

*TEMA 2: INSTALACIÓN,
CONFIGURACIÓN Y
DOCUMENTACIÓN DEL
ENTORNO DE
DESARROLLO Y DEL
ENTORNO DE
EXPLOTACIÓN*

Contenido

EJERCICIO 1	3
LRAUSED	3
Configuración de la maquina	3
Instalación del software.....	3
Configuración servidor ssh.....	4
Configuración apache2.....	5
Administración del sitio	6
LRA-W7ED	7
Configuración inicial.....	7
Instalación y uso de aplicaciones	7
EJERCICIO 2	15
LRA-USEE	15
Configuración inicial.....	15
Instalación del software.....	15
Configuración servidor ssh.....	16
Configuración apache2.....	17
Administración del sitio	18

EJERCICIO 1

1. Instalación y configuración del entorno de desarrollo para la asignatura de Desarrollo web del lado servidor sobre dos máquinas virtuales xxx-USED y xxx-W7ED. Documentando la configuración elegida y los aspectos más importantes del proceso de instalación.

Documentar el proceso de monitorización del correcto funcionamiento de los servicios.

Mantener actualizada esta documentación durante todo el curso.

LRAUSED

Configuración de la maquina

Maquina LRA-USED		
Sistema Operativo	Ubuntu Server 16.04.03	
Memoria	512 MB	
Usuario y Contraseña	miadmin	paso
	operadorweb2	paso
Configuración de red	Ip	192.168.3.108
	Netmask	255.255.255.0
	Gateway	192.168.3.1
	Dns	192.168.20.20
Software	Servidor Web	Apache (con modulo PHP)
	Servidor SSH	Openssh-server
	Servicio SFTP	SFTP
	SGBD	Mysql-server

Instalación del software

El software que necesitaremos instalar en esta máquina es openssh-server, apache2 y php7.0.

Para ello utilizaremos los comandos

- apt-get update → para actualizar el repositorio
- apt-get install openssh-server → para instalar el servidor ssh y poder conectarnos de forma remota
- apt-get install apache2 → para instalar apache y tener nuestro servidor web
- apt-get install php7.0 → para poder programar en php

Configuración servidor ssh

Lo primero que haremos será crear en el directorio “/var/www” donde guardaremos nuestra página, en este caso lo llamaremos “public_html”. Después le daremos los permisos para poder modificar en esta carpeta a nuestro usuario (que crearemos más adelante).

```
root@LRAUSED:~# mkdir public_html
root@LRAUSED:~# chmod 775 public_html/
```

Ahora vamos a crear nuestro usuario que se llamara “operadorweb2”, usaremos el siguiente comando:

```
sudo useradd -g www-data -d /var/www/public_html -s /bin/bash operadorweb
```

Con este comando lo que hacemos es hacer que, nuestro usuario pertenezca al grupo “www-data”, ponerle como directorio personal “/var/www/public_html” y asignarle un shell “/bin/bash”.

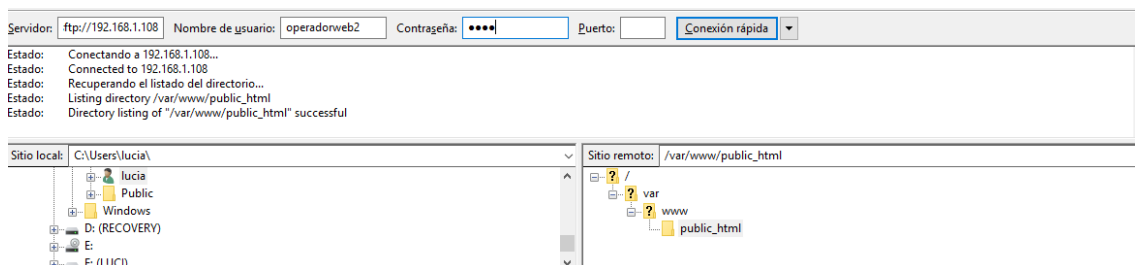
Después le asignaremos una contraseña a nuestro usuario con el comando passwd:

```
root@LRAUSED:~# passwd operadorweb2
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: password updated successfully
```

Y a continuación haremos que sea el propietario del directorio “public_html”:

```
root@LRAUSED:~# chown -R operadorweb2:www-data public_html/
```

Ahora comprobaremos desde nuestro cliente mediante el filezilla si nos podemos conectar con nuestro usuario



Configuración apache2

En el directorio “/etc/apache2/sites-available” vemos que hay un sitio que se llama “000-default.conf”

```
root@LRAUSED:~# ls /etc/apache2/sites-available
000-default.conf  default-ssl.conf
```

Copiamos ese sitio con el comando

```
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/luci.conf
```

Después de copiarlo editamos el fichero “luci.conf” y donde pone “DocumentRoot” pondremos el directorio de nuestro usuario.

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/public_html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Guardamos el fichero y desactivamos el sitio por defecto con el comando a2dissite y activaremos nuestro sitio con el comando a2ensite:

```
root@LRAUSED:~# a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  service apache2 reload
root@LRAUSED:~# a2ensite luci.conf
Enabling site luci.
To activate the new configuration, you need to run:
  service apache2 reload
```

Cuando hayamos hecho esto tendremos que reiniciar apache con el comando “service apache2 restart”. Después de esto ya tendremos configurado nuestro sitio.

Para que nuestro servidor reconozca el lenguaje php tendremos que instalar una librería de apache

apt-get install libapache2-mod-php

Después editaremos el fichero /etc/apache2/mods-enabled/dir.conf

Así estará el fichero ahora mismo

```
GNU nano 2.5.3 Archivo: /etc/apache2/mods-enabled/dir.conf

<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Lo que haremos será mover el index.php al principio para que quede de la siguiente forma

```
GNU nano 2.5.3 Archivo: /etc/apache2/mods-enabled/dir.conf

<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>

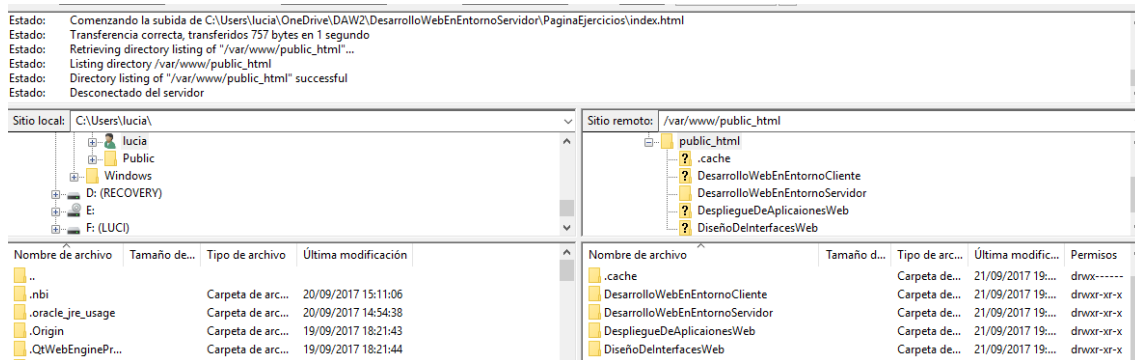
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Después reiniciaremos el servicio apache con “service apache2 restart” y nuestro servidor ya podrá leer código php

Administración del sitio

Desde nuestro cliente y con filezilla y un editor de texto empezaremos a crear nuestro sitio.

Lo primero que haremos será crear una estructura de directorios en la carpeta donde va a estar nuestro sitio, en este caso en “public_html”.



Después crearemos un index.html y comprobaremos desde el navegador si carga nuestra página

```
<?xml version="1.0" encoding="UTF-8" ?>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Welcome</title>
</head>
<body>
    <h1>Desarrollo de aplicaciones web</h1>
    <p><a href="DesarrolloWebEnEntornoServidor/ejercicios.html">Desarrollo Web En Entorno Servidor</a></p>
    <p><a href="DesarrolloWebEnEntornoCliente/mantenimiento.html"></p>Desarrollo Web En Entorno Cliente</a></p>
    <p><a href="DespliegueDeAplicacionesWeb/mantenimiento.html"></p>Desarrollo Web En Entorno Cliente</a></p>
    <p><a href="DiseñoDeInterfacesWeb/mantenimiento.html"></p>Desarrollo Web En Entorno Cliente</a></p>
    <footer><p><em><small>Lucía Rodríguez Álvarez</small></em></p></footer>
</body>
</html>
```

LRA-W7ED

Configuración inicial

Maquina: LRA-W7ED		
Sistema Operativo	Windows 7 profesional	
Memoria	1024MB	
Usuario y contraseña	admin	admin
	usuario	paso
Configuración de red	Ip	192.168.3.128
	Netmask	255.255.255.0
	Gateway	192.168.3.1
	DNS	192.168.20.20
Software	Navegador	Chroome
	IDE	Netbeans
	Ofimática	Libre Office
	Editor texto	Notepad++, komodo 10
	Otros	Filezilla, putty,

Instalación y uso de aplicaciones

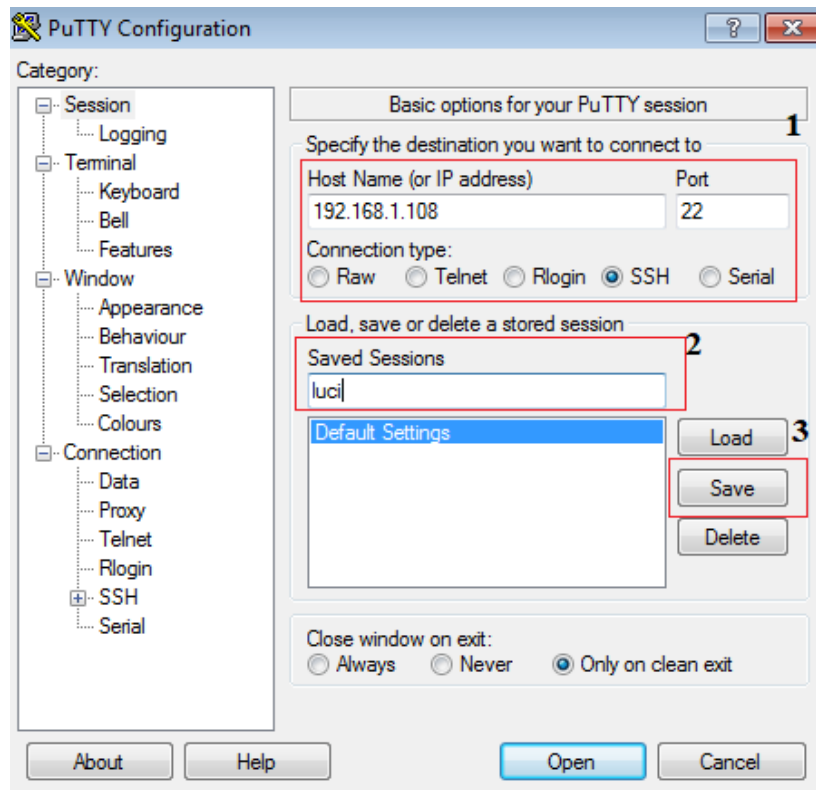
GOOGLE CHROOME (navegador)

Es el navegador que utilizaremos para ver nuestra página web.

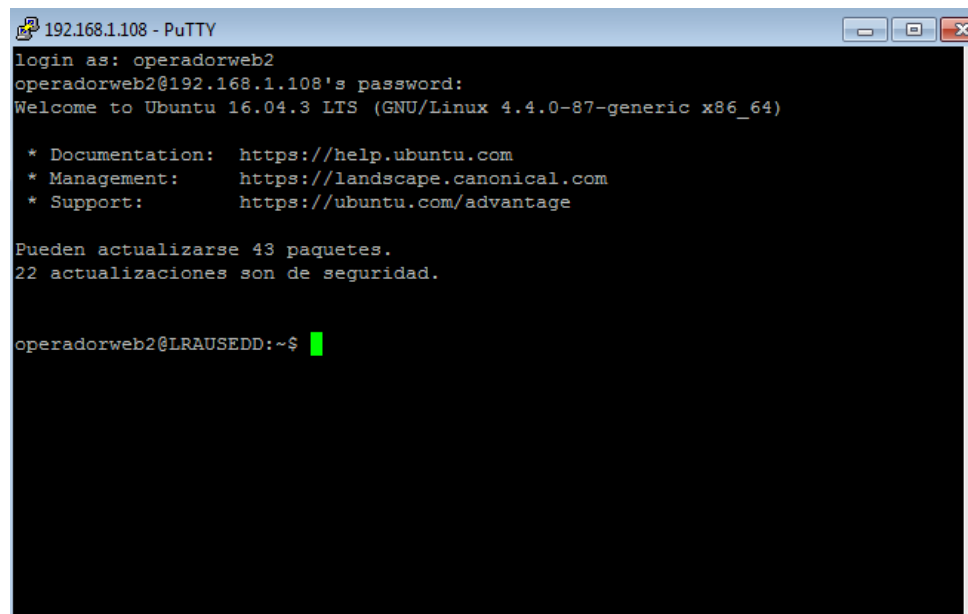


PUTTY

Lo utilizaremos para acceder de forma remota a nuestro servidor



1. Indicaremos la ip del equipo al que queremos conectarnos, el puerto por el que nos conectaremos y el método en nuestro caso será por ssh.
2. Le pondremos un nombre a nuestra conexión
3. Guardaremos nuestra conexión para así la próxima vez entrar directamente solo con clicar en ella

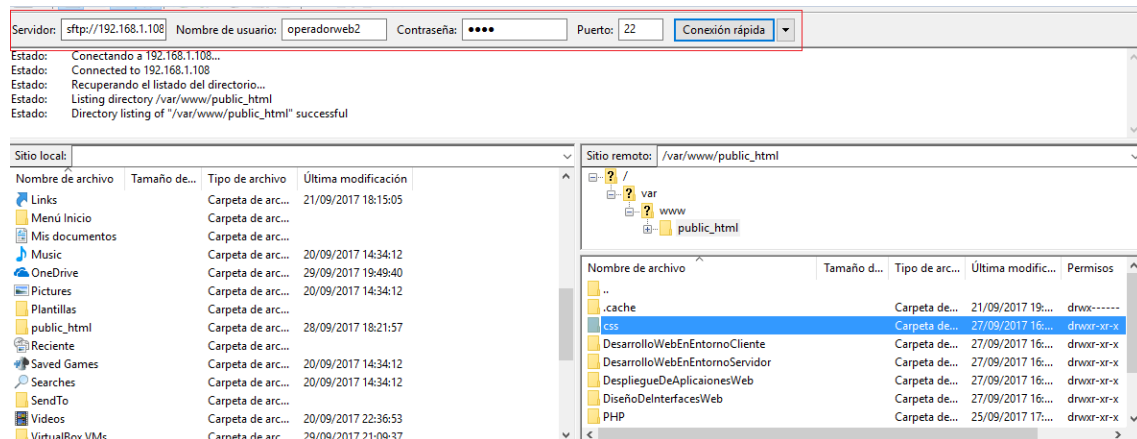


NetBeans(IDE)

IDE que podemos utilizar para editar nuestros archivos del servidor

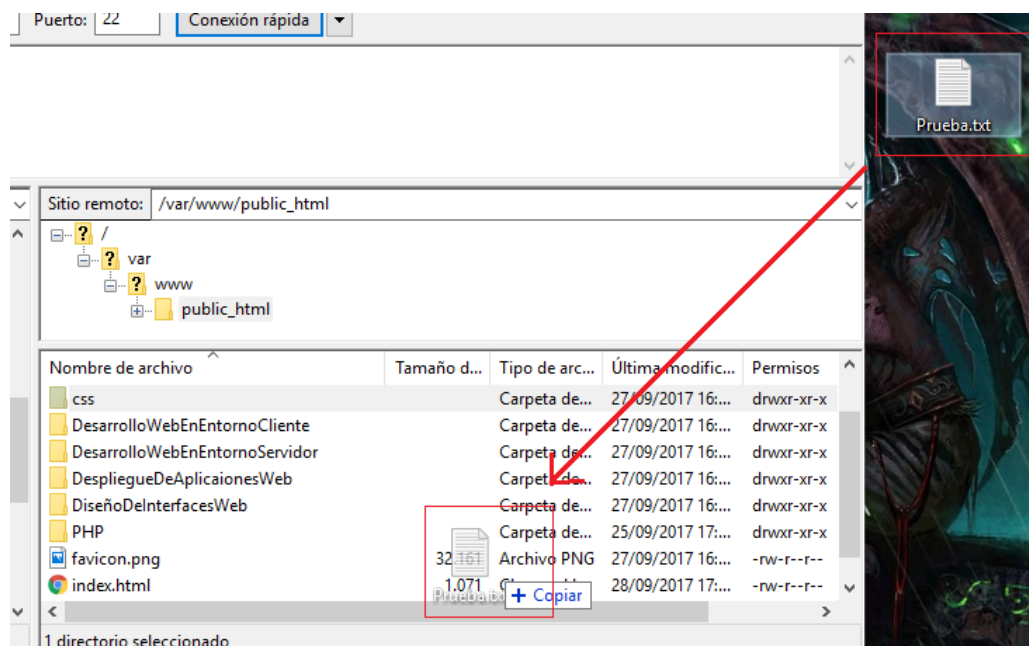
FILEZILLA

Lo usaremos para descargar y transferir ficheros a nuestro servidor

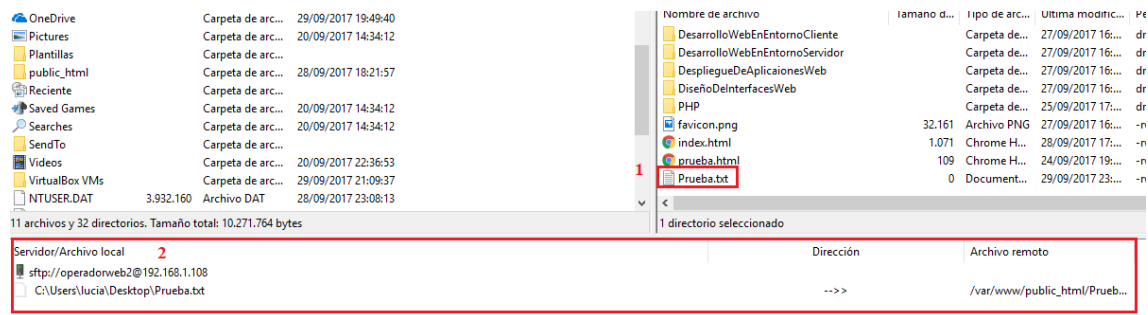


Indicaremos la ip del servidor al que queremos conectarnos, el usuario con el que nos conectaremos y la contraseña de este y además el puerto.

Para pasar archivos de nuestro equipo a nuestro servidor solo tendremos que arrastrarlos



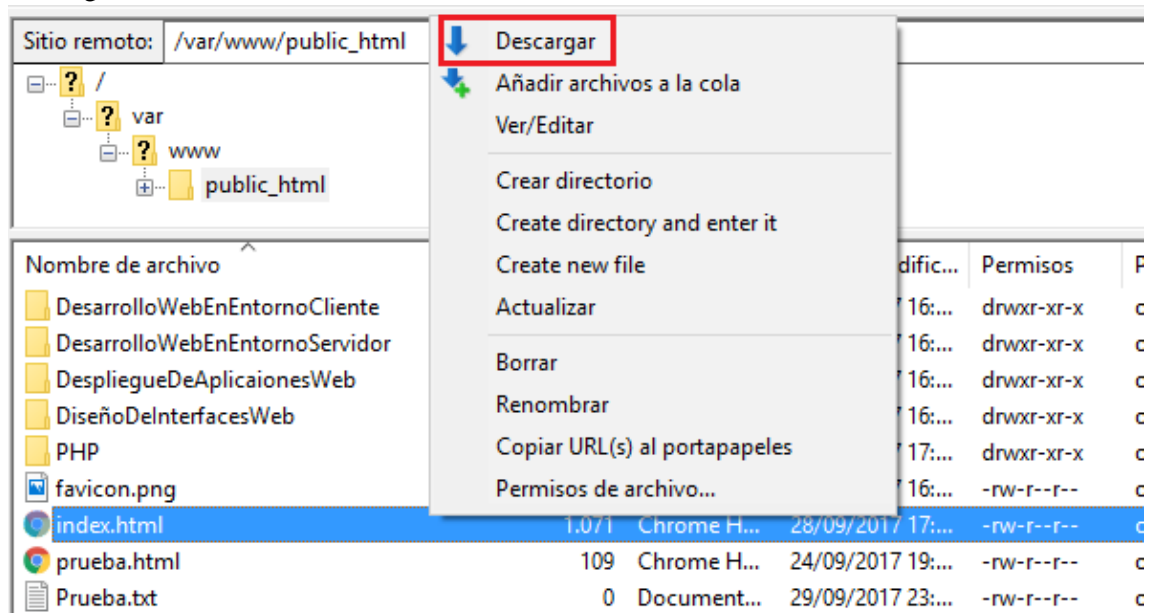
Para comprobar que lo ha pasado nos fijaremos en dos cosas



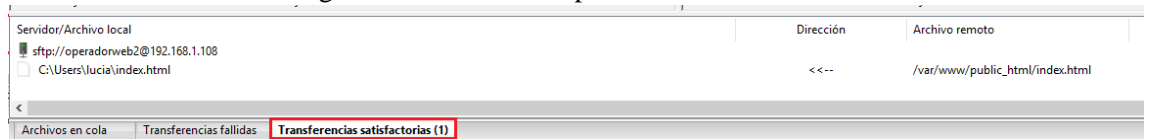
1. Que el fichero está en la carpeta donde lo hemos colocado
2. Que el fichero está en la parte de “transferencias satisfactorias”

Para descargar un fichero del servidor haremos lo siguiente

1. Seleccionamos el fichero que queremos descargar, le damos click izquierdo y le damos a descargar



2. Si el archivo se ha descargado correctamente aparecera en transferencias satisfactorias

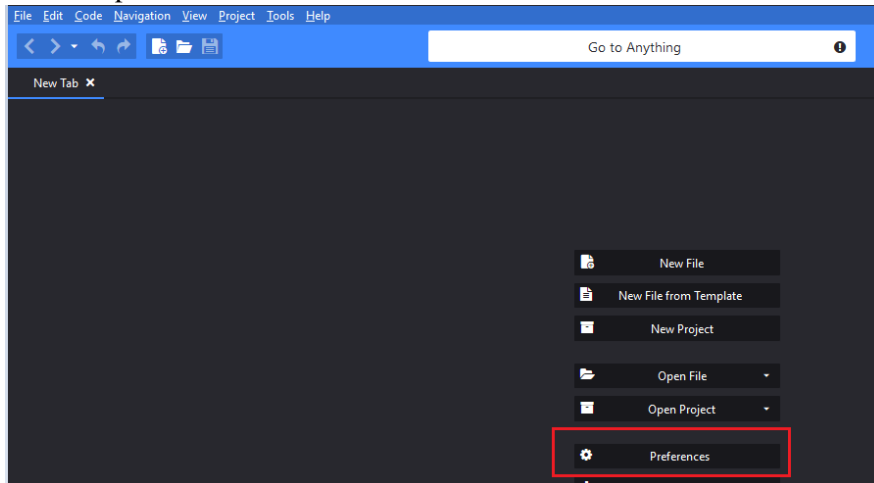


Komodo Edit 10

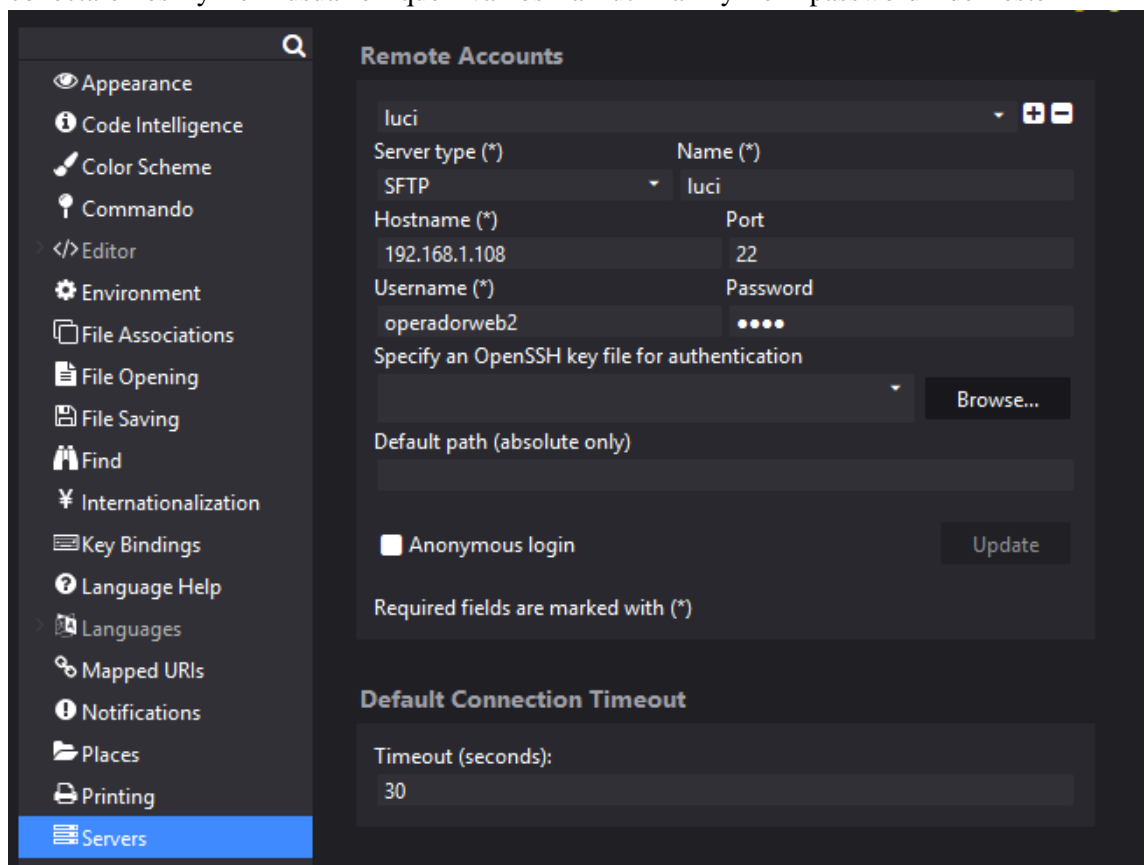
Editor de texto que utilizaremos para editar nuestros ficheros y subirlos al servidor

Para conectarnos a nuestro servidor web desde komodo haremos lo siguiente

1. Iremos a preferences

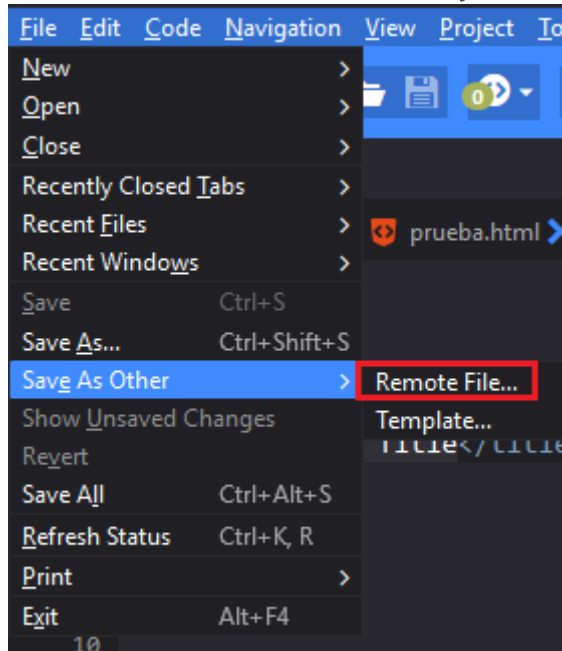


2. Iremos a la pestaña “Servers” y le pondremos un nombre a nuestra conexión para guardarla, indicaremos con que método nos conectaremos, en nuestro caso sftp, indicaremos la ip del servidor al que nos vamos a conectar, el puerto con el que nos conectaremos y el usuario que vamos a utilizar y el password de este

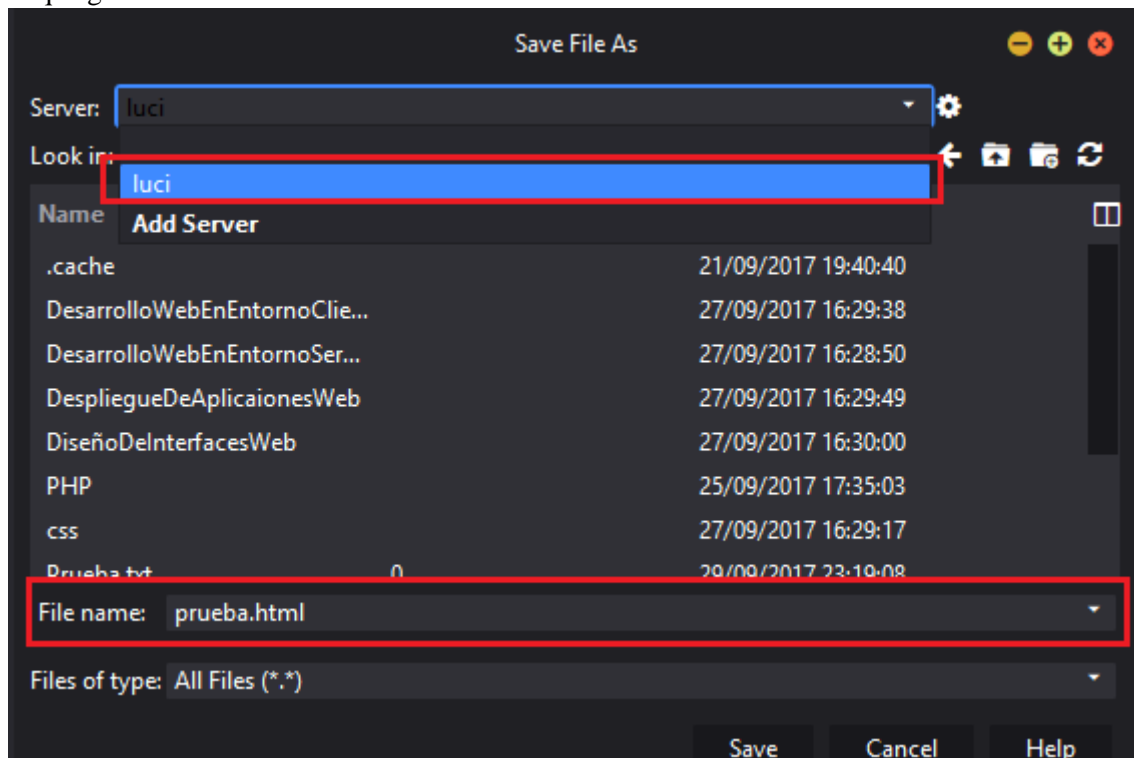


Para subir un nuevo fichero desde el komodo crearemos un fichero nuevo y después haremos lo siguiente

1. Iremos a “File>Save as other>remote file”

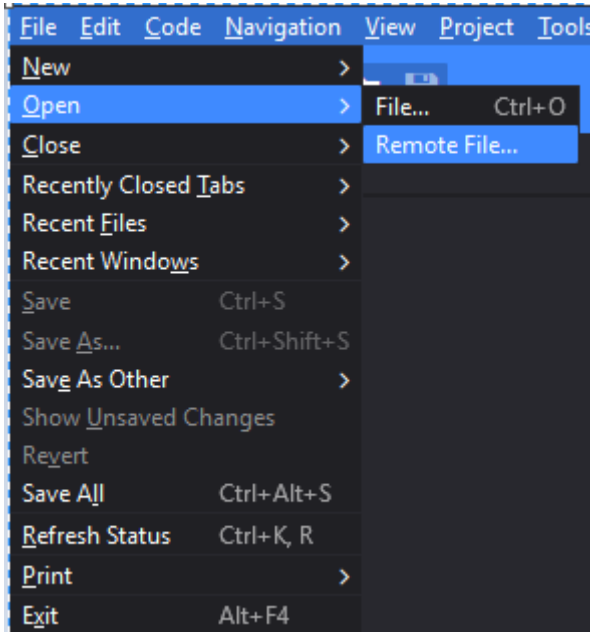


2. Después se nos abrirá una ventana donde elegiremos nuestro servidor y el nombre con el que guardaremos el fichero

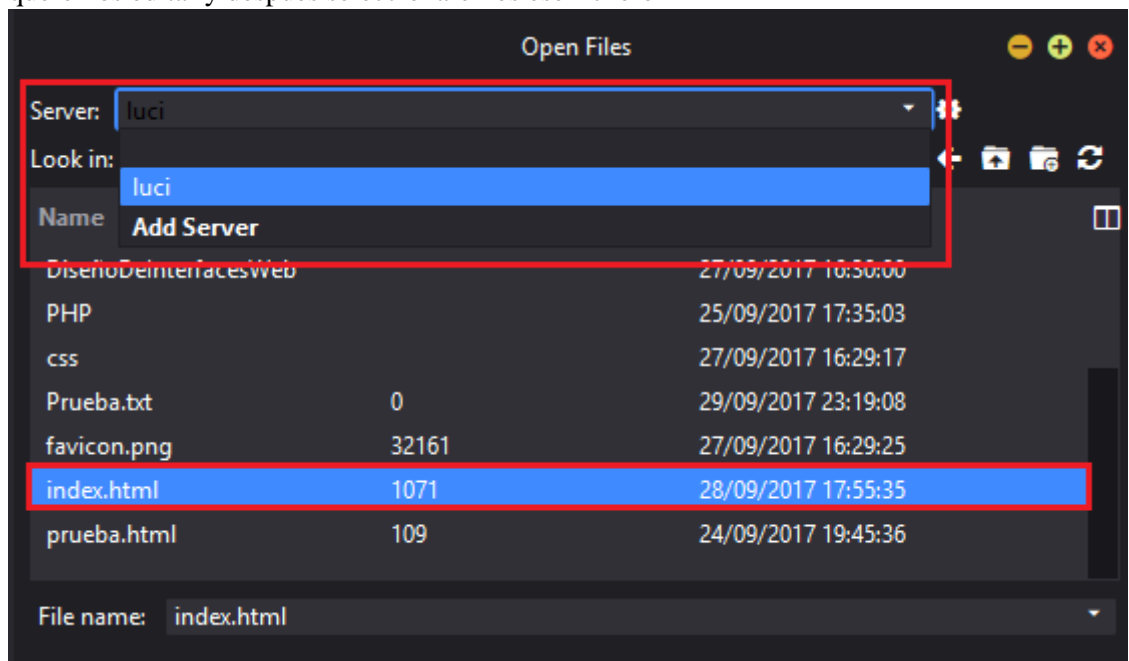


Para editar un fichero que este en el servidor haremos lo siguiente

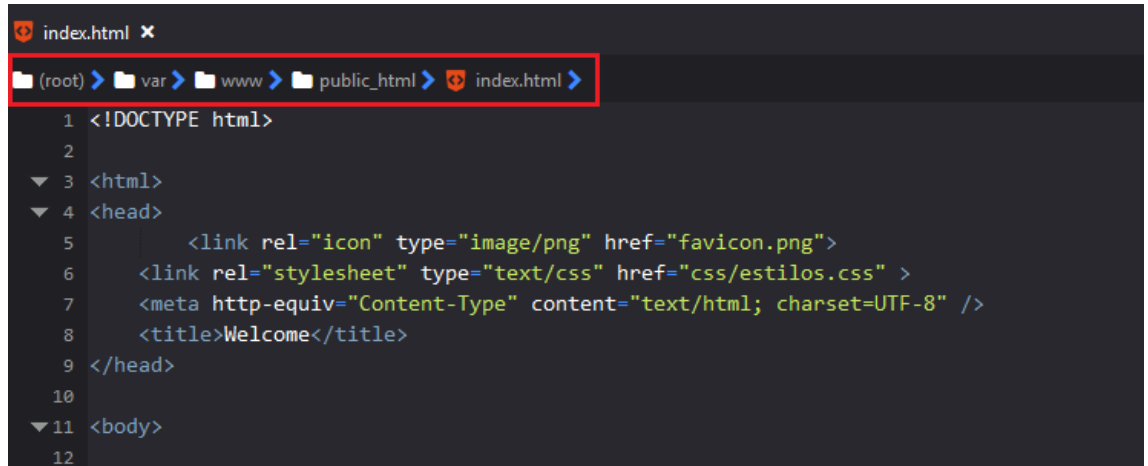
1. Iremos a “File>Open>Remote File”



2. Después se nos abrirá una ventana e indicaremos en que servidor está el fichero que queremos editar y después seleccionaremos ese fichero



3. Cuando lo seleccionemos se abra el fichero



Notepad ++

Editor de texto que podemos utilizar para editar nuestros ficheros y subirlos al servidor

EJERCICIO 2

2. Instalación y configuración del entorno de explotación personal para la asignatura de Desarrollo web del lado servidor sobre una máquina virtual xxx-USEE. Documentando la configuración elegida y los aspectos más importantes del proceso de instalación. Documentar el proceso de monitorización del correcto funcionamiento de los servicios. Creación y actualización del ProyectoDWES durante el curso.

LRA-USEE

Configuración inicial

Maquina LRA-USEE		
Sistema Operativo	Ubuntu Server 16.04.03	
Memoria	512 MB	
Usuario y Contraseña	miadmin	paso
	operadorweb2	paso
Configuración de red	Ip	192.168.3.208
	Netmask	255.255.255.0
	Gateway	192.168.3.1
	Dns	192.168.20.20
Software	Servidor Web	Apache (con modulo PHP)
	Servidor SSH	Openssh-server
	Servicio SFTP	SFTP
	SGBD	Mysql-server

Instalación del software

El software que necesitaremos instalar en esta máquina es openssh-server, apache2 y php7.0.

Para ello utilizaremos los comandos

- apt-get update → para actualizar el repositorio
- apt-get install openssh-server → para instalar el servidor ssh y poder conectarnos de forma remota
- apt-get install apache2 → para instalar apache y tener nuestro servidor web
- apt-get install php7.0 → para poder programas en php

Configuración servidor ssh

Lo primero que haremos será crear en el directorio “/var/www” donde guardaremos nuestra página, en este caso lo llamaremos “public_html”. Después le daremos los permisos para poder modificar en esta carpeta a nuestro usuario (que crearemos más adelante).

```
root@LRA-USEE:/var/www# mkdir public_html
root@LRA-USEE:/var/www# chmod 775 public_html/
```

Ahora vamos a crear nuestro usuario que se llamara “operadorweb2”, usaremos el siguiente comando:

```
sudo useradd -g www-data -d /var/www/public_html -s /bin/bash operadorweb
```

Con este comando lo que hacemos es hacer que, nuestro usuario pertenezca al grupo “www-data”, ponerle como directorio personal “/var/www/public_html” y asignarle un shell “/bin/bash”.

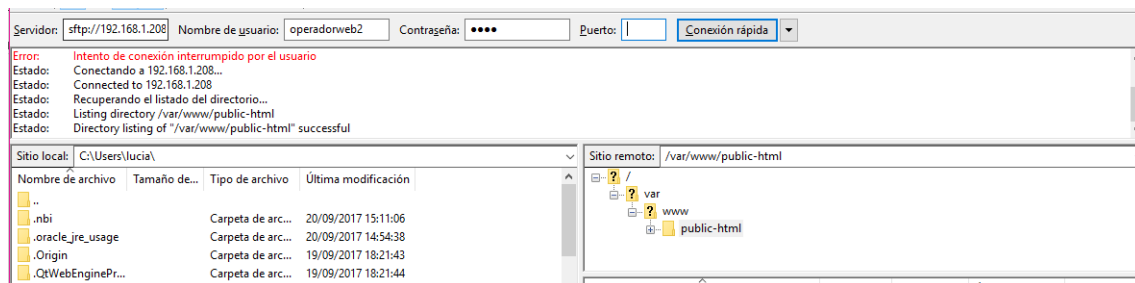
Después le asignaremos una contraseña a nuestro usuario con el comando passwd:

```
root@LRA-USEE:/var/www# passwd operadorweb2
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: password updated successfully
```

Y a continuación haremos que sea el propietario del directorio “public_html”:

```
root@LRA-USEE:/var/www# chown -R operadorweb2:www-data public_html/
```

Ahora comprobaremos desde nuestro cliente mediante el filezilla si nos podemos conectar con nuestro usuario



Configuración apache2

En el directorio “/etc/apache2/sites-available” vemos que hay un sitio que se llama “000-default.conf”

```
root@LRA-USEE:/etc/apache2/sites-available# ls
000-default.conf default-ssl.conf
```

Copiamos ese sitio con el comando

```
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/luci.conf
```

Después de copiarlo editamos el fichero “luci.conf” y donde pone “DocumentRoot” pondremos el directorio de nuestro usuario.

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/public_html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Guardamos el fichero y desactivamos el sitio por defecto con el comando a2dissite y activaremos nuestro sitio con el comando a2ensite:

```
root@LRA-USEE:/etc/apache2/sites-available# a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
    service apache2 reload
root@LRA-USEE:/etc/apache2/sites-available# a2ensite 000-default.conf
Enabling site 000-default.
To activate the new configuration, you need to run:
    service apache2 reload
```

Cuando hayamos hecho esto tendremos que reiniciar apache con el comando “service apache2 restart”. Después de esto ya tendremos configurado nuestro sitio.

Para que nuestro servidor reconozca el lenguaje php tendremos que instalar una librería de apache

apt-get install libapache2-mod-php

Después editaremos el fichero /etc/apache2/mods-enabled/dir.conf

Así estará el fichero ahora mismo

```
GNU nano 2.5.3 Archivo: /etc/apache2/mods-enabled/dir.conf

<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Lo que haremos será mover el index.php al principio para que quede de la siguiente forma

```
GNU nano 2.5.3 Archivo: /etc/apache2/mods-enabled/dir.conf

<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>

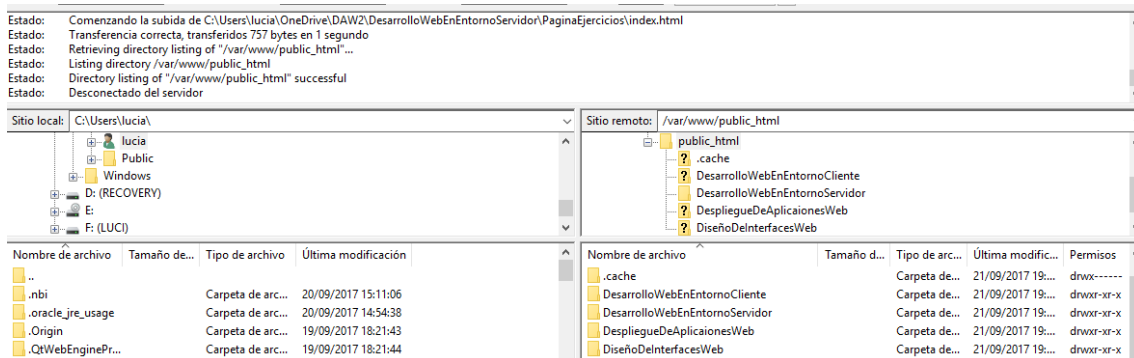
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Después reiniciaremos el servicio apache con “service apache2 restart” y nuestro servidor ya podrá leer código php

Administración del sitio

Desde nuestro cliente y con filezilla y un editor de texto empezaremos a crear nuestro sitio.

Lo primero que haremos será crear una estructura de directorios en la carpeta donde va a estar nuestro sitio, en este caso en “public_html”.



Después crearemos un inex.html y comprobaremos desde el navegador si carga nuestra página

```
2
3 <html>
4 <head>
5   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
6   <title>Welcome</title>
7 </head>
8
9 <body>
10  <h1>Desarrollo de aplicaciones web</h1>
11  <p><a href="DesarrolloWebEnEntornoServidor/ejercicios.html">Desarrollo Web En Entorno Servidor</a></p>
12  <p><a href="DesarrolloWebEnEntornoCliente/mantenimiento.html"></p>Desarrollo Web En Entorno Cliente</a></p>
13  <p><a href="DespliegueDeAplicacionesWeb/mantenimiento.html"></p>Desarrollo Web En Entorno Cliente</a></p>
14  <p><a href="DiseñoDeInterfacesWeb/mantenimiento.html"></p>Desarrollo Web En Entorno Cliente</a></p>
15  <footer><p><em><small>Lucía Rodríguez Álvarez</small></em></p></footer>
16
17
18 </body>
19 </html>
20
```