáticos, que pueden contener trabajos científicos, conjuntos de datos o software.

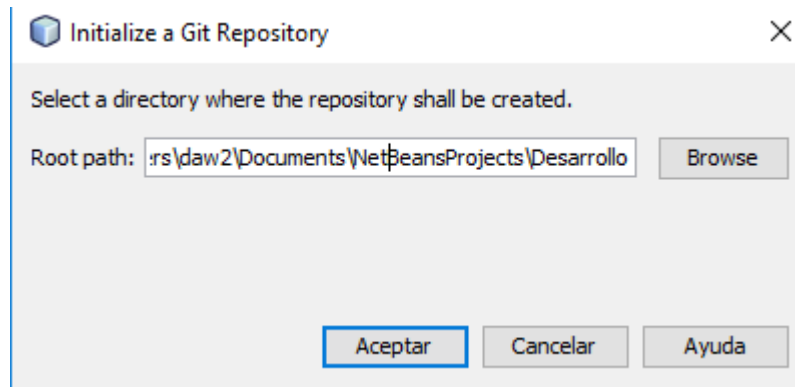
1. Lo primero que haremos será crear un proyecto en el repositorio en este caso GitLab el cual le pondremos el nombre de la asignatura.



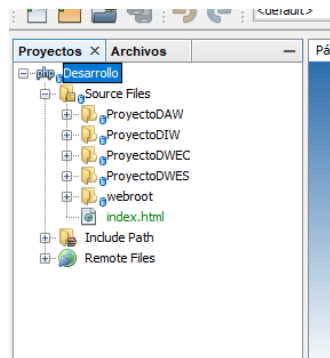
2. Luego en el NetBeans con el proyecto abierto que queremos subir al repositorio nos iremos a git y inicializar el repositorio.



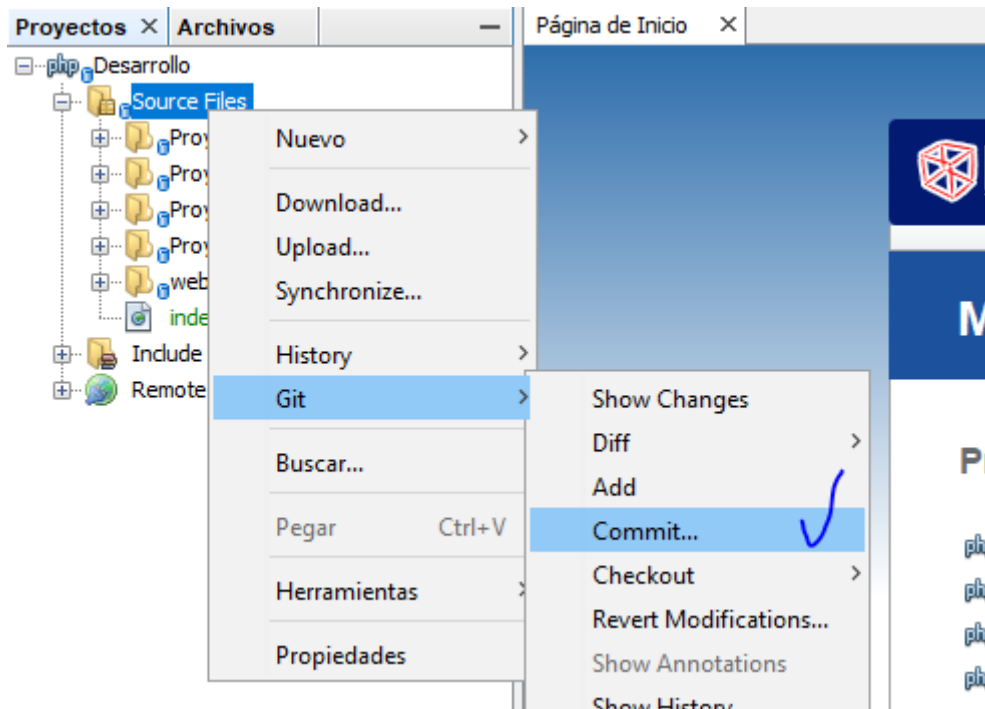
3. Le indicaremos la ruta del proyecto que tenemos en el repositorio para subir los archivos ahí.



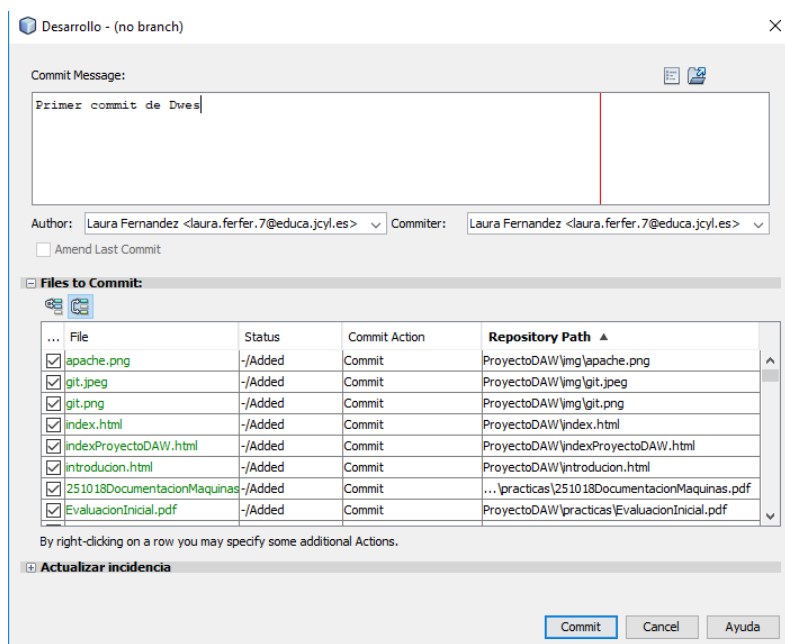
4. Ya tenemos el proyecto inicializado ahora solo nos hará falta subirlo.



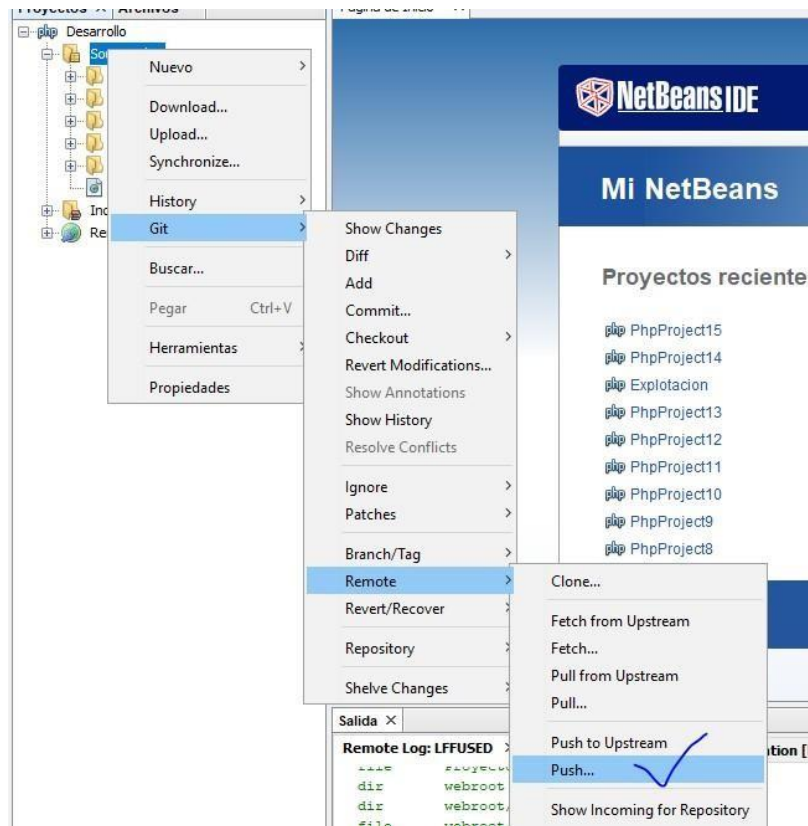
5. Seleccionaremos la pestaña de git y primero haremos un commit para confirmar lo que vamos a subir al repositorio y ponerle un nombre para saber que estamos subiendo.



6. Le ponemos el nombre al commit en este caso como es el primero le daré ese nombre y abajo nos sale los ficheros que vamos a subir al repositorio



7. Ahora iremos a la pestaña del git->remote->Push para subir los archivos que hemos seleccionado en el commit



8. Por ultimo pondremos nuestro usuario y contraseña del git lab y se subirán todos los ficheros

Pasos

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

Remote Repository

☐ Select Configured Git Repository Location:

☒ Specify Git Repository Location:

Remote Name: ☒ Persist Remote

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

User: (leave blank for anonymous access)



Password: ☐ Save Password

GITHUB

1. Creamos el repositorio en el github

Create a new repository

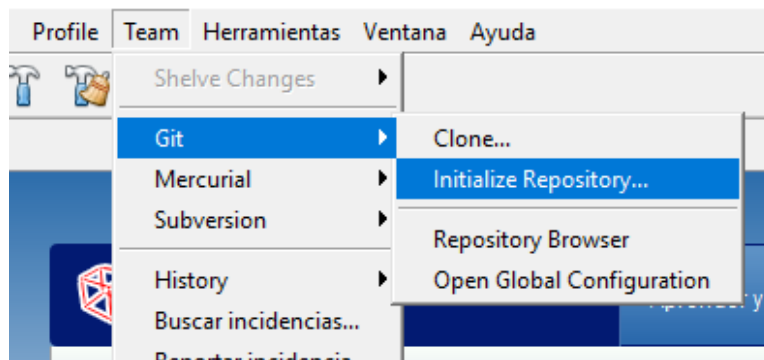
A repository contains all project files, including the revision history.

Owner:  laura1819 / Repository name *: ProyectoDWESLauraFernandez 

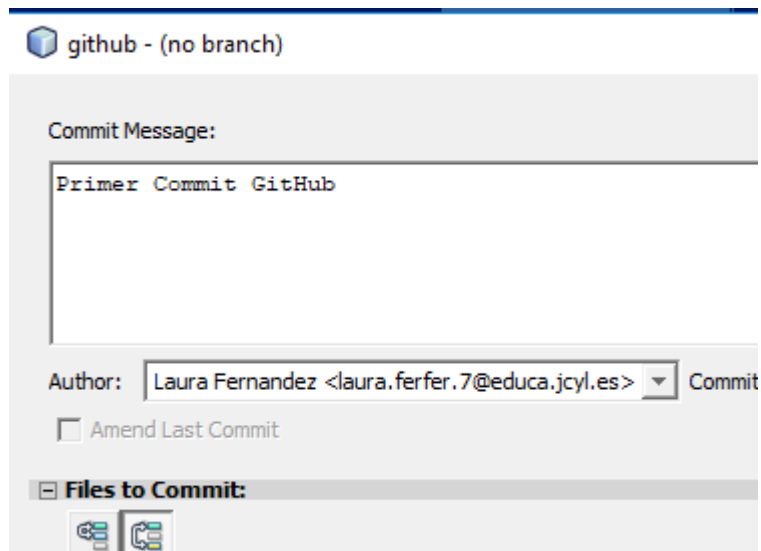
2. Nos creara el proyecto con su URL para ponerla en el NetBeans.



3. Nos vamos a el NetBeans e inicializamos el repositorio.

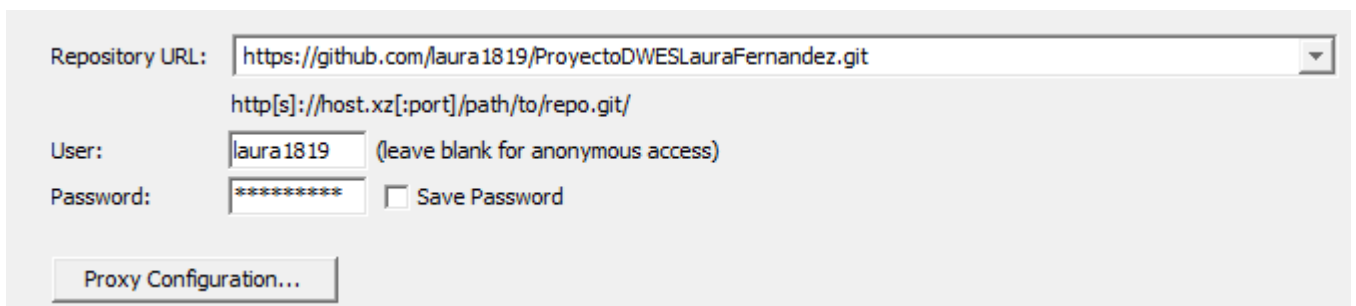


4. Creamos el primer Commit con los archivos que queremos subir al repositorio




A terminal window titled "github - (no branch)" showing the commit process. The "Commit Message:" section contains the text "Primer Commit GitHub". Below this, the "Author:" field is set to "Laura Fernandez <laura.ferfer.7@educa.jcyl.es>". To the right of the author field is a "Commit" button. Below the author field is a checkbox labeled "Amend Last Commit". At the bottom, there is a section titled "Files to Commit:" with two icons below it.

5. Sobre el proyecto vamos a git/remote/push para subir los archivos, ponemos la ruta del GitHub el usuario y la contraseña



A dialog box for configuring the git remote push. It has a "Repository URL:" field with the value "https://github.com/laura1819/ProyectoDWESLauraFernandez.git". Below this is a text field with the placeholder "http[s]://host.xz[:port]/path/to/repo.git/". The "User:" field contains "laura1819" with a note "(leave blank for anonymous access)". The "Password:" field contains "*****" and there is a checkbox labeled "Save Password". At the bottom left is a button labeled "Proxy Configuration...".

6. Por último nos vamos al GitHub para ver si se han subido los archivos correctamente

 **laura1819 / ProyectoDWESLauraFernandez**

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#)

No description, website, or topics provided.

[Manage topics](#)

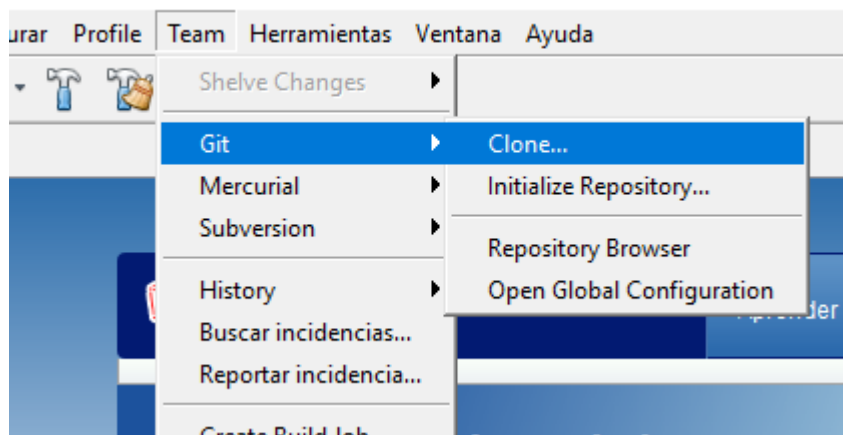
1 commit 1 branch 0 issues

Branch: master [New pull request](#) [Create new branch](#)

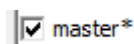
Laura Fernandez Primer Commit GitHub	
ProyectoTema3	Primer Commit GitHub
ProyectoTema4	Primer Commit GitHub
doc	Primer Commit GitHub
nbproject	Primer Commit GitHub
webroot/css	Primer Commit GitHub
indexProyectoDWES.php	Primer Commit GitHub

Trabajar varios en el mismo proyecto(Ramas)

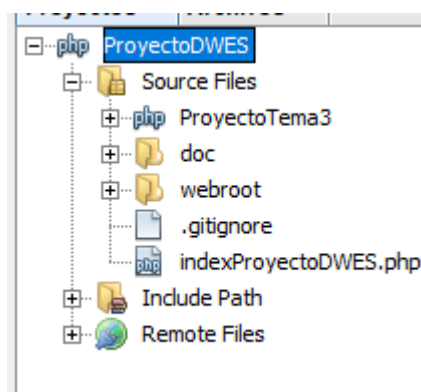
1. Para trabajar en el mismo proyecto varias personas para crear aplicaciones lo primero que haremos será clonar el proyecto que tenemos subido en el Git para descargar el código que tenemos en el repositorio.



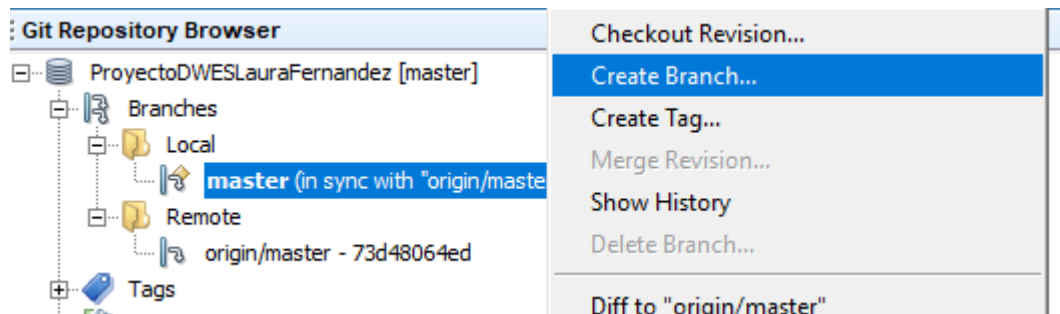
2. Al clonar el proyecto descargaremos lo que tenemos en el máster, que será el que contiene el ultimo commit y es el más actualizado.



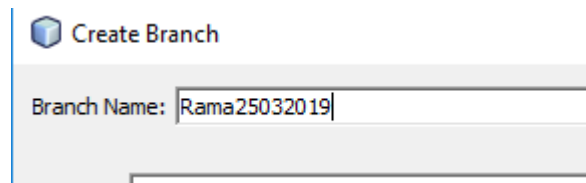
3. Se nos descargara el proyecto que hemos seleccionado.



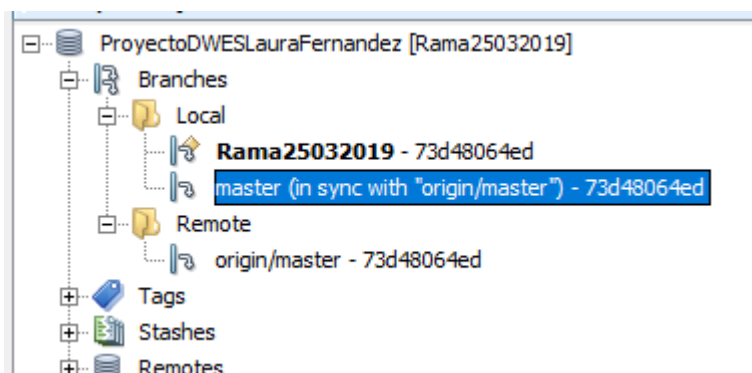
4. Sobre la rama master crearemos una rama para poder trabajar en ella los cambios, cada persona que trabaje en el proyecto deberá de hacer una propia para trabajar sobre ella.



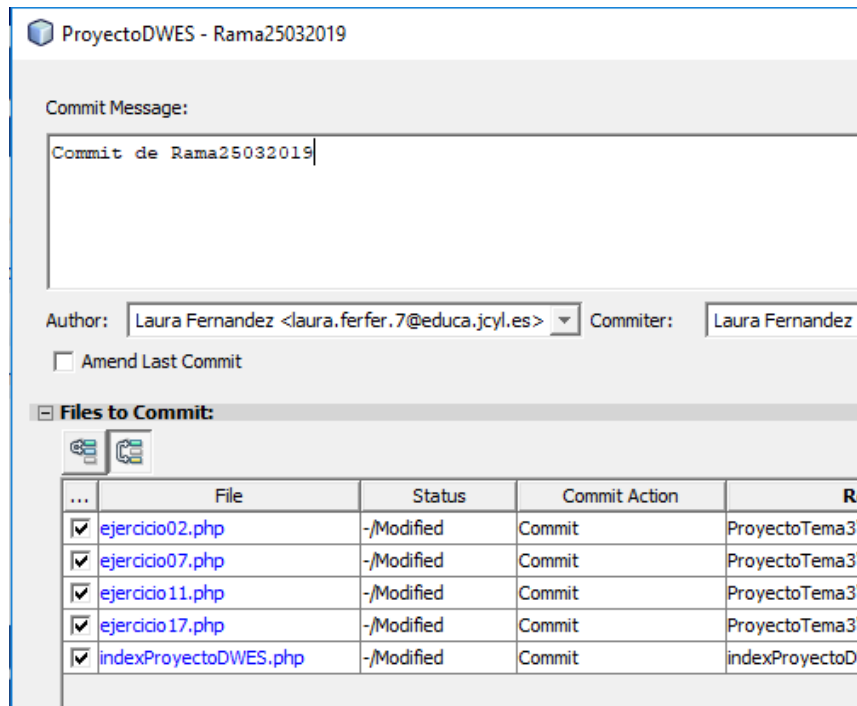
5. Le ponemos el nombre a la rama.



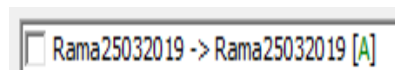
6. Una vez creada nos pondremos sobre ella para realizar los cambios que veamos necesarios en el proyecto.



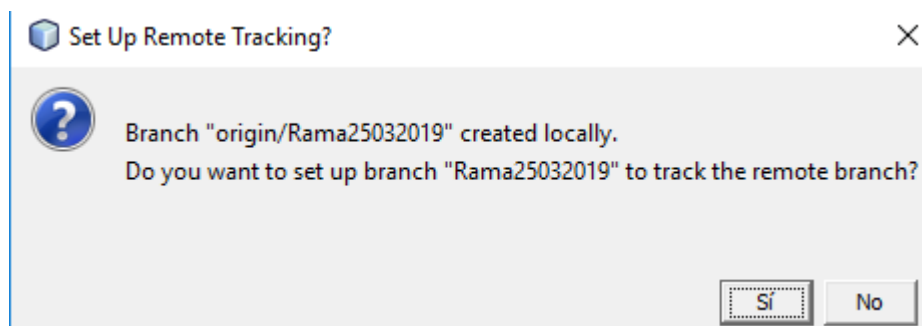
7. Cuando tengamos ya los cambios que queríamos realizar y lo vemos en el servidor es el momento de subir la Rama al Git lo primero es hacer un commit en el cual nos saldrán los archivos que se han modificado y vamos a subir al Git.



8. Y le haremos un Push a la rama para subirla seleccionando la rama que hemos creado.



9. La rama la hemos creado en el local por lo tanto le decimos que si nos la haga en remoto.



10. Vamos al repositorio y vemos que la rama se ha subido correctamente con los ficheros.

Rama25032019 (3 minutes ago) [Compare & pull request](#)

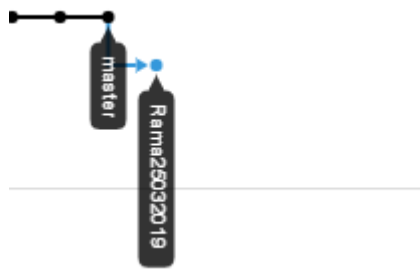
Branch: Rama25032019 [New pull request](#) [Create new file](#) [Upload files](#) [Find File](#) [Clone or download](#)

This branch is 1 commit ahead of master. [Pull request](#) [Compare](#)

Laura Fernandez Commit de Rama25032019 Latest commit c9dabc0 5 minutes ago

File	Commit de Rama25032019	Time
ProyectoTema3	Commit de Rama25032019	5 minutes ago
doc	Primer Commit DWES	7 days ago
nbproject	Mejoras ejercicio del 18 del Tema3 v3 fechas(al master)	4 days ago
webroot/css	Primer Commit DWES	7 days ago
.gitignore	commit tema 3 casa 20032019	5 days ago
indexProyectoDWES.php	Commit de Rama25032019	5 minutes ago

11. En el Network podemos ver la rama que hemos creado como sale del master con el commit que hemos hecho.



12. Por último, vamos a fusionar la rama que hemos creado con el master para tener los cambios actualizados sobre el máster, le daremos a create pull request y nos saldrá una ventana como la de a continuación para solicitar el merge al master y le daremos al botón verde

base: master ← compare: Rama25032019 ✓ Able to merge. These branches can be automatically merged.

Commit de Rama25032019

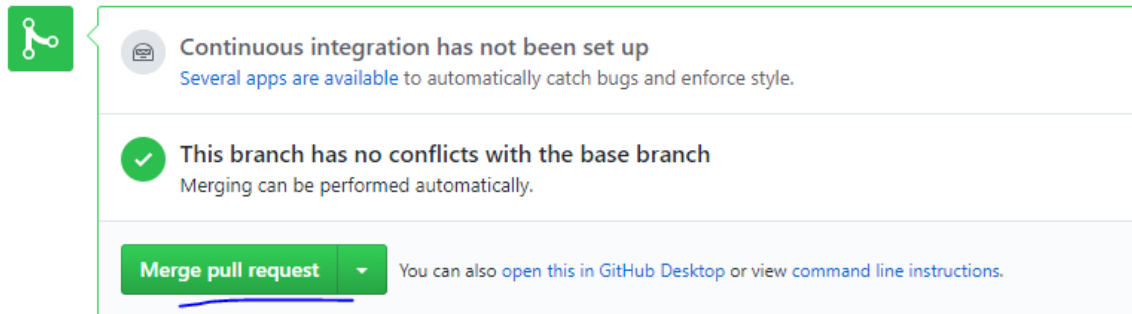
Write Preview **AA B i** **“ > < ©** **⋮ ⋮ ⋮** **@** **🔖** **↶**

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

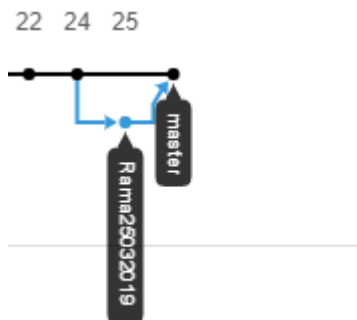
Create pull request

13. Y aceptaremos la pull request para que se haga el merge con la rama que hemos creado y el máster.



A screenshot of a GitHub pull request status box. It features a green icon with a white branching diagram on the left. The main content area has a light gray background and contains three sections: 1) A message 'Continuous integration has not been set up' with a subtext 'Several apps are available to automatically catch bugs and enforce style.' 2) A green checkmark icon followed by the text 'This branch has no conflicts with the base branch' and 'Merging can be performed automatically.' 3) A green button labeled 'Merge pull request' with a dropdown arrow, followed by the text 'You can also open this in GitHub Desktop or view command line instructions.'

14. Por último, veremos cómo se ha hecho bien el Merge y se ha fusionado con el Master y ya tendremos la información actualizada en el master.

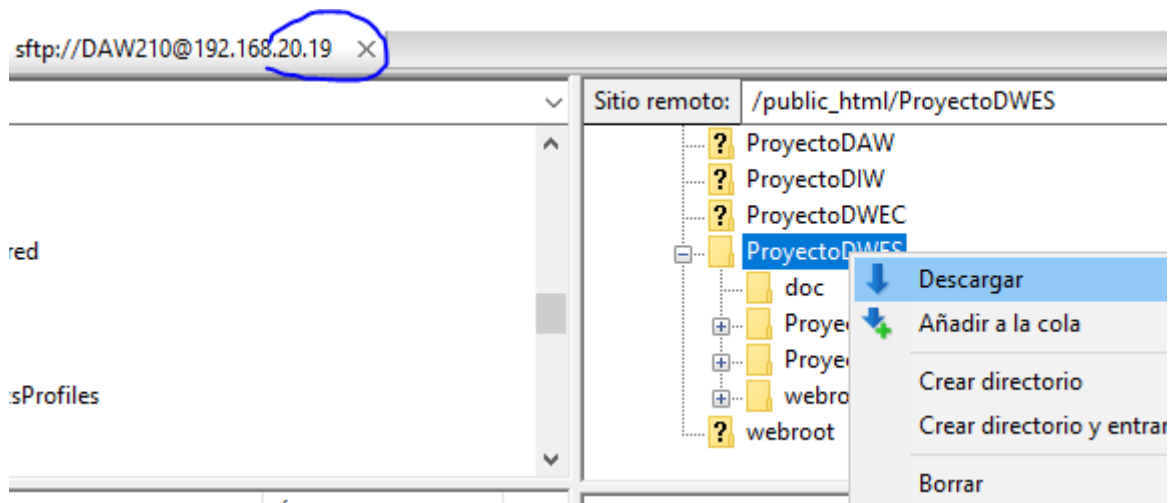


Rama25032019	Updated 16 minutes ago by Laura Fernandez	1 0	#5	Merged	
------------------------------	---	-------	----	--------	--

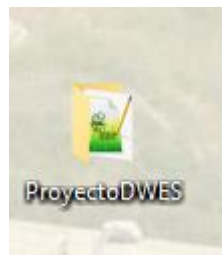
Subir ficheros a explotación

Filezilla

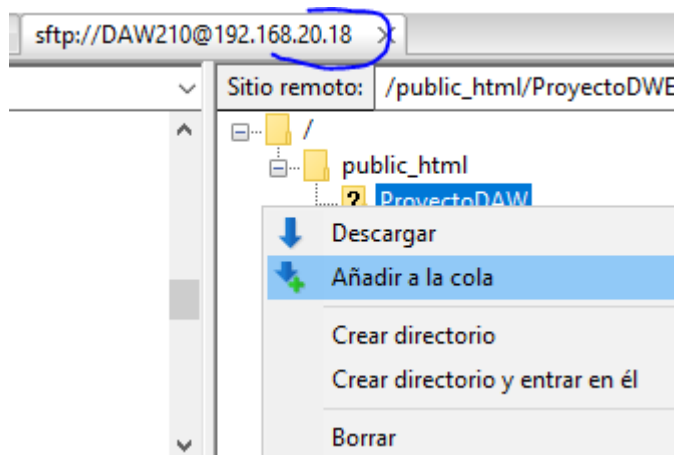
1. Con el filezilla podemos subir los ficheros a explotación nos conectamos a la maquina de desarrollo



2. Los descargamos a la maquina local por ejemplo vamos a dejar la carpeta en el escritorio.

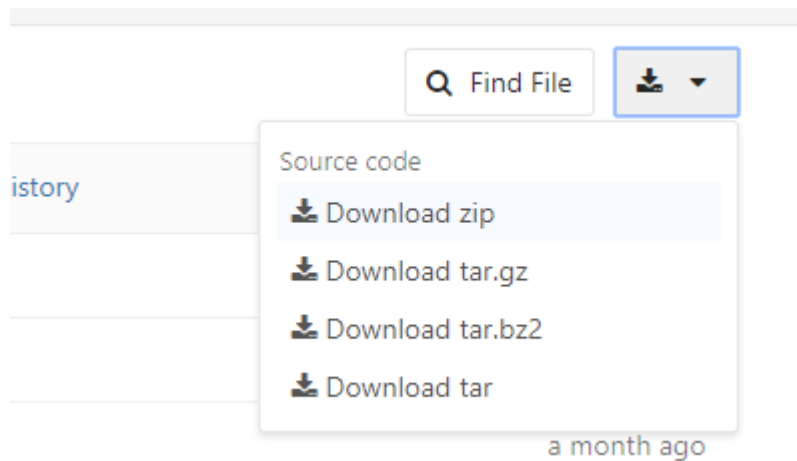


3. Cogemos la carpeta y por el filezilla nos conectamos al entorno de explotacion y la subimos

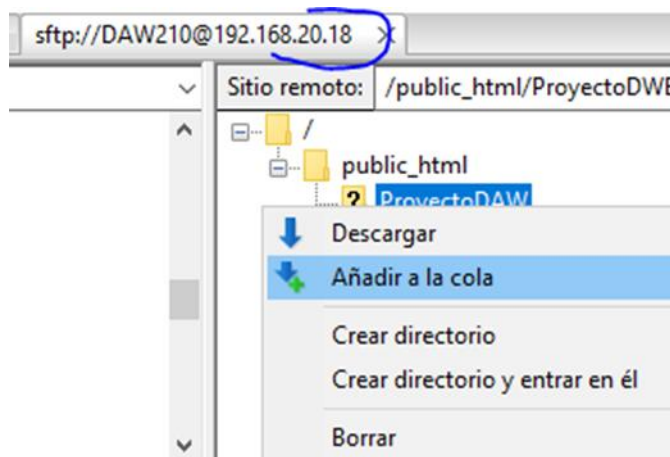


Mediante Repositorio

1. Primero vamos al repositorio y descargamos los archivos/ficheros que estén listos para subir a explotación
2. Tenemos una opción para descargarlo como un zip

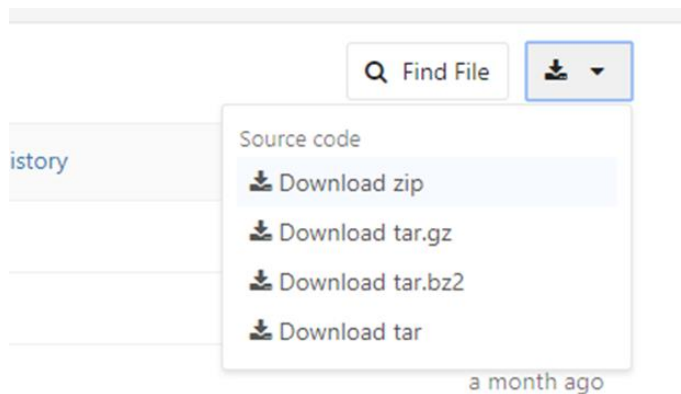


3. Y por último al igual que en el paso del Filezilla subimos al entorno de explotación los archivos/ficheros descargados desde el repositorio

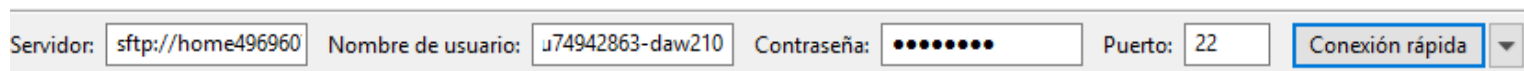


Subir al 1 and 1 las aplicaciones

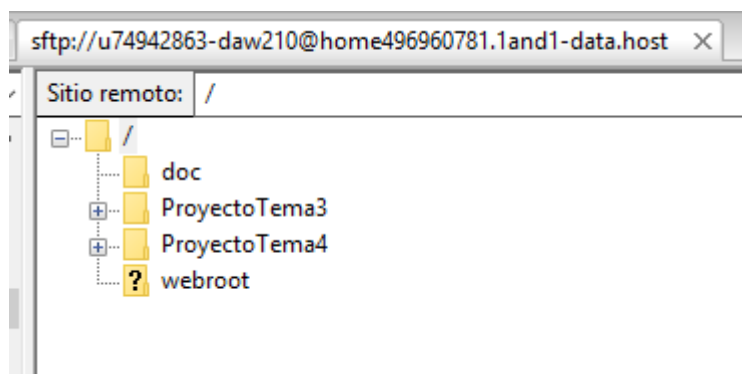
1. Tenemos que descargar los archivos desde el repositorio.



2. Nos vamos al Filezilla para conectarnos por FTP al servidor del 1 and 1

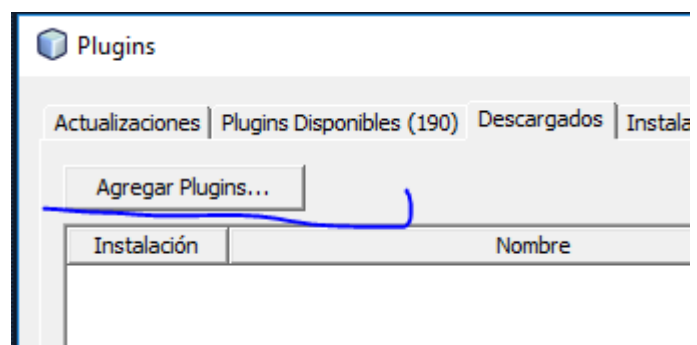
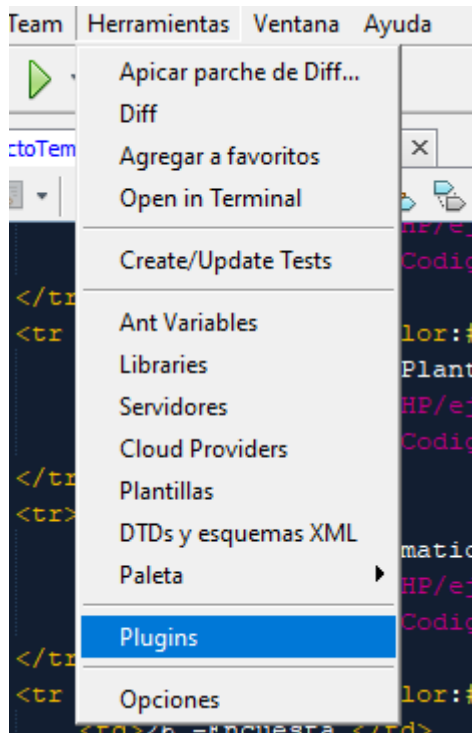


3. Y subimos los archivos que tenemos del repositorio a la carpeta que tenemos en el 1and1

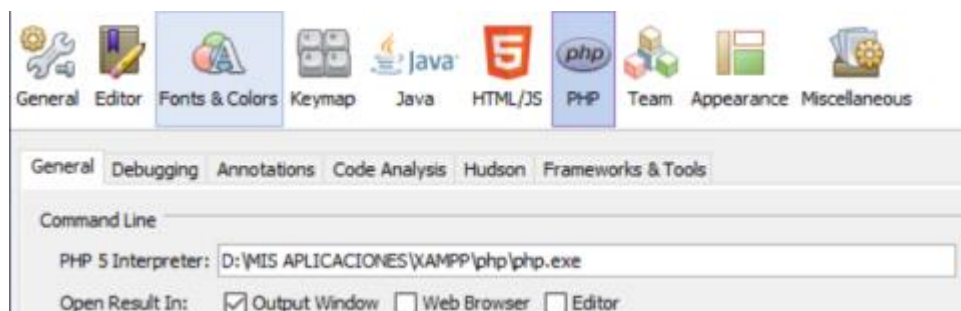


Documentación PHPDoc

1. Lo primero que tenemos que hacer es instalar el Plugin que lo descargaremos desde la siguiente url: <http://plugins.netbeans.org/plugin/43839/php-documentor> vamos a la siguiente ruta para instalar el plugin que hemos descargado y en agregar plugins seleccionamos el archivo .nbm descargado anteriormente.

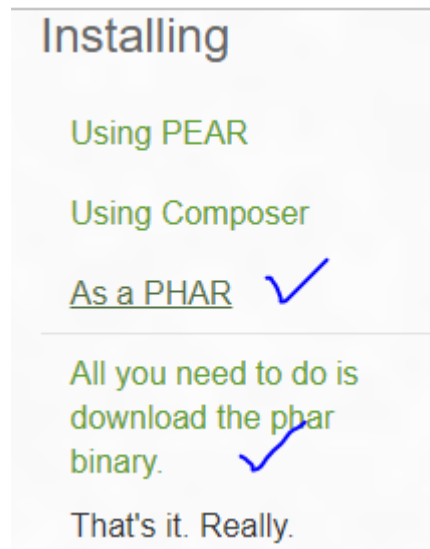


2. Vamos a las opciones de las herramientas para comprobar que la ruta del Interpretador es correcta.
 - En el apartado de php seleccionamos nuestro interprete que es el php.exe que tenemos en nuestro equipo
 - Si no lo tenemos lo descargamos de: <https://windows.php.net/download>

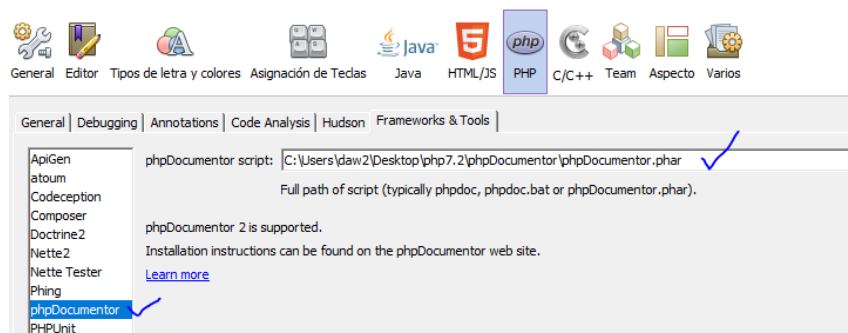


3. Descargamos el phpDocumentor y de la siguiente url: <https://www.phpdoc.org/>

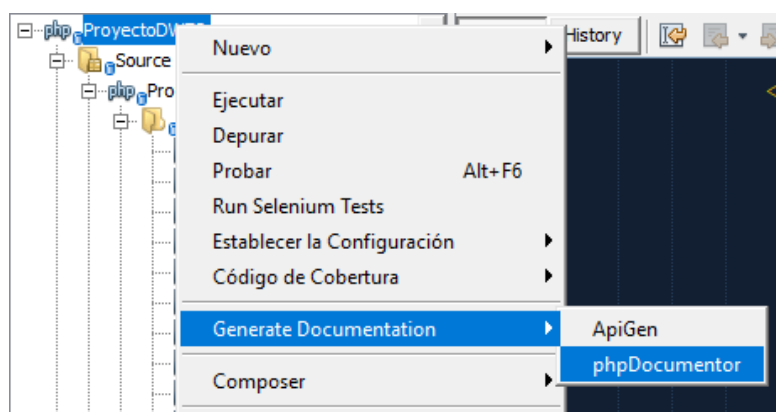
- Primero le daremos a *as a phar* y después se nos desplegará la opción en la que tenemos que pulsar para instalarlo.



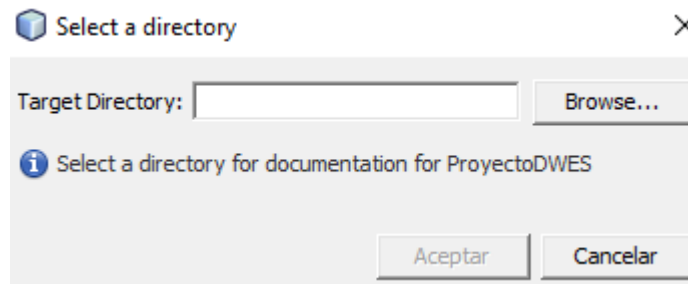
4. En la pestaña de PhpDoc seleccionamos el .phar que hemos descargado anteriormente.



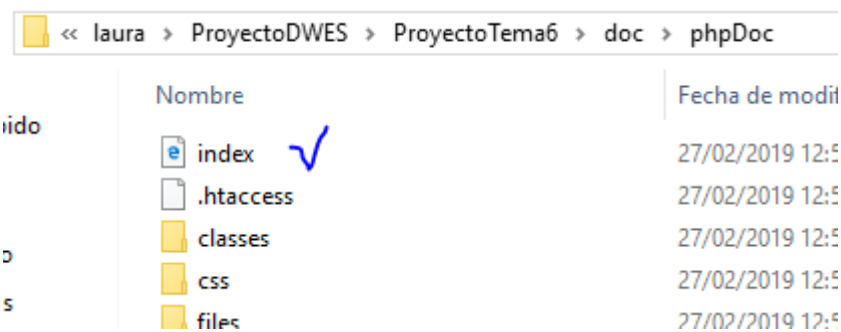
5. Generamos la documentación de un proyecto en la siguiente ruta.



6. Le indicamos en la carpeta que queremos que se nos genere la documentación.



7. En la carpeta nos creara un index.html en el cual podremos entrar para ver lo que ha generado la documentación y los posibles errores que tenemos.



Documentar una clase

- Primero documentaremos el archivo de la siguiente manera pondremos el nombre del archivo lo te va contener, el nombre del @autor y el @package al que corresponde en este caso al del modelo.

Le pondremos el nombre de la interface de la siguiente manera para que no cause un error de interface

- Los comentarios tienen que ir de la Manera que se muestra en la imagen Si no el php nos dará errores.

```
/**
 * Archivo UsuarioDB.php
 *
 * Tiene los metodos que estan en UsuarioPDO
 *
 * @author Laura Fernandez
 * @package modelo
 */

/**
 * Interface UsuarioDB
 */

interface UsuarioDB {
```

- Para documentar una función indicamos el nombre de la función y lo que hace.
- Después documentaremos con las etiquetas siguientes los parámetros que recibe esta función :

- @author: el nombre del autor

- @param: el tipo y el nombre de parámetro que va recibir.

- En este caso esta función no tiene un
- @return : esta etiqueta define lo que va Devolver la función

```
/**
 * Función altaUsuario
 *
 * Implementa la función altaUsuario
 *
 * @author Laura Fernandez
 * @param string $codUsuario pasa el codigo del usuario
 * @param string $password pasa la contraseña del usuario
 * @param string $descUsuario pasa la descripcion del usuario
 */

public static function altaUsuario($codUsuario, $password, $descUsuario);
```


DOCUMENTACION DE LA MAQUINA LFFUSEE

Creación de la maquina limpia Sistema operativo

Utilizaremos el Ubuntu Server 18.04 como servidor.

Le pondremos 500GB de memoria

2 GB de memoria Ram



Particiones

Realizaremos 3 particiones una para la swap con 4GB otra para datos con 100GB y por ultimo el resto para el sistema.

```

MOUNT POINT      SIZE      TYPE      DEVICE TYPE
[ /                395.997G  ext4      partition of local disk ► ]
[ /var            100.000G  ext4      partition of local disk ► ]
[ SWAP             4.000G    swap      partition of local disk ► ]
  
```

Nombre del equipo

El nombre del equipo para el entorno de explotación sera: LFFUSEE

```

miadmin@LFFUSEE:~$ hostname
LFFUSEE
miadmin@LFFUSEE:~$
  
```

Cuentas

Tendremos la cuenta de miadmin con la que administraremos todo el sistema.

Configuración de red

La configuración de red que le pondremos a esta maquina sera la ip del anfitrión mas 200:

```
miadmin@LFFUSEE:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.3.210 netmask 255.255.255.0 broadcast 192.168.3.255
    inet6 fe80::a00:27ff:fee8:aafa prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:e8:aa:fa txqueuelen 1000 (Ethernet)
    RX packets 16499 bytes 20425574 (20.4 MB)
    RX errors 0 dropped 26 overruns 0 frame 0
    TX packets 3208 bytes 239289 (239.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Servicio ssh

Nos conectaremos a esta maquina desde el Putty al igual que en la de desarrollo.

Comprobando que el servicio esta activo:

```
miadmin@LFFUSED:/$ sudo service ssh status
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2018-09-26 19:36:44 UTC; 9min ago
     Process: 1124 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 1128 (sshd)
       Tasks: 1 (limit: 1113)
      CGroup: /system.slice/ssh.service
              └─1128 /usr/sbin/sshd -D
```

Instalación de software

Descripción del programa Apache

Apache es el Servidor Web más utilizado, líder con el mayor número de instalaciones a nivel mundial muy por delante de otras soluciones como el IIS (Internet Information Server) de Microsoft. Apache es un proyecto de código abierto y uso gratuito, multiplataforma (hay versiones para todos los sistemas operativos más importantes), muy robusto y que destaca por su seguridad y rendimiento.

Utilidad en esta maquina

La utilidad en esta maquina es poder tener el servidor apache activo para subir nuestras paginas acabadas, al igual que al entorno de desarrollo.

Comprobar la instalación del servicio

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Wed 2018-09-26 20:08:53 UTC; 2min 2s ago
     Main PID: 2342 (apache2)
       Tasks: 55 (limit: 1113)
    CGroup: /system.slice/apache2.service
            └─2342 /usr/sbin/apache2 -k start
               2345 /usr/sbin/apache2 -k start
               2346 /usr/sbin/apache2 -k start
```

Descripción del programaPHP

PHP, acrónimo recursivo en inglés de PHP Hypertext Preprocessor (preprocesador de hipertexto), es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el HTML resultante.

Utilidad en esta maquina

La utilidad de este servicio en el entorno de explotación es el de poder visualizar el contenido php de nuestras webs.

Comprobar la instalación del servicios

Vamos a comprobar que tenemos la misma versión que en el entorno de desarrollo si no tendremos problemas

```
PHP 7.2.10-0ubuntu0.18.04.1 (cli)
```

Y sus ficheros de configuración

```
miadmin@LFFUSED:/etc/php/7.2$ ls  
apache2  cli  mods-available  
miadmin@LFFUSED:/etc/php/7.2$
```

ARCHIVOS DE CONFIGURACIÓN PHP.INI

Parámetros de php.ini

1. **safe mode**, modo seguro, si está desactivado se habilitan todas las funciones de PHP, para un uso educativo es conveniente ser funcional y no activarlo (off). Sin embargo, para servicios de hosting se recomienda activarlo.
2. **display errors**, muestra los errores en la misma página, es útil para encontrar cualquier fallo con facilidad.
3. **max_execution_time**, tiempo máximo de ejecución de una acción PHP, si dejamos que se ejecute indefinidamente podría colapsar el sistema. ☐
post_max_size, tamaño máximo de datos que se pueden enviar al servidor mediante POST.
4. **upload_max_filesize**, tamaño máximo de archivo que se puede subir al servidor. Si se va a trabajar con archivos grandes es conveniente cambiarlo.
5. **max_file_uploads**, número máximo de archivos que pueden ser subidos al servidor en una petición sencilla

```
; display_errors
;   Default Value: Off
;   Development Value: Off
;   Production Value: Off

; display_startup_errors
;   Default Value: Off
;   Development Value: On
;   Production Value: Off

; error_reporting
;   Default Value: E_ALL & ~E_NOTICE & ~E_STRICT & ~E_DEPRECATED
;   Development Value: E_ALL
;   Production Value: E_ALL & ~E_DEPRECATED & ~E_STRICT

; html_errors
;   Default Value: Off
;   Development Value: Off
;   Production value: Off

; log_errors
;   Default Value: Off
;   Development Value: On
;   Production Value: On

; max_input_time
;   Default Value: -1 (Unlimited)
;   Development Value: 60 (60 seconds)
;   Production Value: 60 (60 seconds)
```

Descripción del programa MySql

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL AB fue adquirida por Sun Microsystems en 2008, y ésta a su vez fue comprada por Oracle Corporation en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL.

Utilidad en esta maquina

Para administrar bases de datos desde el cliente con un software adicional como Workbench

Describir el proceso de instalación

Instalaremos el servicio con el comando apt-get install mysql-server

```
miadmin@LFFUSED:/$ sudo apt-get install mysql-server_
```

Abrimos la consola de mysql, entrando con el comando mysql y la contraseña anteriormente establecida en la instalación.

Entramos como sudo, para administrar los usuarios y contraseñas pertinentes:

```
sudo mysql
```

Asignamos a un usuario una nueva contraseña

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'paso';
```


Creación de nuevos usuarios
Podemos crear usuarios con

```
CREATE USER 'operadorBBDD'@'localhost' IDENTIFIED BY 'contraseña';
```

Y darle privilegios sobre todas las tablas

```
GRANT ALL PRIVILEGES ON *.* TO 'operadorBBDD'@'localhost' WITH GRANT OPTION;
```

O sobre alguna en concreto

```
GRANT ALL PRIVILEGES ON 'nombreBD'.'nombreTabla' TO "operadorBBDD" '@'localhost' WITH GRANT OPTION;
```

Mantenimiento

SSH:

Podemos parar, iniciar o reiniciar el servicio con el comando `sudo service ssh stop/start/reboot/restart` respectivamente.

Apache:

En Apache, para reiniciar el servicio debemos ejecutar el comando `sudo service apache2 restart/reload`, también podemos pararlo o iniciarlo con `stop/start`, para cambiar la carpeta desde la que sirve Apache lo podemos modificar con `sudo nano /etc/apache2/sites-enabled/000-default.conf` y modificando en `DocumentRoot` la ruta predeterminada de Apache `/var/www/html` por la que queramos nosotros.

MySQL:

Podemos reiniciar, parar e iniciar el servicio MySQL con el comando `sudo service MySQL reload/restart/stop/start`, respectivamente.

PHP:

PHP no es un servicio, no podemos pararlo ni iniciarlo, ni reiniciarlo, y en principio no requiere de mantenimiento.

Log / Monitorizacion

SSH:

Podemos comprobar el estado del servicio SSH con el comando `sudo service ssh status`.

Apache:

Podemos ver el estado de Apache con el comando `service apache2 status`:
Podemos ver los puertos que están abiertos con el comando `netstat -ltn`. También podemos ver la lista de app instaladas con `sudo ufw app list`.

MySQL:

En MySQL podemos ver el estado del servicio con el comando `sudo service mysql status`.

Copia de seguridad

SSH:

Para guardar la configuración de SSH en caso de querer cargarla en otra máquina o por razones de seguridad deberemos copiar el archivo de configuración `/etc/ssh/sshd_config` en nuestra máquina anfitriona.

Apache:

Si queremos quedarnos con la configuración de Apache debemos guardar en el anfitrión los archivos ubicados en `/etc/apache2/sites-enabled/000-default.conf` y `/etc/apache2/ports.conf`.

MySQL:

Si queremos guardar la configuración de MySQL debemos guardar los archivos ubicados en `/etc/mysql/my.cnf`, `/etc/mysql/mysql.cnf` y `/etc/mysql/mysql.conf.d/mysqld.cnf`.

PHP:

Si queremos guardar la configuración de PHP debemos hacer una copia del archivo ubicado en el path `/etc/php/7.2/apache2/php.ini`.

Transferir archivos de desarrollo a explotación.

Nos vamos a la carpeta donde actuá apache que lógicamente es la misma que antes `varwww/html` con el comando `cd` para viajar por las carpetas y allí ejecutamos el comando `get index.html`, siendo `index` el archivo a descargar y tras ejecutarlo, nos descargará el archivo a la carpeta del entorno de explotación en la que habíamos ejecutado el comando SFTP, en este caso, `/var/www/html`.

Por último usamos el comando `exit` para salir de la conexión con la máquina de desarrollo.

```
sftp> cd /var/www/html
sftp> get index.html
Fetching /var/www/html/index.html to index.html
/var/www/html/index.html 1
sftp> exit
```

