

# TEMA 2

## INSTALACIÓN, CONFIGURACIÓN Y DOCUMENTACIÓN DEL ENTORNO DE DESARROLLO Y DEL ENTORNO DE EXPLOTACIÓN

### EJERCICIOS

Autora original de la documentación: Rebeca Sánchez

a. USED – UBUNTU SERVER .....	4
1. Configuración inicial.....	4
1.1 Características de la máquina virtual .....	4
1.2 Instalación.....	6
1.3 Nombre y configuración de red .....	13
1.4 Actualizar el sistema.....	15
1.5 Cortafuegos local .....	15
2. Cuentas de administración.....	16
3. Apache .....	17
3.1 ¿Qué es y para qué sirve?.....	17
3.2 Módulos a instalar.....	17
3.3 Comandos de control del servicio.....	17
3.4 Instalación y configuración .....	18
3.5 Estructura de directorios y ficheros de configuración .....	18
3.6 Configuración de Logs de errores y acceso .....	19
3.7 Habilitar el uso de .htaccess .....	20
3.8 Modulo Alias .....	21
3.9 Modulo Rewrite .....	22
3.10 HTTPS.....	22
4.1 Módulos.....	27
4.2 Instalación de php y configuración de php.ini .....	27
5. MySQL .....	30
5.1 Instalación y configuración.....	30
5.2 PHPMyAdmin.....	32
6. XDebug .....	35
6.1 Instalación.....	35
7. Servidores DNS .....	36
7.1 Instalación del servidor DNS.....	36
7.2 Declaración de la zona directa .....	37
7.3 Declaración de la zona inversa.....	38
7.4 Comprobación de funcionamiento .....	38
8. Cuentas de desarrollo y hosting virtual.....	40
8.1 Creación de operadorweb .....	40
8.2 Creación del grupo ftpuser y enjaulamiento .....	40
b. WXED – WINDOWS X.....	43

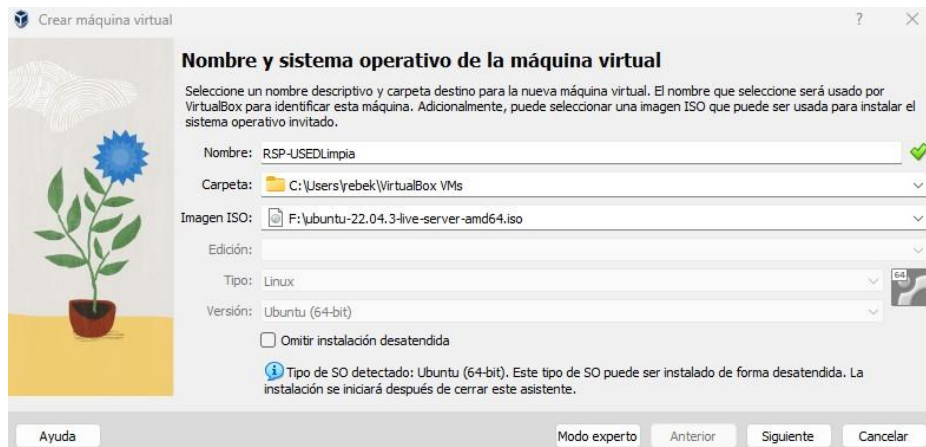
1. Nombre y configuración de red.....	43
2. Cuentas administradoras y cuenta de desarrollador .....	44
2.1 Conexión SSH desde Windows 10.....	44
2.2 Cuentas administradoras del sistema operativo .....	45
3. Navegadores.....	46
3.1 Google Chrome .....	46
3.2 Microsoft Edge .....	46
3.2 Mozilla Firefox .....	46
4. FileZilla .....	47
4.1 Versión.....	47
4.2 Conexión a un servidor.....	47
4.3 Transferencia de archivos.....	48
4.4 Gestión de sitios.....	48
5. NetBeans .....	50
5.1 Instalación y configuración inicial (plugings) .....	50
5.2 Creación de proyectos, modificación, borrado, prueba.....	50
5.3 Conexión al servidor remoto SFTP. (Almacenamiento local/almacenamiento remoto).....	53
5.4 Administración de la base de datos .....	53
5.5 Depuración - Configuración de la ejecución para la depuración .....	56
5.6 Paso a explotación .....	56
5.7 PHP Doc .....	56
5.8 CSS / JS / AJAX / XML / JSON.....	56
c. GITHUB – INTERNET .....	57
1. Creación de token desde GitHub.....	57
2. Comandos básicos de Git .....	60
3. Creación de un repositorio desde NetBeans.....	61
4. Clonar un repositorio .....	65
6. Crear una release.....	69
7.1 Generar clave SSH .....	71
7.2 Conexión SSH desde NetBeans .....	73

## a. USED – UBUNTU SERVER

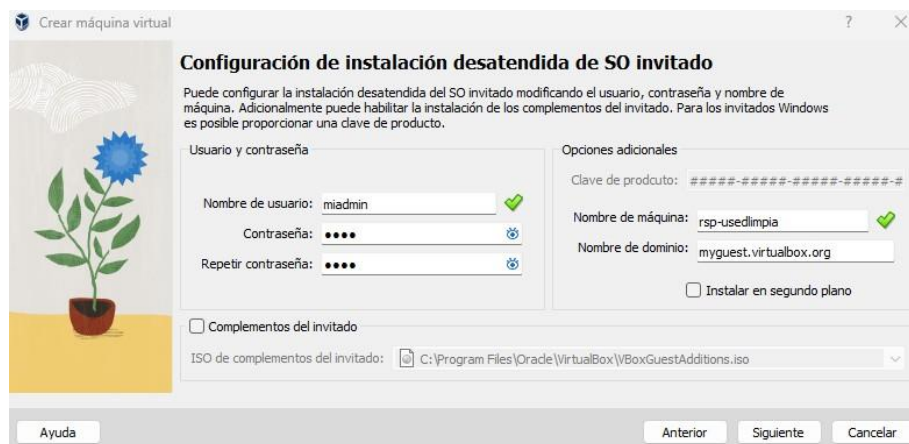
### 1. Configuración inicial

#### 1.1 Características de la máquina virtual

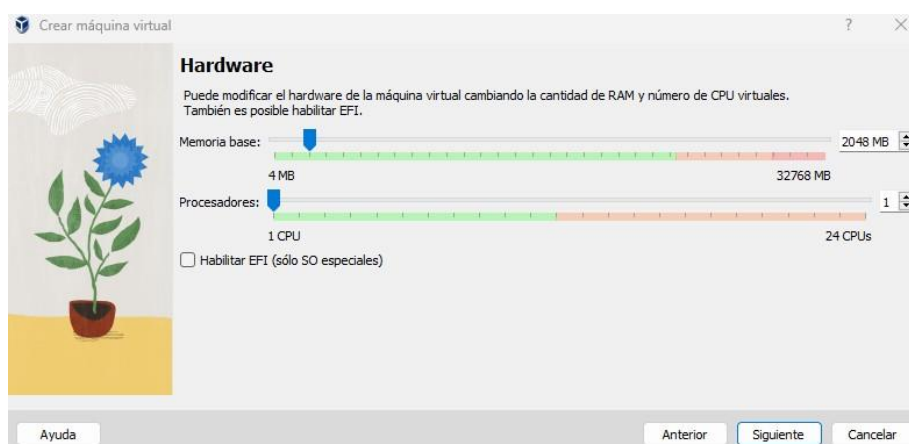
En primer lugar, abriremos el VirtualBox y crearemos una nueva máquina virtual que se llame RSP-USEDLimpia (Rebeca Sánchez Pérez - Ubuntu Server Entorno de Desarrollo Maquina limpia) con una ISO de un Ubuntu server 22.04.3



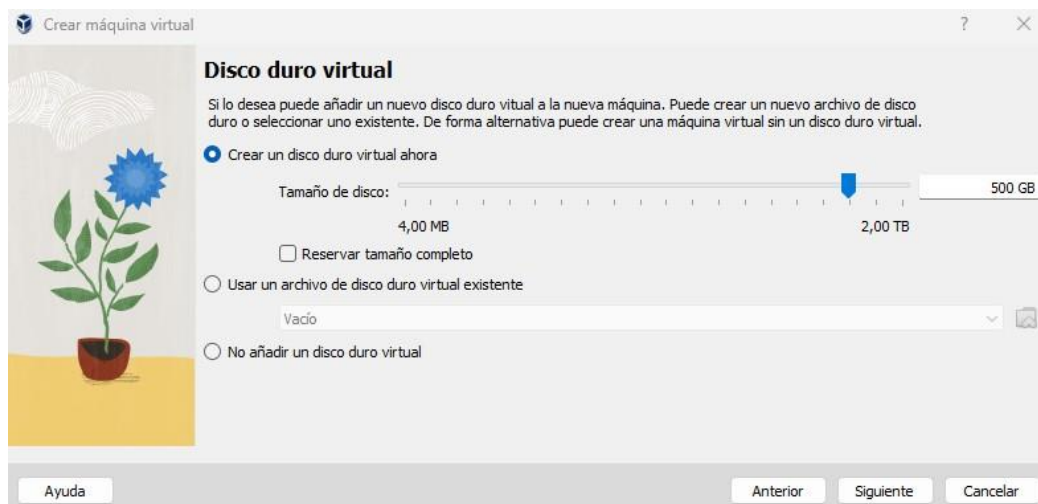
A continuación, elegiremos un nombre de usuario y una contraseña, en mi caso miadmin y paso.



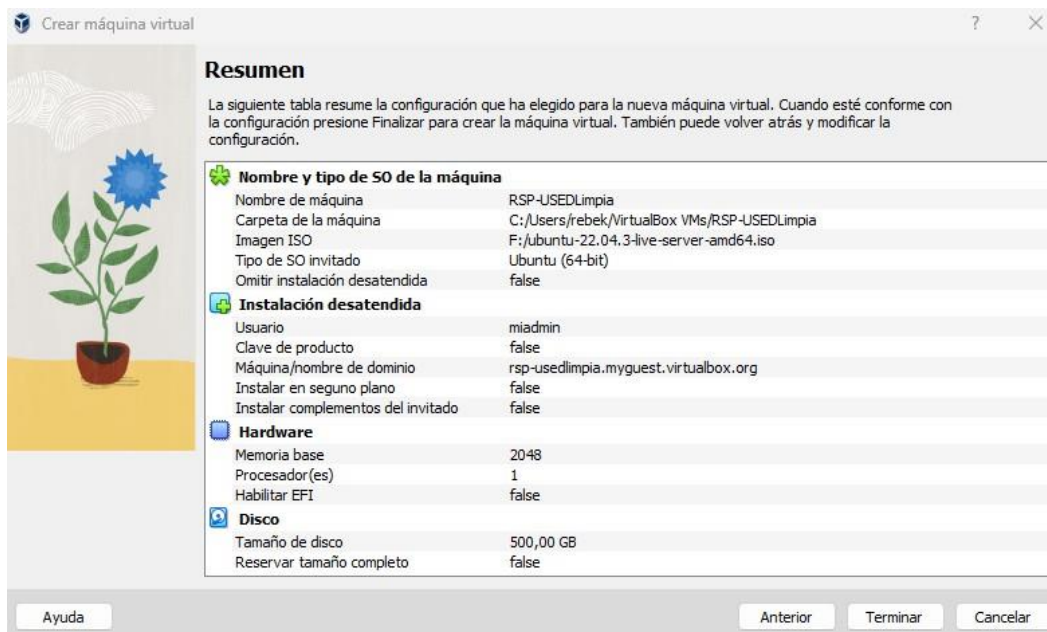
Seguimos con la configuración y reservamos 2048MB para la memoria RAM



Reservamos 500GB para la memoria del Disco Duro



Este será el resumen de la configuración de nuestra máquina de Ubuntu Server:



## 1.2 Instalación

Una vez creada la máquina, la iniciamos y comenzamos con la instalación seleccionando el idioma que queremos (español)

```
Willkommen! Bienvenue! Welcome! Добро пожаловать! Welkom! [ Help ]
Use UP, DOWN and ENTER keys to select your language.

[ Asturianu                ▾ ]
[ Bahasa Indonesia         ▾ ]
[ Català                  ▾ ]
[ Deutsch                  ▾ ]
[ English                  ▾ ]
[ English (UK)             ▾ ]
[ Español                  ▾ ]
[ Français                 ▾ ]
[ Galego                   ▾ ]
[ Deutsch                  ▾ ]
```

Seguiremos con la instalación sin actualizar a la versión que nos recomiendan

```
Actualización del instalador disponible [ Help ]
Version 23.09.1 of the installer is now available (23.08.1 is currently
running).
You can read the release notes for each version at:
https://github.com/canonical/subiquity/releases
If you choose to update, the update will be downloaded and the installation
will continue from here.

[ Actualizar al instalador nuevo ]
[ Continuar sin actualizar ]
[ Atrás ]
```

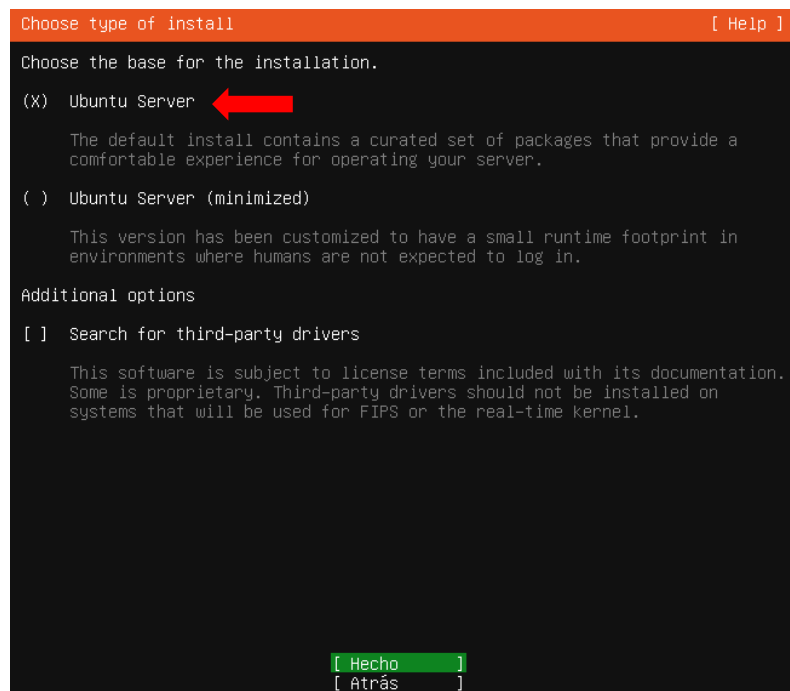
```
Configuración del teclado [ Help ]
Seleccione a continuación la disposición del teclado o elija «Identificar
teclado» para detectarla automáticamente.

Disposición: [ Spanish ▾ ]
Variant: [ Spanish ▾ ]

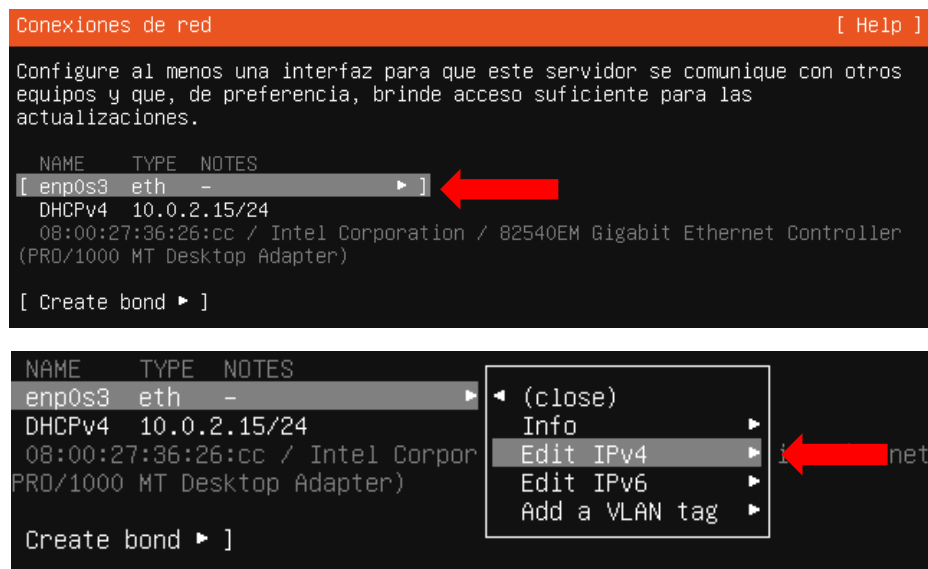
[ Identificar teclado ]

[ Hecho ]
[ Atrás ]
```

Seleccionamos el Ubuntu Server que aparece en primer lugar



A continuación, realizaremos la configuración de red. Esta configuración cambiará dependiendo de si la instalación la realizamos en casa o en clase ya que no coincidirán las direcciones IP de las máquinas anfitrionas (pertenecen a redes distintas). Configuramos la IPv4 de forma manual dándole la subred de 192.168.1.0/24 (en clase le daríamos 192.168.3.0/24), la dirección IP de 192.168.1.204 (en clase sería la 192.168.3.204) y la puerta de enlace de 192.168.1.1 (y en clase sería la 192.168.3.1). En ambos casos utilizaremos los servidores de DNS de 8.8.8.8 de Google.



Edit enp0s3 IPv4 configuration

Método de IPv4: Automático (DCHP)    
Manual  
Desactivado

[ Cancelar ]

---

Edit enp0s3 IPv4 configuration

Método de IPv4: [ Manual ▼ ]

Subred: 192.168.1.0/24

Dirección: 192.168.1.204

Puerta de enlace: 192.168.1.1

Servidores de nombres: 8.8.8.8  
IP addresses, comma separated

Dominios de búsqueda:   
Domains, comma separated

[ Guardar ]  
[ Cancelar ]

---

Configure al menos una interfaz para que este servidor se comunique con otros equipos y que, de preferencia, brinde acceso suficiente para las actualizaciones.

NAME	TYPE	NOTES
enp0s3	eth	
[ Create bond ► ]		
static 192.168.1.204/24		
08:00:27:38:28:CC / Intel Corporation / 82540EM Gigabit Ethernet Controller (PRO/1000 MT Desktop Adapter)		
[ Create bond ► ]		

En la siguiente pantalla nos pedirán una dirección proxy, simplemente tenemos que darle a “Hecho”

Configure proxy [ Help ]

If this system requires a proxy to connect to the internet, enter its details here.

Proxy address:

If you need to use a HTTP proxy to access the outside world, enter the proxy information here. Otherwise, leave this blank.

The proxy information should be given in the standard form of "http://[user][:pass]@host[:port]/".

[ Hecho ]  
[ Atrás ]



Ahora esperamos a que se instalen las nuevas configuraciones y le damos a “Hecho”

```
Configure Ubuntu archive mirror [ Help ]

If you use an alternative mirror for Ubuntu, enter its details here.

Mirror address: http://es.archive.ubuntu.com/ubuntu
You may provide an archive mirror that will be used instead of
the default.

This mirror location does not seem to work. The output below may help explain
the problem. You can try again once the issue has been fixed (common problems
are network issues or the system clock being wrong).

[ Try again now ]

Ign:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease
Ign:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease
Ign:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease
Err:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
      Fallo temporal al resolver «es.archive.ubuntu.com»
Err:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
      Fallo temporal al resolver «es.archive.ubuntu.com»

[ Hecho ]
[ Atrás ]
```

```
Guided storage configuration [ Help ]

Configure a guided storage layout, or create a custom one:

(X) Use an entire disk

   [ VBOX_HARDDISK_VBc6d02768-4cd20ceb local disk 500.000G ▼ ]

[X] Set up this disk as an LVM group

   [ ] Encrypt the LVM group with LUKS

      Passphrase:

      Confirm passphrase:

( ) Custom storage layout

[ Hecho ]
[ Atrás ]
```

En la siguiente pantalla nos aparece un listado de las particiones que tiene nuestro disco duro. Tendremos que cambiarlas para que cumplan los requisitos de la máquina, para ello seleccionamos “Restablecer”

```
Storage configuration [ Help ]

RESUMEN DEL SISTEMA DE ARCHIVOS

PUNTO DE MONTAJE  TAMAÑO  TIPO  TIPO DE DISPOSITIVO
[ /               100.000G  new ext4  new LVM logical volume ► ]
[ /boot          2.000G  new ext4  new partition of disco local ► ]

DISPOSITIVOS DISPONIBLES

DISPOSITIVO  TIPO  TAMAÑO
[ ubuntu-vg (new)  LVM volume group  497.996G ► ]
espacio disponible  397.996G ►

[ Create software RAID (md) ► ]
[ Crear grupo de volúmenes (LVM) ► ]

DISPOSITIVOS UTILIZADOS

DISPOSITIVO  TIPO  TAMAÑO
[ ubuntu-vg (new)  LVM volume group  497.996G ► ]
ubuntu-lv      new, to be formatted as ext4, mounted at 100.000G ► ]
/              /

[ VBOX_HARDDISK_VBc6d02768-4cd20ceb  disco local  500.000G ► ]
partition 1  new, BIOS grub spacer  1.000M ► ]
partition 2  new, to be formatted as ext4, mounted at 2.000G ► ]
/boot
partition 3  new, PV of LVM volume group ubuntu-vg  497.997G ► ]

[ Hecho ]
[ Restablecer ]
[ Atrás ]
```

Una vez pulsado el botón, aparece esta pantalla en la que tendremos que formatear las particiones del disco de forma que el Sistema (/) ocupe 150Gb y los datos (/var) el resto del disco, es decir, 349.997Gb.

```
Storage configuration [ Help ]

To continue you need to: Mount a filesystem at /
                        Select a boot disk

RESUMEN DEL SISTEMA DE ARCHIVOS

No se montó ningún disco o partición.

DISPOSITIVOS DISPONIBLES

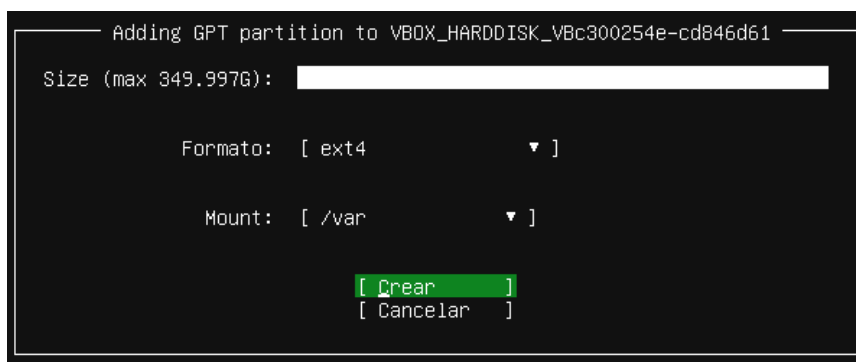
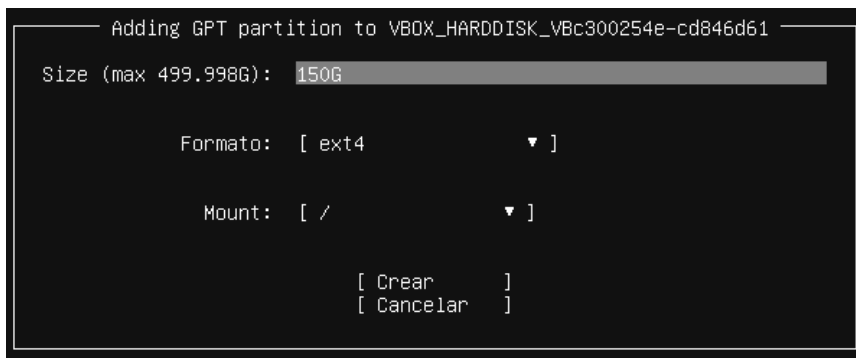
DISPOSITIVO  TIPO  TAMAÑO
[ VBOX_HARDDISK_VBc6d02768-4cd20ceb  disco local  500.000G ► ]
espacio disponible  499.998G ►

[ Create software RAID (md) ► ]
[ Crear grupo de volúmenes (LVM) ► ]

DISPOSITIVOS UTILIZADOS

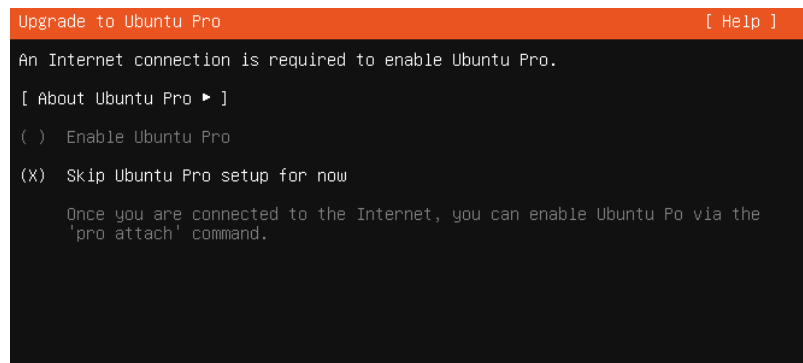
No used devices

[ Hecho ]
[ Restablecer ]
[ Atrás ]
```

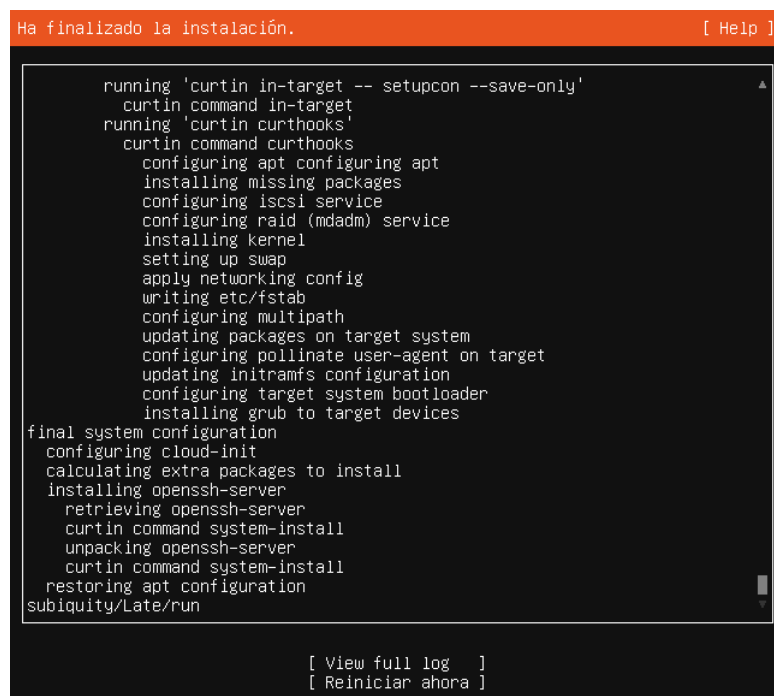
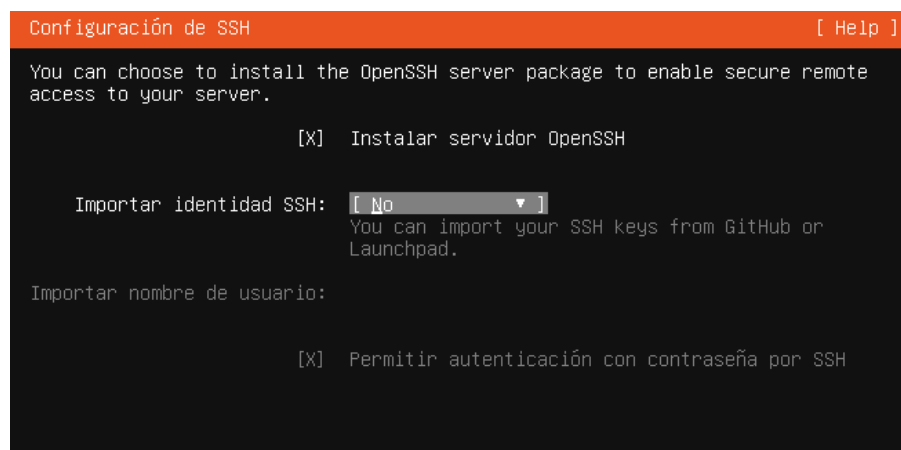


A continuación, elegiremos el nombre de la máquina, el usuario y la contraseña





Instalamos el servidor OpenSSH, esperamos a que se complete la instalación y reiniciamos



### 1.3 Nombre y configuración de red

Para cambiar el nombre del equipo, abrimos el nuevo Ubuntu Server, usamos el comando “sudo hostnamectl set-hostname rsp-used” y modificamos el fichero host con “sudo nano /etc/hosts” sustituyendo “rsp-usedlimpia” por “rsp-used”

```
miadmin@rsp-usedlimpia:~$ sudo hostnamectl set-hostname rsp-used
[sudo] password for miadmin:
miadmin@rsp-usedlimpia:~$ sudo nano /etc/hosts_
```

```
127.0.0.1 localhost
127.0.1.1 rsp-usedlimpia_
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Una vez guardados los cambios, reiniciamos con “reboot” y escribimos “hostname” para confirmar que se han realizado los cambios

```
miadmin@rsp-usedlimpia:~$ reboot_
```

```
miadmin@rsp-used:~$ hostname
rsp-used
```

Si queremos acceder a la configuración de la red, nos dirigimos a la ruta de los archivos de configuración escribiendo “cd /etc/netplan”. Para modificar el archivo de red usaremos en comando “sudo nano 00-installer-config.yaml”

```
miadmin@rsp-usedlimpia:~$ cd /etc/netplan/
miadmin@rsp-usedlimpia:/etc/netplan$ sudo nano 00-installer-config.yaml _
```

A continuación, aparecerá esta pantalla con el contenido de ese archivo. Si queremos cambiar la IP de nuestra maquina basta con editar la que se muestra en el archivo (es muy importante respetar la tabulación y el espaciado de este archivo). Para aplicar los cambios usamos el comando “sudo netplan apply” y se cambiará la dirección IP por la nueva.

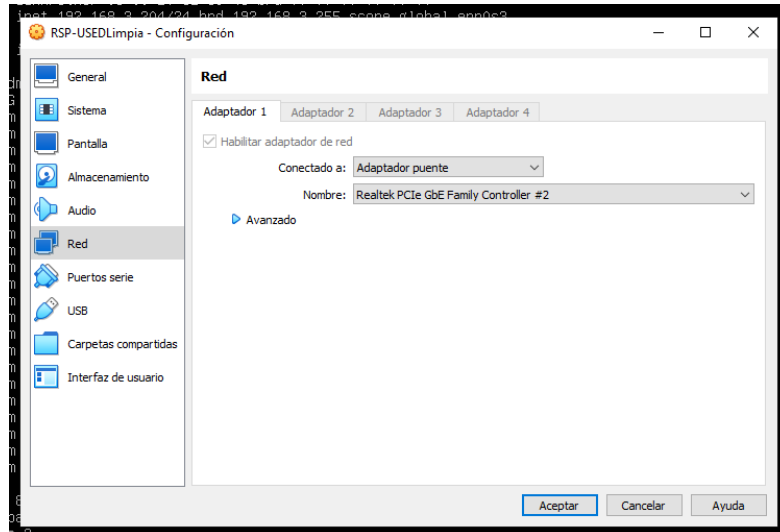
```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      addresses:
        - 192.168.1.204/24
      nameservers:
        addresses:
          - 8.8.8.8
          - 8.8.8.8
        search: []
      routes:
        - to: default
          via: 192.168.1.1
      version: 2
```

```
miadmin@rsp-usedlimpia:/etc/netplan$ sudo netplan apply
```

Iniciamos sesión con el usuario “miadmin” y comprobamos que tenemos conexión a internet con el comando “ping google.es”. En el caso de que no se conecte tendremos que cambiar

la configuración de red de la maquina desde VirtualBox a adaptador puente y reiniciar la maquina con el comando “reboot”

```
miadmin@rsp-usedlimpia:~$ ping google.es  
ping: google.es: Temporary failure in name resolution
```



## 1.4 Actualizar el sistema

Cuando tengamos conexión a internet escribimos “sudo apt update” y “sudo apt upgrade” para actualizar los repositorios y librerías.

Para configurar la fecha y la hora antes debemos conocer cuál es la fecha y hora activa en nuestro server, para ello escribimos “date” y para cambiar a la franja horaria de Madrid usaremos el comando “sudo timedatectl set-timezone Europe/Madrid”

```
miadmin@rsp-usedlimpia:~$ date
mar 03 oct 2023 09:01:54 UTC
```

```
miadmin@rsp-usedlimpia:~$ sudo timedatectl set-timezone Europe/Madrid
[sudo] password for miadmin:
miadmin@rsp-usedlimpia:~$ date
mar 03 oct 2023 11:07:55 CEST
miadmin@rsp-usedlimpia:~$ _
```

## 1.5 Cortafuegos local

En primer lugar, tenemos que saber cuál es el estado del cortafuegos con el comando “sudo ufw status”, si esta inactivo lo activamos con “systemctl status ufw” y “sudo ufw enable”

```
miadmin@rsp-usedlimpia:~$ sudo ufw status
Status: inactive
```

```
miadmin@rsp-usedlimpia:~$ systemctl status ufw
• ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset: enabled)
   Active: active (exited) since Tue 2023-10-03 08:17:01 UTC; 17min ago
     Docs: man:ufw(8)
    Main PID: 555 (code=exited, status=0/SUCCESS)
      CPU: 1ms

oct 03 08:17:01 rsp-usedlimpia systemd[1]: Starting Uncomplicated firewall...
oct 03 08:17:01 rsp-usedlimpia systemd[1]: Finished Uncomplicated firewall.
miadmin@rsp-usedlimpia:~$ _
```

```
miadmin@rsp-usedlimpia:~$ sudo ufw enable
Firewall is active and enabled on system startup
miadmin@rsp-usedlimpia:~$
```

Cuando tenemos el cortafuegos activo abrimos el puerto 22 con “sudo ufw allow 22” y comprobamos que está abierto con “sudo ufw status”

```
miadmin@rsp-usedlimpia:~$ sudo ufw allow 22
Rule added
Rule added (v6)
miadmin@rsp-usedlimpia:~$ sudo ufw status
Status: active

To Action From
--
22 ALLOW Anywhere
22 (v6) ALLOW Anywhere (v6)

miadmin@rsp-usedlimpia:~$ _
```

## 2. Cuentas de administración

Para la creación de un usuario nuevo usamos el comando `"sudo adduser miadmin2"` y rellenamos los campos que queramos por ejemplo la contraseña

```
miadmin@rsp-used:~$ sudo adduser miadmin2
[sudo] password for miadmin:
Adding user `miadmin2' ...
Adding new group `miadmin2' (1001) ...
Adding new user `miadmin2' (1001) with group `miadmin2' ...
Creating home directory `/home/miadmin2' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for miadmin2
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n]
miadmin@rsp-used:~$ _
```

Si queremos añadir el nuevo usuario al grupo de super usuarios (es decir, el grupo de los admins o root) escribiremos el comando `"sudo usermod -aG sudo miadmin2"`

```
miadmin@rsp-used:~$ sudo usermod -aG sudo miadmin2
miadmin@rsp-used:~$ _
```

Para comprobar que hemos creado el usuario correctamente y que pertenece al grupo que se le ha especificado usamos el comando `"grep "miadmin2" /etc/group"`

```
miadmin@rsp-used:~$ grep "miadmin2" /etc/group
sudo:x:27:miadmin,miadmin2
miadmin2:x:1001:
```



### 3. Apache

#### 3.1 ¿Qué es y para qué sirve?

Apache es un servidor web HTTP de código abierto que se usa para poder desarrollar y acceder a aplicaciones web con modelos cliente/servidor a través de los protocolos HTTP y HTTPS. La funcionalidad principal de este servicio web es abastecer a los usuarios de todos los ficheros necesarios para visualizar una web. Las solicitudes de los usuarios se hacen normalmente mediante un navegador (Chrome, Firefox, Safari, etc.).

#### 3.2 Módulos a instalar

Para listar los módulos activos de apache escribiremos “apache2ctl -M”. A continuación, se muestra un listado de todos los módulos de Apache que se han instalado en nuestro servidor web:

- Status
- Alias
- Rewrite
- Php

```
miadmin@rsp-used:~$ sudo apache2ctl -M
Loaded Modules:
  core_module (static)
  so_module (static)
  watchdog_module (static)
  http_module (static)
  log_config_module (static)
  logio_module (static)
  version_module (static)
  unixd_module (static)
  access_compat_module (shared)
  alias_module (shared)
  auth_basic_module (shared)
  authn_core_module (shared)
  authn_file_module (shared)
  authz_core_module (shared)
  authz_host_module (shared)
  authz_user_module (shared)
  autoindex_module (shared)
  deflate_module (shared)
  dir_module (shared)
  env_module (shared)
  filter_module (shared)
  mime_module (shared)
  mpm_prefork_module (shared)
  negotiation_module (shared)
  php_module (shared)
  reqtimeout_module (shared)
  setenvif_module (shared)
  status_module (shared)
```

#### 3.3 Comandos de control del servicio

- sudo apache2ctl -S: muestra los sitios activos
- sudo apache2ctl -M: muestra los módulos activos

- `sudo apache2ctl {start|stop|restart|gracefullconfigtest}`: en orden arranca, para, reinicia, reinicia de forma breve y valida los ficheros de configuración (este último comando, te muestra los posibles errores o warning que has podido cometer en algún fichero de configuración)
- `a2ensite nombre-del-sitio`: activa el sitio virtual que se le especifica
- `a2dissite nombre-del-sitio`: desactiva el sitio virtual que se le especifica
- `a2enmod nombre-del-modulo`: activa el módulo que se le especifica
- `a2dismod nombre-del-modulo`: desactiva el módulo que se le especifica
- `a2enconfmi nombre-del-archivo`: habilita archivos de configuración

### 3.4 Instalación y configuración

En la maquina RSP-USED instalaremos el servicio Apache2, pero antes de eso tenemos que actualizar los repositorios de Linux con el comando “`sudo apt-get update`”.

Para instalar Apache escribiremos “`sudo apt-get install apache2`”:

```
miadmin@rsp-used:~$ sudo apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
apache2 ya está en su versión más reciente (2.4.52-1ubuntu4.6).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 5 no actualizados.
```

Una vez instalado Apache, tenemos que permitir el acceso a Apache al puerto 80 del cortafuegos de nuestro server mediante el comando “`sudo ufw allow apache`” y para comprobarlo escribimos “`sudo ufw status`”

```
miadmin@rsp-used:~$ sudo ufw allow apache
Rule added
Rule added (v6)
```

```
miadmin@rsp-used:~$ sudo ufw status
Status: active

To Action From
--
22 ALLOW Anywhere
Apache ← ALLOW Anywhere
22 (v6) ALLOW Anywhere (v6)
Apache (v6) ← ALLOW Anywhere (v6)
```

### 3.5 Estructura de directorios y ficheros de configuración

Para comprobar cuál es la estructura de directorios que tiene por defecto apache, basta con editar el fichero de configuración con “`sudo nano /etc/apache2/sites-available/000-default.conf`” y leer las líneas comentadas

```
# It is split into several files forming the configur
# below, all located in the /etc/apache2/ directory:
#
#   /etc/apache2/
#   |-- apache2.conf
#   |   \-- ports.conf
#   |-- mods-enabled
#   |       |-- *.load
#   |       |-- *.conf
#   |-- conf-enabled
#   |       |-- *.conf
#   |-- sites-enabled
#   |       |-- *.conf
#
```

- apache2.conf: es el archivo de configuración principal. Reúne las piezas al incluir todos los archivos de configuración restantes al iniciar el servidor web.
- ports.conf: siempre se incluye en el archivo de configuración principal. Determina los puertos de escucha para las conexiones entrantes que se pueden personalizar en cualquier momento.
- Los archivos de configuración en los directorios mods-enabled/, conf-enabled/ y sites-enabled/ contienen fragmentos de configuración particulares que administran módulos, fragmentos de configuración global o configuraciones de host virtual, respectivamente.
- Los archivos de configuración en los directorios mods-available/, conf-available/ y sites-available / contienen fragmentos de configuración particulares que administran módulos disponibles, fragmentos de configuración global disponibles o configuraciones de host virtual disponibles, respectivamente.

### 3.6 Configuración de Logs de errores y acceso

Modificamos con “sudo nano /etc/apache2/sites-available/000-default.conf” el fichero de configuración para grabar los registros logs en un directorio específico. Para ello escribimos estas 2 líneas:

```
# It is also possible to configure the LogLevel for particu
# modules, e.g.
#LogLevel info ssl:warn

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

# For most configuration files from conf-available/, which
```

- La línea ErrorLog /var/www/html/log/error.log sirve para generar un archivo (error.log) con registros logs de los errores en la ruta que se especifica (en /var/www/html/log).
- La línea CustomLog /var/www/html/log/access.log combined sirve para generar una serie de registros de solicitudes o acceso a archivos en un servidor web.

Después de esto creamos el directorio para almacenar los logs que hemos especificado, para ello usamos “sudo mkdir /var/www/html/log” y reiniciamos el servicio de apache.

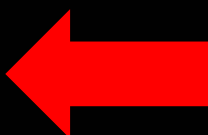
### 3.7 Habilitar el uso de .htaccess

Configuraremos el archivo de configuración de apache con “sudo nano /etc/apache2/apache2.conf” y escribimos lo siguiente:

```
AllowOverride None
Require all granted
</Directory>

<Directory /var/www/>
    Options -Indexes +FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

#<Directory /srv/>
#     Options Indexes FollowSymLinks
```



- Options -Indexes +FollowSymLinks: Define las opciones para el directorio. Significa que se desactiva la función de mostrar automáticamente el listado de archivos si no hay un archivo de índice en el directorio (-Indexes) y se permite el seguimiento de enlaces simbólicos (+FollowSymLinks)
- AllowOverride All: Permite la anulación de configuraciones mediante archivos .htaccess en ese directorio. Esto permite a los usuarios locales del directorio personalizar algunas configuraciones para sus propios fines.
- Require all granted: Esto establece que se permite el acceso a este directorio a cualquier usuario o cliente que intente acceder

Ahora creamos el archivo .htaccess escribiendo con “sudo nano /var/www/html/.htaccess” y escribimos:

```
GNU nano 6.2
DirectoryIndex index.php index.html
```

- La directiva “DirectoryIndex index.php index.html” indica el nombre del archivo que se debe servir como página de inicio cuando se accede a un directorio en el servidor web, tiene preferencia el primer archivo (index.php) y en el caso de que no se encuentre el fichero, se buscara el segundo nombre (index.html) y así consecutivamente.

Para validar los archivos de configuración de Apache escribimos el comando “sudo apache2ctl configtest” y en nuestro caso aparece un warning que tenemos que subsanar modificando el fichero de configuración del servicio

```
miadmin@rsp-used:~$ sudo apache2ctl configtest
[sudo] password for miadmin:
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0
.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
```

Usamos “sudo nano /etc/apache2/apache2.conf” y al final del archivo escribimos “ServerName rsp-used” y reiniciamos el servicio

```
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
ServerName rsp-used
```

- La directiva “ServerName rsp-used” se utiliza para especificar el nombre del host o el dominio al que el servidor web debe responder (en nuestro caso es el mismo nombre que el de la máquina virtual del Ubuntu server: rsp-used)

Para instalar un navegador de texto usamos el comando “sudo apt install lynx” y para activar el módulo de apache status usamos “sudo a2enmod status” y reiniciamos apache2

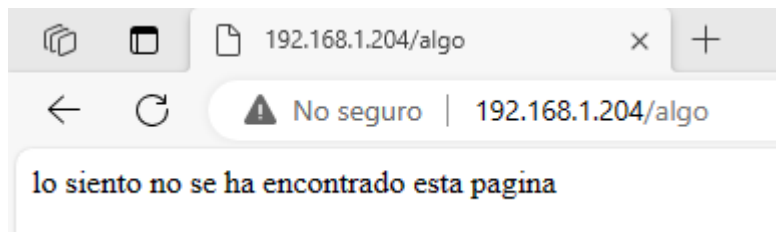
```
miadmin@rsp-used:~$ sudo a2enmod status
Module status already enabled
```

- El módulo mod\_status sirve para saber el rendimiento de nuestro servidor Apache de forma remota a través de una página HTML que accedemos desde el navegador

Para ver los informes escribimos en la terminal “apache2ctl fullstatus” y para la personalización de errores podemos editar el fichero .htaccess (con “sudo nano /var/www/html/.htaccess”)

```
miadmin@rsp-used:~$ sudo nano /var/www/html/.htaccess
DirectoryIndex index.php index.html
ErrorDocument 404 "lo siento no se ha encontrado esta pagina"
```

- La directiva “ErrorDocument 404 “lo siento...”” sirve para personalizar los mensajes de error. En primer lugar, se especifica el código de error HTTP que queremos personalizar (404) y a continuación escribimos el mensaje de error que queremos que aparezca en el navegador entre comillas (“lo siento, esta página no está disponible”)



### 3.8 Modulo Alias

Instalaremos un módulo de apache llamado Alias que nos permite el acceso a directorios superiores a los que puede acceder un usuario (directorios virtuales). En primer lugar, comprobamos que el módulo Alias esté activo con “ls /etc/apache2/mods-enabled/ | grep alias”.

```
miadmin@rsp-used:~$ ls /etc/apache2/mods-enabled/ | grep alias
alias.conf
alias.load
```

Si está activo, modificamos el fichero de configuración del sitio virtual (“sudo nano /etc/apache2/sites-enabled/000-default.conf”) escribiendo las siguientes líneas:

```
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
Alias /documentos /home/miadmin/doc
<Directory /home/miadmin/doc>
    Options +FollowSymLinks +Indexes
    AllowOverride None
    Require all granted
</Directory>
```

- La directiva “Alias /documentos /home/miadmin/doc” vincula una carpeta existente en nuestro proyecto (en nuestro caso /documentos) con un directorio virtual o externo al proyecto especificando su ruta absoluta (/home/miadmin/doc)

Para hacer una prueba creamos una carpeta *doc* en */home/miadmin* y dentro de esta creamos 2 archivos de texto *file1.txt* y *file2.txt*. Usamos el comando “sudo nano /etc/apache2/mods-enabled/alias.conf” para modificar el archivo de configuración de alias y escribimos las siguientes directivas y usamos “sudo service apache2 restart” para reiniciar Apache

```
</Directory>
Alias /documentos /home/miadmin/doc
<Directory /home/miadmin/doc>
    Options FollowSymLinks Indexes
    AllowOverride None
    Require all granted
</Directory> -
```

### 3.9 Modulo Rewrite

El módulo Rewrite de Apache nos permite crear direcciones URL alternativas a las generadas por nuestro sitio web. Habilitamos el módulo con “sudo a2enmod rewrite” y reiniciamos el servicio

```
miadmin@rsp-used:~$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
systemctl restart apache2
```

Como ejemplo, editaremos un archivo *.htaccess* y le escribiremos la siguiente línea:

```
Redirect 301 /educa https://www.educa.jcyl.es/es
```

- Esta directiva redirige la primera ruta (/educa) automáticamente a la segunda (https://www.educa.es) como un enlace

A continuación, escribimos en el navegador la URL de nuestro proyecto seguido de */educa* y cuando carguemos la página, nos redimirá automáticamente a la página web de educa.

### 3.10 HTTPS

Para trabajar con un protocolo seguro de transferencia de hipertexto (HTTPS), deberemos configurar el protocolo SSL/TLS. Con esto lo que hacemos es generar una clave pública y un certificado en nuestro servidor.

En primer lugar, generamos una clave privada en nuestro servidor con el comando “openssl genrsa 2048 > claveprivada.key”

```
miadmin@rsp-used:~$ pwd
/home/miadmin
miadmin@rsp-used:~$ openssl genrsa 2048 > claveprivada.key
miadmin@rsp-used:~$
```

A continuación, haremos una solicitud del certificado a partir de la clave privada que hemos creado con el comando `"openssl req -new -key claveprivada.key > certificado.csr"` especificando el nombre del archivo con extensión csr (esta es la extensión del certificado) que queremos darle. A medida que se crea nos va pidiendo datos que tendremos que facilitarle como el nombre del país, la provincia, el municipio, el dominio...

```
miadmin@rsp-used:~$ openssl req -new -key claveprivada.key > certificado.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Zamora
Locality Name (eg, city) []:Benavente
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IES Los Sauces
Organizational Unit Name (eg, section) []:Informatica
Common Name (e.g. server FQDN or YOUR name) []:daw204.rebeca.local
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
miadmin@rsp-used:~$
```

Se firma el certificado con una autofirma que genera un archivo con la extensión ".crt" con el comando `"openssl x509 -req -days 365 -in certificado.csr -signkey claveprivada.key -out certificado.crt"` y comprobamos con un `ls` que se han generado los 3 ficheros correctamente

```
miadmin@rsp-used:~$ openssl x509 -req -days 365 -in certificado.csr -signkey claveprivada.key -out certificado.crt
Certificate request self-signature ok
subject=C = ES, ST = Zamora, L = Benavente, O = IES Los Sauces, OU = Informatica, CN = daw204.rebeca.local
miadmin@rsp-used:~$ ls
204DAWProyectoDAW certificado.crt certificado.csr claveprivada.key doc
```

A continuación, pasaremos a configurar apache para poder usar la clave privada y el certificado que hemos generado. Para ello, en primer lugar, activamos el módulo SSL con el comando `"sudo a2enmod ssl"`

```
miadmin@rsp-used:~$ sudo a2enmod ssl
[sudo] password for miadmin:
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
systemctl restart apache2
miadmin@rsp-used:~$
```

Movemos la clave privada al directorio `/etc/ssl/private` con `"sudo mv claveprivada.key /etc/ssl/private"`, cambiamos el propietario para que pertenezca al grupo `root:ssl-cert` con `"sudo chown root:ssl-cert /etc/ssl/private/claveprivada.key"`, cambiamos los permisos para que tenga 640 con `"sudo chmod 640 /etc/ssl/private/claveprivada.key"` y comprobamos listando el contenido de ese directorio



```
miadmin@rsp-used:~$ sudo mv claveprivada.key /etc/ssl/private
miadmin@rsp-used:~$ sudo chown root:ssl-cert /etc/ssl/private/claveprivada.key
miadmin@rsp-used:~$ sudo ls -l /etc/ssl/private/claveprivada.key
-rw-rw-r-- 1 root ssl-cert 1704 nov 24 08:48 /etc/ssl/private/claveprivada.key
miadmin@rsp-used:~$ sudo chmod 640 /etc/ssl/private/claveprivada.key
miadmin@rsp-used:~$ sudo ls -l /etc/ssl/private/claveprivada.key
-rw-r----- 1 root ssl-cert 1704 nov 24 08:48 /etc/ssl/private/claveprivada.key
```

Movemos el certificado al directorio `certs` con “`sudo mv certificado.crt /etc/ssl/certs`” y cambiamos el propietario para que pertenezca a `root:root` con “`sudo chown root:root /etc/ssl/certs/certificado.crt`” y comprobamos listando el contenido de ese directorio

```
miadmin@rsp-used:~$ sudo mv certificado.crt /etc/ssl/certs
miadmin@rsp-used:~$ sudo chown root:root /etc/ssl/certs/certificado.crt
miadmin@rsp-used:~$ sudo ls -l /etc/ssl/certs/certificado.crt
-rw-rw-r-- 1 root root 1281 nov 24 09:06 /etc/ssl/certs/certificado.crt
```

A continuación, vamos a `/etc/apache2/sites-available` y copiamos el sitio “`default-ssl.conf`” creando uno nuevo llamado “`rebeca-ssl.conf`” con el comando “`sudo cp default-ssl.conf rebeca-ssl.conf`”

```
miadmin@rsp-used:/etc/apache2/sites-available$ sudo cp default-ssl.conf rebeca-ssl.conf
miadmin@rsp-used:/etc/apache2/sites-available$ sudo nano rebeca-ssl.conf
```

Editamos “`rebeca-ssl.conf`”: insertamos `SSLEngine on`, modificamos las líneas `SSLCertificateFile` y `SSLCertificateKeyFile` especificando la ruta donde se encuentran el certificado y la clave privada respectivamente y añadimos un dominio que tenga de puerto el 443 (HTTPS)

```
# Enable/Disable SSL for this virtual host.
SSLEngine on

# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/certificado.crt
SSLCertificateKeyFile /etc/ssl/private/claveprivada.key

# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# full certificate chain for this SSL virtual host.
# For more about this feature, see the mod_ssl documentation.
#
#<VirtualHost *:443>
#    ServerAdmin webmaster@localhost
#    ServerName daw204.rebeca.local
#    DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1,
# error, crit, alert, emerg.
```

Guardamos el documento y comprobamos que no hay errores con “`sudo apache2ctl configtest`”

```
miadmin@rsp-used:/etc/apache2/sites-available$ sudo apache2ctl configtest
Syntax OK
```

Si todo está bien activamos el sitio con “`sudo a2ensite rebeca-ssl.conf`” y reiniciamos apache

```
miadmin@rsp-used:/etc/apache2/sites-available$ sudo a2ensite rebeca-ssl.conf
Enabling site rebeca-ssl.
To activate the new configuration, you need to run:
systemctl reload apache2
miadmin@rsp-used:/etc/apache2/sites-available$ sudo systemctl reload apache2
```



Abrimos el puerto 443 para poder trabajar con https con “sudo ufw allow 443” y comprobamos que este abierto con “sudo ufw status”

```
miadmin@rsp-used:/etc/apache2/sites-available$ sudo ufw allow 443
Rule added
Rule added (v6)
miadmin@rsp-used:/etc/apache2/sites-available$ sudo ufw status
Status: active

To Action From
--
22 ALLOW Anywhere
Apache ALLOW Anywhere
9003 ALLOW Anywhere
3306 ALLOW Anywhere
443 ALLOW Anywhere
22 (v6) ALLOW Anywhere (v6)
Apache (v6) ALLOW Anywhere (v6)
9003 (v6) ALLOW Anywhere (v6)
3306 (v6) ALLOW Anywhere (v6)
443 (v6) ALLOW Anywhere (v6)
```

Para comprobar que hemos activado el certificado escribimos en el navegador https://192.168.3.204 y debería aparecer lo siguiente:



### La conexión no es privada

Es posible que los atacantes estén intentando robar tu información de **192.168.3.204** (por ejemplo, contraseñas, mensajes o tarjetas de crédito). [Más información](#)

NET::ERR\_CERT\_AUTHORITY\_INVALID



Para disfrutar del máximo nivel de seguridad en Chrome, [activa la protección mejorada](#).

Configuración avanzada

Volver para estar a salvo

Modificamos el fichero 000-default y escribimos estas líneas:

```
miadmin@rsp-used:/etc/apache2/sites-available$ sudo nano 000-default.conf
[sudo] password for miadmin:
# value is not recursive as it is used as a loop
# However, you must set it for any further virtu
#ServerName www.example.com

ServerName daw204.rebeca.local
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debu
# error, crit, alert, emerg.
# It is also possible to configure the loglevel
```

Nos vamos a /var/www/html/.htaccess y escribimos las siguientes líneas en el archivo para que cuando escribamos el dominio en el navegador nos redireccione automáticamente a https

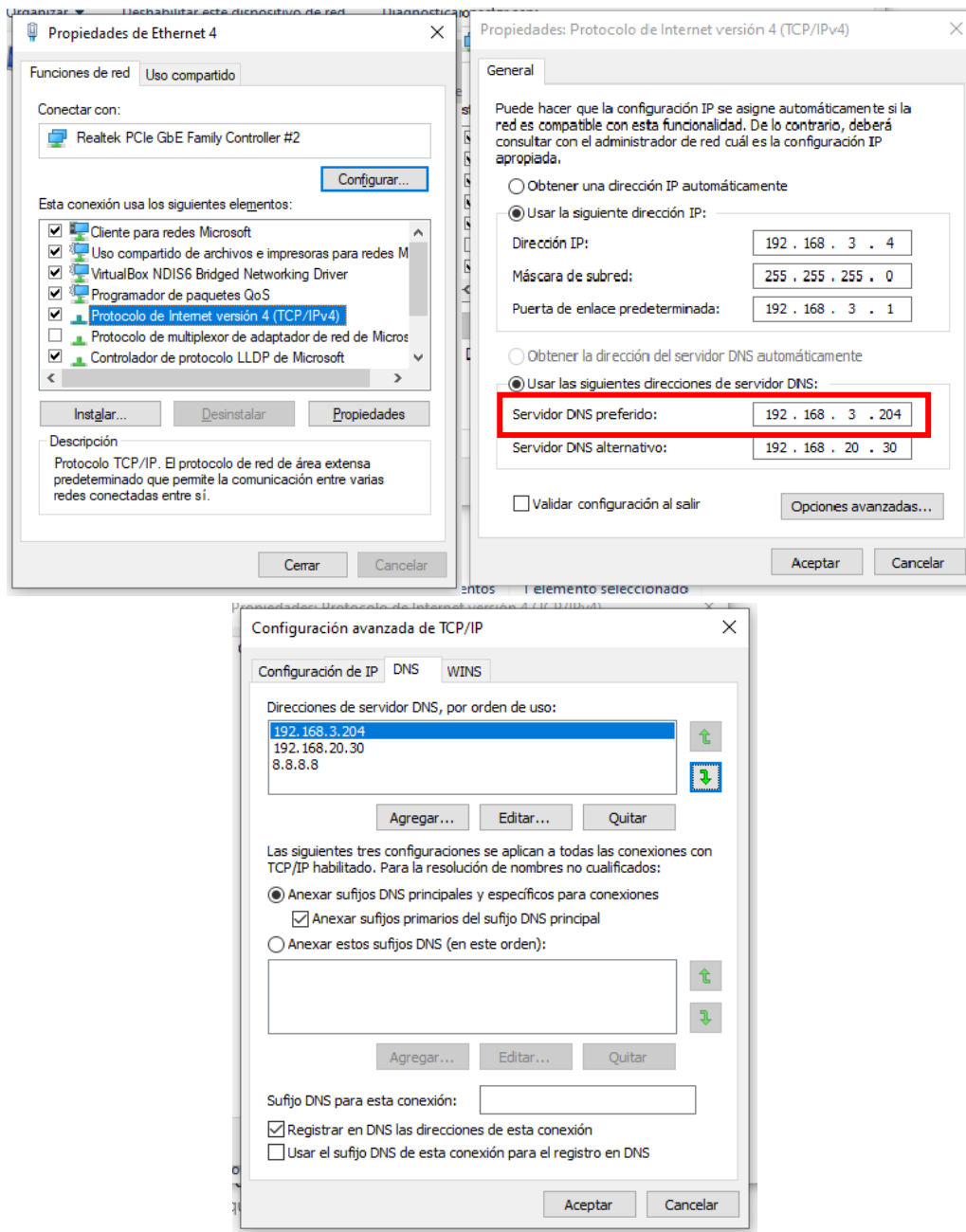
```
ETOT DOCUMENT 404 LO SIEMPRE, ESE DIRECTORIO NO EXISTE
Redirect 301 /educa https://www.educa.jcyl.es/es
RewriteEngine On
RewriteCond %{SERVER_PORT} 80
RewriteRule ^(.*)$ https://daw204.rebeca.local/$1 [R,L]
```

Guardamos el fichero y comprobamos que el módulo rewrite este activo con “`sudo a2enmod rewrite`” y reiniciamos apache y hacemos un configtest para ver que todo ha ido bien

```
miadmin@rsp-used:/var/www/html$ sudo a2enmod rewrite
Module rewrite already enabled

miadmin@rsp-used:/var/www/html$ sudo apache2ctl configtest
Syntax OK
```

Por último, nos vamos a la configuración de red de nuestra maquina anfitriona añadiendo un servidor DNS con la IP de nuestro servidor y aceptamos los cambios.



## 4. PHP

### 4.1 Módulos

A continuación, se muestran los módulos instalados de php mostrados por el comando `"dpkg --get-selections | grep php"`:

```
miadmin@rsp-used:~$ dpkg --get-selections | grep php8.1
libapache2-mod-php8.1      install
php8.1                    install
php8.1-bz2                install
php8.1-cli                install
php8.1-common             install
php8.1-curl               install
php8.1-gd                 install
php8.1-mbstring           install
php8.1-mysql              install
php8.1-opcache            install
php8.1-readline           install
php8.1-soap               install
php8.1-xdebug             install
php8.1-xml                install
php8.1-zip                install
miadmin@rsp-used:~$
```

### 4.2 Instalación de php y configuracion de php.ini

Antes de instalar php tendremos que actualizar el sistema con `"sudo apt update"` y para instalarlo escribimos `"sudo apt -y install php8.1"` (instalamos la versión 8.1 porque es la más nueva que es estable) y para comprobar la versión podemos utilizar `"sudo php -v"`


```
miadmin@rsp-used:~$ sudo php -v
PHP 8.1.2-1ubuntu2.14 (cli) (built: Aug 18 2023 11:41:11) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.14, Copyright (c), by Zend Technologies
```

Instalamos la librería de módulos de apache con `"sudo apt install libapache2-mod-php"` y para crear un archivo info.php usamos `"sudo nano /var/www/html/info.php"` y escribimos lo siguiente:

```
<?php
    phpinfo();
?>
```

- `phpinfo()` es una función que muestra información detallada sobre la configuración de PHP y el entorno del servidor web

A continuación, reiniciamos el servicio de apache con `"sudo service apache2 restart"` y para comprobar que todo está bien escribimos en nuestro navegador `192.168.3.204/info.php` y si aparece esta pantalla significa que se ha creado correctamente:

PHP Version 8.1.2-1ubuntu2.14	
	
System	Linux rasp-used 5.15.0-86-generic #96-Ubuntu SMP Wed Sep 20 08:23:49 UTC 2023 x86_64
Build Date	Aug 18 2023 11:41:11
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-gd.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-imagick.ini, /etc/php/8.1/apache2/conf.d/20-imagick.ini, /etc/php/8.1/apache2/conf.d/20-ldap.ini, /etc/php/8.1/apache2/conf.d/20-mbstring.ini, /etc/php/8.1/apache2/conf.d/20-mcrypt.ini, /etc/php/8.1/apache2/conf.d/20-mysqlnd.ini, /etc/php/8.1/apache2/conf.d/20-openssl.ini, /etc/php/8.1/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tidy.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini, /etc/php/8.1/apache2/conf.d/20-xml.ini, /etc/php/8.1/apache2/conf.d/20-xmlrpc.ini, /etc/php/8.1/apache2/conf.d/20-zip.ini, /etc/php/8.1/apache2/conf.d/20-zlib.ini

Instalamos los módulos de SOAP y de xml con el comando `"sudo apt install -y php8.1-xml php8.1-soap"` y reiniciamos el servicio de apache2. En `192.168.3.204/info.php` podemos comprobar que se han instalado correctamente

SNMP	Rasmus Lerdorf, Harrie Hazewinkel, Mike Jackson, Steven Lawrence, Jonnari H.
SOAP	Brad Lafountain, Shane Caraveo, Dmitry Stogov
Sockets	Chris Vandomelen, Sterling Hughes, Daniel Beulshausen, Jason Greene
Sodium	Frank Denis
SPL	Marcus Boerger, Etienne Kneuss
SQLite 3.x driver for PDO	Wez Furlong
SQLite3	Scott MacVicar, Ilia Alshanetsky, Brad Dewar
System V Message based IPC	Wez Furlong
System V Semaphores	Tom May
System V Shared Memory	Christian Cartus
tidy	John Coggeshall, Ilia Alshanetsky
tokenizer	Andrei Zmievski, Johannes Schlueter
XML	Stig Bakken, Thies C. Arntzen, Sterling Hughes
XMLReader	Rob Richards
XMLWriter	Rob Richards, Pierre-Alain Joye
YAML	Christian Stachler, Rob Richards

Ahora cambiaremos el fichero de configuración php.ini pero antes de esto haremos una copia de seguridad. Cambiamos de directorio a `"cd /etc/php/8.1/apache2"` y usamos el comando `"sudo cp php.ini php.ini.backup"` para hacer la copia.

```
miadmin@rsp-used:/etc/php/8.1/apache2$ sudo cp php.ini php.ini.backup
miadmin@rsp-used:/etc/php/8.1/apache2$ ls
conf.d  php.ini  php.ini.backup
```

Usamos el comando `"sudo nano php.ini"` para modificar el fichero de configuración de php. Buscamos en el archivo la línea `"display errors"` y cambiamos la línea `display_errors=Off` y la ponemos en On

```
; Development Value: On
; Production Value: Off
; https://php.net/display-errors
display_errors = On
; The display of errors which occur
; separately from display_errors. W
```

En ese mismo archivo buscamos la línea `"display_startup_errors"` y lo activamos igual que el anterior

```
; Development Value: On
; Production Value: Off
; https://php.net/display-startup-errors
display_startup_errors = On_
```

Guardamos el archivo, reiniciamos el servicio de apache y para comprobarlo volvemos al navegador escribiendo 192.168.3.204/info.php y buscamos las configuraciones que acabamos de activar para comprobar que se han cambiado correctamente

disable_functions	no value	no value
display_errors	On	On
display_startup_errors	On	On
doc_root	no value	no value

Por último, volvemos a modificar el php.ini y cambiamos los valores de “memory\_limit” para que tenga una memoria de 256M

```
; Maximum amount of memory a scrip
; https://php.net/memory-limit
memory_limit = 256M
```

Como hemos hecho anteriormente hacemos “sudo service apache2 restart” para reiniciar apache y comprobamos que se haya cambiado la memoria en nuestro navegador

max_multipart_body_parts	-1
memory_limit	256M
open_basedir	no value

## 5. MySQL

### 5.1 Instalación y configuración

Primero actualizamos las librerías de Linux con “sudo apt update” y “sudo apt upgrade” y después instalamos el servicio de MySQL con “sudo apt install mysql-server”. Para visualizar la versión que se ha instalado, usamos el comando “mysql --version”

```
miadmin@rsp-used:~$ mysql --version
mysql Ver 8.0.34-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
miadmin@rsp-used:~$ _
```

Para permitir la conexión desde cualquier IP necesitamos entrar al fichero de configuración de MySQL (“sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf”) y comentar las siguientes líneas: “bind-address = 127.0.0.1” y “mysqlx-bind-address = 127.0.0.1”

```
#
# Instead of skip-networking the default is now to listen on *
# localhost which is more compatible and is not less secure
#bind-address            = 127.0.0.1
#mysqlx-bind-address     = 127.0.0.1
#
# * Fine Tuning
#
```

Guardamos el archivo y reiniciamos mysql con “sudo service mysql restart” y después abrimos el puerto 3306 con “sudo ufw allow 3306”

```
miadmin@rsp-used:~$ sudo ufw status
Status: active

To Action From
--
22 ALLOW Anywhere
Apache ALLOW Anywhere
9003 ALLOW Anywhere
3306 ALLOW Anywhere
22 (v6) ALLOW Anywhere (v6)
Apache (v6) ALLOW Anywhere (v6)
9003 (v6) ALLOW Anywhere (v6)
3306 (v6) ALLOW Anywhere (v6)
```

A continuación, ejecutaremos un script para realizar una configuración inicial de mysql con el comando “sudo mysql\_secure\_installation” y responderemos “yes” a todos los pasos excepto el de nivel de seguridad de contraseña que elegiremos el “LOW”

```
Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.
- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
```

Ahora volvemos a reiniciar MySQL e instalamos las librerías de php que son necesarias para la conexión de mysql con php con el comando “`sudo apt install libapache2-mod-php8.1 php8.1-mysql`” y reiniciamos apache. Ejecutamos mysql con “`sudo mysql`” y una vez dentro creamos el usuario administrador de las bases de datos de nuestro servidor. Para ello escribimos:

- Create user ‘adminsqli’@’%’ identified by ‘passo’;
- Grant all privileges on \*.\* to ‘adminsqli’@’%’ with grant option;

```
mysql> select user,plugin,host from user where user like 'adminsqli';
+-----+-----+-----+
| user      | plugin                | host |
+-----+-----+-----+
| adminsqli | caching_sha2_password | %    |
+-----+-----+-----+
1 row in set (0,00 sec)
```

En el caso de que no nos permita crear el usuario saltando el error de que la contraseña no cumple con los requisitos, tendremos que cambiar las variables globales de mysql para admitir contraseñas con 4 caracteres como mínimo. Para ello dentro de la consola de mysql escribiremos “`show variables like ‘validate_password%’`” y se mostrara la tabla de las variables que controlan los requisitos de las contraseñas de los usuarios.

```
mysql> show variables like 'validate_pass%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| validate_password.changed_characters_percentage | 0 |
| validate_password.check_user_name | ON |
| validate_password.dictionary_file | |
| validate_password.length | 8 |
| validate_password.mixed_case_count | 1 |
| validate_password.number_count | 1 |
| validate_password.policy | MEDIUM |
| validate_password.special_char_count | 1 |
+-----+-----+
8 rows in set (0,00 sec)
```

Una vez aquí, cambiaremos el valor de “length” y “policy” con “`set global validate_password.length=4;`” y “`set global validate_password.policy=LOW;`”

```
mysql> set global validate_password.length=4;
Query OK, 0 rows affected (0,00 sec)

mysql> set global validate_password.policy=LOW;
Query OK, 0 rows affected (0,00 sec)

mysql> show variables like 'validate_password%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| validate_password.changed_characters_percentage | 0 |
| validate_password.check_user_name | ON |
| validate_password.dictionary_file | |
| validate_password.length | 4 |
| validate_password.mixed_case_count | 1 |
| validate_password.number_count | 1 |
| validate_password.policy | LOW |
| validate_password.special_char_count | 1 |
+-----+-----+
8 rows in set (0,00 sec)
```

## 5.2 PHPMyAdmin

Antes de la instalación de PHPMyAdmin tenemos que cambiar la contraseña del usuario de MySQL a “paso” para que cumpla con los requisitos requeridos. Para ello entraremos en MySQL con “sudo mysql”

1. Filtramos las variables que contengan parámetros para las contraseñas con “show variables like 'validate\_password%';”

```
mysql> show variables like 'validate_password%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| validate_password.changed_characters_percentage | 0 |
| validate_password.check_user_name | ON |
| validate_password.dictionary_file | |
| validate_password.length | 8 |
| validate_password.mixed_case_count | 1 |
| validate_password.number_count | 1 |
| validate_password.policy | MEDIUM |
| validate_password.special_char_count | 1 |
+-----+-----+
8 rows in set (0,00 sec)
```

2. Cambiamos los valores de longitud (para que tenga un máximo de 4 caracteres) y de política (para que sea menos segura) con “set global validate\_password.length=4;” y “set global validate\_password.policy=LOW;” respectivamente;

```
mysql> set global validate_password.length=4;
Query OK, 0 rows affected (0,00 sec)

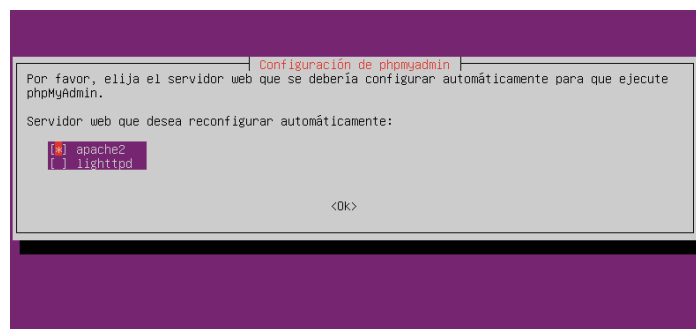
mysql> set global validate_password.policy=LOW;
Query OK, 0 rows affected (0,00 sec)
```

3. Ahora creamos el usuario “create user 'adminsqli'@'%' identified by 'paso';” y le damos los permisos “grant all privileges on \*.\* to 'adminsqli'@'%' with grant option;”

```
mysql> create user 'adminsqli'@'%' identified by 'paso';
Query OK, 0 rows affected (0,00 sec)

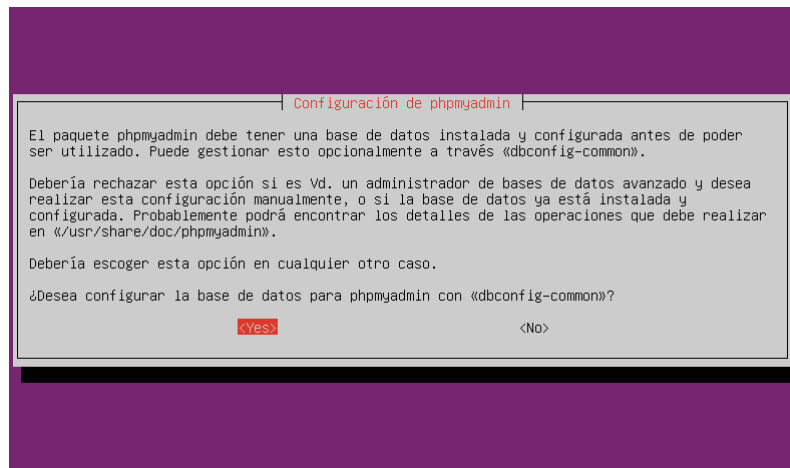
mysql> grant all privileges on *.* to 'adminsqli'@'%' with grant option;
Query OK, 0 rows affected (0,01 sec)
```

A continuación, salimos de mysql con “exit”. Para instalar PHPMyAdmin y los módulos que necesitamos escribiremos el comando “sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl”. Cuando lo ejecutemos, aparecerá una interfaz del Ubuntu diferente en la que nos harán una serie de preguntas:

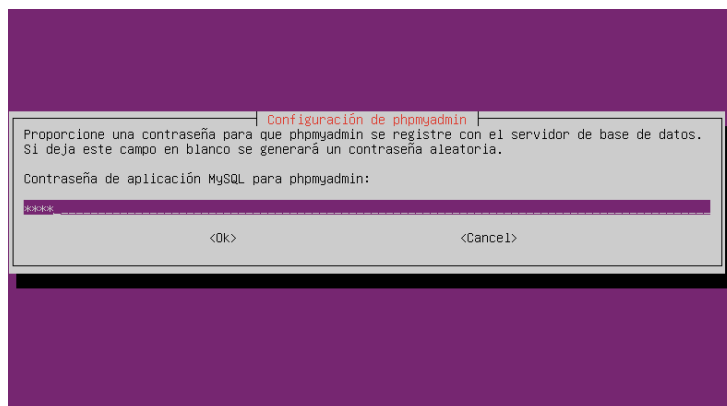




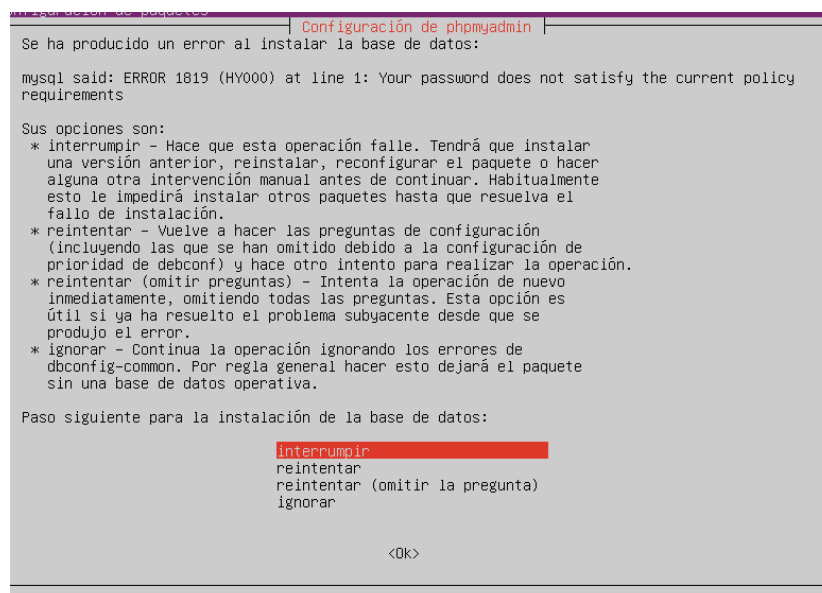
- Seleccionamos con la barra espaciadora la opción de *apache2*



- Ahora seleccionamos la opción yes para configurar la base de datos



- Introducimos la contraseña, en nuestro caso: *paso*



- Seleccionamos *interrumpir*

Una vez terminada la instalación, nos volvemos a conectar a mysql pero con el nuevo usuario “mysql -u adminsql -p”

```
miadmin@rsp-used:~$ mysql -u adminsql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.35-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Ejecutamos el comando `"uninstall component 'file://component_validate_password';"` y volvemos a cerrar mysql. Ahora es necesaria una purga de phpmyadmin `"sudo apt purge phpmyadmin"` y lo volvemos a instalar de igual manera con `"sudo apt install phpmyadmin"`. Llegados a este punto volveremos a acceder a mysql con el usuario adminsql y volveremos a hacer el comando `"install component 'file://component_validate_password';"`. Para comprobar que todo ha ido bien, escribiremos en el navegador la IP de nuestro servidor seguida de `"phpmyadmin"` y debería aparecer esta ventana



## 6. XDebug

### 6.1 Instalación

Antes de instalar XDebug comprobaremos que no esté instalado anteriormente con “`php -m | grep xdebug`”, si la terminal no devuelve nada significa que ese modulo no está instalado.

Para instalarlo escribimos “`sudo apt install php8.1-xdebug`”

```
miadmin@rsp-used:~$ sudo apt install php8.1-xdebug
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
php8.1-xdebug ya está en su versión más reciente (3.1.2)
0 actualizados, 0 nuevos se instalarán, 0 para eliminar
```

A continuación, cambiaremos de directorio a “`cd /etc/php/8.1/mods-available/`” y modificamos el archivo 20-xdebug.ini con “`sudo nano 20-xdebug.ini`” y escribimos las siguientes líneas:

```
zend_extension=xdebug.so
xdebug.extended_info=on
xdebug.remote_host=192.168.3.204
xdebug.remote_port=9003
xdebug.remote_handler=dbgp
xdebug.remote_autostart=on
xdebug.remote_enable = on
xdebug.mode=debug
xdebug.client_host=localhost
xdebug.client_port=9003
xdebug.idekey="netbeans-xdebug"
```

Para que se pueda conectar por el puerto 9003 tendremos que abrirlo mediante “`sudo ufw allow 9003`” y con “`sudo ufw status`” vemos los puertos que están abiertos

```
miadmin@rsp-used:/etc/php/8.1/apache2/conf.d$ sudo ufw allow 9003
Rule added
Rule added (v6)
miadmin@rsp-used:/etc/php/8.1/apache2/conf.d$ sudo ufw status
Status: active
```

To	Action	From
22	ALLOW	Anywhere
Apache	ALLOW	Anywhere
9003	ALLOW	Anywhere
22 (v6)	ALLOW	Anywhere (v6)
Apache (v6)	ALLOW	Anywhere (v6)
9003 (v6)	ALLOW	Anywhere (v6)

Para terminar, reiniciamos el servicio con “`sudo service apache2 restart`”.

## 7. Servidores DNS

### 7.1 Instalación del servidor DNS

Comprobamos que el paquete de DNS está instalado con `"aptitude show bind9"` y visualizamos el paquete con `"dpkg -l | grep bind9"`

```
miadmin@rsp-used:~$ aptitude show bind9
Command 'aptitude' not found, did you mean:
  command 'aptitude' from deb aptitude (0.8.13-3ubuntu1)
Try: sudo apt install <deb name>
miadmin@rsp-used:~$ dpkg -l | grep bind9
ii  bind9-dnsutils      1:9.18.18-0ubuntu0.22.04.1      amd64      Client
ts provided with BIND 9
ii  bind9-host          1:9.18.18-0ubuntu0.22.04.1      amd64      DNS L
ookup Utility
ii  bind9-libs:amd64    1:9.18.18-0ubuntu0.22.04.1      amd64      Share
d Libraries used by BIND 9
miadmin@rsp-used:~$
```

Procedemos a actualizar el sistema con `"sudo apt update"` e instalamos el servicio con el comando `"sudo apt install bind9"`. Comprobamos que el estado de bind9 esté activo con `"sudo service bind9 status"`

```
miadmin@rsp-used:~$ sudo service bind9 status
• named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-12-14 15:36:19 CET; 1min 30s ago
     Docs: man:named(8)
  Process: 33551 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 33553 (named)
    Tasks: 4 (limit: 2221)
   Memory: 6.8M
      CPU: 29ms
   CGroup: /system.slice/named.service
           └─33553 /usr/sbin/named -u bind

dic 14 15:36:19 rsp-used named[33553]: network unreachable resolving './DNSKEY/IN': 2001:500:2::
dic 14 15:36:19 rsp-used named[33553]: network unreachable resolving './NS/IN': 2001:500:2::c#53
```

Ahora debemos editar la configuración de red, que se aloja en la siguiente ruta, el archivo que tenga como extensión `‘.yaml’`, en mi caso se llama `‘00-installer-config.yaml’`, para ello, usamos el comando `"sudo nano /etc/netplan/00-installer-config.yaml"` y modificamos lo siguiente:

```
GNU nano 6.2 /etc/netplan/00-in
# This is the network config written by 'subiquity'
network:
  ethernet:
    enp0s3:
      addresses:
        - 192.168.1.204/24
      nameservers:
        addresses:
          - 192.168.1.204
          search: [rebeca.local]
      routes:
        - to: default
          via: 192.168.1.1
  version: 2
```

Aplicamos los cambios con `"sudo netplan apply"`, abrimos el puerto 53 para permitir la conexión con bind9 `"sudo ufw allow 53"` y comprobamos si funcionamiento haciendo ping a google

```
miadmin@rsp-used:~$ sudo netplan apply
** (generate:33813): WARNING **: 15:44:45.746: Permissions for /etc/netplan/00-installer-config.yaml
are too open. Netplan configuration should NOT be accessible by others.
WARNING:root:Cannot call Open vSwitch: ovssdb-server.service is not running.

** (process:33811): WARNING **: 15:44:46.037: Permissions for /etc/netplan/00-installer-config.yaml
are too open. Netplan configuration should NOT be accessible by others.

** (process:33811): WARNING **: 15:44:46.098: Permissions for /etc/netplan/00-installer-config.yaml
are too open. Netplan configuration should NOT be accessible by others.

** (process:33811): WARNING **: 15:44:46.098: Permissions for /etc/netplan/00-installer-config.yaml
are too open. Netplan configuration should NOT be accessible by others.
miadmin@rsp-used:~$ sudo ufw allow 53
Rule added
Rule added (v6)
miadmin@rsp-used:~$ ping www.google.es
PING www.google.es (216.58.215.163) 56(84) bytes of data.
64 bytes from mad41s07-in-f3.1e100.net (216.58.215.163): icmp_seq=1 ttl=115 time=12.0 ms
64 bytes from mad41s07-in-f3.1e100.net (216.58.215.163): icmp_seq=2 ttl=115 time=11.3 ms
64 bytes from mad41s07-in-f3.1e100.net (216.58.215.163): icmp_seq=3 ttl=115 time=11.9 ms
```

## 7.2 Declaración de la zona directa

Modificaremos el archivo con “sudo nano /etc/bind/named.conf.local” añadiendo una nueva zona como se muestra a continuación:

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

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
zone "rebeca.local" {
    type master;
    file "/etc/bind/db.rebeca.local";
};
```

Comprobamos los ficheros de configuración con “sudo named-checkconf”

```
miadmin@rsp-used:~$ sudo named-checkconf
```

A continuación, crearemos una copia de seguridad con el comando “sudo cp /etc/bind/db.local /etc/bind/db.rebeca.local” y editaremos esa copia para que resulte de la siguiente manera:

```
;
; BIND data file for rebeca.local
;
$TTL      604800
@         IN      SOA      rsp-used.rebeca.local. marodriguez.educa.jcyl.es. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                3600      ) ; Negative Cache TTL
;
@         IN      NS       rsp-used.rebeca.local.

; Registro Host
@         IN      A        192.168.1.204
rsp-used  IN      A        192.168.1.204

; Registro Alias
daw201    IN      CNAME    rsp-used.rebeca.local.
daw202    IN      CNAME    rsp-used.rebeca.local.
daw204    IN      CNAME    rsp-used.rebeca.local.
```

Comprobamos que la sintaxis esta correcta con “sudo named-checkzone rebeca.local /etc/bind/db.rebeca.local” y reiniciamos el servicio con “sudo service bind9 restart”

```
miadmin@rsp-used:~$ sudo named-checkzone rebeca.local /etc/bind/db.rebeca.local
zone rebeca.local/IN: loaded serial 1
OK
miadmin@rsp-used:~$
```

```
miadmin@rsp-used:~$ sudo service bind9 restart
miadmin@rsp-used:~$
```

### 7.3 Declaración de la zona inversa

Volvemos a repetir los pasos, modificaremos el archivo con “sudo nano /etc/bind/named.conf.local” y escribimos:

```
};
zone "3.168.192.in-addr.local"{
    type master;
    file "/etc/bind/db.3.168.192.in-addr.arpa";
};
```

Como medida de seguridad hacemos una copia de este archivo “sudo cp /etc/bind/db.rebeca.local /etc/bind/db.3.168.192.in-addr.arpa” y lo editamos “sudo nano /etc/bind/db.3.168.192.in-addr.arpa” de la siguiente forma:

```
;
; BIND data for 3.168.192.in-addr.arpa
;
$TTL      604800
@         IN      SOA      rsp-used.rebeca.local. root.localhost. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                3600 )     ; Negative Cache TTL
;
@         IN      NS       rsp-used.rebeca.local.
; Registros punteros o pointers
204      IN      PTR      rsp-used.rebeca.local.
204      IN      PTR      rebeca.local.
```

Comprobamos la zona con “sudo named-checkzone 204.168.192.in-addr.arpa /etc/bind/db.3.168.192.in-addr.arpa”y reiniciamos bind9

```
miadmin@rsp-used:~$ sudo named-checkzone 204.168.192.in-addr.arpa /etc/bind/db.3.168.192.in-addr.arpa
zone 204.168.192.in-addr.arpa/IN: loaded serial 1
OK
miadmin@rsp-used:~$ sudo service bind9 restart
miadmin@rsp-used:~$
```

### 7.4 Comprobación de funcionamiento

Escribimos en la terminal “host rsp-used.rebeca.local”

```
miadmin@rsp-used:~$ host rsp-used.rebeca.local
rsp-used.rebeca.local has address 192.168.1.204
```

Ahora comprobamos con este nombre "host daw204.rebeca.local"

```
miadmin@rsp-used:~$ host daw204.rebeca.local
daw204.rebeca.local is an alias for rsp-used.rebeca.local.
rsp-used.rebeca.local has address 192.168.1.204
```

Y por último "host rebeca.local"

```
miadmin@rsp-used:~$ host rebeca.local
rebeca.local has address 192.168.1.204
```

A continuación, comprobamos con el comando "nslookup" -> "rsp-used.rebca.local"

```
miadmin@rsp-used:~$ nslookup
> rsp-used.rebeca.local
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   rsp-used.rebeca.local
Address: 192.168.1.204
>
```

Ahora hacemos "daw204.rebeca.local"

```
> daw204.rebeca.local
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
daw204.rebeca.local canonical name = rsp-used.rebeca.local.
Name:   rsp-used.rebeca.local
Address: 192.168.1.204
```

Y por último "rebeca.local"

```
> rebeca.local
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   rebeca.local
Address: 192.168.1.204
```

## 8. Cuentas de desarrollo y hosting virtual

### 8.1 Creación de operadorweb

Tenemos que crear un usuario que se encargue de conectarse al servidor mediante http/s desde la maquina anfitriona que usaremos de cliente. Para eso usamos el comando “sudo adduser --home /var/www/html --no-create-home --ingroup www-data operadorweb” que creara el usuario operadorweb en el grupo www-data (grupo por defecto de usuarios de apache) y con el home en el directorio html.

```
miadmin@rsp-used:~$ sudo adduser --home /var/www/html --no-create-home --ingroup www-data operadorweb
Adding user `operadorweb' ...
Adding new user `operadorweb' (1002) with group `www-data' ...
Not creating home directory `/var/www/html'.
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for operadorweb
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n]
miadmin@rsp-used:~$ _
```

Para confirmar que se ha creado usamos el comando “id operadorweb” y se mostrará la información sobre ese usuario.

```
miadmin@rsp-used:~$ id operadorweb
uid=1002(operadorweb) gid=33(www-data) groups=33(www-data)
```

A continuación, cambiamos el propietario del directorio /var/www/html para que sea operadorweb con “sudo chown -R operadorweb:www-data /var/www/html” y seguido de esto cambiamos los permisos del archivo que se encuentra en ese directorio para que pueda ser legible, modificado y ejecutado por ese usuario con “sudo chmod -R 2775 /var/www/html”

```
miadmin@rsp-used:~$ sudo chown -R operadorweb:www-data /var/www/html
miadmin@rsp-used:~$ sudo chmod -R 2775 /var/www/html
```

```
miadmin@rsp-used:~$ ll /var/www/html/
total 20
drwxrwsr-x 2 operadorweb www-data 4096 oct 4 00:25 ./
drwxr-xr-x 3 root         root    4096 oct 4 00:25 ../
-rwxrwsr-x 1 operadorweb www-data 10671 oct 4 00:25 index.html*
```

### 8.2 Creación del grupo ftpuser y enjaulamiento

Creamos el grupo ftpuser al que pertenecerá el operadorweb con el comando “sudo groupadd ftpuser”

```
miadmin@rsp-used:~$ sudo groupadd ftpuser
[sudo] password for miadmin:
```

Después de esto, cambiaremos de directorio a “cd /etc/ssh” y crearemos una copia de seguridad del fichero de configuracion sshd\_config.d con el comando “sudo cp -r sshd\_config.d sshd\_config.d.backup”. Una vez creada la copia, editaremos el archivo original (con “sudo nano sshd\_config”) de la siguiente manera:



```
# override default of no subsystems
#Subsystem      sftp      /usr/lib/openssh/sftp-server
Subsystem sftp internal-sftp

# Example of overriding settings on a per-user basis
#Match User anoncvs
#      X11Forwarding no
#      AllowTcpForwarding no
#      PermitTTY no
#      ForceCommand cvs server
Match Group ftpuser
ChrootDirectory %h
ForceCommand internal-sftp -u 2
AllowTcpForwarding yes
PermitTunnel no
X11Forwarding no_
```

- Comentamos la línea “Subsystem sftp /usr/lib/openssh/sftp-server” que viene por defecto y escribimos la que viene a continuación “Subsystem sftp internal-sftp”
- Al final del fichero agregamos las líneas de texto que vienen indicadas por la segunda flecha

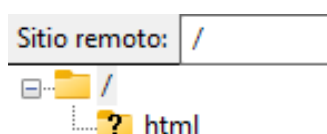
Una vez guardada la configuración, reiniciaremos ssh con “sudo service ssh restart”. Esta configuración nos permite controlar el servicio SFTP con el grupo de usuarios “ftpuser”. Ahora procedemos a enjaular a nuestro usuario operadorweb en el directorio “/var/www”, esto significa que el usuario operadorweb nunca a va a tener acceso a los archivos y directorios que estén por encima del directorio en el que se encuentra enjaulado. Para ello debemos parar el servicio de apache con “sudo service apache2 stop” y después usaremos el comando “sudo usermod operadorweb -d /var/www” que modifica el usuario para que su home sea /var/www

```
miadmin@rsp-used:/etc/ssh$ sudo usermod operadorweb -d /var/www
miadmin@rsp-used:/etc/ssh$ _
```

A continuación, usamos el comando “sudo chown root:root /var/www” para cambiar el propietario de ese directorio y seguido de esto con “sudo chmod 555 /var/www” y “sudo usermod -G ftpuser operadorweb” le daremos permisos de lectura y ejecución y agregaremos el usuario operadorweb al grupo ftpuser.

```
drwxrwxrwt  6 root root    4096 oct 30 18:48 tmp/
dr-xr-xr-x  3 root root    4096 oct  4 00:25 www/
miadmin@rsp-used:/var$
```

Si queremos comprobar que el enjaulamiento está correcto, nos conectamos desde FileZilla el servidor con el usuario operadorweb y si el directorio raíz contiene /www significa que ha ido correctamente.



Ahora haremos un ejemplo de creación de un usuario nuevo DAW204 y de su enjaulamiento. Usamos el comando “sudo useradd -g www-data -G ftpuser -m -d /var/www/DAW201 DAW201” para crear un usuario con directorio principal el especificado y dentro de los grupos especificados.

Comprobamos que esta creado con “id DAW201”.

Cambiamos el propietario de la carpeta DAW201 para que sea de root con “sudo chown root:root /var/www/DAW201” y le cambiamos los permisos con “sudo chmod 555 /var/www/DAW201”

Ahora creamos la carpeta donde estará alojado el proyecto “sudo mkdir /var/www/DAW201/public\_html” y le damos los permisos sobre esa carpeta con “sudo chmod 2775 /var/www/DAW201/public\_html”

Cambiamos el propietario de esa carpeta con “ht”

Comprobamos desde filezilla

## b. WXED – WINDOWS X

### 1. Nombre y configuración de red



```
C:\Users\daw2>ipconfig /all
```

```
Configuración IP de Windows
```

```
Nombre de host. . . . . : IS32WX04
Sufijo DNS principal . . . . . :
Tipo de nodo. . . . . : híbrido
Enrutamiento IP habilitado. . . . . : no
Proxy WINS habilitado . . . . . : no
```

```
Adaptador de Ethernet Ethernet 2:
```

```
Sufijo DNS específico para la conexión. . . :
Descripción . . . . . : VirtualBox Host-Only Ethernet Adapter
Dirección física. . . . . : 0A-00-27-00-00-0F
DHCP habilitado . . . . . : no
Configuración automática habilitada . . . : sí
Vínculo: dirección IPv6 local. . . : fe80::9c7d:3248:7df:cc37%15(Preferido)
Dirección IPv4. . . . . : 192.168.56.1(Preferido)
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . :
IAID DHCPv6 . . . . . : 269090855
DUID de cliente DHCPv6. . . . . : 00-01-00-01-28-CC-D8-2E-D8-CB-8A-73-7E-79
Servidores DNS. . . . . : fec0:0:0:ffff::1%1
                        fec0:0:0:ffff::2%1
                        fec0:0:0:ffff::3%1
NetBIOS sobre TCP/IP. . . . . : habilitado
```

```
Adaptador de Ethernet Ethernet 4:
```

```
Sufijo DNS específico para la conexión. . . :
Descripción . . . . . : Realtek PCIe GbE Family Controller #2
Dirección física. . . . . : 18-31-8F-2F-02-6A
DHCP habilitado . . . . . : no
Configuración automática habilitada . . . : sí
Vínculo: dirección IPv6 local. . . : fe80::d2a2:1e40:913f:d894%13(Preferido)
Dirección IPv4. . . . . : 192.168.3.4(Preferido)
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.3.1
IAID DHCPv6 . . . . . : 471347647
DUID de cliente DHCPv6. . . . . : 00-01-00-01-28-CC-D8-2E-D8-CB-8A-73-7E-79
Servidores DNS. . . . . : 192.168.3.204
                        192.168.20.30
                        8.8.8.8
NetBIOS sobre TCP/IP. . . . . : habilitado
```

## 2. Cuentas administradoras y cuenta de desarrollador

### 2.1 Conexión SSH desde Windows 10

Ahora pasamos a trabajar desde el Windows 10 de nuestra maquina anfitriona. Abrimos el cmd y escribimos “ssh miadmin@192.168.3.204” para conectarnos desde nuestro windows al terminal de nuestro entorno de desarrollo. Si al intentar conectarte te da problemas porque ya has usado la IP de la maquina nueva para otra máquina, tendremos que entrar en C: > Usuarios > TuUsuario > .ssh y eliminamos un archivo llamado “known\_hosts” y volver a ejecutar el comando “ssh miadmin@192.168.3.204”.

```
C:\Users\daw2>ssh miadmin@192.168.3.204
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@   WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!   @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:P2PA+gsJ4IG2G9BLN0VggONunE0/lENT5JRi6htw/8M.
Please contact your system administrator.
Add correct host key in C:\Users\daw2\.ssh\known_hosts to get rid of this message.
Offending ECDSA key in C:\Users\daw2\.ssh\known_hosts:1
ECDSA host key for 192.168.3.204 has changed and you have requested strict checking.
Host key verification failed.
```

equipo > SISTEMA (C:) > Usuarios > daw2 > .ssh

Nombre	Fecha de modificación	Tipo	Tamaño
known_hosts	03/10/2023 10:50	Archivo	1 KB

```
Active ESM Apps para recibir futuras actualizaciones de
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Failed to connect to https://changelogs.ubuntu.com/me

*** System restart required ***
Last login: Tue Oct 3 08:17:46 2023
miadmin@rsp-usedlimpia:~$
```

## 2.2 Cuentas administradoras del sistema operativo

```
C:\Users\daw2>net users
```

```
Cuentas de usuario de \\IS32WX14
```

```
-----  
admin                Administrador          bachillerato  
daw1                 daw2                 DefaultAccount  
eso                  Invitado             smr1  
smr2                 WDAGUtilityAccount  
Se ha completado el comando correctamente.
```

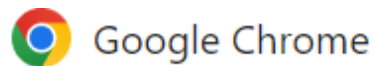
```
C:\Users\daw2>net localgroup
```

```
Alias para \\IS32WX14
```

```
-----  
*Administradores  
*Administradores de Hyper-V  
*Duplicadores  
*IIS_IUSRS  
*Invitados  
*Lectores del registro de eventos  
*Operadores criptográficos  
*Operadores de asistencia de control de acceso  
*Operadores de configuración de red  
*Operadores de copia de seguridad  
*System Managed Accounts Group  
*Usuarios  
*Usuarios avanzados  
*Usuarios COM distribuidos  
*Usuarios de administración remota  
*Usuarios de escritorio remoto  
*Usuarios del monitor de sistema  
*Usuarios del registro de rendimiento  
Se ha completado el comando correctamente.
```

### 3. Navegadores

#### 3.1 Google Chrome




La actualización ya casi ha terminado. Reinicia Chrome para completar la actualización.

✓

Versión 120.0.6099.72 (Build oficial) (64 bits)

#### 3.2 Microsoft Edge

Acerca de

 **Microsoft Edge**  
Versión 120.0.2210.61 (Compilación oficial) (64 bits)

✓ Microsoft Edge está actualizado.

**Descargar las actualizaciones en conexiones de uso medido** ☐

Descargar automáticamente las actualizaciones a través de redes de uso medido (por ejemplo, redes de telefonía móvil) e instalarlas al reiniciar el explorador. Pueden aplicarse cargos.

Este explorador se ha hecho gracias al proyecto de código abierto [Chromium](#) y a otro [software de código abierto](#).

Microsoft Edge  
© 2023 Microsoft Corporation. Todos los derechos reservados.  
[Términos de uso](#) - [Declaración de privacidad](#) - [Contrato de servicios de Microsoft](#)

#### 3.2 Mozilla Firefox

##### Actualizaciones de Firefox

Mantenga Firefox actualizado para un rendimiento, estabilidad y seguridad óptimos.

Versión 120.0.1 (64-bit) [Novedades](#)

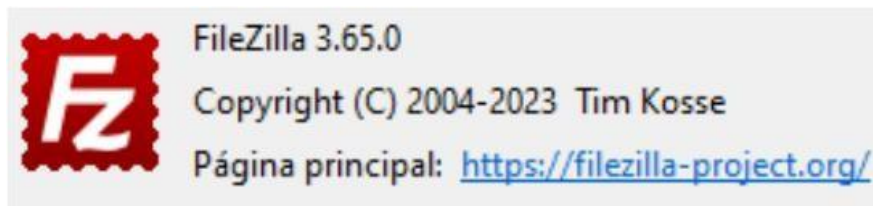
[Mostrar historial de actualizaciones...](#)

😊 Firefox está actualizado

[Buscar actualizaciones](#)

## 4. FileZilla

### 4.1 Versión



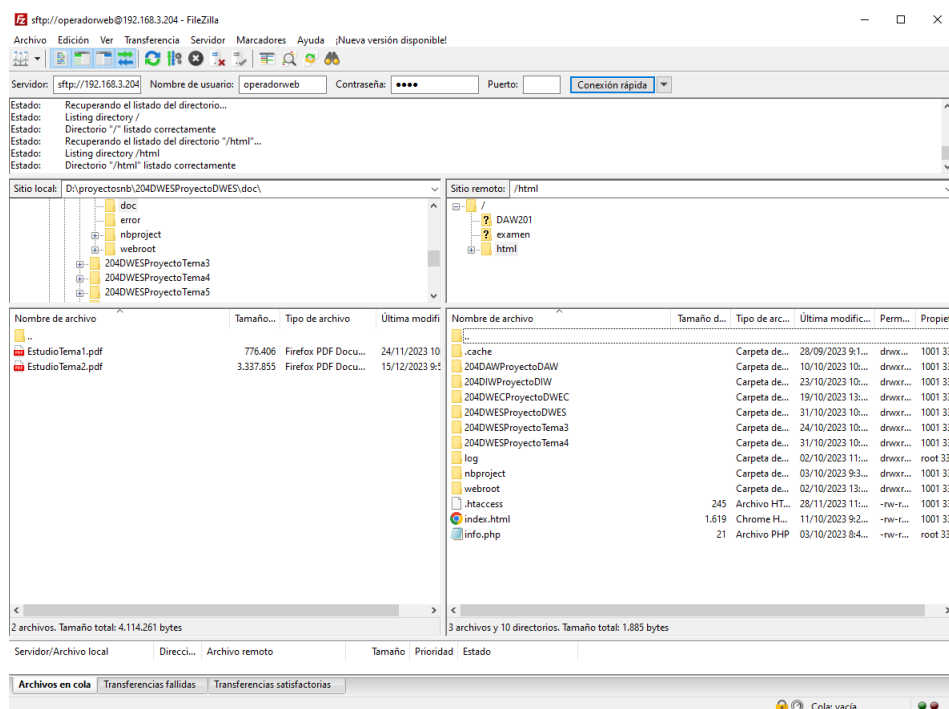
### 4.2 Conexión a un servidor

Para acceder a los ficheros de nuestro servidor web con un entorno gráfico, usaremos FileZilla. Para realizar una conexión tenemos que ir rellenando los campos que aparecen en la parte superior de la ventana siguiendo estas pautas:



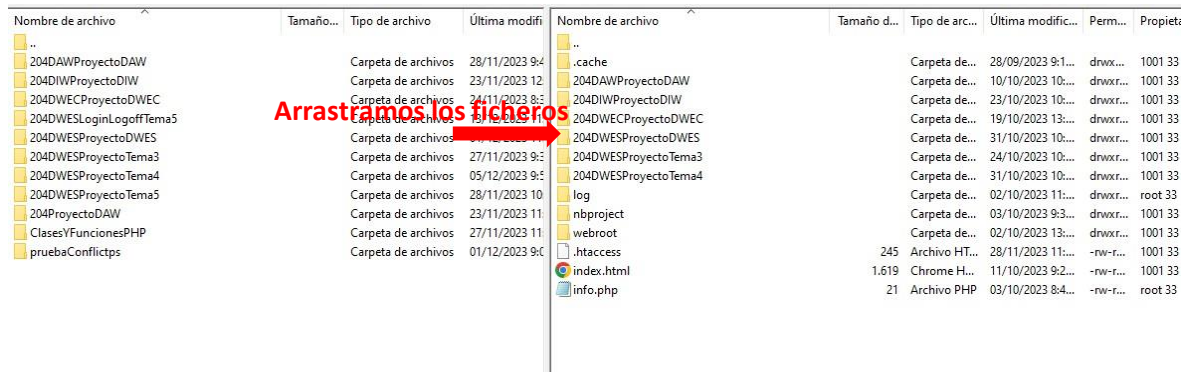
- En campo “Servidor” escribimos la IP de nuestro servidor (192.168.1.204)
- En “Nombre de usuario” escribimos nuestra cuenta de desarrollo y [hosting virtual](#) (operadorweb)
- El campo “contraseña” lo rellenamos con la contraseña del usuario (paso)
- En el campo “Puerto” escribimos el puerto 22 que es el que utiliza SSH

Una vez, rellenos los campos, Hacemos click sobre “Conexión Rápida” y en la ventana de la derecha aparece la estructura de directorios que tiene el directorio raíz del servidor entroneo al usuario operadorweb, en este caso como está enjaulado, solo podemos visualizar el interior de la carpeta public\_html.



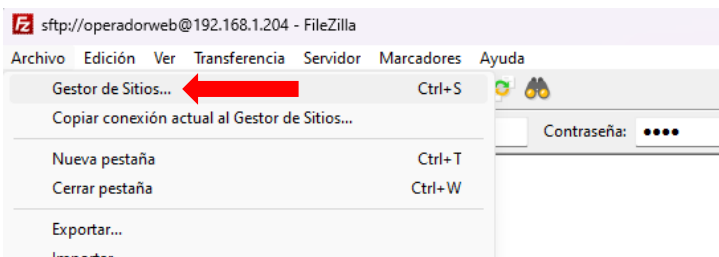
### 4.3 Transferencia de archivos

Para realizar transferencias de archivos tenemos que tener en la ventana de la izquierda nuestro directorio local y en la derecha el directorio correspondiente del servidor. Para subirlos o bajarlos basta con arrastrar los archivos que queramos subir (o descargar) y soltarlos en su destino

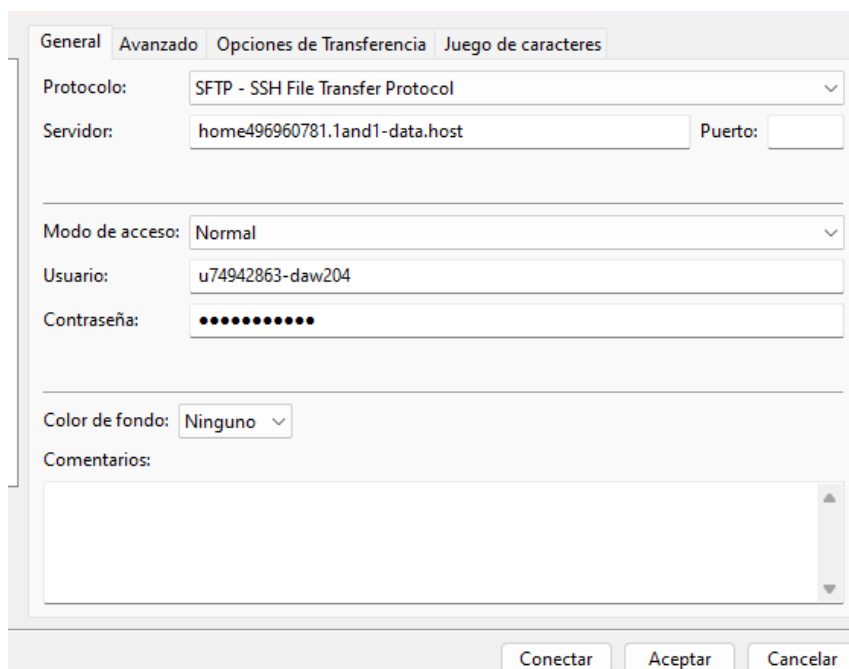


### 4.4 Gestión de sitios

Para hacer una conexión de FileZilla con nuestro EE tenemos que crear un sitio virtual donde se encontrará la configuración de conexión con IONOS. En primer lugar, iremos a la pestaña izquierda superior “Archivo>Gestor de sitios”

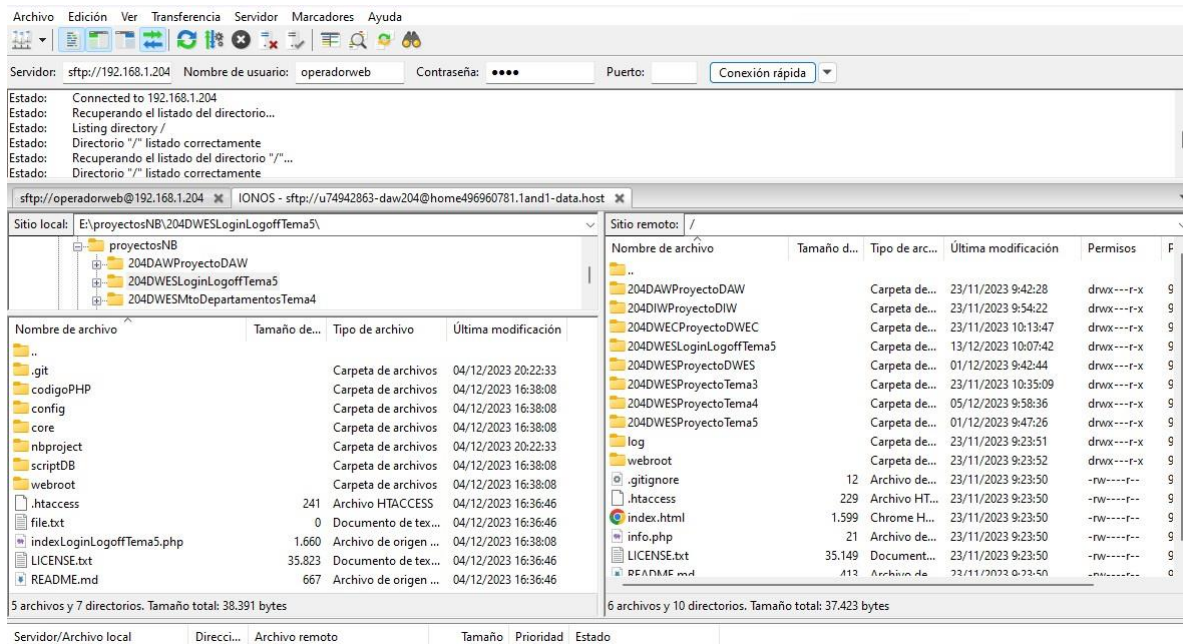


En la nueva ventana, seleccionamos el botón “Nuevo sitio”, elegimos un nombre y completamos la configuración de la siguiente manera:



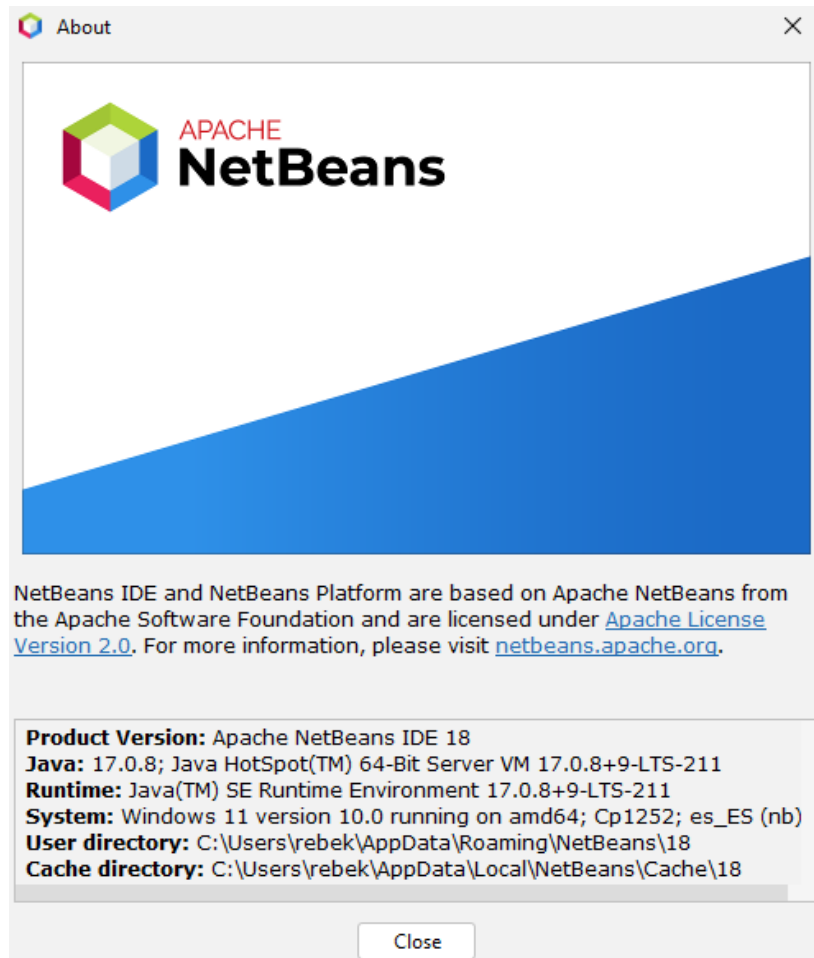


Cuando pulsemos el botón “Conectar” se establecerá la conexión y la interfaz de FileZilla se adaptará al contenido de nuestro servidor de one&one



## 5. NetBeans

### 5.1 Version



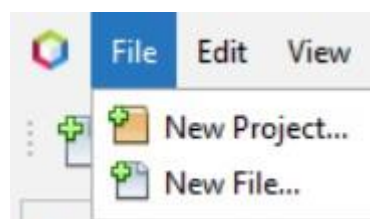
### 5.1 Instalación y configuración inicial (plugings)

??????? No la hemos realizado en clase ???????

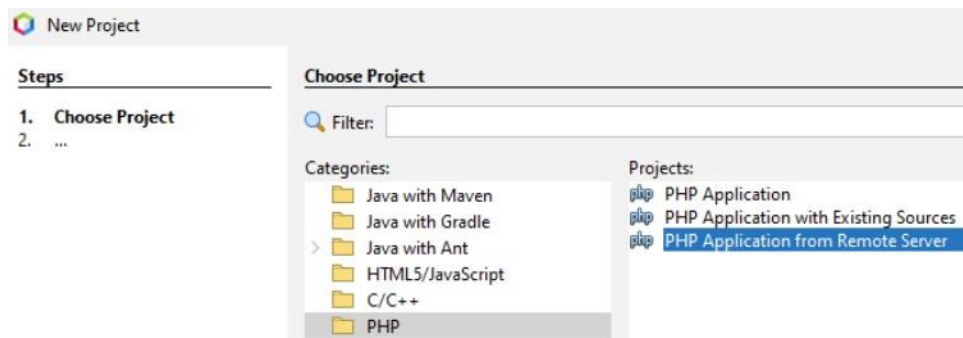
### 5.2 Creación de proyectos, modificación, borrado, prueba

Para crear un proyecto en NetBeans basado en una arquitectura AMP seguiremos los siguientes pasos: (capturas de pantalla por Carlos García Cachón)

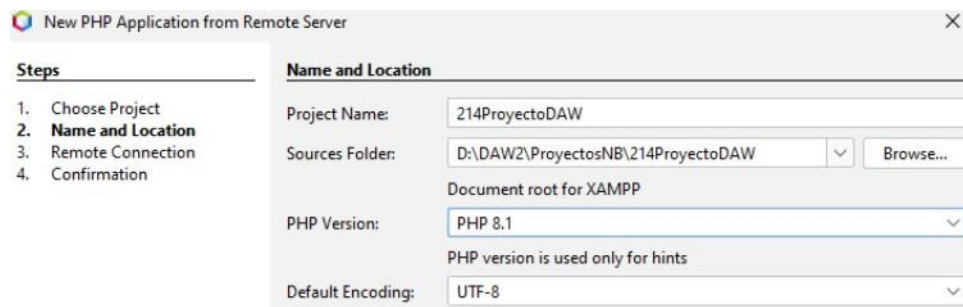
Seleccionamos "File>New Project"



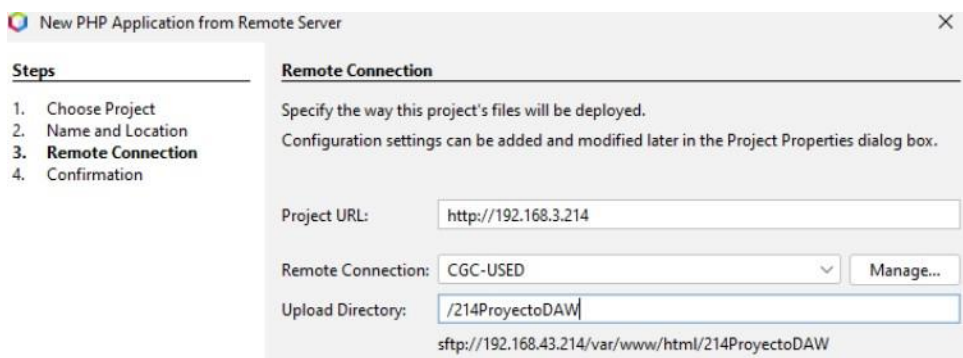
En la siguiente ventana, elegiremos PHP como lenguaje de programación y la opción "Aplicación PHP desde servidor remoto"



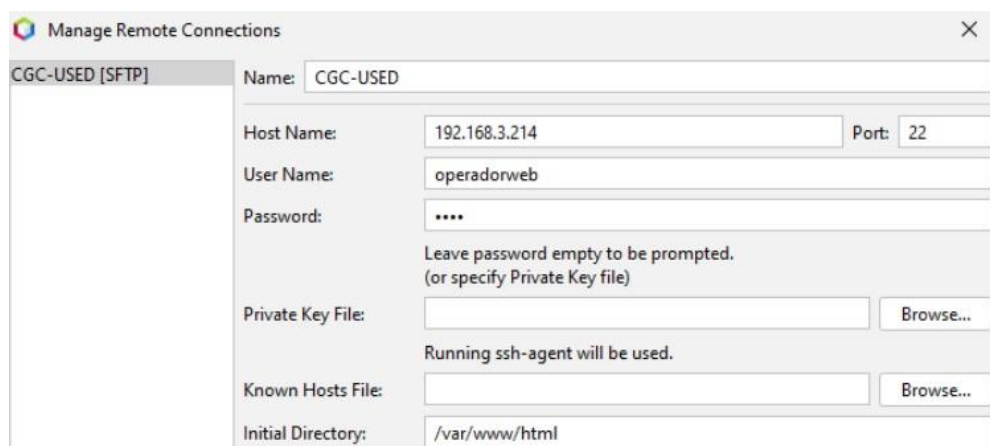
A continuación, elegiremos el nombre del proyecto, la carpeta contenedora del proyecto en nuestra maquina anfitriona y la version de php que vamos a utilizar (en nuestro caso la 8.1)



En la siguiente ventana, introducimos la IP de nuestro servidor, elegimos la carpeta donde se alojará el proyecto en el servidor (tendrá en mismo nombre que el proyecto) y creamos una nueva conexión a un servidor remoto

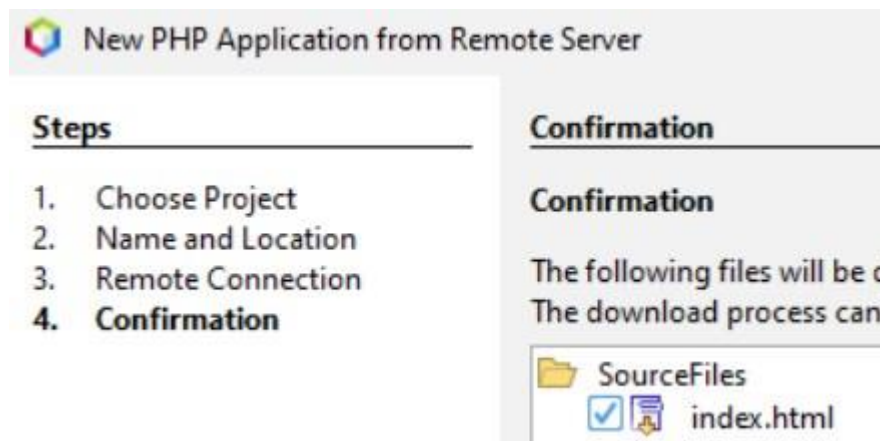


Aquí se muestran los parámetros que debe de tener nuestra conexión:

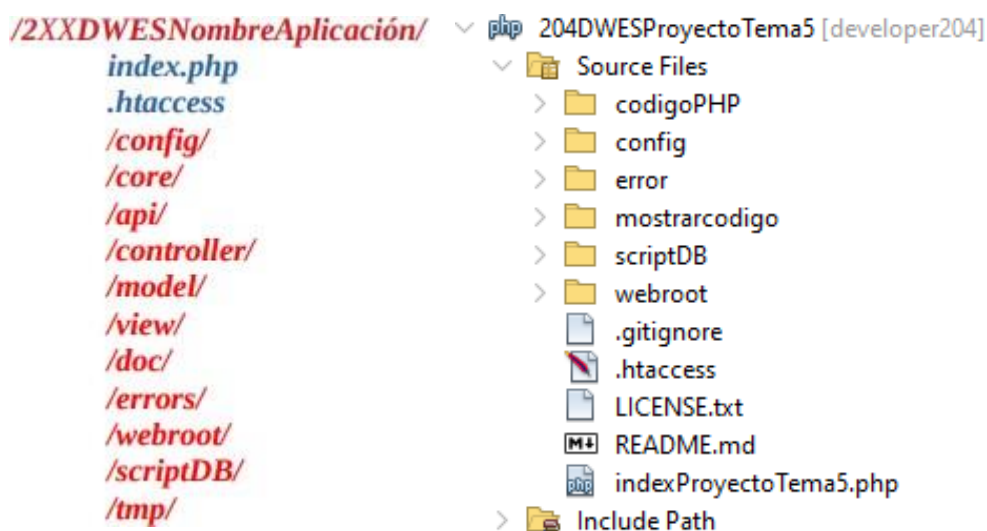


- Name: el nombre personalizado de la conexión
- Host Name: IP del servidor
- Port: 22 (por el que establece la conexión SSH)
- User Name: el nombre del usuario administrador del hosting virtual
- Password: contraseña del usuario
- Initial Directory: directorio padre

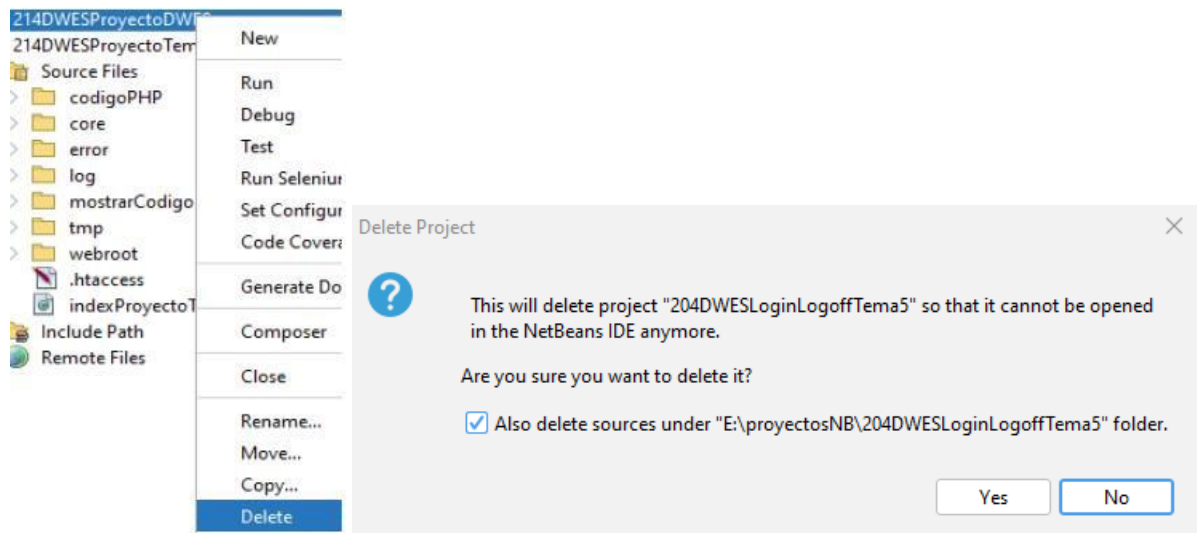
Hacemos un “Test Connection” y si todo ha salido bien, seguiremos con la creación del proyecto. En la siguiente ventana se conectará al servidor y solo te podrá dejar crear el proyecto si ese directorio (html/204DAWProyectoDAW) en el servidor contiene algún archivo. Para solucionar esto, podemos crear un fichero auxiliar (index.html o file.txt) desde FileZilla para más tarde eliminarlo



Una vez creado comenzaremos a realizar modificaciones para que la estructura de directorios del proyecto se adapte a los estándares y en los archivos de código para darle funcionalidad al proyecto

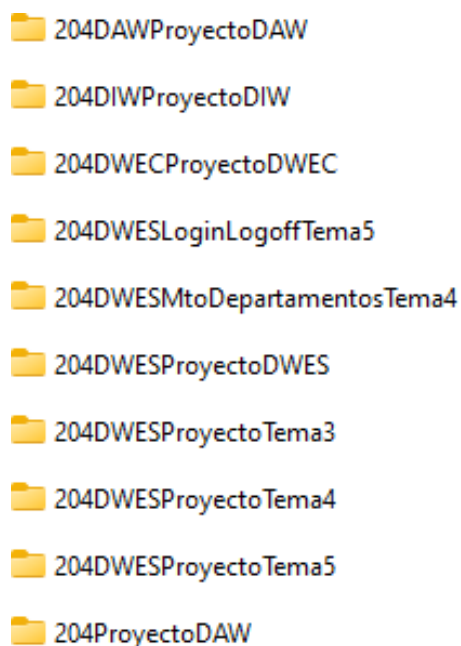


Para realizar el borrado de proyectos es tan simple como hacer click derecho encima del proyecto y seleccionar la opción “delete” (es muy importante que seleccionemos la opción de eliminado seguro para que NetBeans se asegure también de borrar todos los ficheros que se encuentran en el directorio del proyecto en nuestro equipo).

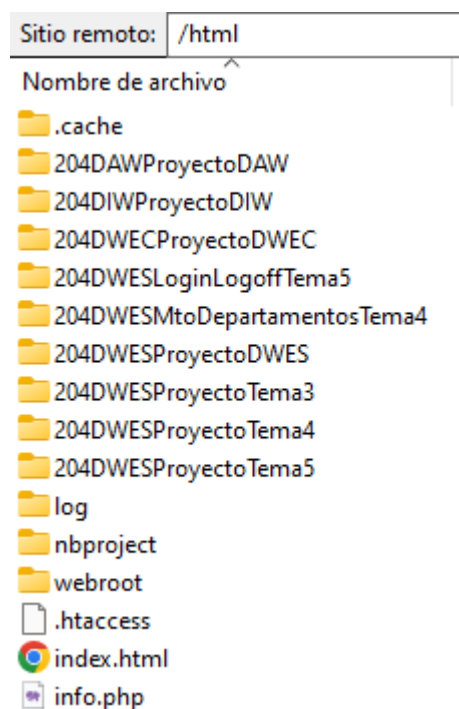


### 5.3 Conexión al servidor remoto SFTP. (Almacenamiento local/almacenamiento remoto)

1. Directorios en local (/proyectosNB)

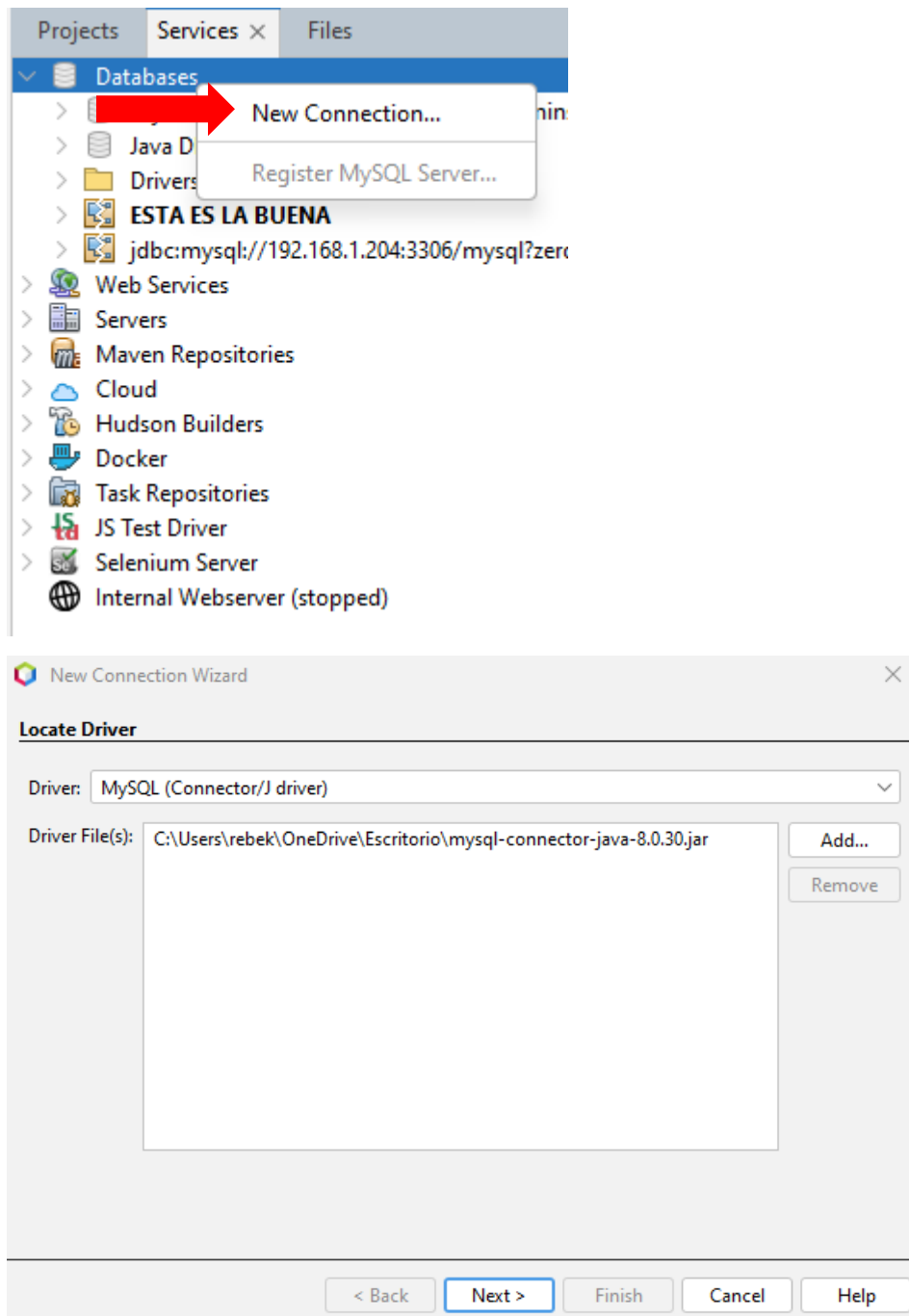


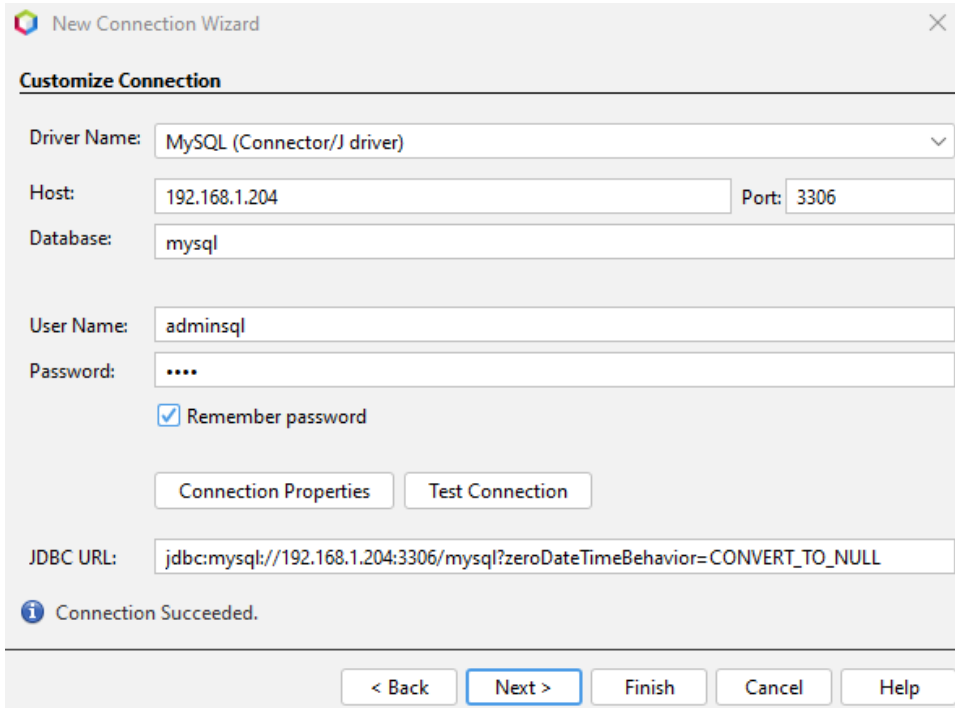
2. Directorios en remoto (/html)



### 5.4 Administración de la base de datos

Dentro de NetBeans nos iremos a la pestaña de "services" del menú de la izquierda y una vez allí, podemos apreciar un apartado llamado "Databases". Para realizar una nueva conexión haremos click derecho sobre este y seleccionamos "New connection"





**New Connection Wizard**

**Customize Connection**

Driver Name: MySQL (Connector/J driver)

Host: 192.168.1.204 Port: 3306

Database: mysql

User Name: adminsql

Password: \*\*\*\*

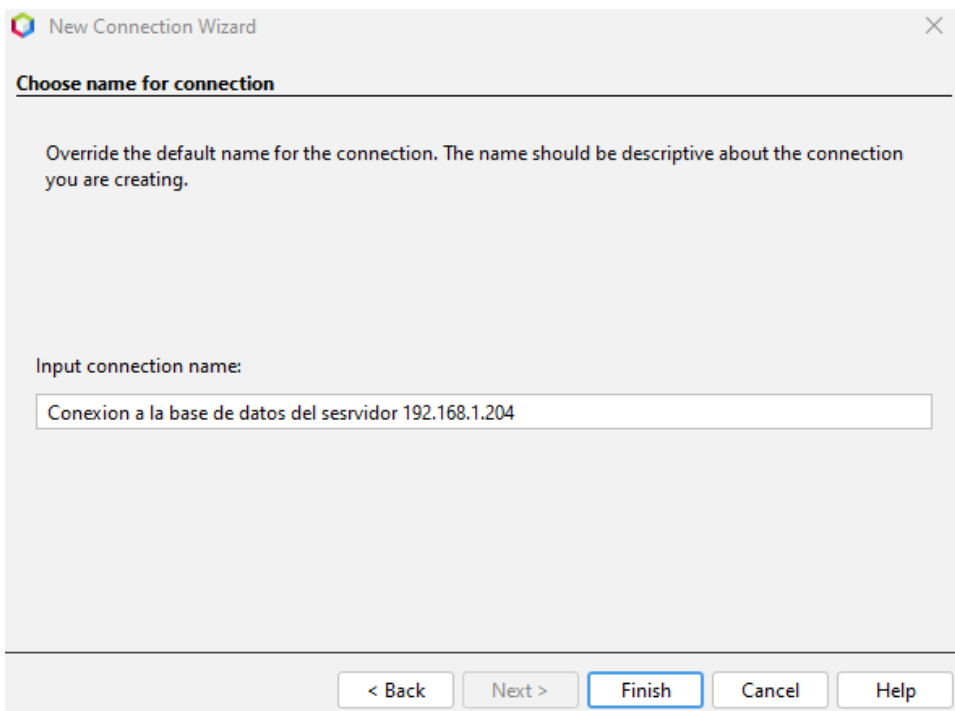
☒ Remember password

Connection Properties Test Connection

JDBC URL: jdbc:mysql://192.168.1.204:3306/mysql?zeroDateTimeBehavior=CONVERT\_TO\_NULL

**i** Connection Succeeded.

< Back Next > Finish Cancel Help



**New Connection Wizard**

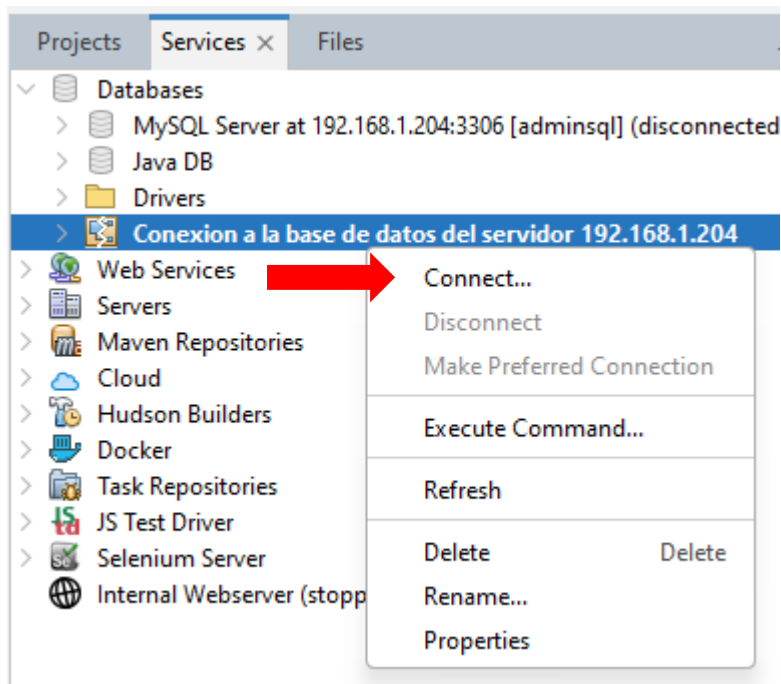
**Choose name for connection**

Override the default name for the connection. The name should be descriptive about the connection you are creating.

Input connection name:

Conexion a la base de datos del servidor 192.168.1.204

< Back Next > Finish Cancel Help



## 5.5 Depuración - Configuración de la ejecución para la depuración

## 5.6 Paso a explotación

Puedes encontrar el proceso de paso a explotación en el apartado de Git de creación de una release

## 5.7 PHP Doc

??????? No la hemos realizado en clase ???????

## 5.8 CSS / JS / AJAX / XML / JSON

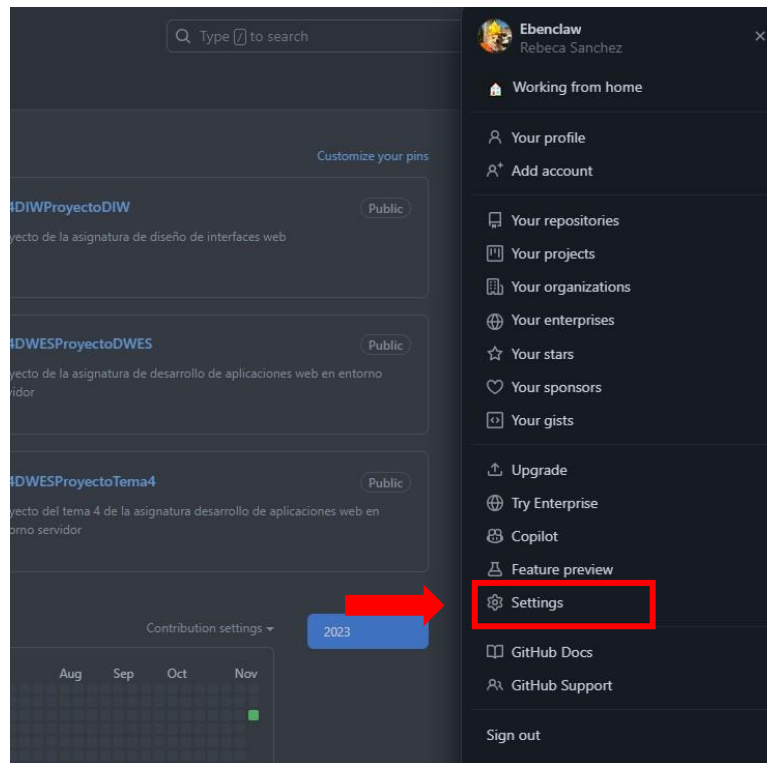
??????? No la hemos realizado en clase ???????



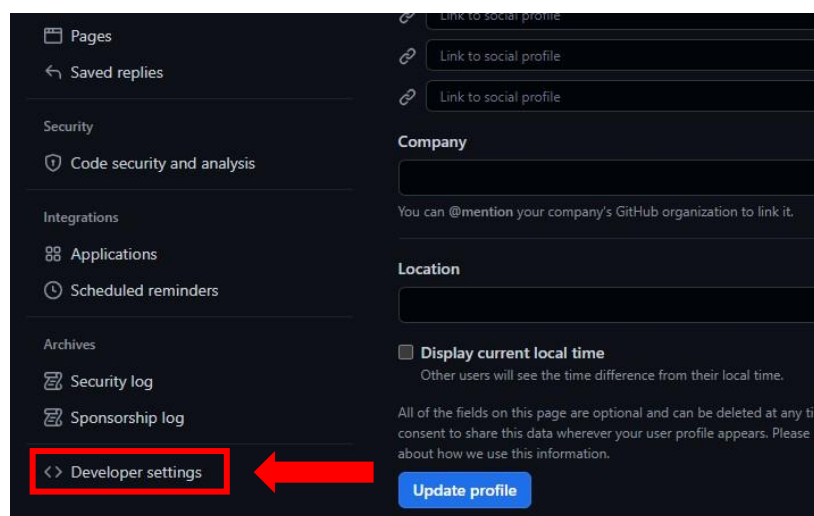
## c. GITHUB – INTERNET

### 1. Creación de token desde GitHub

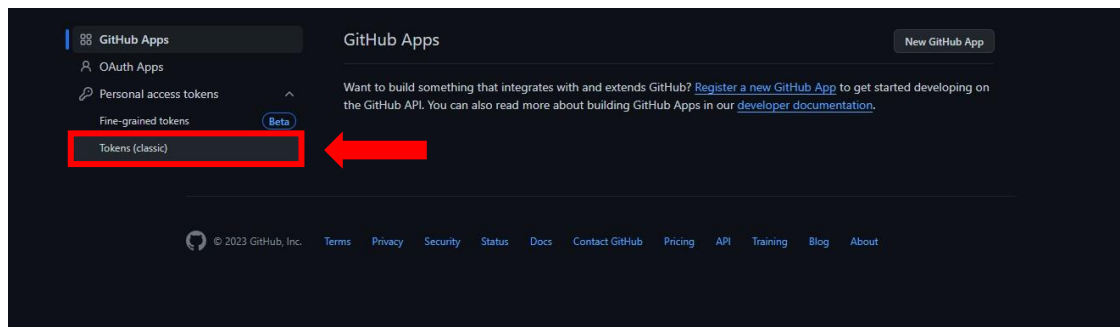
Para comenzar a trabajar en GitHub, necesitamos tener una cuenta registrada y una vez la tengamos, comenzaremos creando un token. Un token es una contraseña de autenticación usada para manejar la API de GitHub, para crearlo nos dirigiremos a la derecha de la parte superior de la página donde aparece nuestro perfil y cuando le hagamos clic, se desplegará un menú con distintas opciones. Nos dirigiremos a la opción de “settings”



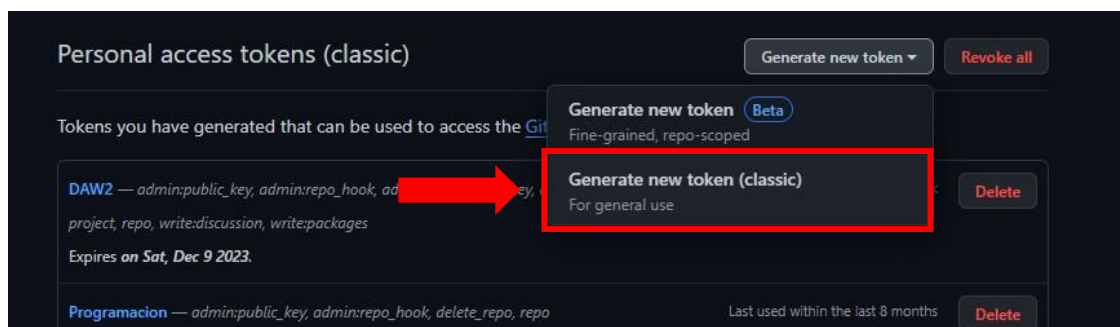
En la siguiente ventana, nos desplazaremos hasta la parte inferior y seleccionaremos la opción “developer settings” del menu de la izquierda



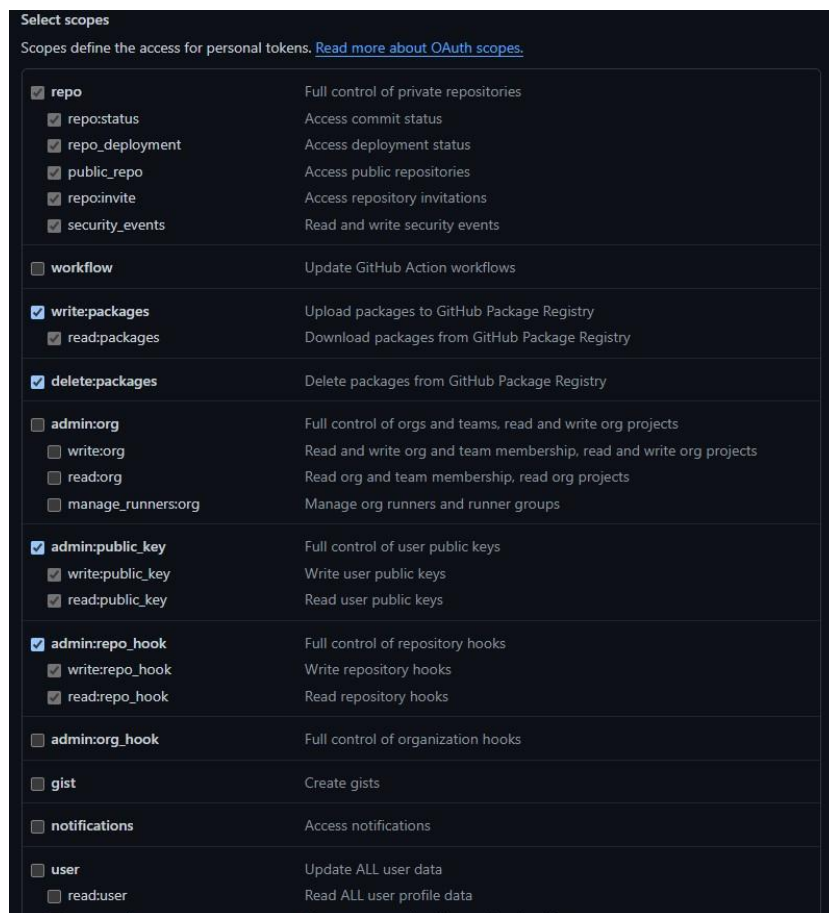
Nos dirigimos a “Personal access tokens>Tokens (clasic)”



A continuación, nos aparecerá la lista de nuestros tokens. Para crear uno nuevo, desplegamos el botón “Generate new token” y seleccionamos la opción de “Generate new token (classic)”



Aparecerá una ventana en la elegiremos el nombre del token, el tiempo de expiración y las especificaciones. Seguiremos las siguientes especificaciones:



<input type="checkbox"/> user	Update ALL user data
<input type="checkbox"/> read:user	Read ALL user profile data
<input type="checkbox"/> user:email	Access user email addresses (read-only)
<input type="checkbox"/> user:follow	Follow and unfollow users
<input checked="" type="checkbox"/> delete_repo	Delete repositories
<input checked="" type="checkbox"/> write:discussion	Read and write team discussions
<input checked="" type="checkbox"/> read:discussion	Read team discussions
<input type="checkbox"/> admin:enterprise	Full control of enterprises
<input type="checkbox"/> manage_runners:enterprise	Manage enterprise runners and runner groups
<input type="checkbox"/> manage_billing:enterprise	Read and write enterprise billing data
<input type="checkbox"/> read:enterprise	Read enterprise profile data
<input type="checkbox"/> audit_log	Full control of audit log
<input type="checkbox"/> read:audit_log	Read access of audit log
<input type="checkbox"/> codespace	Full control of codespaces
<input type="checkbox"/> codespace:secrets	Ability to create, read, update, and delete codespace secrets
<input type="checkbox"/> copilot	Full control of GitHub Copilot settings and seat assignments
<input type="checkbox"/> manage_billing:copilot	View and edit Copilot for Business seat assignments
<input checked="" type="checkbox"/> project	Full control of projects
<input checked="" type="checkbox"/> read:project	Read access of projects
<input type="checkbox"/> admin:gpg_key	Full control of public user GPG keys
<input type="checkbox"/> write:gpg_key	Write public user GPG keys
<input type="checkbox"/> read:gpg_key	Read public user GPG keys
<input checked="" type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input checked="" type="checkbox"/> write:ssh_signing_key	Write public user SSH signing keys
<input checked="" type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys

Generate token Cancel


Una vez seleccionadas las características, generamos el token y nos aparecerá la ventana anterior con la clave del nuevo token. Esta clave será la contraseña que nos pedirán siempre que intentemos realizar cambios en el repositorio remoto, por lo cual, es muy importante copiarla en un fichero y guardarla para no perderla.

Personal access tokens (classic)

Generate new token ▼ Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp\_Q5fM6ZL31nupTYYJDjtuUtk7xySQ410JB4eK 

Delete

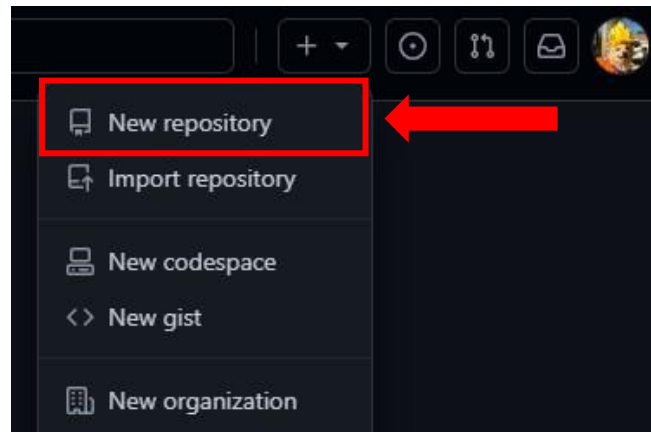
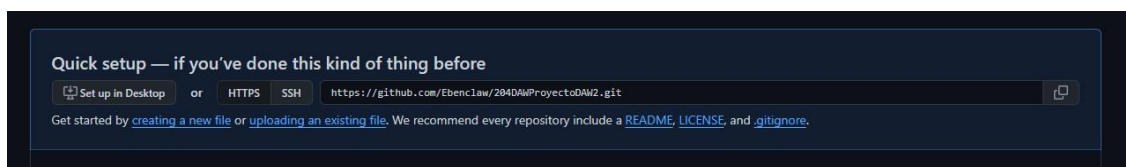
## 2. Comandos básicos de Git

A continuación, se muestra una lista con los comandos básicos que podemos usar en git bash o en NetBeans:

- git init creará un nuevo repositorio local GIT.
- git clone se usa para clonar un repositorio en local
- git add se usa para agregar archivos al área de preparación o staging
- git commit creará una instantánea de los cambios y la guardará en el directorio git
- git status muestra la lista de los archivos que se han cambiado junto con los archivos que están por ser preparados o confirmados
- git push se usa para enviar confirmaciones locales a las ramas del repositorio remoto.
- git checkout cambia el área de trabajo a la rama especificada.
- git branch se usa para listar, crear o borrar ramas
- git pull fusiona todos los cambios que se han hecho en el repositorio remoto con el directorio de trabajo local
- git merge se usa para fusionar una rama con otra rama activa
- git diff se usa para hacer una lista de conflictos
- git log se usa para ver el historial del repositorio listando ciertos detalles de la confirmación
- git fetch le permite al usuario buscar todos los objetos de un repositorio remoto que actualmente no se encuentran en el directorio de trabajo local, es decir, muestra los cambios existentes entre el repositorio remoto y el local

### 3. Creación de un repositorio desde NetBeans

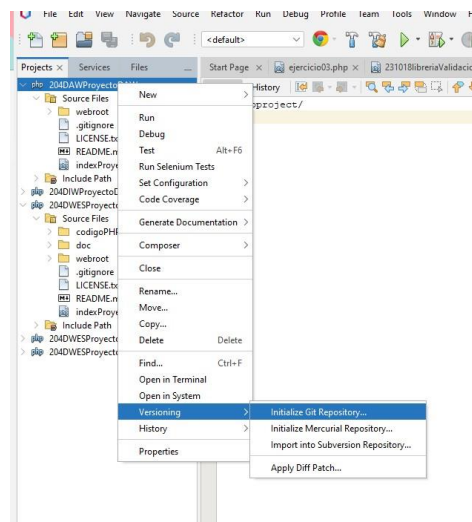
En primer lugar, crearemos un repositorio vacío en GitHub desde nuestro navegador con el nombre que va a tener nuestro proyecto. Para ello hacemos click en el icono de + en la parte superior derecha de la página y seleccionamos “New repository” donde elegiremos el nombre del proyecto y una breve descripción de este. También podemos especificar la privacidad de este, si queremos que sea público o privado, en este caso elegiremos público. Después se nos generará una URL que tendremos que asocia a nuestro repositorio local.

A screenshot of the 'Create a new repository' page on GitHub. The page has a dark theme. At the top, it says 'Create a new repository' followed by a brief explanation. Below this, there are input fields for 'Owner' (set to 'Ebenclaw') and 'Repository name' (set to '204DAWProyectoDAW2'). A message indicates '204DAWProyectoDAW2 is available.' There is a text area for 'Description' with the text 'Proyecto de desarrollo de aplicaciones web'. At the bottom, there are two radio buttons for 'Public' (selected) and 'Private'. The 'Public' option is described as 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option is described as 'You choose who can see and commit to this repository.'

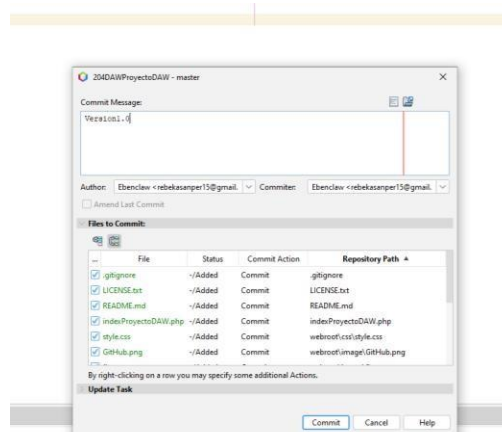
A continuación, desde NetBeans o en la carpeta de local, añadimos un README.md y una LICENCE.txt en cada proyecto y un “.gitignore” con el contenido “/nbproject/” para que no suba esa carpeta (que se genera automáticamente cuando creas un proyecto en NetBeans) al repositorio remoto

codigoPHP	24/11/2023 10:03	Carpeta de archivos	
doc	23/11/2023 10:17	Carpeta de archivos	
error	24/11/2023 9:59	Carpeta de archivos	
nbproject	11/10/2023 9:33	Carpeta de archivos	
webroot	11/10/2023 9:33	Carpeta de archivos	
.gitignore	14/11/2023 9:07	Documento de te...	1 KB
.htaccess	24/11/2023 9:59	Archivo HTACCESS	1 KB
indexProyectoDWES.php	24/11/2023 10:03	Archivo PHP	3 KB
LICENSE.txt	14/11/2023 8:45	Documento de te...	35 KB
README.md	23/11/2023 10:17	Archivo MD	1 KB

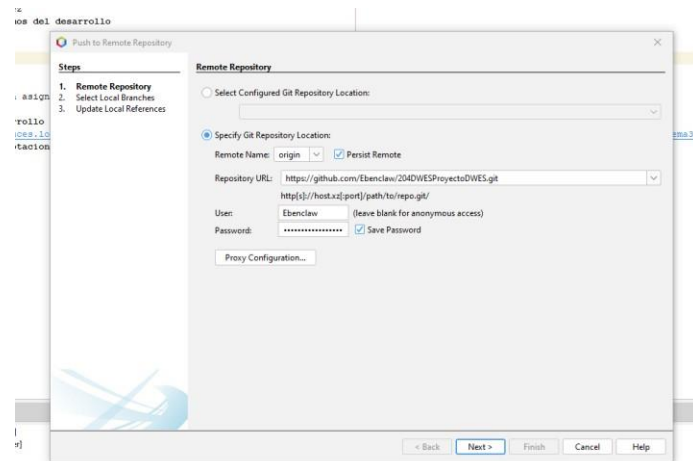
Desde NetBeans, abrimos el proyecto que deseamos versionar y le hacemos click derecho>"versioning">"Initialize Git Repository"



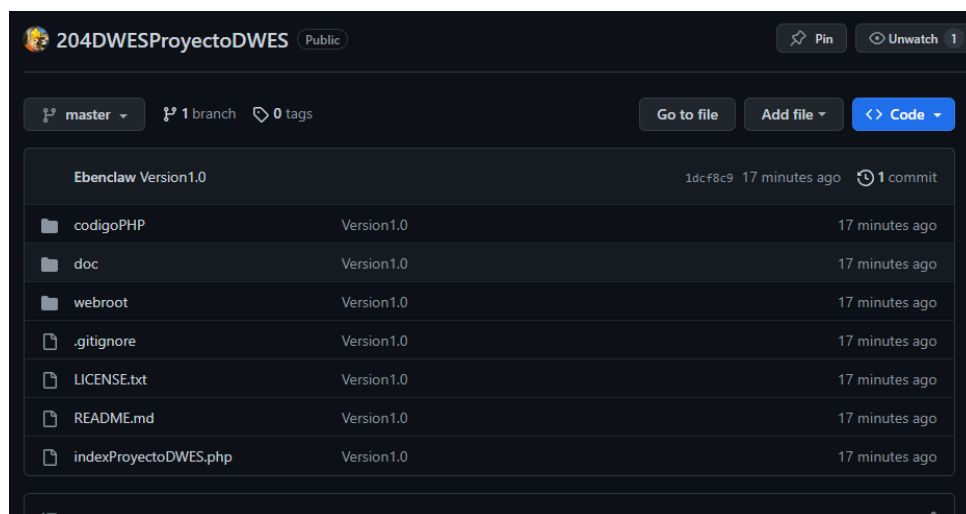
hacemos el primer commit en master con la Version1.0 con click derecho>Git>Commit. Podemos escribir un pequeño texto en el que aparezca el contenido de la primera version.



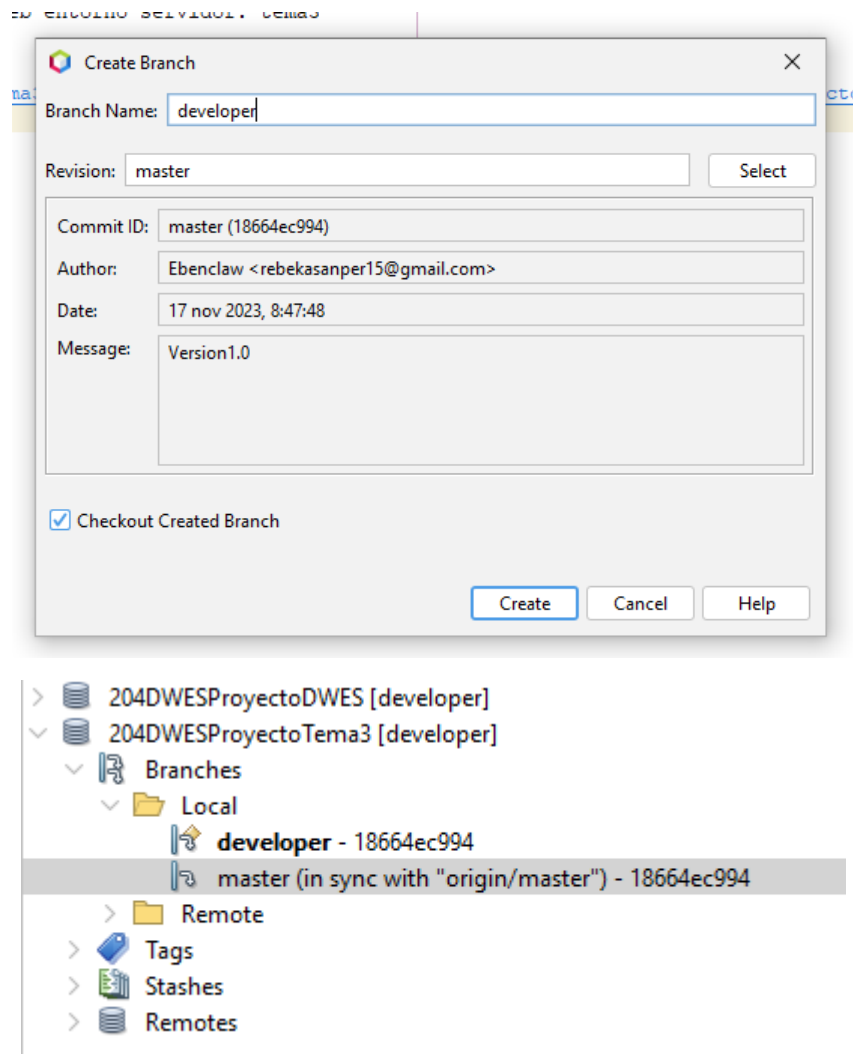
Una vez hecho el commit, hacemos un push al repositorio de GitHub indicando la URL del proyecto que habíamos creado vacío, el usuario y la password que será nuestro token



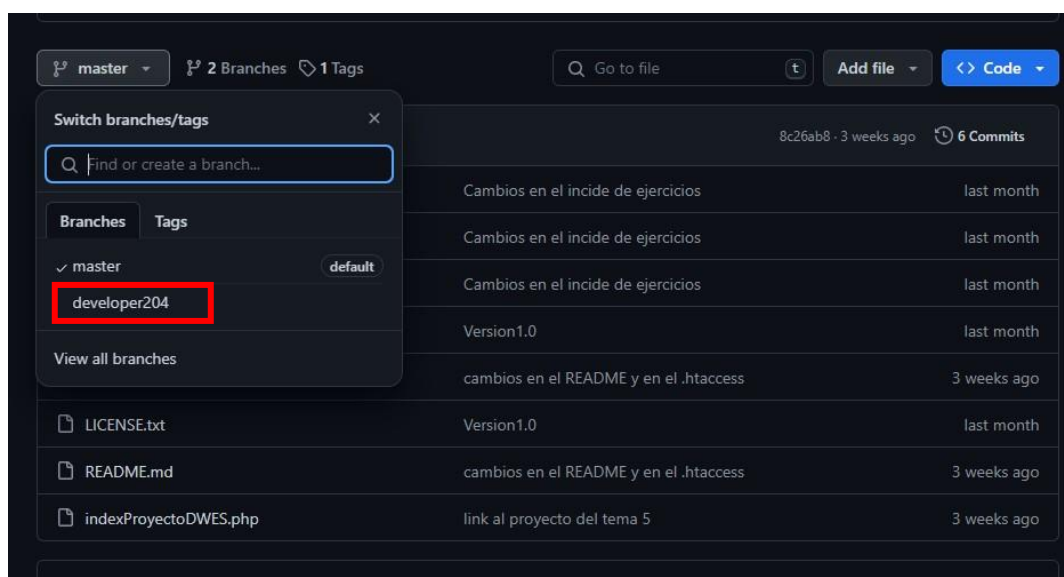
Comprobamos que el proyecto se ha subido correctamente a GitHub



Creamos una rama a partir de la version 1.0 de la master en local llamada “developer204” con `git>branch/tag>create branch`. Esta rama la usaremos para publicar todos los commit que realicemos en el desarrollo. Para movernos entre ramas usaremos la opción checkout



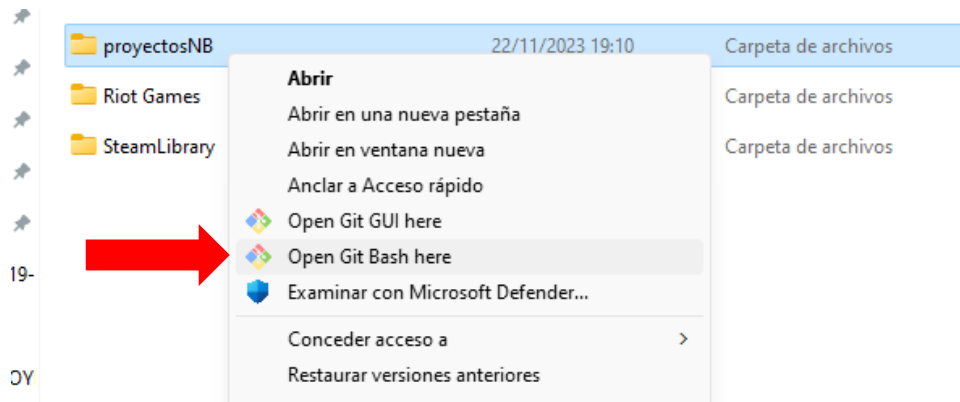
Dentro de esa rama realizamos cambios, cuando el proyecto este actualizado hacemos commit y después hacemos un push para actualizar el repositorio remoto con la rama nueva



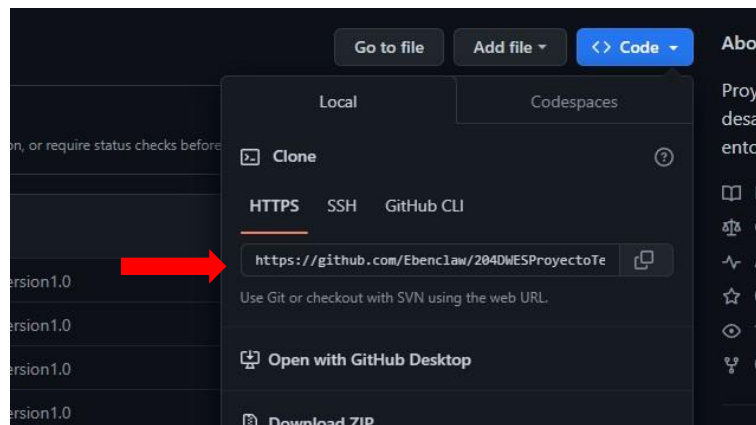


## 4. Clonar un repositorio

Si queremos clonar un repositorio remoto a nuestro repositorio local, en primer lugar, haremos click derecho sobre la carpeta padre en la que queremos que esté situado nuestro proyecto y seleccionamos “open Git Bash here”.

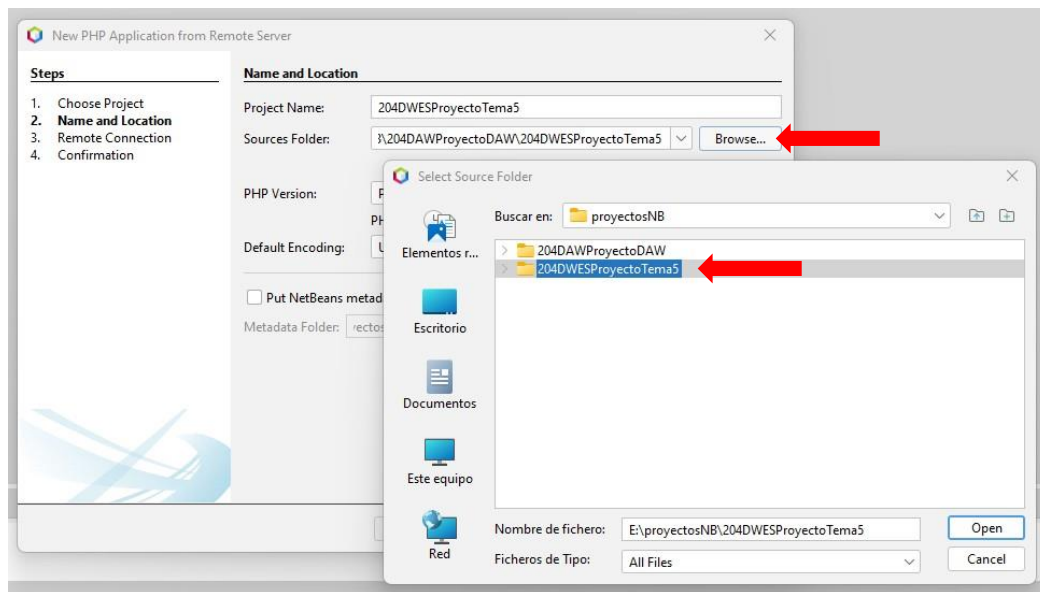


A continuación, se abrirá el cmd de Windows y escribiremos “git clone <<url de nuestro repositorio remoto>>”. La url de nuestro repositorio ponemos copiándola desde la barra de navegación de nuestro navegador o desde la ventana “code” del proyecto en GitHub.

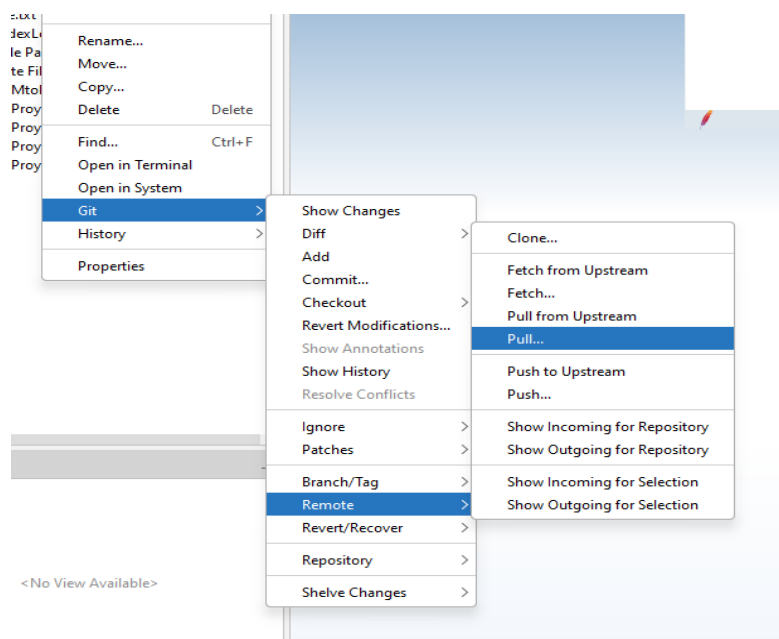


```
rebek@Pc_de_Rebeca MINGW64 /e/proyectosNB
$ git clone https://github.com/Ebenc1aw/204DWESProyectoTema5.git
Cloning into '204DWESProyectoTema5'...
remote: Enumerating objects: 45, done.
remote: Counting objects: 100% (45/45), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 45 (delta 12), reused 42 (delta 9), pack-reused 0
Receiving objects: 100% (45/45), 2.48 MiB | 1.65 MiB/s, done.
Resolving deltas: 100% (12/12), done.
```

Una vez clonado, iremos a NetBeans y crearemos un nuevo proyecto como hasta ahora, pero seleccionaremos el repositorio clonado cuando nos pida la localización del proyecto



Con el proyecto en NetBeans abierto, haremos click derecho “versioning>remote>fetch” para comparar los commit que hay en el repositorio remoto. Para descargar las ramas remotas en las ramas locales en las que queremos hacer cambios, usamos el comando `git pull` sobre la rama a desarrollar



## 5. Crear una version estable en la rama master

Cuando hemos alcanzado una version estable de nuestra aplicación, es el momento de pasarla al entorno de explotación, pero no subiremos un commit cualquiera, sino aquel que ha sido mergeado a la rama master e identificado como una version estable. Lo realizaremos de la siguiente manera:

Desde git bash cambiamos a la rama master con “git checkout master” (nos colocamos en la rama destino en la que se va a mergear la version) y realizamos el comando “git merge --no-ff developer204” (el comando --no-ff significa que cuando realice el merge a su vez haga un commit)

```
daw2@IS32WX04 MINGW64 /d/proyectosnb/204DWESProyectoDWES (developer204)
$ git branch
* developer204
  master

daw2@IS32WX04 MINGW64 /d/proyectosnb/204DWESProyectoDWES (developer204)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

daw2@IS32WX04 MINGW64 /d/proyectosnb/204DWESProyectoDWES (master)
$ git merge --no-ff developer204
Merge made by the 'ort' strategy.
 .htaccess          | 5 +++++
 README.md          | 3 ++-
 codigoPHP/indexTema1.php | 6 +++--
 codigoPHP/indexTema2.php | 7 +++---
 doc/EstudioTema2.pdf   | Bin 930044 -> 1671719 bytes
 indexProyectoDWES.php | 18 ++++++++-----
 webroot/css/style.css  | 24 ++++++++-----
 7 files changed, 46 insertions(+), 17 deletions(-)
 create mode 100644 .htaccess
```

A continuación, cambiamos a la rama developer con un checkout y seguido de esto hacemos un “git merge --ff master” para colocar el cursor sobre la version estable de la rama master. Esta es la forma más idónea de mover el puntero gracias al comando --ff que anula el autocommit cuando hacemos un merge. Podemos escribir “git log --graph --all” para mostrar una línea temporal muy gráfica en la que se distinguen las ramas y todas las versiones

```
daw2@IS32WX04 MINGW64 /d/proyectosnb/204DWESProyectoDWES (master)
$ git branch
  developer204
* master

daw2@IS32WX04 MINGW64 /d/proyectosnb/204DWESProyectoDWES (master)
$ git checkout developer204
Switched to branch 'developer204'
Your branch is ahead of 'origin/developer' by 2 commits.
(use "git push" to publish your local commits)

daw2@IS32WX04 MINGW64 /d/proyectosnb/204DWESProyectoDWES (developer204)
$ git merge --ff master
Updating ee0a19f..8c26ab8
Fast-forward

daw2@IS32WX04 MINGW64 /d/proyectosnb/204DWESProyectoDWES (developer204)
$ git log --graph --all
* commit 8c26ab8dd6c9a1d7b4394dec736c0618a47b2bea (HEAD -> developer204, master)
| \ Merge: 1dcf8c9 ee0a19f
| | Author: Ebenclaw <rebekasanper15@gmail.com>
| | Date: Thu Nov 23 10:17:50 2023 +0100
| |
| | Merge branch 'developer204'
```

Para terminar, subiremos los cambios al repositorio de GitHub con “`git push origin -u --all`”

```
* commit f418c80450e71bf8caa609abc0169288d342e7dd
/ Author: Ebenclaw <rebekasanper15@gmail.com>
  Date:   Fri Nov 17 09:09:40 2023 +0100

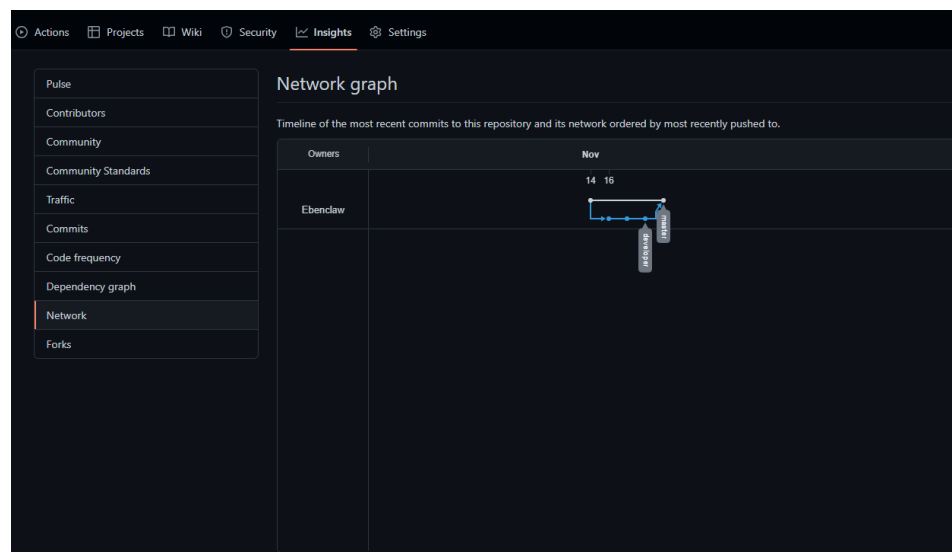
    Primer commit en developer

* commit 1dcf8c9752f6780cd6a3182fcfef77daf86553bd (origin/master)
  Author: Ebenclaw <rebekasanper15@gmail.com>
  Date:   Fri Nov 17 08:38:35 2023 +0100

    Version1.0

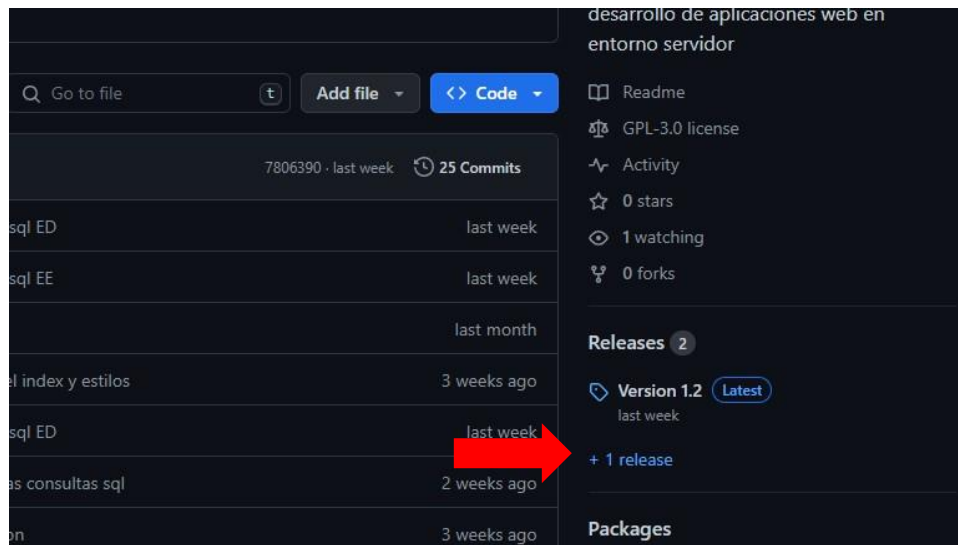
daw2@IS32WX04 MINGW64 /d/proyectosnb/204DWESProyectoDWES (developer204)
$ git push origin -u --all
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 227 bytes | 227.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Ebenclaw/204DWESProyectoDWES.git
   ee0a19f..8c26ab8  developer204 -> developer204
   1dcf8c9..8c26ab8  master -> master
branch 'developer204' set up to track 'origin/developer204'.
branch 'master' set up to track 'origin/master'.
```

Comprobamos que se ha subido a GitHub correctamente, para ello existe la herramienta “Network” en las estadísticas de los proyectos que nos muestra un diagrama de flujo muy visual

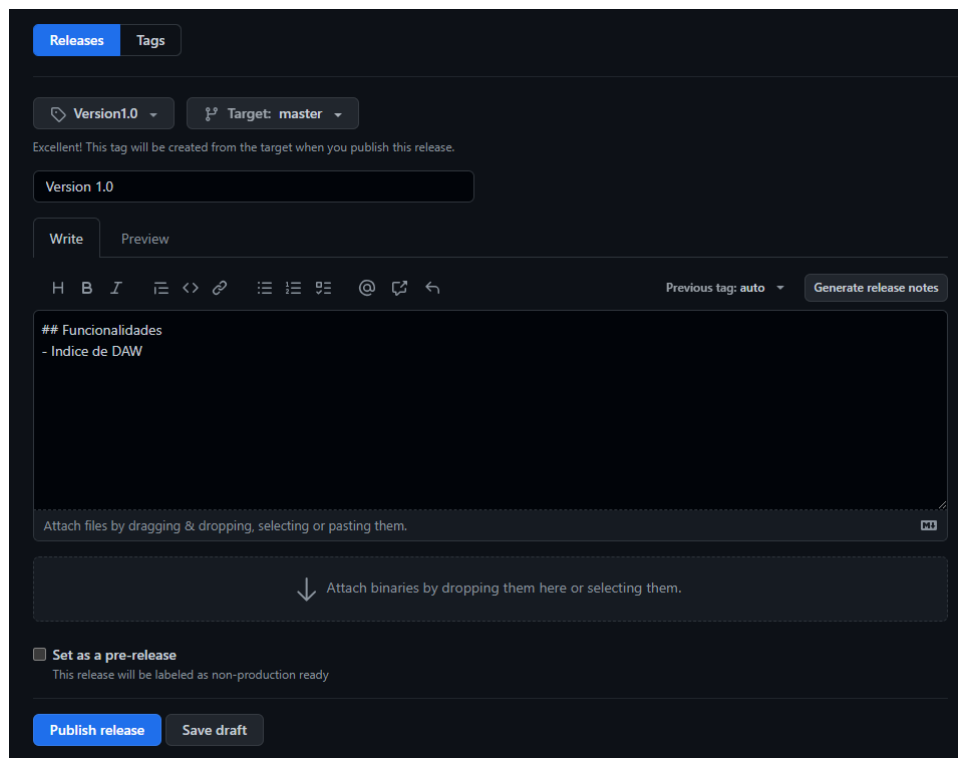


## 6. Crear una release

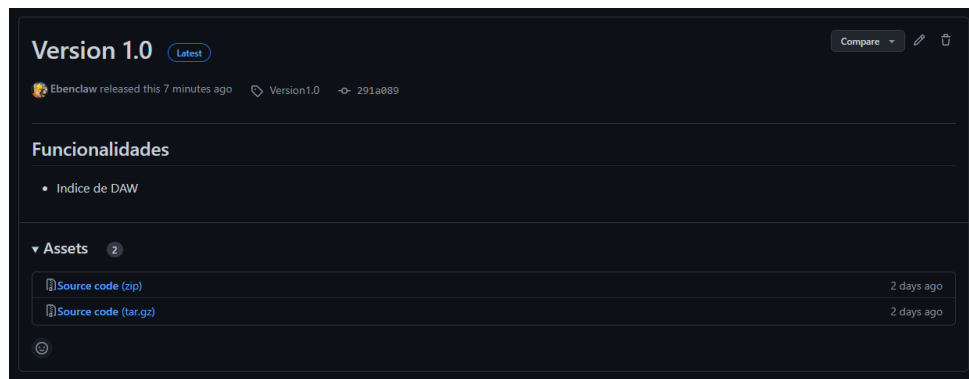
Para crear una release de una version estable de nuestra app, iremos a nuestro proyecto de GitHub a la parte de la derecha donde aparece el título Release y “create a new release”



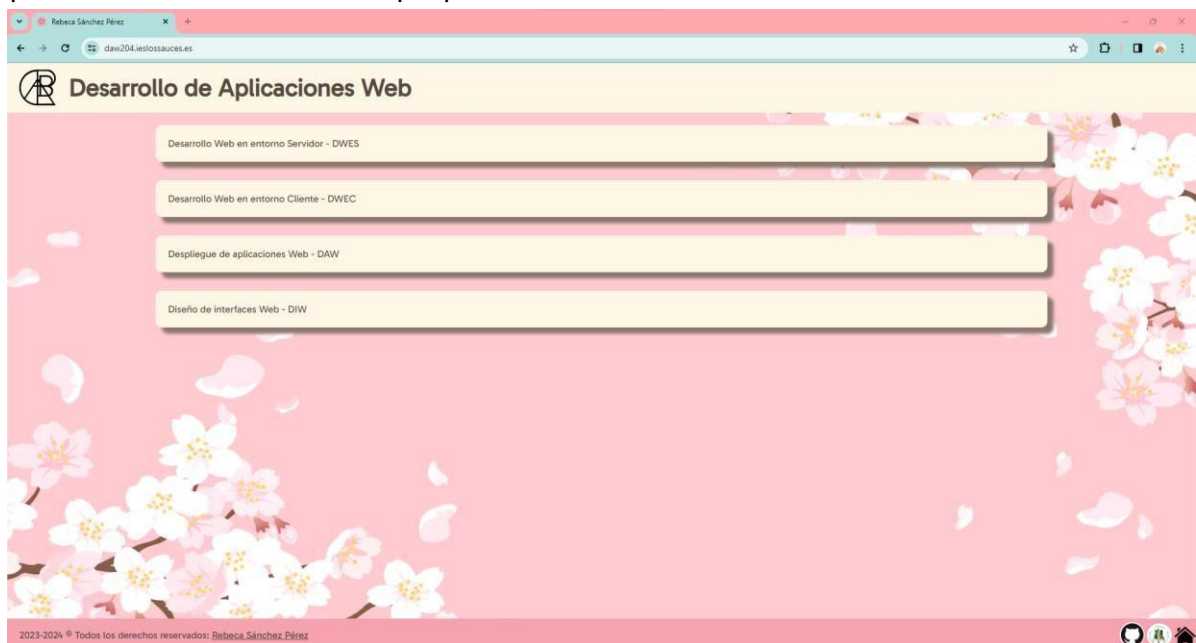
En la siguiente ventana seleccionamos una Tag (si es nuestra primera release, tendremos que crearla), elegiremos un nombre para nuestra release (en nuestro caso la llamaremos como el nombre de la version) y detallaremos una breve descripción a cerca de los cambios que se ha llevado a cabo en esa version del código



Cuando este creada, se generarán 2 ficheros, uno con extensión zip y otro tar.gz. Descargamos el zip y lo descomprimos en nuestro repositorio local. Esta carpeta corresponde a las líneas de código del último commit que ha sido mergeado en la rama master. Esta carpeta es la que desplegaremos en el entorno de explotación mediante [filezilla](#)



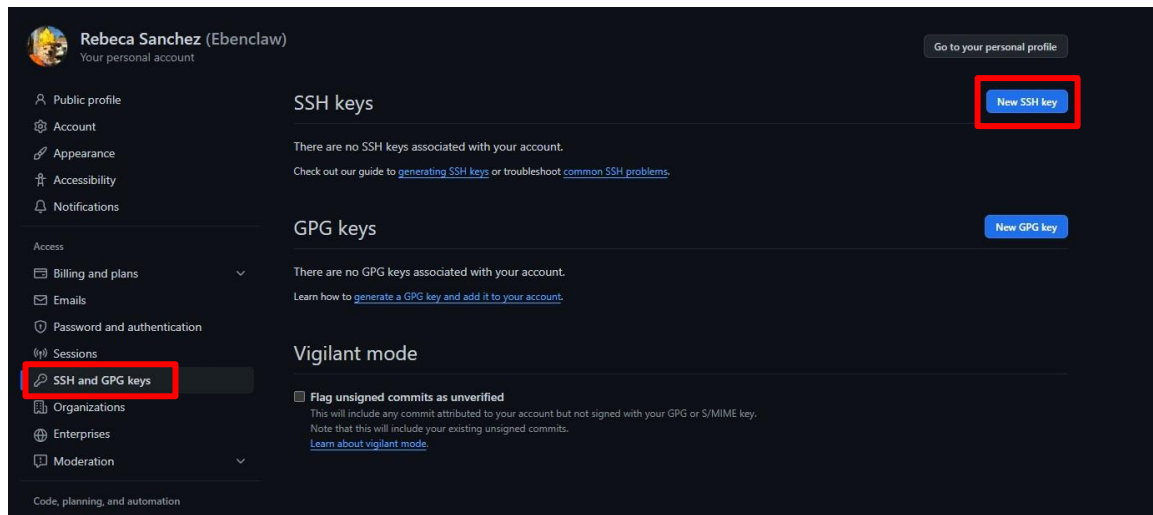
Para comprobar que la release se ha subido correctamente al entorno de explotación, escribiríamos en el navegador "daw204.ieslossauces.es" y desde ese dominio comprobarías que esa es la version estable que pretendías subir



## 7. Conexión con SSH

### 7.1 Generar clave SSH




La conexión SSH nos permite vincular nuestros repositorios de manera segura utilizando SSH mediante una clave publica y una clave privada. Para generarla, abriremos el menú de “Settings”, la opcion “SSH and GPG keys” y seleccionamos “New SSH key”



Nombraremos a nuestra clave con el título que queramos y seleccionaremos el tipo de clave de autenticación. En el recuadro de abajo tenemos que escribir nuestra clave privada pero como en la maquina anfitriona no tenemos, la generaremos con el comando “`ssh-keygen -t ed25519 -C tucorreoelectronico@hotmail.com`” desde el cmd de Windows y se generarán 2 claves (pública y privada) en el directorio que te especifica la sinopsis.

```
daw2@IS32WX04 MINGW64 ~  
$ ssh-keygen -t ed25519 -C "rebeca.sanper.1@educa.jcyl.com"  
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/c/Users/daw2/.ssh/id_ed25519):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /c/Users/daw2/.ssh/id_ed25519  
Your public key has been saved in /c/Users/daw2/.ssh/id_ed25519.pub  
The key fingerprint is:  
SHA256:t0u97Sn84uRernAKixJ2m5nHPRvyjRda9ul7g3Xns5I rebeca.sanper.1@educa.jcyl.com  
The key's randomart image is:  
+--[ED25519 256]--+  
|  
|      S .  
|    o . . o+ ..o  
|  . o *O.==+E++o  
|  . =.O*+@=*-oo  
|    .... **BBO+o  
+-----[SHA256]-----+
```

Abriremos el fichero que contiene la clave pública y la copiaremos en el recuadro de antes de GitHub

 id_ed25519	01/12/2023 9:20	Archivo	1 KB
 id_ed25519.pub	01/12/2023 9:20	Documento de Mi...	1 KB
 known_hosts	26/10/2023 8:56	Archivo	1 KB

### Add new SSH Key

**Title**

**Key type**  

Authentication Key

**Key**  

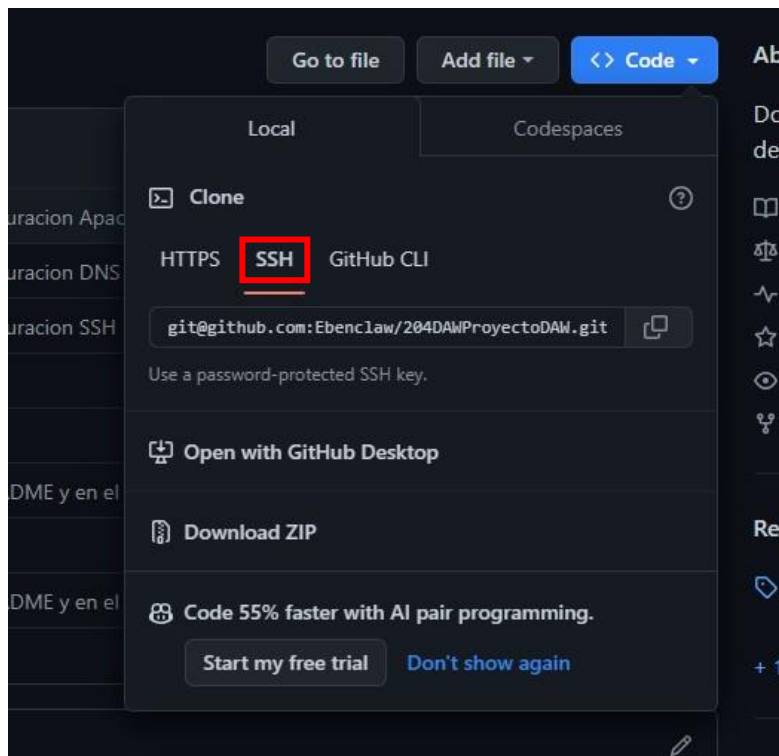
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFOLvgJknjVXC5DXNwsrncLJP51bEI7UBsvYN9StN9ke rebeca.sanper.1@educa.jcyl.com

Add SSH key



## 7.2 Conexión SSH desde NetBeans

Copiamos el enlace SSH de nuestro repositorio remoto



A la hora de establecer una conexión en NetBeans del repositorio local con el remoto, elegiremos la opción “Specify Git Repository Location” y en la URL del directorio pegamos la del paso anterior de GitHub. Usaremos la clave privada en “Private Key Files” y se establecerá la conexión SSH

