

# DOCUMENTACION DE LAS MAQUINAS

Laura Fernández

DESARROLLO DE APLICACIONES WEB

<b>DOCUMENTACION DE LA MAQUINALFFUSED</b>	4
Sistema operativo	4
Particiones	4
Nombre del equipo	5
Cuentas	5
Configuracion 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
Descripción del programa PHP	10
Utilidad en esta maquina	10
Describir el proceso de instalación	10
Descripción del programa MySql	12
Utilidad en esta maquina	12
Describir el proceso de instalación	12
Mantenimiento	13
Copia de seguridad	14
Repositorio	14

<b>DOCUMENTACION DE LA MAQUINALFFW10ED</b> .....	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

<b>DOCUMENTACION DE LA MAQUINA LFFUSEE</b>	30
Creación de la maquina limpia.....	30
Sistema operativo .....	30
Particiones .....	30
Nombre del equipo .....	30
Cuentas .....	31
Configuración de red.....	31
Servicio ssh .....	31
Instalación de software.....	32
Descripción del programa Apache .....	32
Utilidad en esta maquina .....	32
Comprobar la instalación del servicio .....	32
Utilidad en esta maquina .....	33
Comprobar la instalacion del servicios.....	33
Transferir archivos de desarrollo a explotación. ....	34

# DOCUMENTACION DE LA MAQUINALFFUSED

## Creacion 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 sera 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 ultimo 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

Otra cuenta que sera operadorweb para organizar los servicios Web

## Configuracion de red

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

Edit enp0s3 IPv4 configuration

IPv4 Method: [ Manual ▼ ]

Subnet: 192.168.3.0/24

Address: 192.168.3.110

Gateway: 192.168.3.1

Name servers: 192.168.20.20, 8.8.8.8  
IP addresses, comma separated

Search domains:   
Domains, comma separated

[ Guardar ]  
[ Cancelar ]

Para ver la configuracion 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 esta activo para poder conectarnos desde fuera.

Vemos que esta 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 maquina sera 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 susejercicios.

### 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 wi
# 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
```

## 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

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".

## ARCHIVOS DE CONFIGURACIÓN

Parametros de php.

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.

## 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_
```

## 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`.

### 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`.

## 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`.

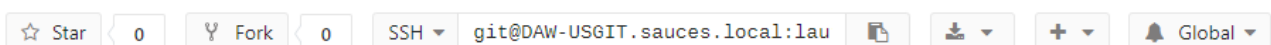
## Repositorio

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.

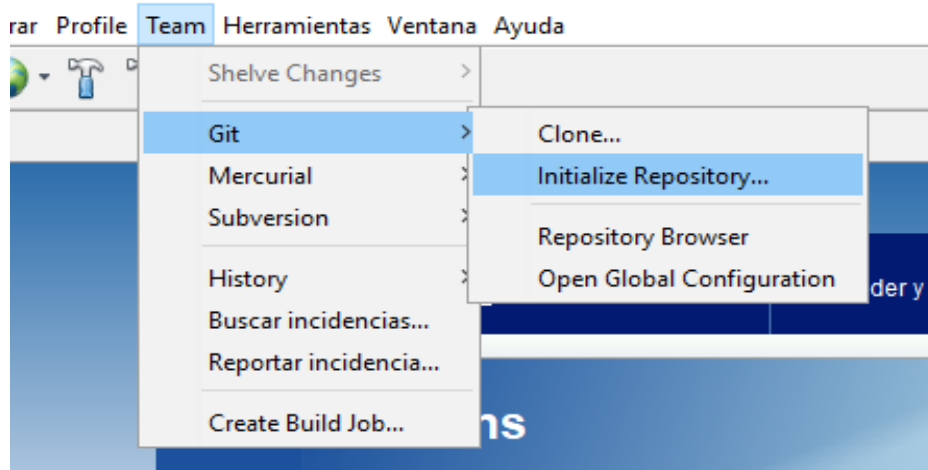
1. Lo primero que haremos será crear un proyecto en el repositorio en este caso GitLab el cual le pondremos el nombre de la asignatura.



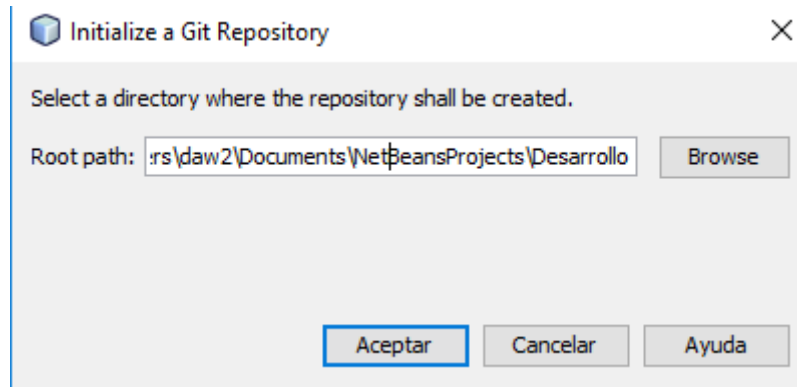
ProyectoDWESLauraFernandez



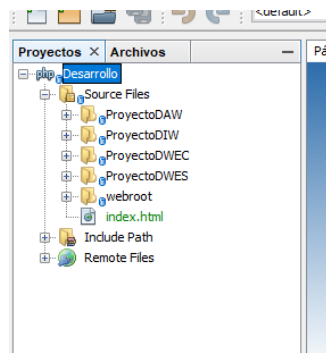
2. Luego en el NetBeans con el proyecto abierto que queremos subir al repositorio nos iremos a git y inicializar el repositorio.



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

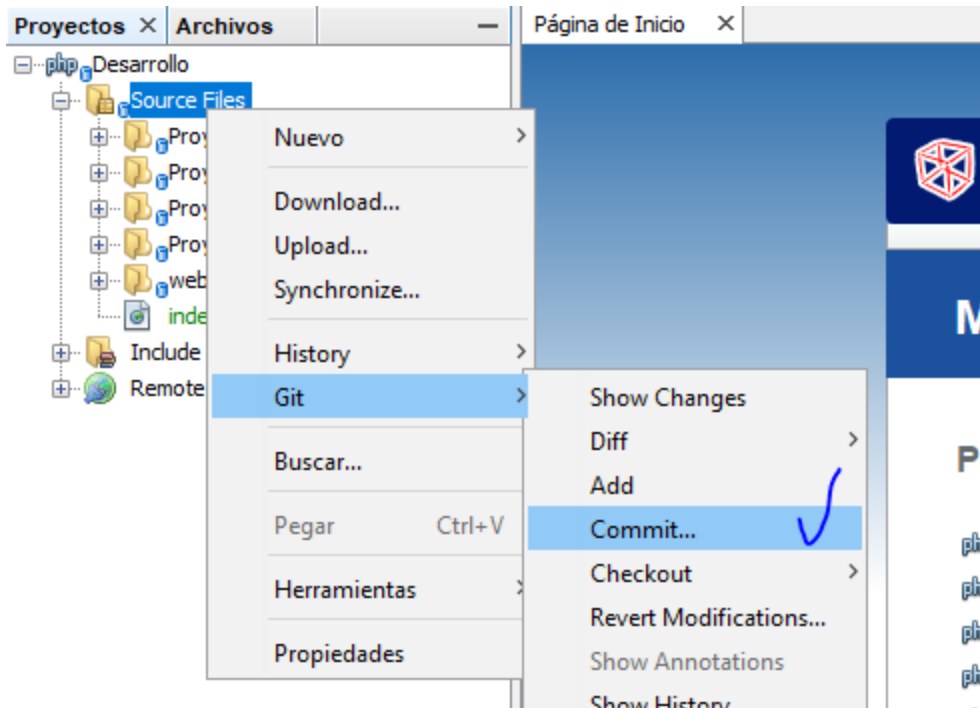


4. Ya tenemos el proyecto inicializado ahora solo nos hará falta subirlo.

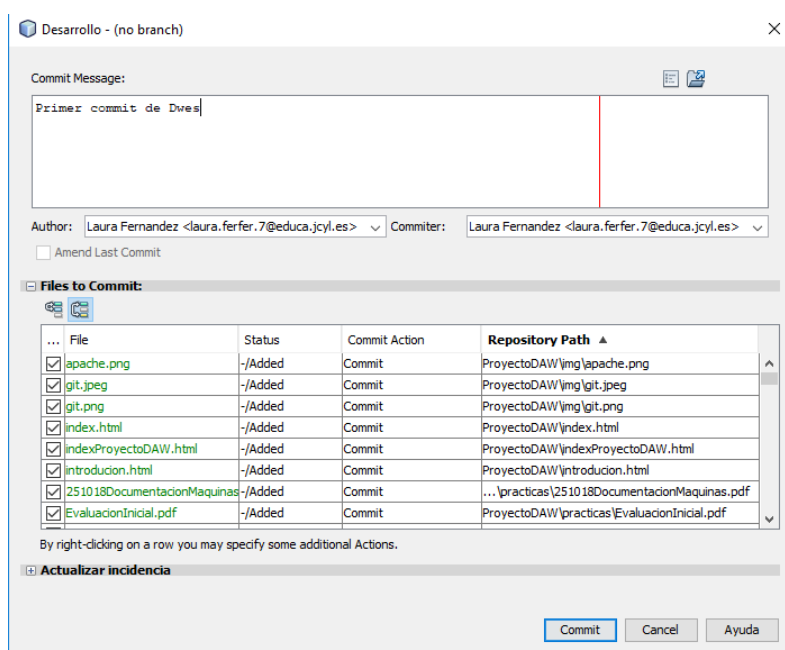




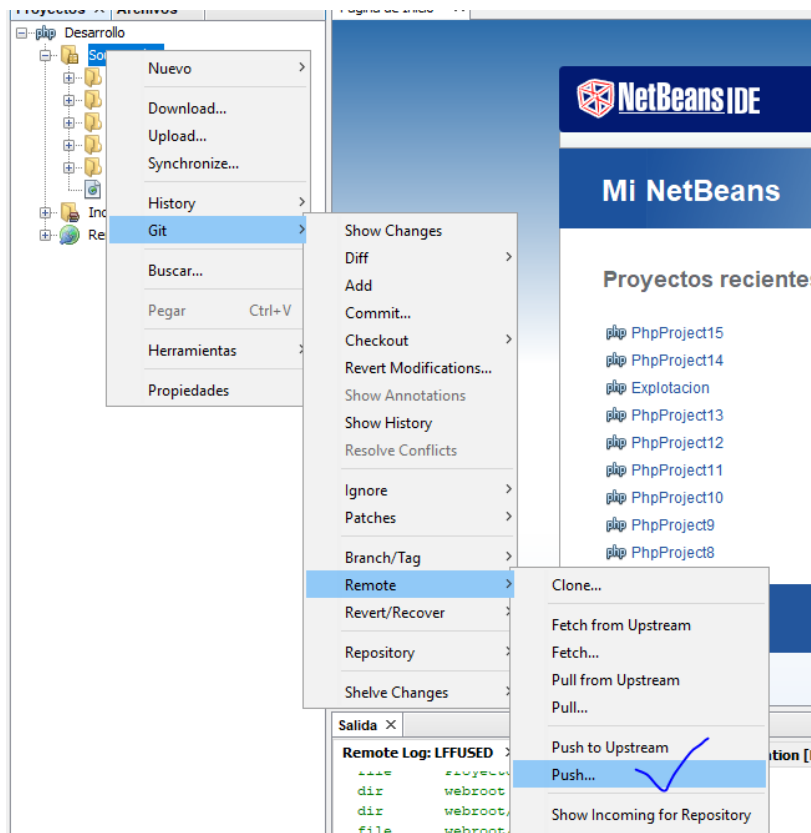
5. Seleccionaremos la pestaña de git y primero haremos un commit para confirmar lo que vamos a subir al repositorio y ponerle un nombre para saber que estamos subiendo.



6. Le ponemos el nombre al commit en este caso como es el primero le dare 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:

User:  (leave blank for anonymous access)

Password:  ☐ Save Password



# DOCUMENTACION DE LA MAQUINALFFW10ED

## Creación de la maquina limpia

### Sistema operativo

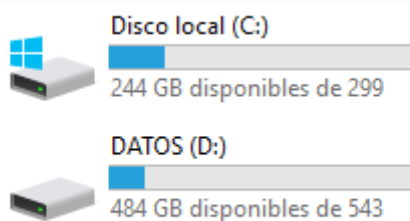
Hemos elegido para el entorno de desarrollo en entorno cliente un Windows 10.



### Particiones

Las particiones que tendra este sistema operativo seran:

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



### Nombre del equipo

El nombre del equipo sera: **LFFW10ED**

## Cuentas

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

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

Escriba un nombre de equipo:  
miadmin-PC

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: 192 . 168 . 3 . 10

Máscara de subred: 255 . 255 . 255 . 0

Puerta de enlace predeterminada: 192 . 168 . 3 . 1

☐ Obtener la dirección del servidor DNS automáticamente

☒ Usar las siguientes direcciones de servidor DNS:

Servidor DNS preferido: 192 . 168 . 20 . 20

Servidor DNS alternativo: 8 . 8 . 8 . 8

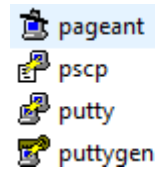
☐ Validar configuración al salir Opciones avanzadas...

Aceptar Cancelar

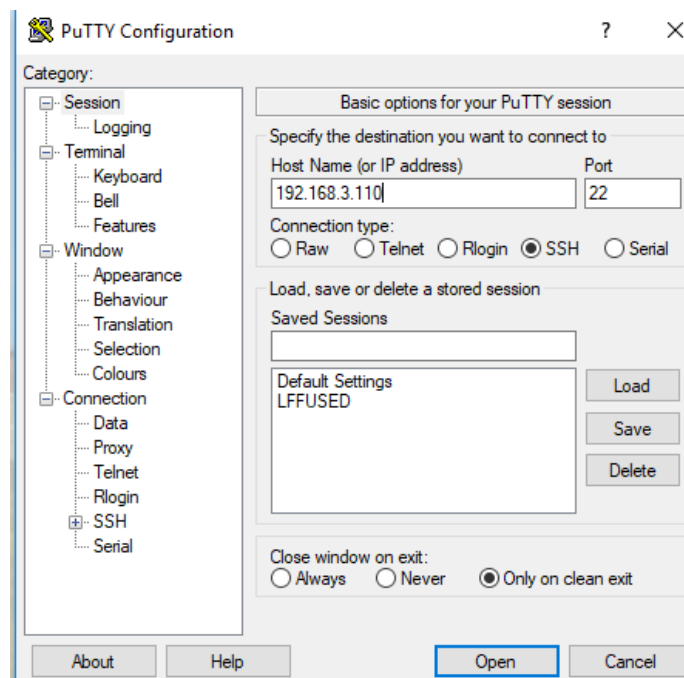
## Servicio ssh

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).

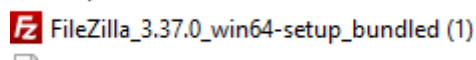
## Utilidad en esta maquina

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

## Describir el proceso de instalación

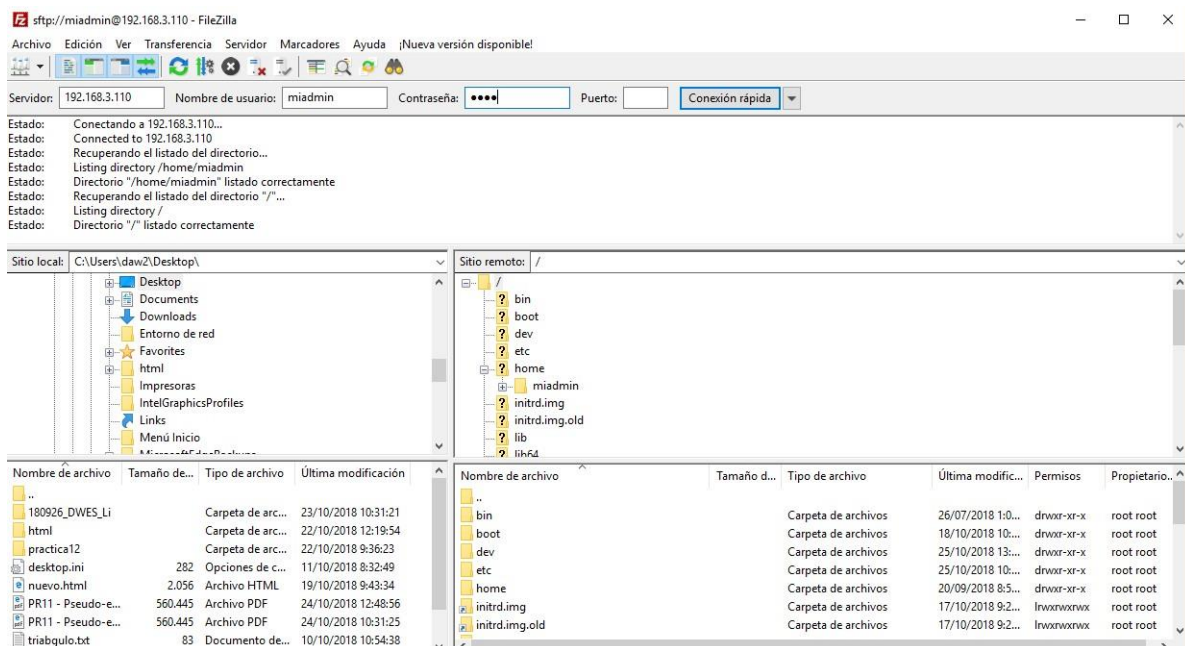
Descargaremos el Filezilla desde la pagina.

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.

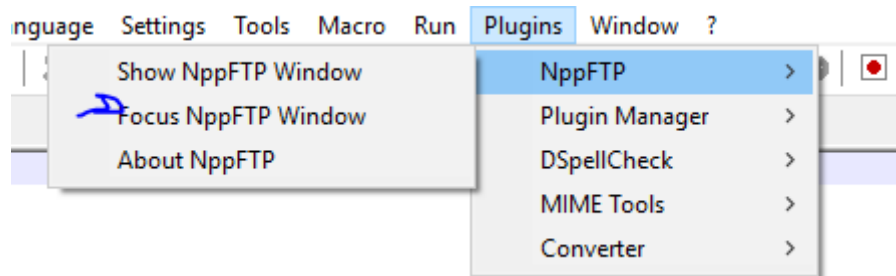
### Utilidad en esta maquina

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++

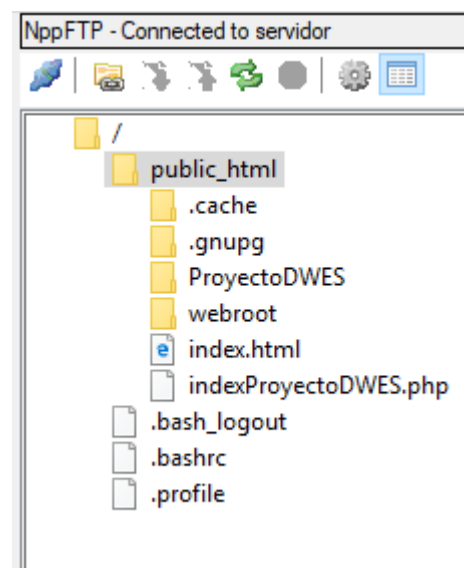
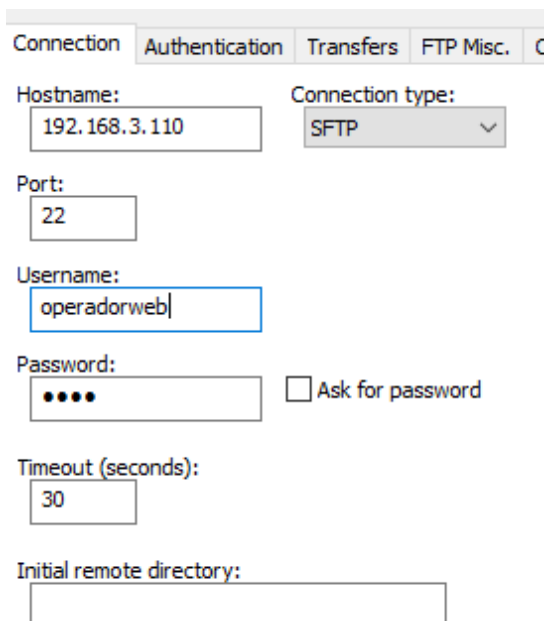
Nos conectaremos por Notepad++ al servidor:

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

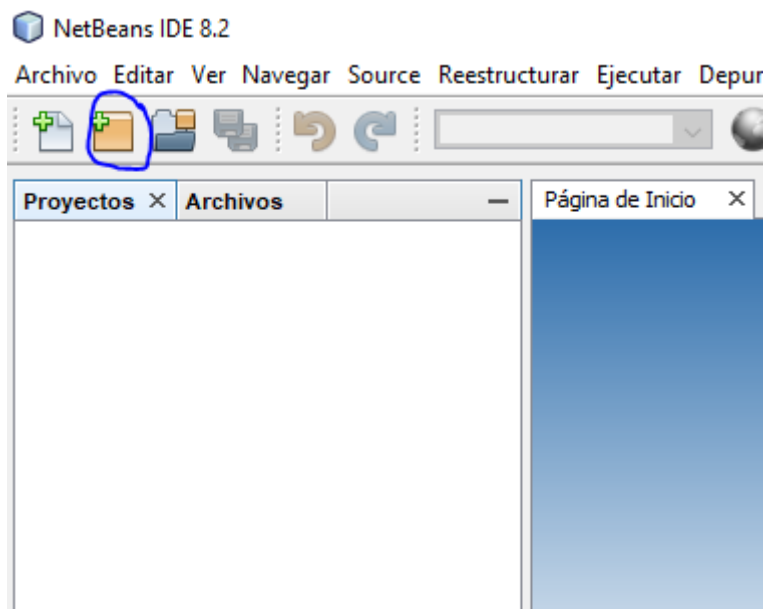
NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

### Utilidad en esta maquina

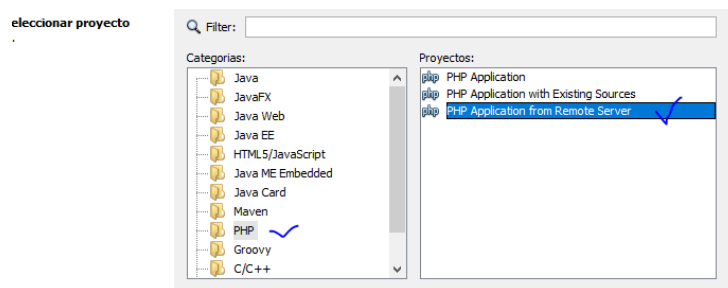
La utilidad del NetBeans en esta maquina sera la de conectarnos al servidor para editar los ficheros del servidor desde el cliente.

### 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 qué 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... Manage...

Running ssh-agent will be used.

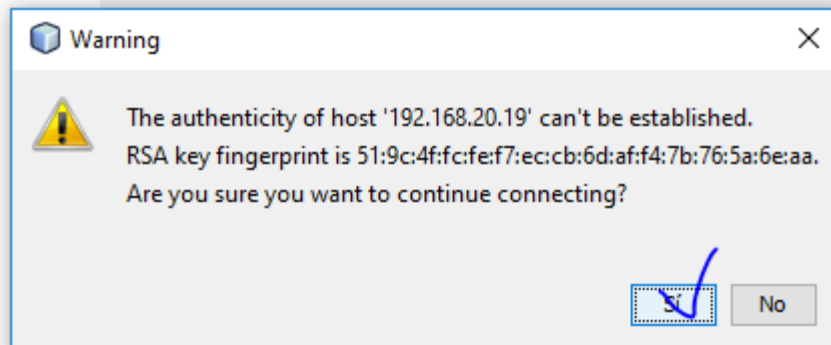
Known Hosts File:  Browse...

Initial Directory:

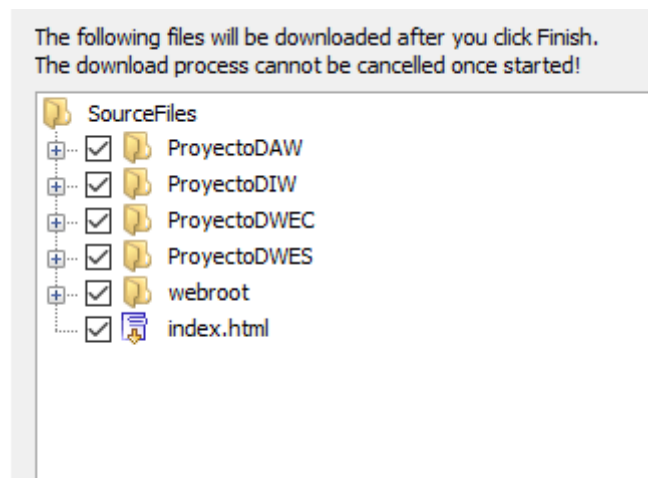
Timeout (s):

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

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



7. Nos salen los ficheros que vamos a descargar para verlos o modificarlos



8. Y ya tendremos los ficheros en nuestro anfitrión para desde el ide hacer modificaciones o ver ficheros crearlos etc...

## Descripción del programa WorkBench

MySQLWorkbench es una herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, gestión 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

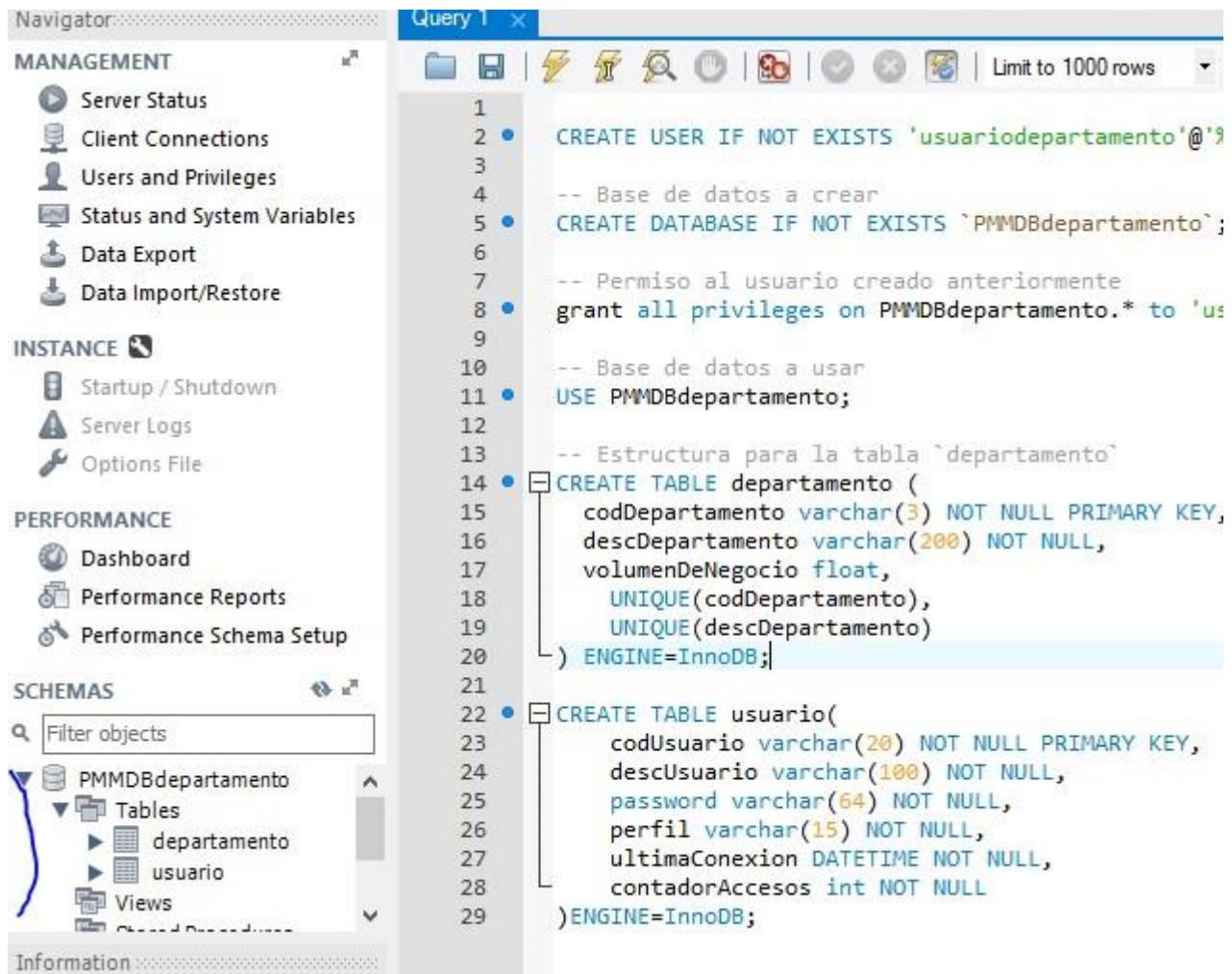
MySQL Server Port: 3306 TCP/IP

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 'SCHEMAS' pane is expanded, showing a tree view of the database structure. The 'PMMDBdepartamento' database is selected, and its 'Tables' folder is expanded, showing 'departamento' and 'usuario' tables. The 'Query 1' pane on the right contains the following SQL code:

```

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 'usuariodepartamento'@'%';
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;
  
```



# 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 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

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

## Comprobar la instalacion del servicios

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

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

Y sus ficheros de configuración

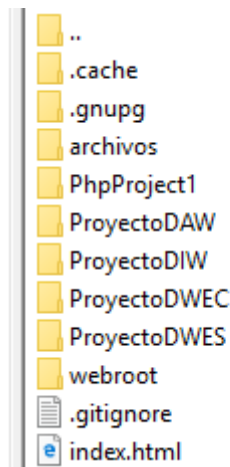
```
miadmin@LFFUSED:/etc/php/7.2$ ls
apache2  cli  mods-available
```

## Transferir archivos de desarrollo a explotación.

Una opción para ello seria conectarnos con el Filezilla y descargar los archivos de la maquina de desarrollo para conectarnos al servidor de explotación y subirlos ahí.

Otra opción seria conectarnos con el Putty o directamente desde el servidor a la maquina de explotación, nos vamos a la carpeta en la que actuá directamente Apache `var/www/html` y desde ahi nos conectamos a la maquina de desarrollo con el comando `sftp miadmin@192.168.3.110` nos pedirá la contraseña del usuario y ya estaríamos dentro de la maquina.

Descargar los archivos que tenemos a continuación y subirlos a el servidor de 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 alli 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
```

