INSTALACIÓN UBUNTU SERVER

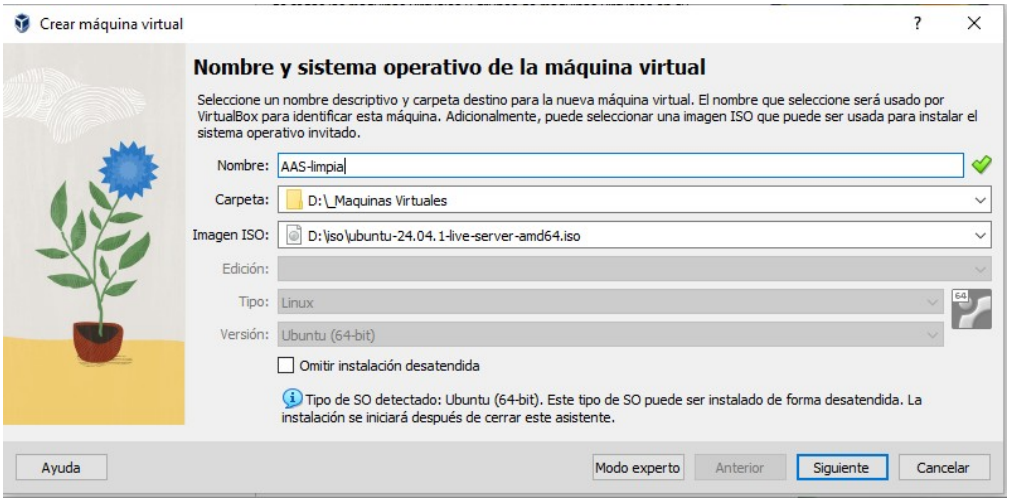
Ultima actualización 10/12/2024

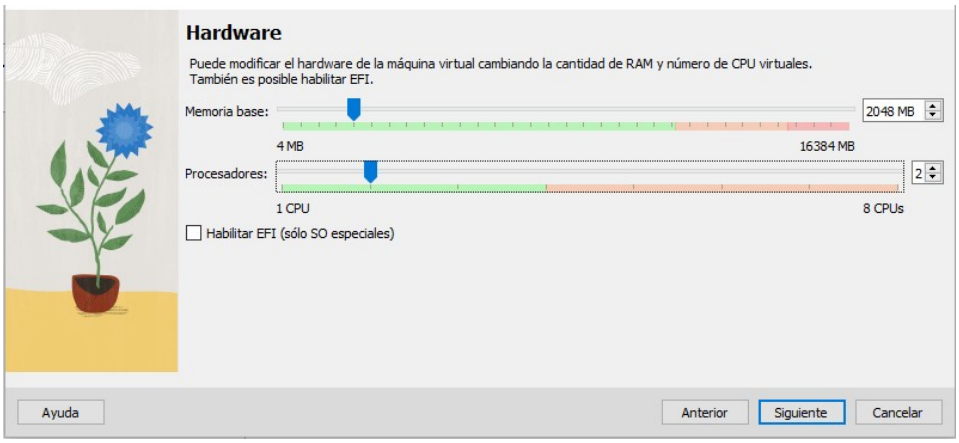
Sumario

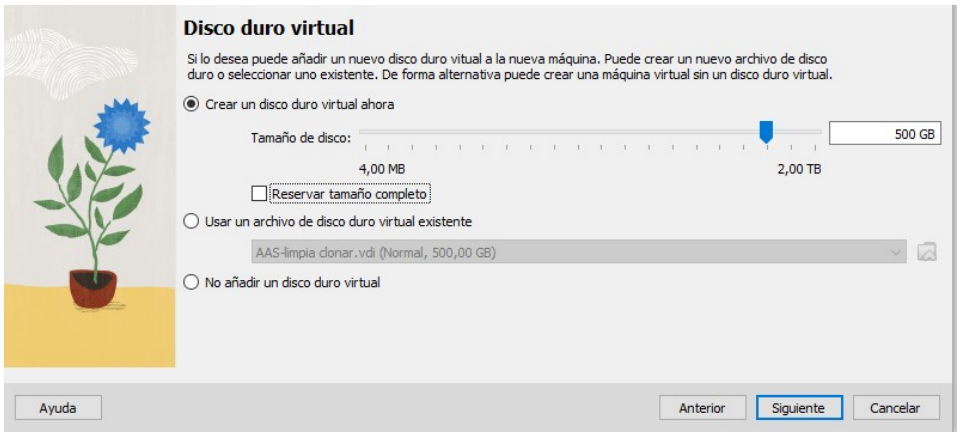
USED-Ubuntu server

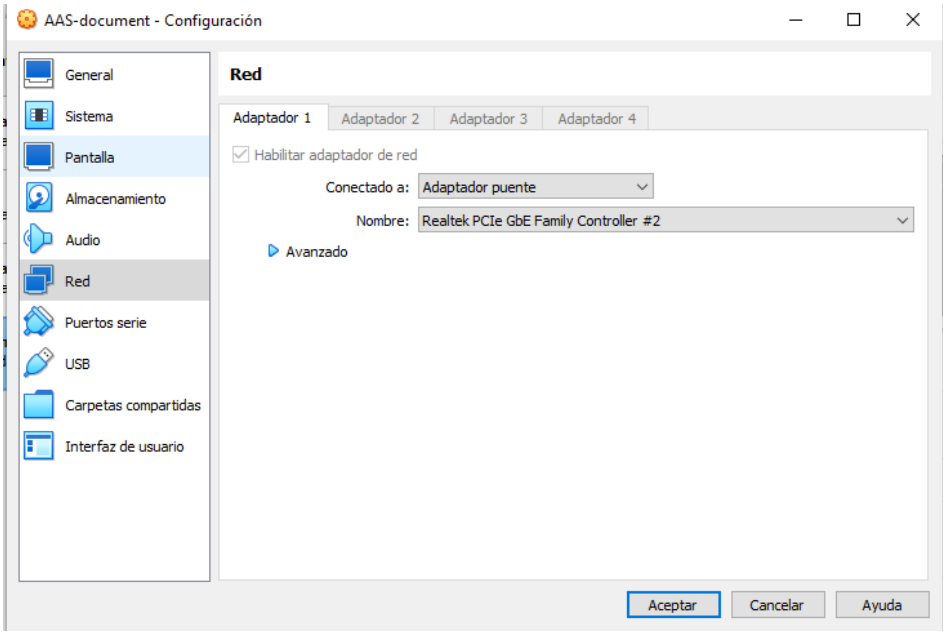
Configuración inicial

La maquina contara con 2GB de ram y 2 CPU, 500GB de almacenamiento, ademas la configuracion de red estara colocada en adaptador puente

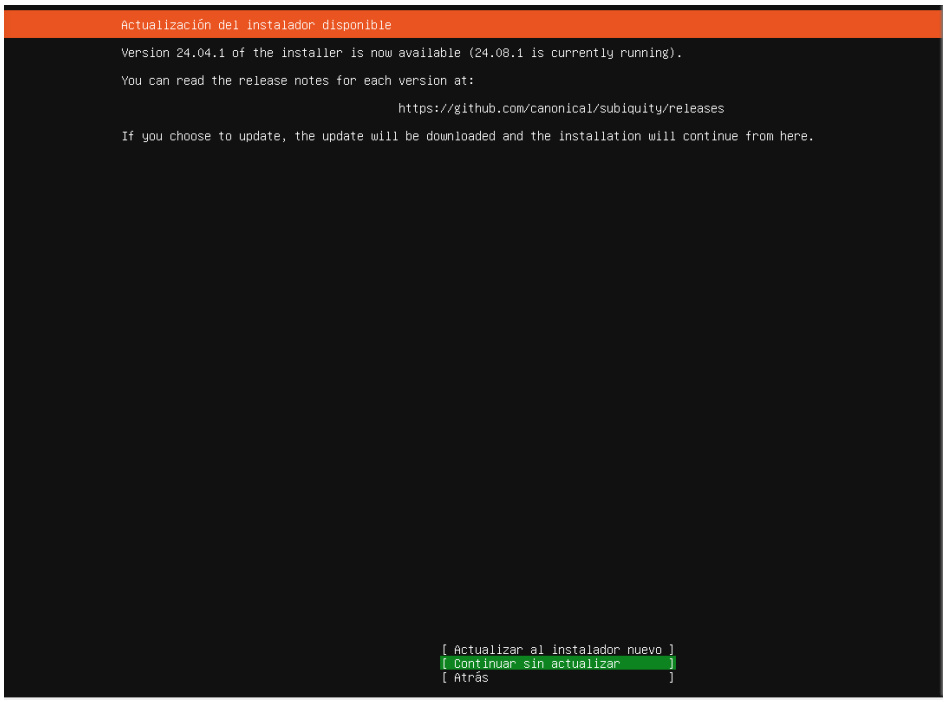





Disco duro virtual
Si lo desea puede añadir un nuevo disco duro vitual a la nueva máquina. Puede crear un nuevo archivo de disco duro o seleccionar uno existente. De forma alternativa puede crear una máquina virtual sin un disco duro virtual.
Crear un disco duro virtual ahora
Tamaño de disco:
500 GB
4,00 MB
2,00 TB
Reservar tamaño completo
Usar un archivo de disco duro virtual existente
AAS-limpia clonar.vdi (Normal, 500,00 GB)
No añadir un disco duro virtual
Ayuda
Anterior
Siguiente
Cancelar

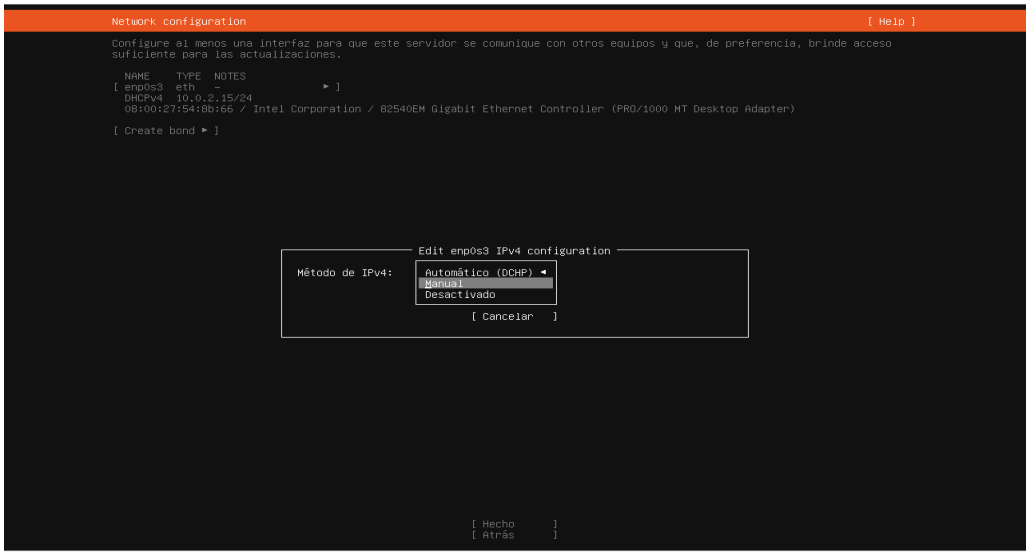

AAS-document - Configuración
General
Sistema
Pantalla
Almacenamiento
Audio
Red
Puertos serie
USB
Carpetas compartidas
Interfaz de usuario
Red
Adaptador 1
Adaptador 2
Adaptador 3
Adaptador 4
Habilitar adaptador de red
Conectado a:
Adaptador puente
Nombre:
Realtek PCIe GbE Family Controller #2
Avanzado
Aceptar
Cancelar
Ayuda

Ya podremos arrancar la maquina, seleccionamos el idioma que queramos. Continuamos sin actualizar



Elegimos instalar Ubuntu server

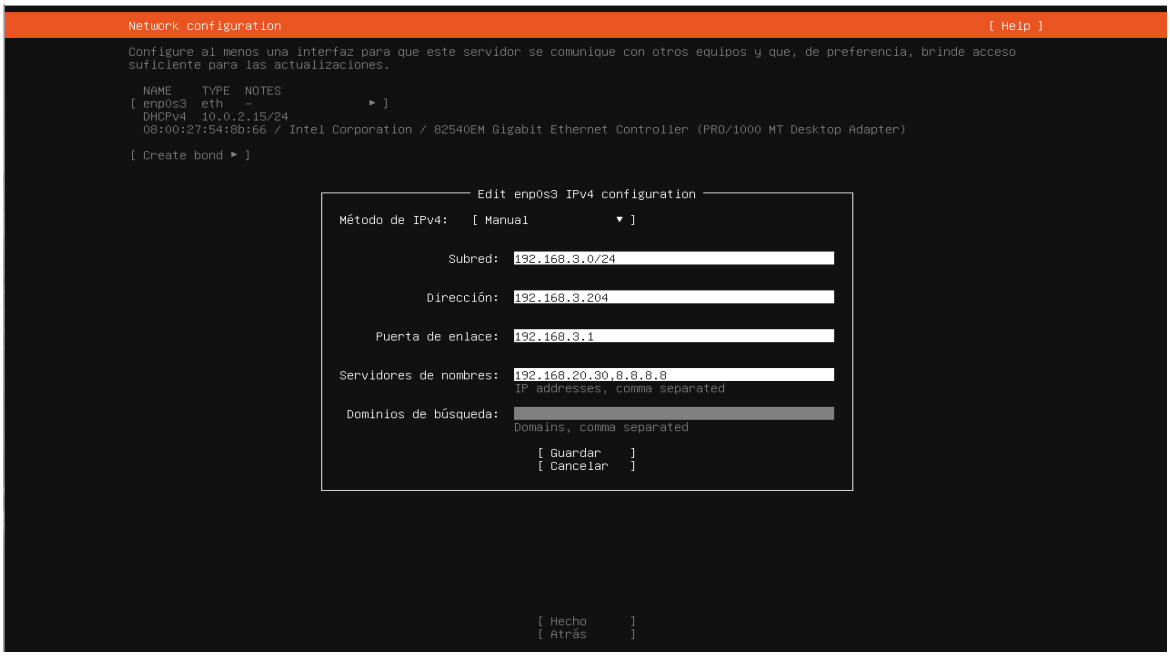
En la configuración de red seleccionaremos una red ipv4 personalizada e introducimos los siguientes valores



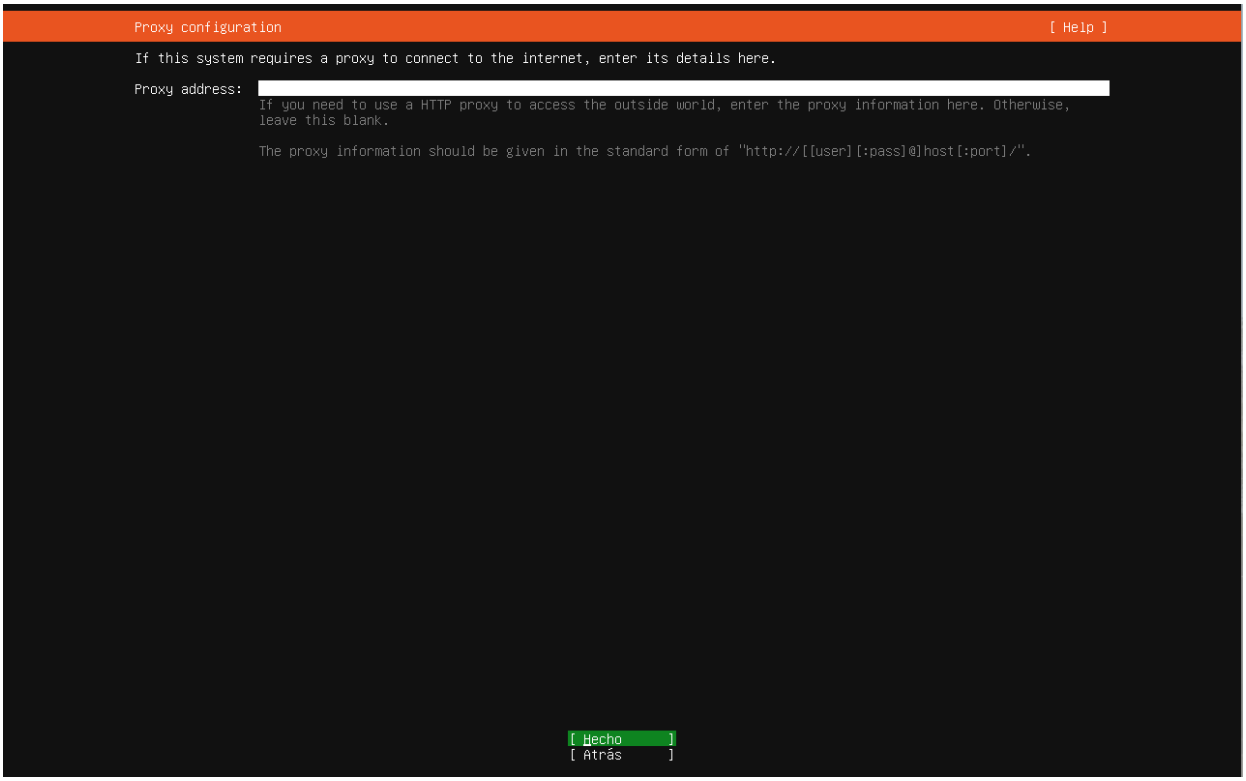
En caso de hacer esto en casa:

La IP no se pisara con la nuestra, la mascara de red sera de 24 bits usando la ip de la maquina anfitriona (xxx.xxx.xxx.0/24), y la puerta de enlace sera la misma que la maquina anfitriona

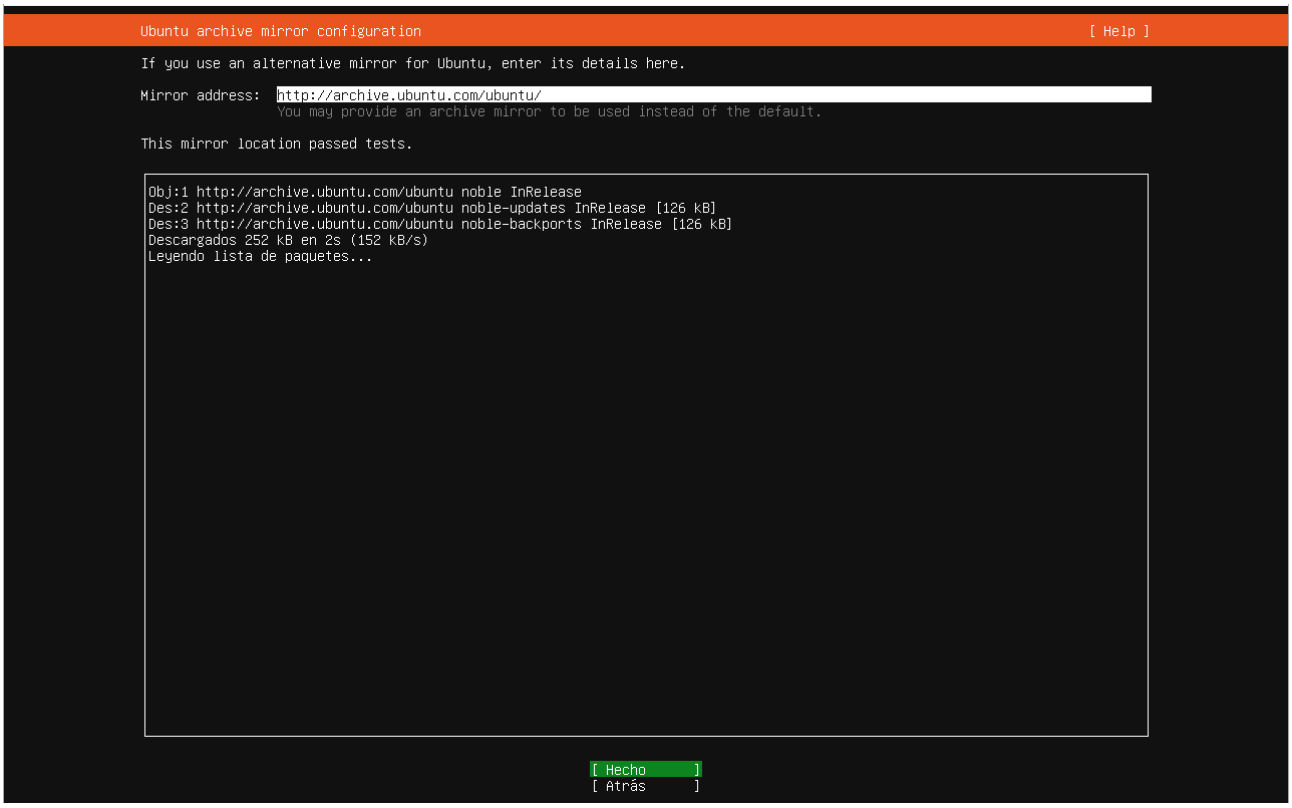
Esperaremos a que la opción “continuar sin red” cambie a “continuar” y la seleccionaremos



No haremos nada con esto



Ni con esto, simplemente continuamos

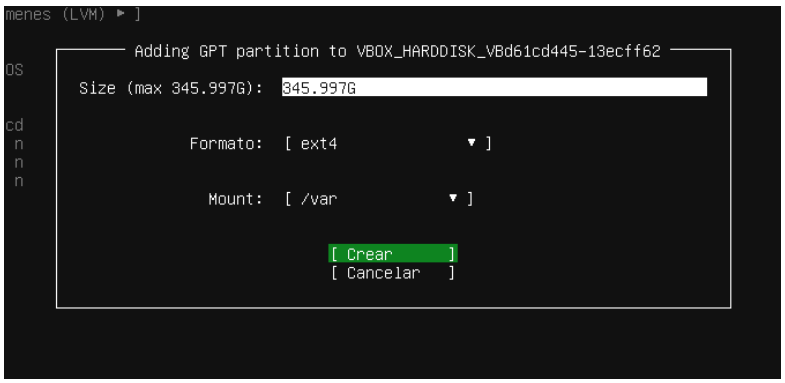


Seleccionaremos la opción “custom store layout”

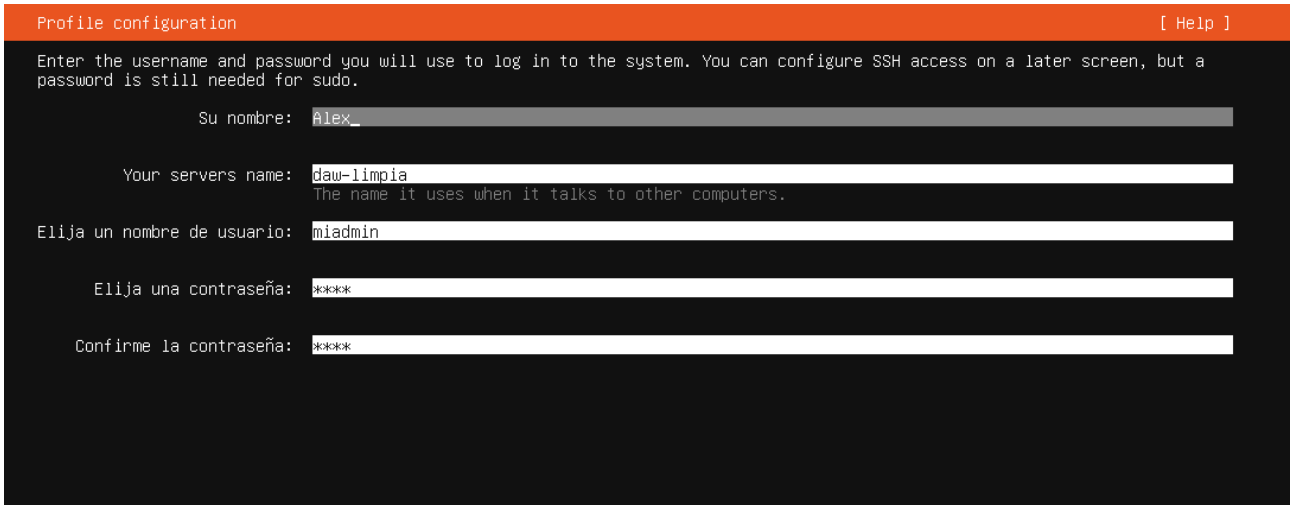
Crearemos 3 particiones, una de 150GB para el sistema

Una de 4GB de tipo swap

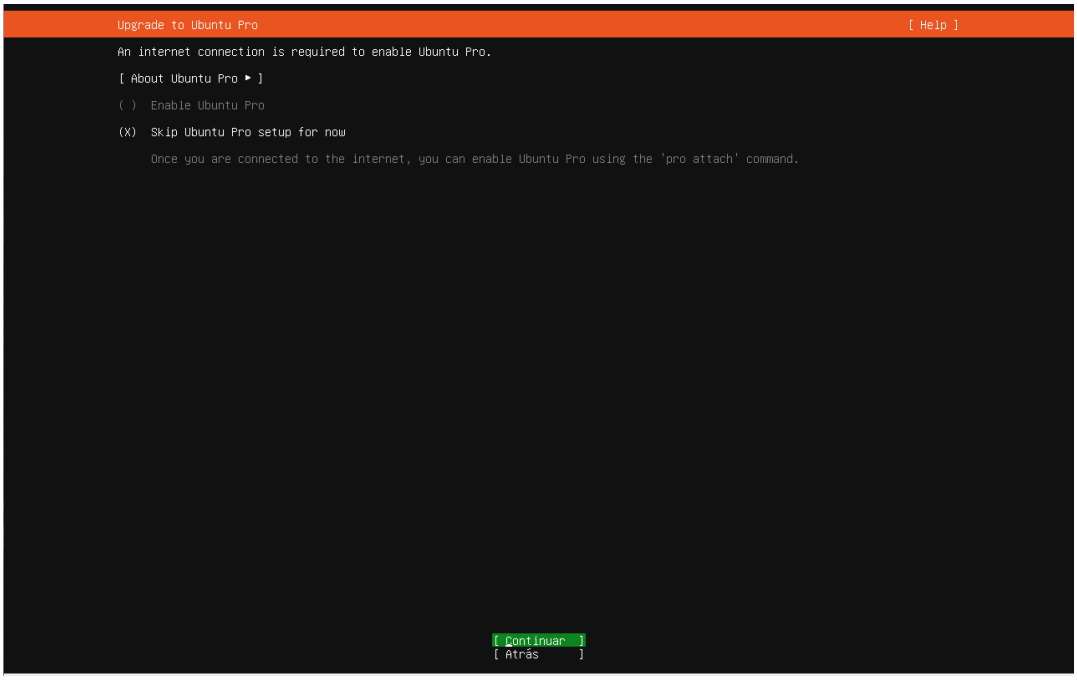
Y una con el espacio restante para /var



Introducimos nuestros credenciales

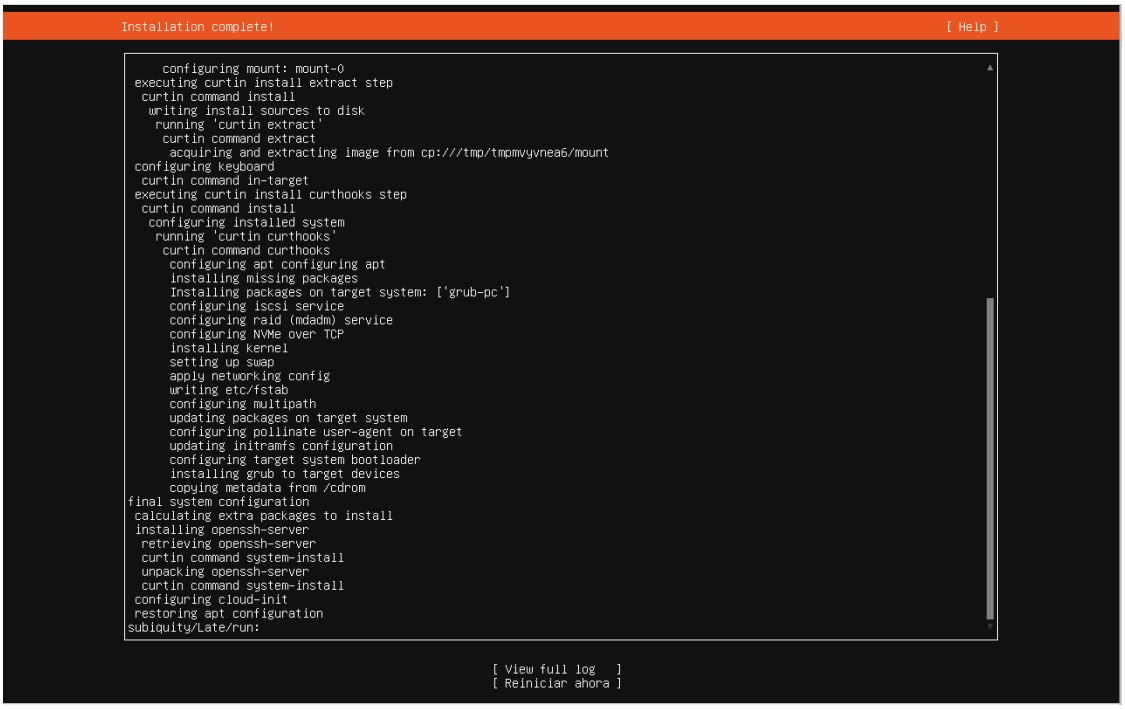


Nos saltamos la instalacion de ubuntu pro



Y, muy importante, instalaremos ssh (la casilla viene sin marcar por defecto)

Dejamos que se instale la maquina y la reiniciaremos cuando termine



Ya reiniciada, comprobaremos que tenemos acceso a internet haciendo ping

Si es así, haremos un “sudo apt update” y “sudo apt upgrade”, que descargan e instalan las ultimas actualizaciones

A continuación cambiaremos el nombre de la maquina

Cambiamos el nombre

Introducimos el siguiente comando y reiniciamos la maquina

Para comprobar la zona horaria de la maquina introduciremos el comando “timedatectl”

Si queremos una zona horaria en especifico, introduciremos el siguiente comando:
“sudo timedatectl set-timezone Continente/Ciudad”

Crearemos 2 usuarios, el primero sera uno con permisos de administrador llamado miadmin2

Le concedemos permisos

Ahora añadimos un usuario al directorio /var/www/html y al grupo www-data

```
miadmin@aas-used:~$ sudo adduser --home /var/www/html/ --no-create-home --ingroup www-data operadorweb
info: Adding user `operadorweb' ...
info: Selecting UID from range 1000 to 59999 ...

info: Adding new user `operadorweb' (1002) with group `www-data (33)' ...
info: 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] y
info: Adding new user `operadorweb' to supplemental / extra groups `users' ...
info: Adding user `operadorweb' to group `users' ...
```

Le concedemos permisos solo en el directorio html

```
miadmin@aas-used:~$ cd /var/www
miadmin@aas-used:/var/www$ sudo chown -R operadorweb:www-data /var/www/html
miadmin@aas-used:/var/www$ ls -l
total 4
drwxr-xr-x 2 operadorweb www-data 4096 oct 11 13:38 html
```

con ls -l podemos ver que el usuario aparece

```
miadmin@aas-used:/var/www$ sudo chmod -R 2775 /var/www/html
miadmin@aas-used:/var/www$
```

Cuentas de administración

Maquina anfit, administrador y usuario

Apache

Primero habilitamos el cortafuegos (comprobaremos su estatus con ufw enable y ufw status)

```
miadmin@daw-limpia:~$ sudo ufw enable
Firewall is active and enabled on system startup
miadmin@daw-limpia:~$ sudo ufw status
Status: active
```

Abrimos los puertos 80 y 22

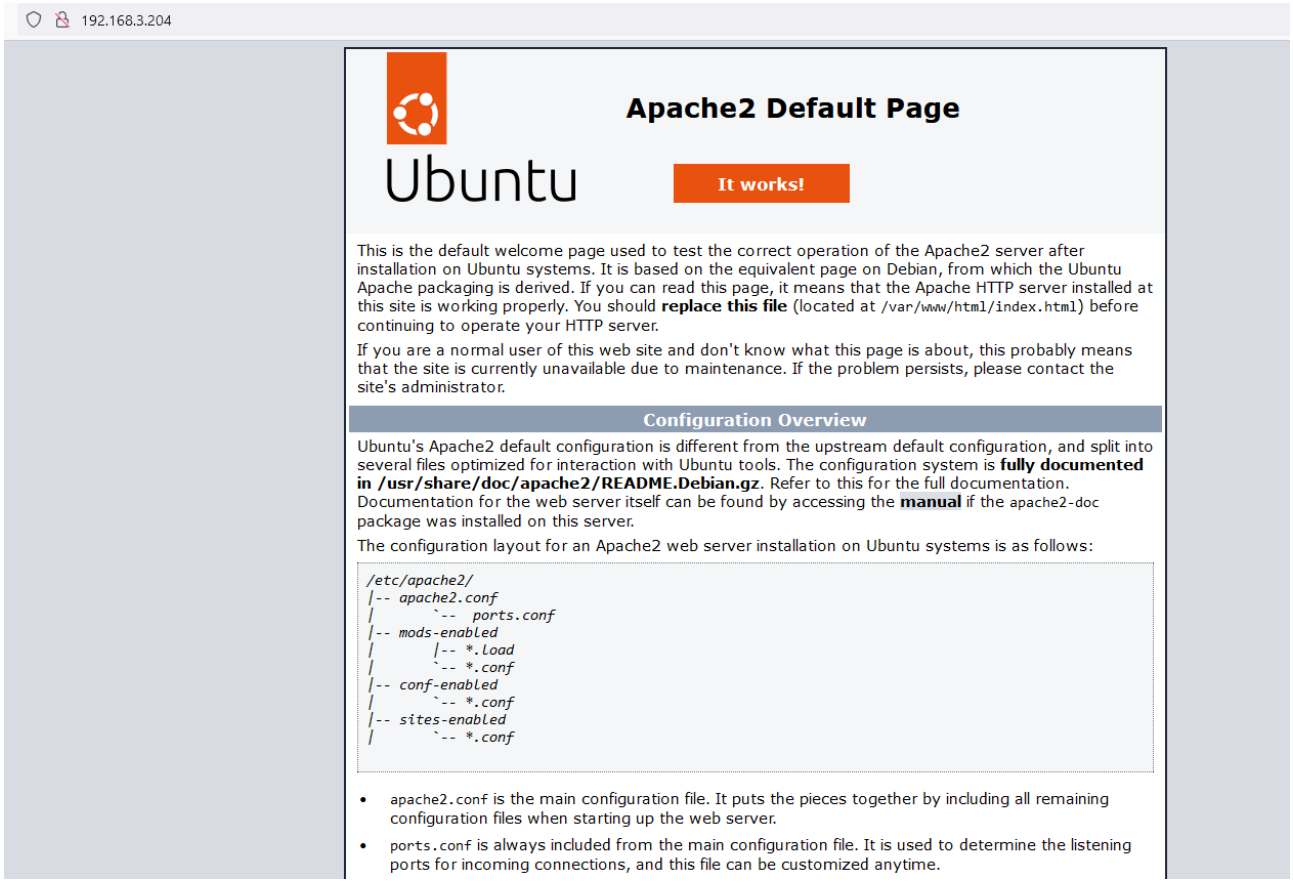
```
miadmin@daw-limpia:~$ sudo ufw allow 80
Rule added
Rule added (v6)
miadmin@daw-limpia:~$ sudo ufw allow 22
Rule added
Rule added (v6)
miadmin@daw-limpia:~$ sudo ufw status
Status: active

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

Ahora instalaremos el servidor apache, introduciremos el siguiente comando

```
miadmin@daw-limpia:~$ sudo apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
```

Para comprobar que funciona, introduciremos la ip de la maquina en cualquier buscador, deberia aparecer esta pagina:



PHP

Para instalar PHP en nuestra maquina ejecutaremos el siguiente comando

```
miadmin@aas-used:~$ sudo apt install php
```

(El comando instala la ultima versión de php, si quisiéramos una en especifico podríamos indicarselo así: “sudo apt install php8.3”)

Tras esto, comprobaremos que se ha instalado correctamente y que version se ha instalado

```
miadmin@aas-used:~$ php -v
PHP 8.3.6 (cli) (built: Jun 13 2024 15:23:20) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
miadmin@aas-used:~$
```

Ahora editaremos el archivo php.ini, que se encuentra en el siguiente directorio

```
miadmin@aas-used:/etc/php/8.3/apache2$ ls
conf.d  php.ini
miadmin@aas-used:/etc/php/8.3/apache2$
```

```
miadmin@aas-used:/etc/php/8.3/apache2$ sudo nano php.ini
[sudo] password for miadmin:
```

Cambiaremos el parametro “display_errors” de “off” a “on”

```
; https://php.net/displa
display_errors = Off
```

```
; https://php.net/display
display_errors = On
```

Haremos lo mismo con el parametro “display_startup_errors”

```
; https://php.net/display-start
display_startup_errors = On_
```

Al acabar, guardaremos los cambios con ctrl+x, reiniciaremos el servicio apache y comprobaremos que esta en marcha

```
miadmin@aas-used:/etc/php/8.3/apache2$ sudo service apache2 restart
miadmin@aas-used:/etc/php/8.3/apache2$ sudo service apache2 status
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Fri 2024-11-08 12:54:56 CET; 11s ago
       Docs: https://httpd.apache.org/docs/2.4/
```

Instalaremos una serie de modulos para php:

```
miadmin@aas-used:/etc/php/8.3/apache2/conf.d$ sudo apt install libapache2-mod-php
```

(Si

```
miadmin@aas-used:/etc/php/8.3/apache2/conf.d$ sudo apt install mbstring
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
```

este comando nos dice que no puede encontrar el paquete para instalar mbstring, introduciremos el siguiente comando)

```
sudo apt-get install php-mbstring
```

```
miadmin@aas-used:/etc/php/8.3/apache2/conf.d$ sudo apt install php8.3-xml
```

```
miadmin@aas-used:/etc/php/8.3/apache2/conf.d$ sudo apt install php8.3-zip
```

```
miadmin@aas-used:/etc/php/8.3/apache2/conf.d$ sudo apt install php8.3-gd
```

```
miadmin@aas-used:/etc/php/8.3/apache2/conf.d$ sudo apt install php8.3-curl
```

Podemos comprobar todos los modulos instalados de la siguiente manera

```
miadmin@aas-used:/etc/php/8.3/apache2/conf.d$ dpkg --get-selections | grep php8.3
libapache2-mod-php8.3                           install
php8.3                                          install
php8.3-cli                                      install
php8.3-common                                   install
php8.3-curl                                     install
php8.3-gd                                       install
php8.3-mbstring                                 install
php8.3-opcache                                  install
php8.3-readline                                 install
php8.3-xml                                      install
php8.3-zip                                      install
```

Para acabar, editaremos de nuevo el archivo php.ini que se encontraba en la ruta /etc/php/8.3/apache2/php.ini

Aqui indicaremos aumentaremos el limite de memoria de 128M a 256M

```
; Maximum amount of me
; https://php.net/memo
memory_limit = 128M
```

```
; https://php.net/memor
memory_limit = 256M
```

MySQL

Instalamos mysql-server

```
miadmin@aas-used:~$ sudo apt install mysql-server
```

Comprobamos que esta activo

```
miadmin@aas-used:~$ sudo service mysql status
● mysql.service - MySQL Community Server
     Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
     Active: active (running) since Tue 2024-10-15 11:44:03 CEST; 21s ago
    Process: 2368 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
   Main PID: 2387 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 2276)
     Memory: 365.3M (peak: 379.6M)
        CPU: 1.073s
     CGroup: /system.slice/mysql.service
             └─2387 /usr/sbin/mysqld

oct 15 11:44:02 aas-used systemd[1]: Starting mysql.service - MySQL Community Server...
oct 15 11:44:03 aas-used systemd[1]: Started mysql.service - MySQL Community Server.
```

(REVISAR A PARTIR DE AQUÍ)

A continuación ejecutaremos el siguiente comando seleccionando las opciones aquí mostradas

```
miadmin@aas-used:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary                  file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.


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) :
```

```
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!
miadmin@aas-used:~$ _
```

Posteriormente abrimos el puerto 3306 (el que utiliza MySql)

```
miadmin@aas-used:~$ sudo ufw allow 3306/tcp
Rule added
Rule added (v6)
```

Copiaremos el archivo de configuración de mysql (para tener una copia de seguridad)

```
miadmin@aas-used:~$ sudo cp /etc/mysql/mysql.conf.d/mysqld.cnf /etc/mysql/mysql.conf.d.mysqld.cnf.backup
miadmin@aas-used:~$
```

Editaremos este mismo archivo y comentaremos estas dos líneas

```
# Instead of skip-networking the default is now to listen only 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
```

```
# localhost which is more compatible and
#bind-address            = 127.0.0.1
#mysqlx-bind-address     = 127.0.0.1
#
```

Reiniciamos el servicio de MySQL

```
miadmin@aas-used:~$ sudo service mysql restart
miadmin@aas-used:~$
```

A continuación entramos en Mysql con el comando “sudo mysql”

Ejecutamos los siguiente comando

```
mysql> set global validate_password.length=4;
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                         | MEDIUM |
| validate_password.special_char_count             | 1      |
+--------------------------------------------------+--------+
8 rows in set (0,01 sec)
```

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

Creamos un usuario administrador y le concedemos todos los permisos en todas las bases de datos

```
mysql> create user 'adminsql'@'%' identified by 'paso';
Query OK, 0 rows affected (0,06 sec)

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

Salimos de mysql

```
mysql> exit
Bye
```

Intentamos entrar a mysql usando ese usuario para comprobar si se ha creado correctamente

```
miadmin@aas-used:~$ sudo mysql -u adminsql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

Ya habriamos instalado MySQL

PHP Myadmin

XDebug

Instalamos xdebug

```
miadmin@aas-used:/etc/php/8.3/apache2$ sudo apt install php8.3-xdebug
```

Editamos el archivo de configuración de xdebug

```
miadmin@aas-used:/etc/php/8.3/apache2/conf.d$ sudo nano /etc/php/8.3/mods-available/xdebug.ini
```

En este archivo escribiremos lo siguientes

```
zend_extension=xdebug.so
xdebug.discover_client_host=1
xdebug.mode=debug
xdebug.client_host=localhost
xdebug.client_port=9003
xdebug.idekey="netbeans-xdebug"
xdebug.show_error_trace=1
xdebug.remote_autostart=on
xdebug.start_with_request=yes
```

Una vez hecho esto editaremos el archivo php.ini y en el apartado “output_buffering” cambiaremos el valor de “Default value” a On

```
miadmin@aas-used:/etc/php/8.3/apache2$ sudo nano php.ini
```

```
; output_buffering
;   Default Value: Off
;   Development Value: 4096
;   Production Value: 4096
```

```
; output_buffering
;   Default Value: On
;   Development Value: 40
```

Reiniciamos Apache y ya tendríamos instalado xdebug

Cuentas de desarrollo y hosting virtual

Configurar DNS

Zona directa

Primero instalaremos el modulo bind con el comando “sudo apt install bind9”

A continuación habilitaremos el puerto 53

```
miadmin@aas-used:~$ sudo ufw allow 53
Rule added
Rule added (v6)
miadmin@aas-used:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
80                         ALLOW       Anywhere
22                         ALLOW       Anywhere
53                         ALLOW       Anywhere
80 (v6)                    ALLOW       Anywhere (v6)
22 (v6)                    ALLOW       Anywhere (v6)
53 (v6)                    ALLOW       Anywhere (v6)
```

Editaremos el archivo de configuración de sitios y añadiremos una zona con nuestro nombre

```
miadmin@aas-used:~$ sudo nano /etc/bind/named.conf.local
```

```
miadmin@aas-used: ~
 GNU nano 7.2                                /etc/bind/named.conf.local *
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "alex.local" {
        type master;
        file "/etc/bind/db.alex.local";
};
```

Comprobaremos que no tenemos ningún fallo de sintaxis ni de otro tipo

```
miadmin@aas-used:~$ sudo named-checkconf /etc/bind/named.conf.local
miadmin@aas-used:~$
```

Copiamos el archivo db.local a uno con nuestro nombre y lo editaremos siguiendo estos parametros

```
miadmin@aas-used:~$ sudo cp /etc/bind/db.local /etc/bind/db.alex.local
miadmin@aas-used:~$
```

```
;
; BIND data file for alex.local
;
$TTL    604800
@       IN      SOA     aas-used.alex.local. alexasesan.educa.jcyl.es. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         3600 ) ; Negative Cache TTL
;
@       IN      NS      aas-used.alex.local.
@       IN      A       192.168.3.204
aas-used        IN       A           192.168.3.204

daw201  IN CNAME        aas-used.alex.local.
daw204  IN CNAME        aas-used.alex.local.
```

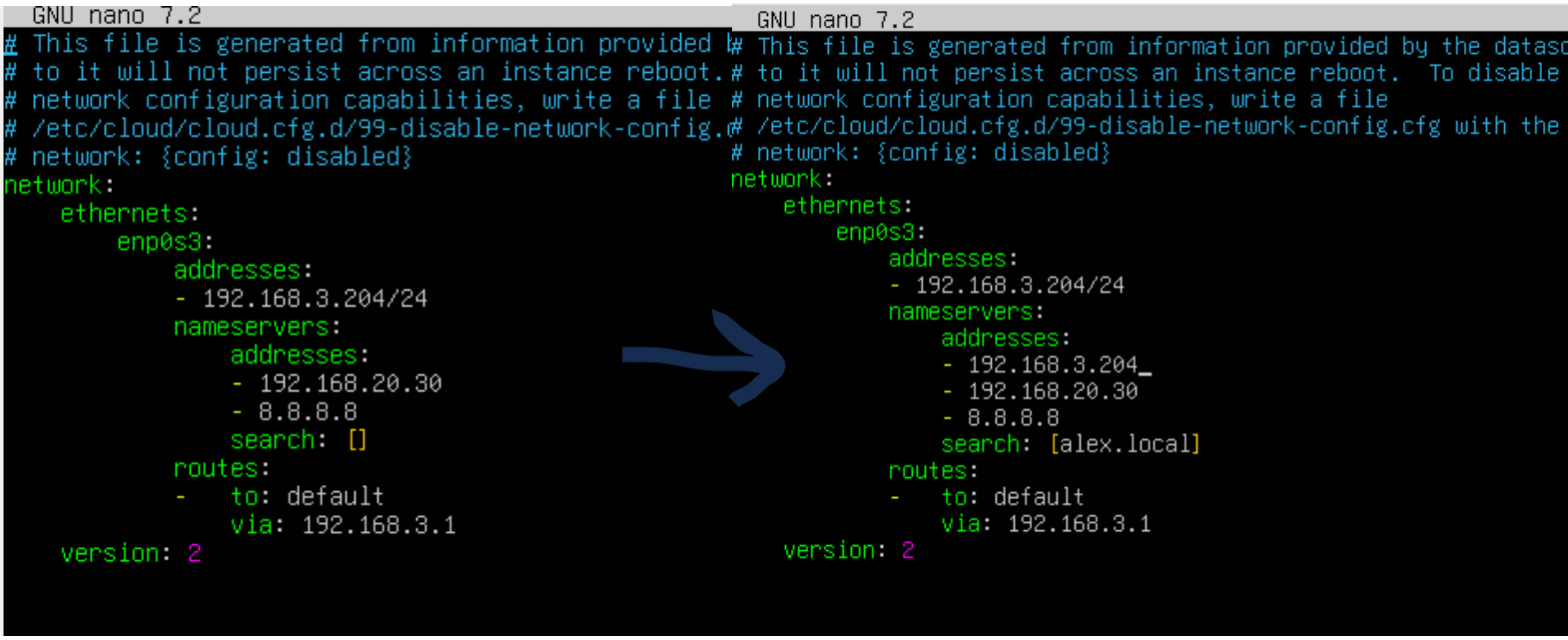
Comprobaremos que no hay ningún error

```
miadmin@aas-used:~$ sudo named-checkzone alex.local /etc/bind/db.alex.local
zone alex.local/IN: loaded serial 2
OK
```

Ahora editaremos el fichero de configuración de red

```
miadmin@aas-used:/etc/netplan$ sudo nano 50-cloud-init.yaml
```

Introduciremos la ip de la maquina como servidor dns preferido



Aplicaremos los cambios

```
miadmin@aas-used:/etc/netplan$ sudo netplan apply
miadmin@aas-used:/etc/netplan$
```

Zona inversa

Añadimos una nueva zona con estos parámetros

```
miadmin@aas-used:~$ sudo nano /etc/bind/named.conf.local
```

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

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "alex.local" {
        type master;
        file "/etc/bind/db.alex.local";
};

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

Comprobaremos que no hay errores

```
miadmin@aas-used:~$ sudo named-checkconf /etc/bind/named.conf.local
miadmin@aas-used:~$
```

Copiaremos el archivo que creamos antes

```
miadmin@aas-used:~$ sudo cp /etc/bind/db.alex.local /etc/bind/db.3.168.192.in-addr.arpa
miadmin@aas-used:~$
```

Introducimos estos parámetros

```
;
; BIND data file for 3.168.192.in-addr.arpa
;
$TTL    604800
@       IN      SOA     aas-used.alex.local. alexasesan.educa.jcyl.es. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         3600 ) ; Negative Cache TTL
;
@       IN      NS      aas-used.alex.local.

204  IN PTR             aas-used.alex.local.
204     IN PTR          alex.local.
```

Comprobaremos que no hay errores y reiniciamos el servicio bind

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

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

Configurar HTTPS

Habilitamos el modulo ssl y reiniciamos apache

```
miadmin@aas-used:~$ sudo a2enmod ssl
Considering dependency mime for ssl:
Module mime already enabled
```

```
miadmin@aas-used:~$ sudo service apache2 restart
miadmin@aas-used:~$
```

Habilitamos el puerto 443

```
miadmin@aas-used:~$ sudo ufw allow 443
Rule added
Rule added (v6)
```

Generaremos la clave y el certificado necesarios para habilitar http

```
miadmin@aas-used:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

Le indicamos los datos que nos pidan

```
-----
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) []:www.alex.local
Email Address []:alexasensio04clase@gmail.com
```

Copiamos el archivo default-ssl.conf y lo editaremos

```
miadmin@aas-used:/etc/apache2/sites-available$ sudo cp default-ssl.conf alex-local.conf
miadmin@aas-used:/etc/apache2/sites-available$ ls
000-default.conf  000-default.conf.backup  alex-local.conf  default-ssl.conf  p8080.conf
```

Añadiremos estas 3 lineas, habilitando SSLEngine, e indicandole la ruta de la clave y el certificado generados

Habilitaremos el sitio que acabamos de configurar

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

Reiniciamos el servicio apache

```
miadmin@aas-used:/etc/apache2/sites-available$ sudo service apache2 restart
miadmin@aas-used:/etc/apache2/sites-available$
```

Enjaular usuarios

Añadiremos un grupo

```
miadmin@aas-used:~$ sudo addgroup ftpusers
[sudo] password for miadmin:
info: Selecting GID from range 1000 to 59999 ...
info: Adding group `ftpusers' (GID 1002) ...
miadmin@aas-used:~$
```

Hacemos que root sea propietario de la carpeta en la que queremos enjaular al usuario

```
miadmin@aas-used:~$ sudo chown root:root /var/www/prestashop
miadmin@aas-used:~$
```

Añadimos un usuario indicándole el grupo y la carpeta a la que pertenece

```
miadmin@aas-used:~$ sudo useradd -g www-data -G ftpusers -m -d /var/www/prestashop userprestashop
miadmin@aas-used:~$
```

Le ponemos una contraseña

```
miadmin@aas-used:/etc/apache2/sites-available$ sudo passwd userprestashop
New password:
Retype new password:
passwd: password updated successfully
miadmin@aas-used:/etc/apache2/sites-available$
```

Indicamos que los cambios en la carpeta solo pueden ser hecho por el usuario root

```
miadmin@aas-used:~$ sudo chmod 555 /var/www/prestashop
miadmin@aas-used:~$
```

Hacemos que el usuario root sea dueño de la carpeta, creamus la carpeta en la que queremos enjaular al usuario, y luego hacemos que dicho usuario sea el dueño y tenga permisos en la carpeta en la que lo queremos enjaular

```
miadmin@aas-used:~$ sudo chown root:root /var/www/prestashop
miadmin@aas-used:~$ sudo chmod 555 /var/www/prestashop
miadmin@aas-used:~$ sudo mkdir /var/www/prestashop/public_html
miadmin@aas-used:~$ sudo chown userprestashop:www-data -R /var/www/prestashop/public_html
miadmin@aas-used:~$ sudo chmod 2775 -R /var/www/prestashop/public_html
miadmin@aas-used:~$
```

Creamos una carpeta para los logs y hacemos al usuario dueño de esa carpeta

```
miadmin@aas-used:/etc/apache2/sites-available$ sudo mkdir /var/www/prestashop/logs
miadmin@aas-used:/etc/apache2/sites-available$ sudo chown userprestashop:root /var/www/prestashop/logs
miadmin@aas-used:/etc/apache2/sites-available$ sudo chmod 700 /var/www/prestashop/logs
miadmin@aas-used:/etc/apache2/sites-available$
```

Copiamos el archivo 000-default.conf a uno con el nombre de la carpeta y lo editamos

```
miadmin@aas-used:/etc/apache2/sites-available$ sudo cp 000-default.conf prestashop.conf
miadmin@aas-used:/etc/apache2/sites-available$ sudo nano prestashop.conf
```

Introducimos los siguientes parámetros

```
miadmin@aas-used: /etc/apache2/sites-available
  GNU nano 7.2                                prestashop.conf *
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        ServerName prestashop.alex.local

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/prestashop/public_html

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

        ErrorLog ${APACHE_LOG_DIR}/error.log
        ErrorLog /var/www/prestashop/logs/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        CustomLog /var/www/prestashop/logs/access.log combined
```

Habilitamos el sitio

```
miadmin@aas-used:/etc/apache2/sites-available$ sudo a2ensite prestashop.conf
Site prestashop already enabled
miadmin@aas-used:/etc/apache2/sites-available$
```

Reiniciamos apache

```
miadmin@aas-used:/etc/apache2/sites-available$ sudo service apache2 restart
miadmin@aas-used:/etc/apache2/sites-available$
```

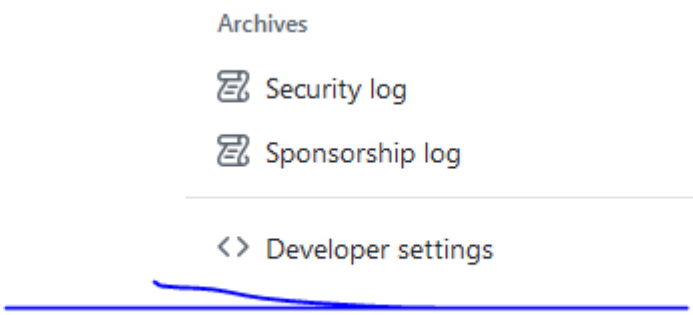
GITHub-Internet

Cuentas de desarrollador

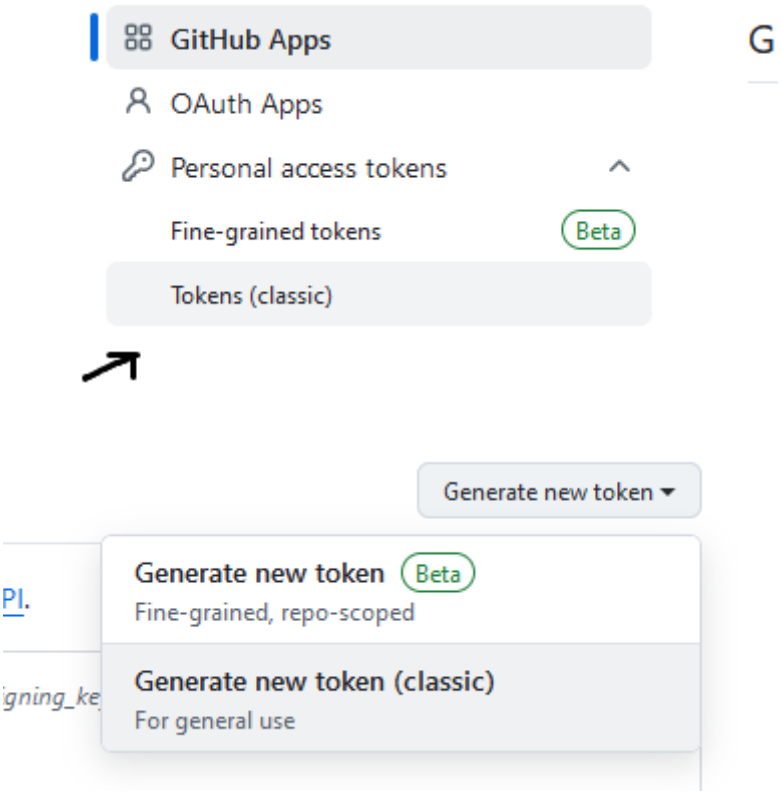
Necesitaremos una cuenta de github, nos registraremos (no tiene perdida) y obtendremos un token personal

Dicho token nos permitirá conectarnos a un repositorio de GitHub a través de NetBeans

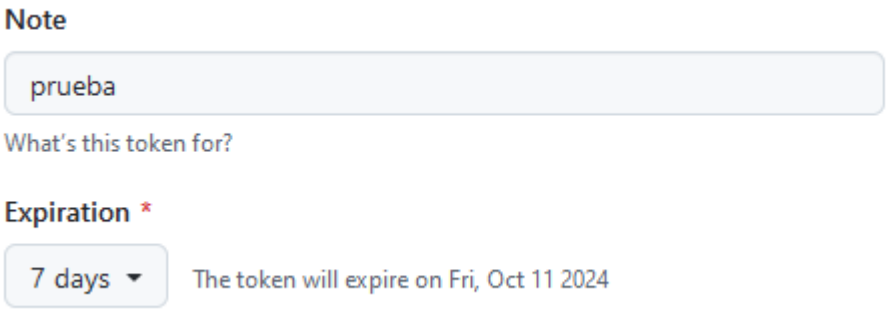
Esta opcion se encuentra abajo del todo en la pestaña de configuracion de nuestro perfil, en developer settings



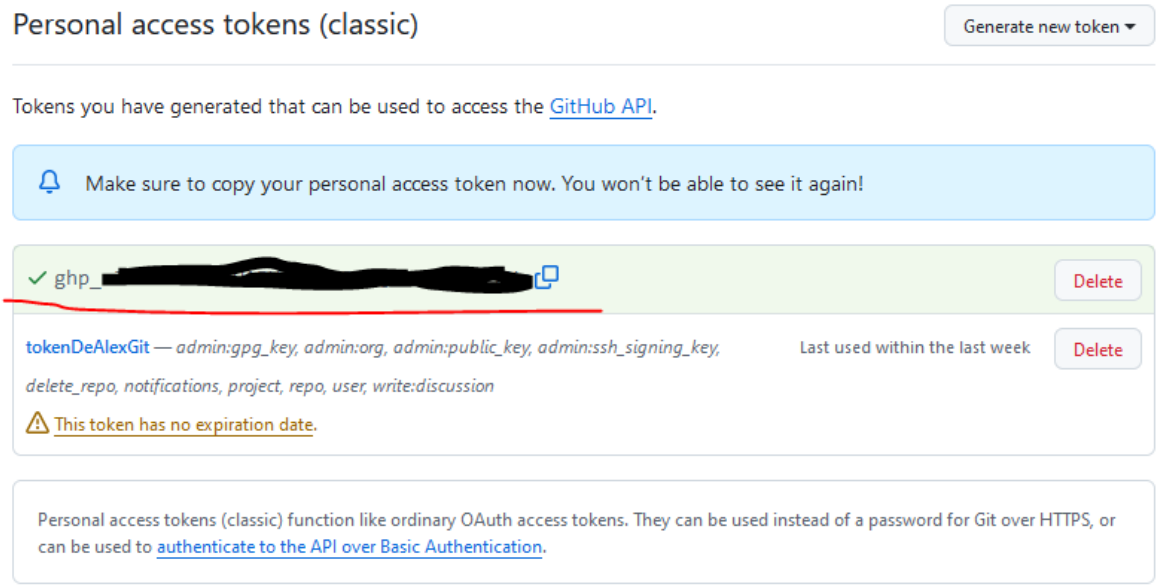
Despues iremos a la pestaña de claves de acceso, a la opcion clasic



Indicaremos para que usaremos el token (puede ser cualquier cosa, con no dejar el cuadro en blanco sirve) y cuando expirara el token, el minimo son 7 dias y el maximo es que no expire



Y listo, ya tendremos nuestro token, debemos guardarlo bien pues esta sera la unica vez que github nos lo muestre


Personal access tokens (classic)
Generate new token
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_
Delete
tokenDeAlexGit — admin:gpg_key, admin:org, admin:public_key, admin:ssh_signing_key, delete_repo, notifications, project, repo, user, write:discussion
Last used within the last week
Delete
This token has no expiration date.
Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

WXED-Windows X

Instalación y configuración inicial de la máquina

Cuentas administradoras y cuenta de desarrollador

Filezilla

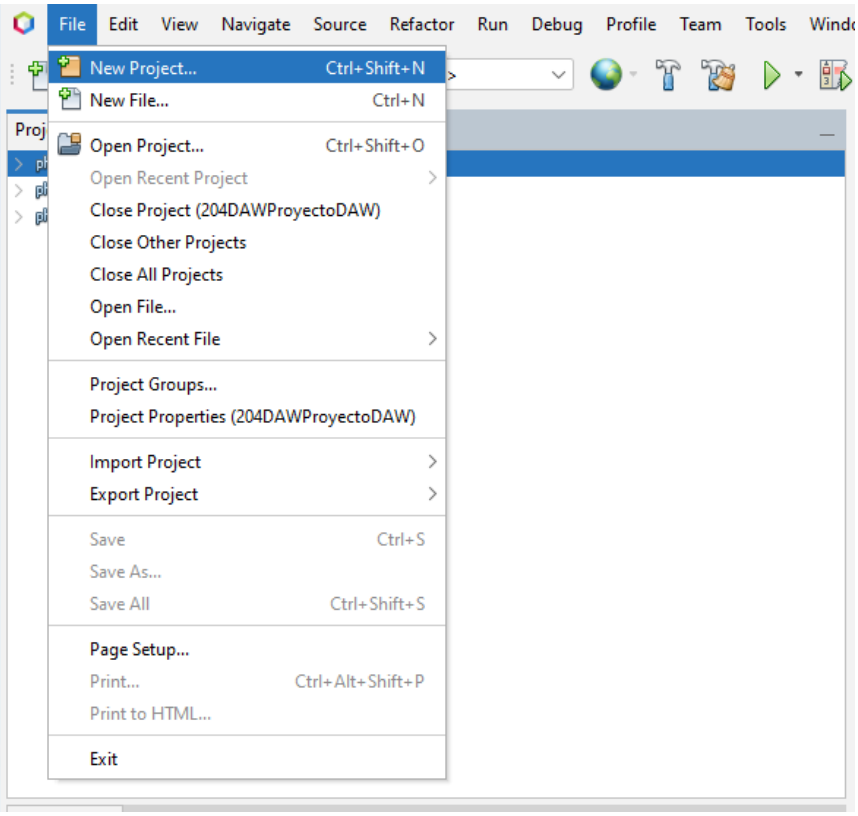
NetBeans

Instalación y configuración inicial (plugins)

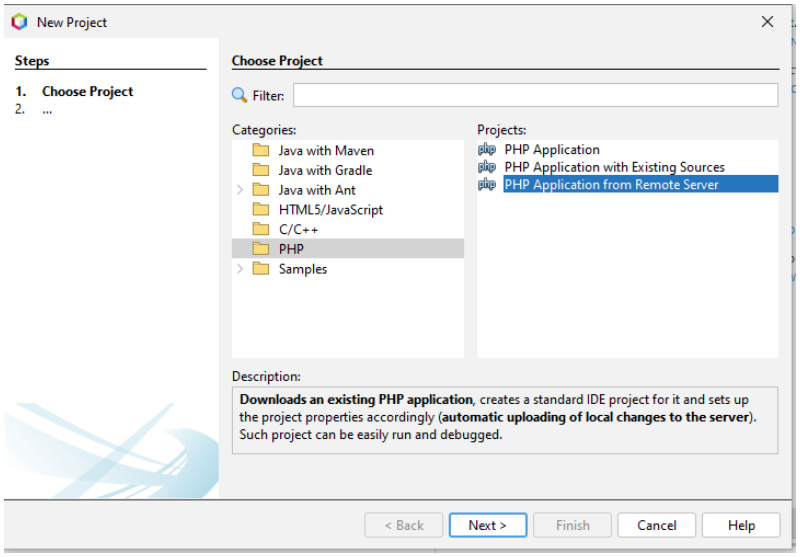
Creación de proyectos, modificación, borrado, prueba.

Desde cero

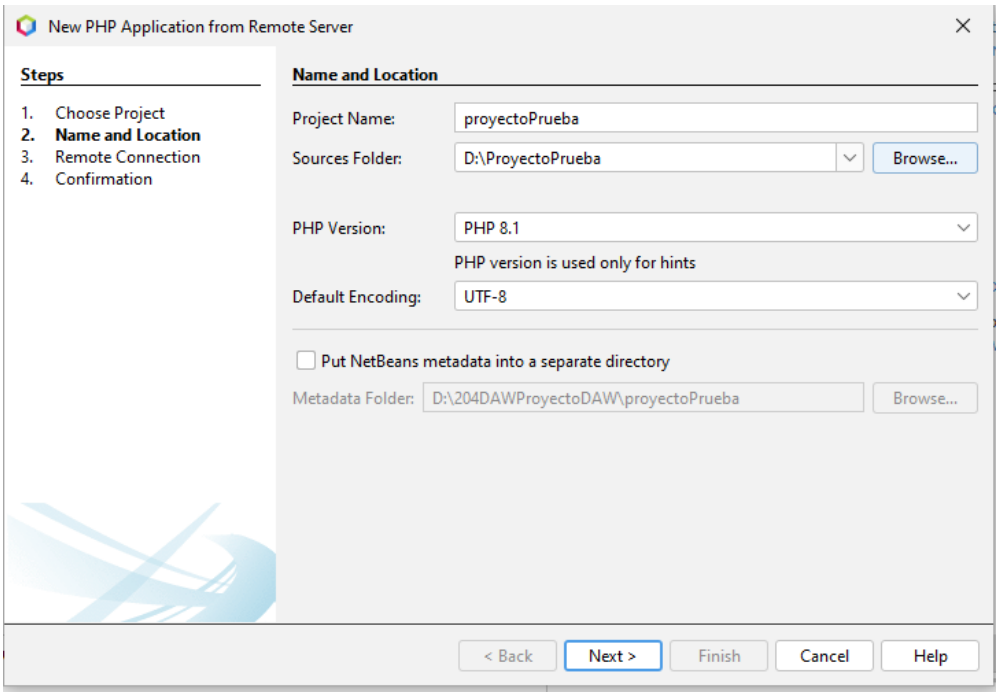
Lo primero sera crear un proyecto de php



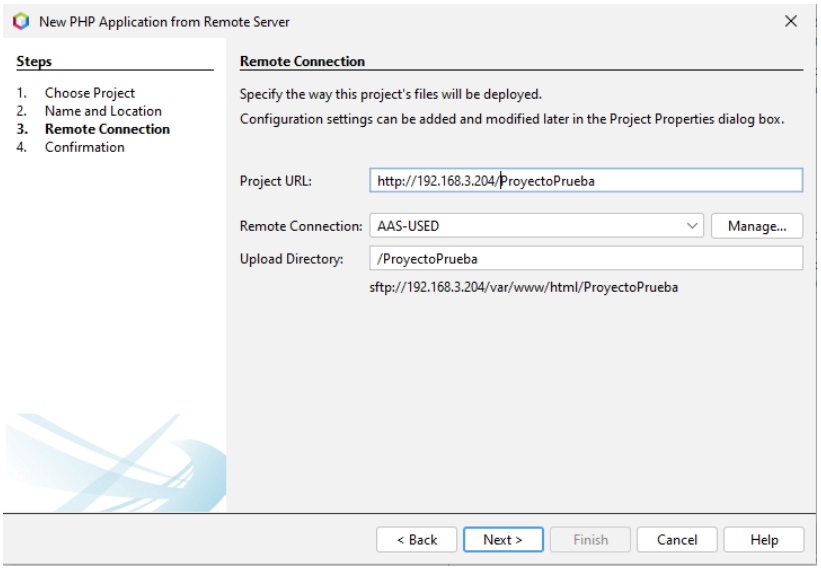
Para crear un proyecto desde 0 seleccionaremos la opción “aplicación desde servidor remoto”

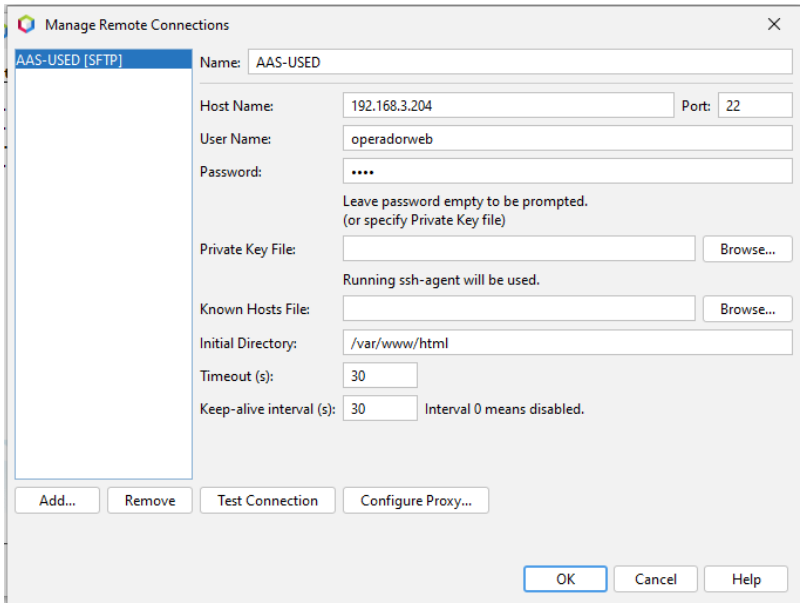


Elegimos el nombre y la ruta del proyecto

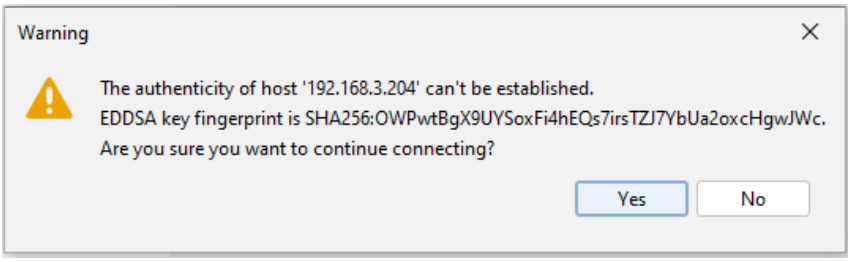


Indicamos la url del directorio del servidor en el que deseamos subir los archivos, también añadimos una conexión remota al servidor por sftp

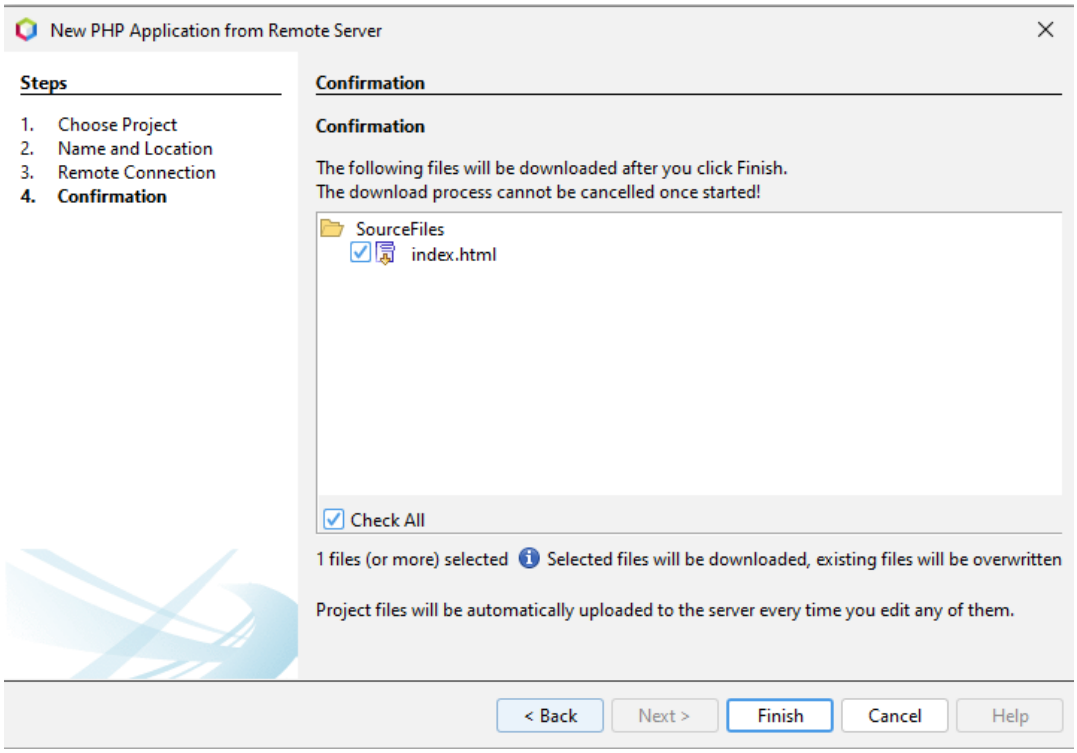




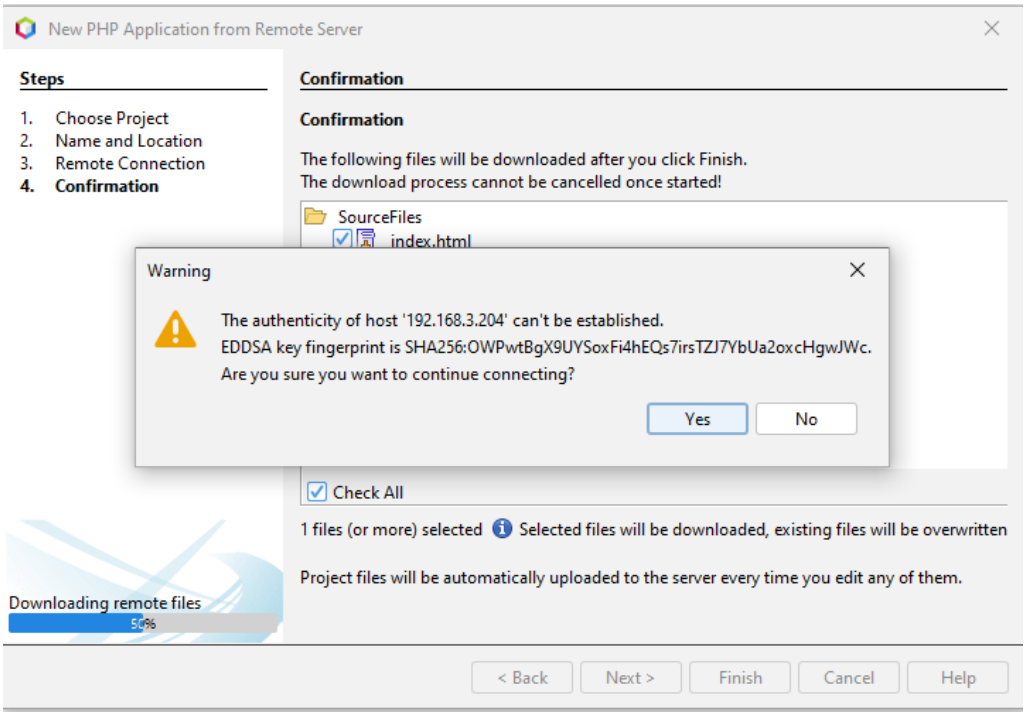
Tras esto seleccionamos que “si”



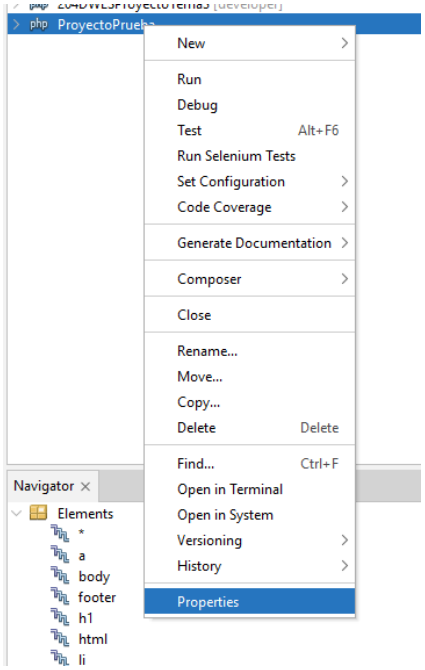
Seleccionamos los archivos que queremos descargar del servidor



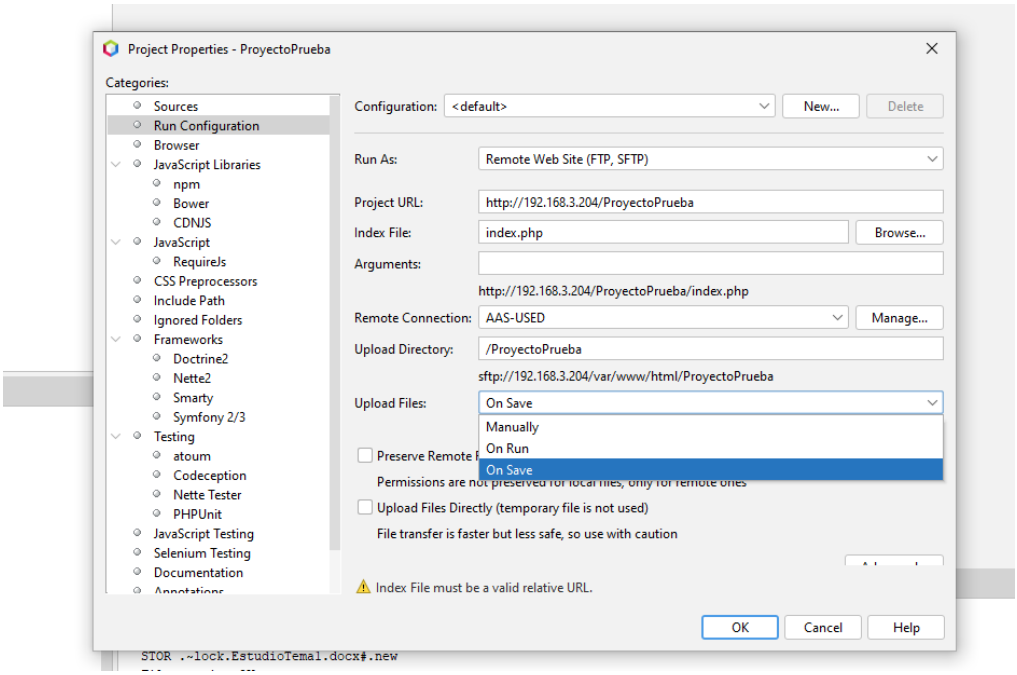
Seleccionamos que si de nuevo



Para configurar el que los archivos se suban al servidor cada vez que guardemos, iremos a la configuración del proyecto



En la pestaña “run configuration” seleccionaremos “upload on save”

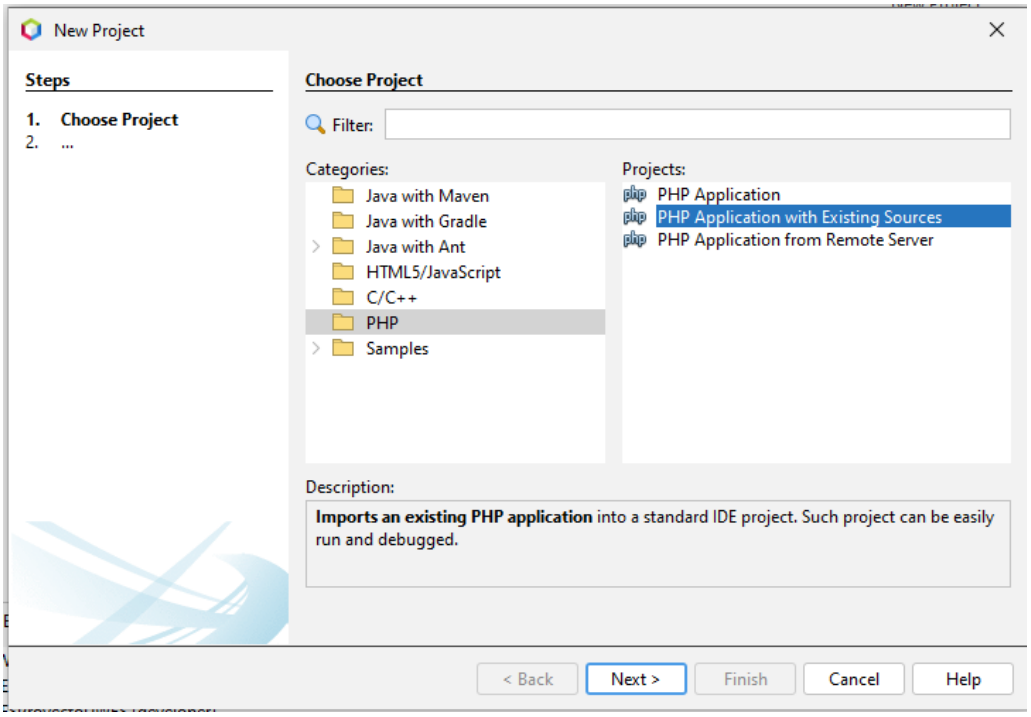


Aceptamos y ya habriamos creado un proyecto desde cero

Proyecto con archivos ya existentes en local

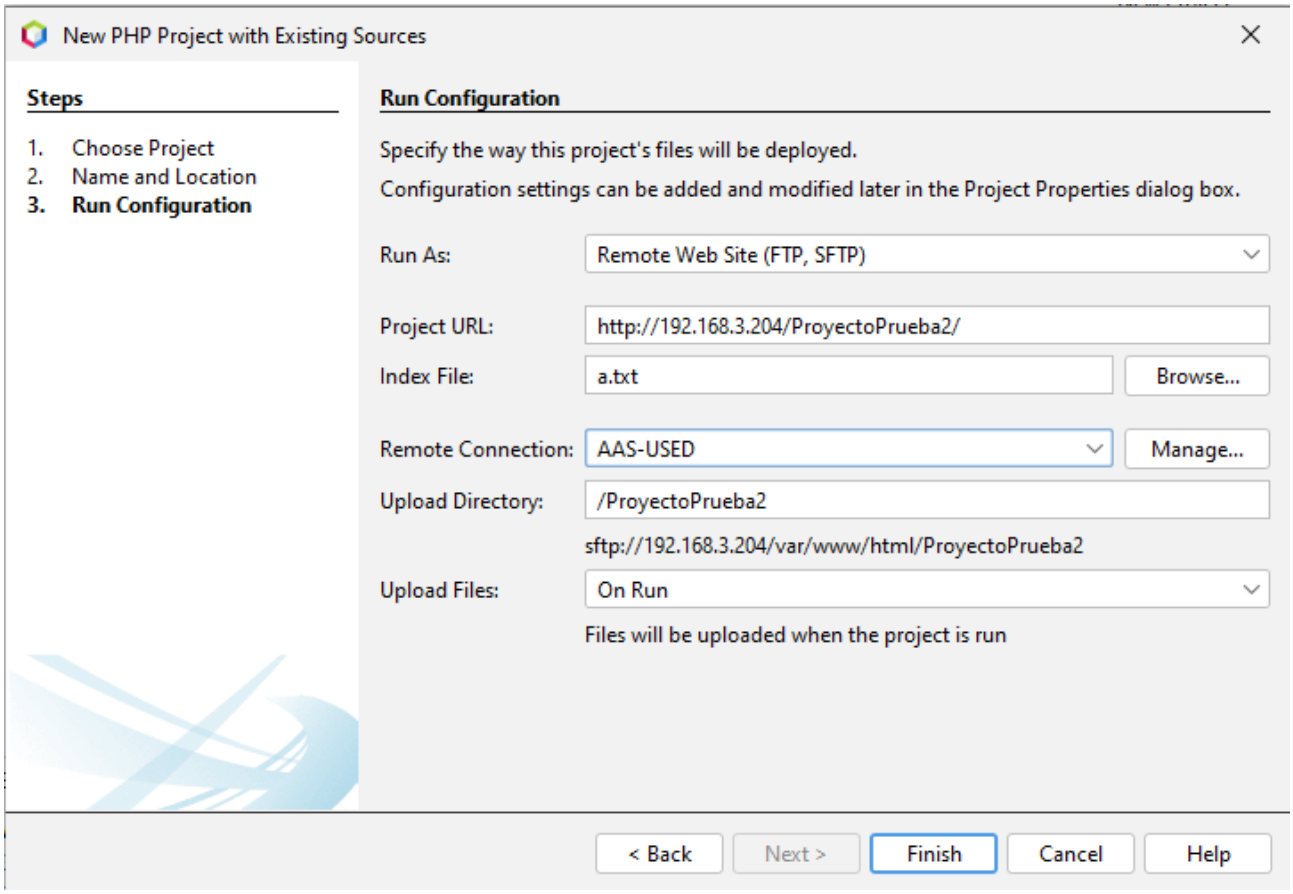
Si por el contrario, queremos usar archivos ya existentes, los pasos seran los siguientes:

Elegimos php with existing sources



Indicamos la ruta:

Indicamos la IP y conexión de la maquina (igual que en el apartado anterior)



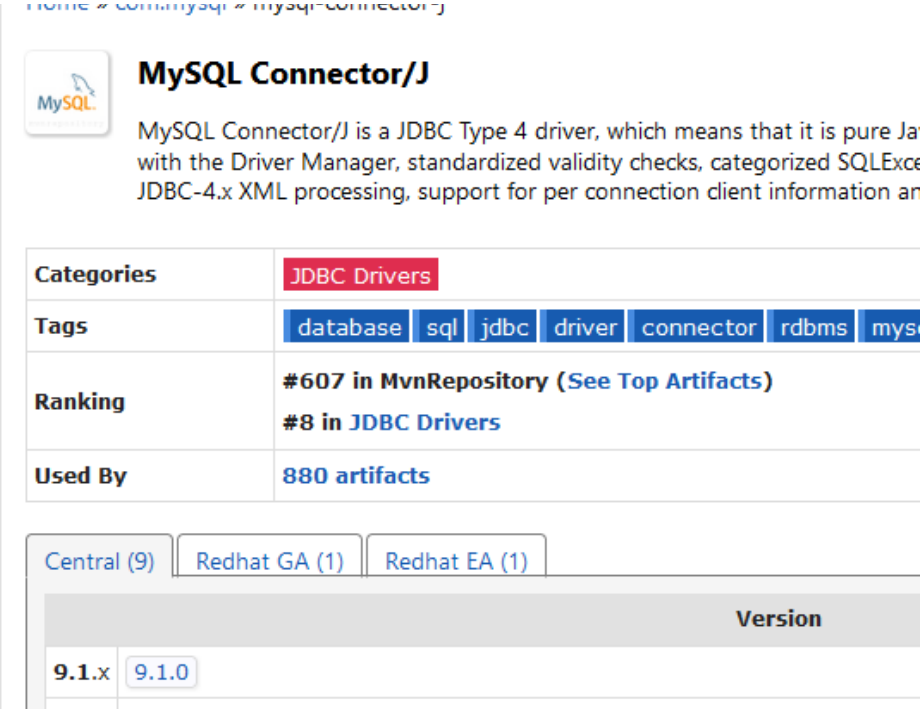
Y con eso ya tendriamos nuestro proyecto

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

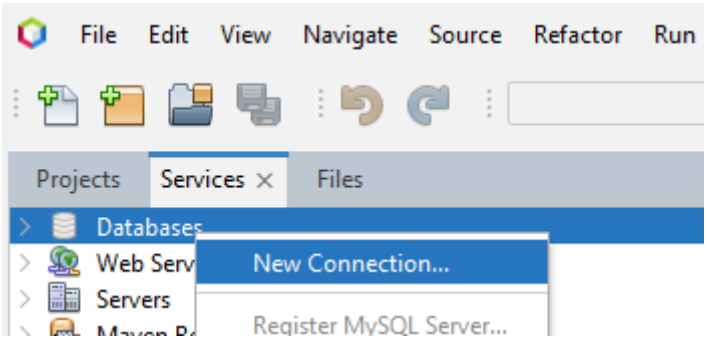
Administración de la base de datos

Lo primero que haremos sera descargar el driver conector de mysql desde la pagina de maven: https://mvnrepository.com/artifact/com.mysql/mysql-connector-j

en esta ocasión, descargaremos la versión 9.1



Ya descargado, iremos a NetBeans, en el apartado de la izquierda (en el que aparecen los proyectos) iremos a el apartado “services”, haremos click derecho en “databases” y en “new connection”



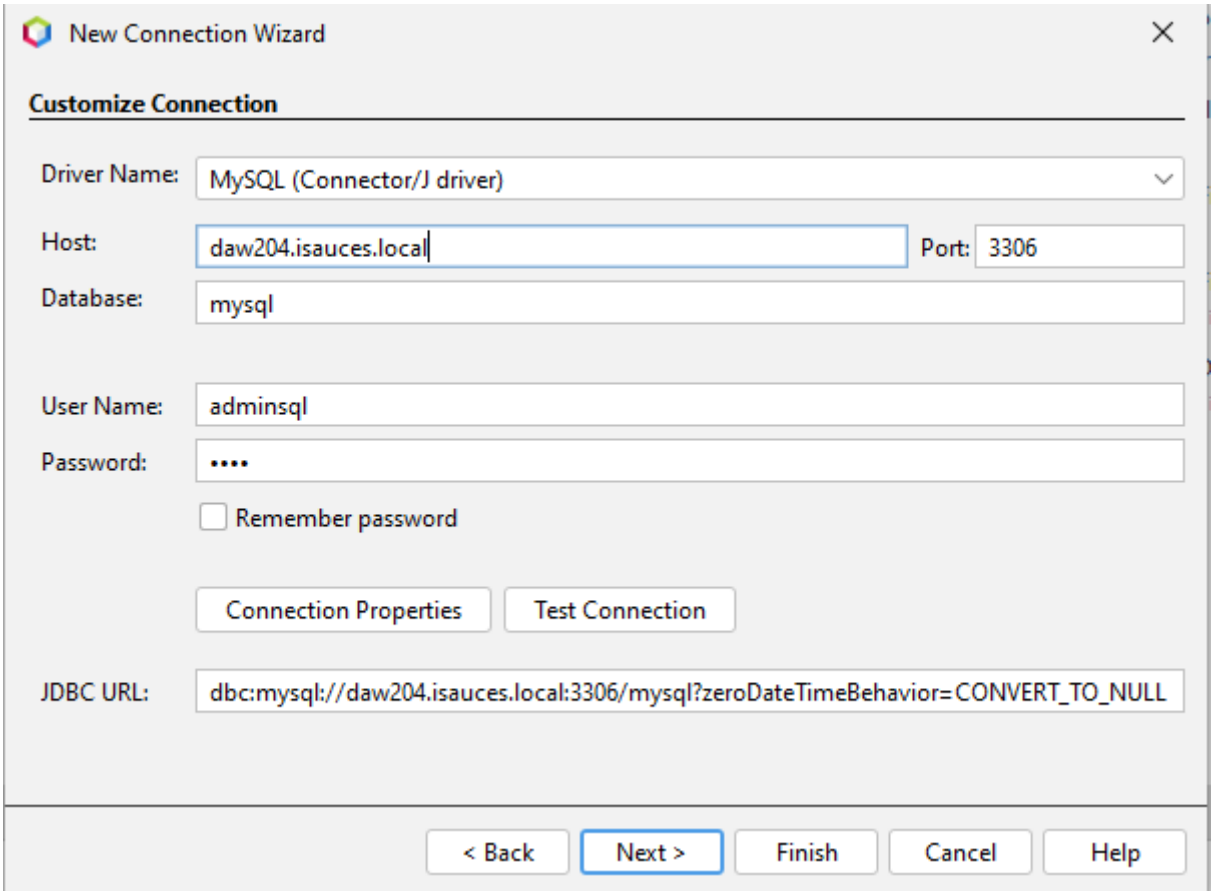
En la siguiente pestaña, seleccionaremos la opción “mysql connector/J driver” y selecionaremos “add”

Seleccionaremos el driver que descargamos hace un momento y continuaremos.

En esta nueva pestaña configuraremos la conexión

En host indicaremos la ip/alias de la maquina en la que este la base de datos, la base de datos puede ser la generica (mysql) o una ya creada, el puerto sera el predefinido para mysql (3306) y el usuario sera uno ya creado por el administrador de la base de datos

Si

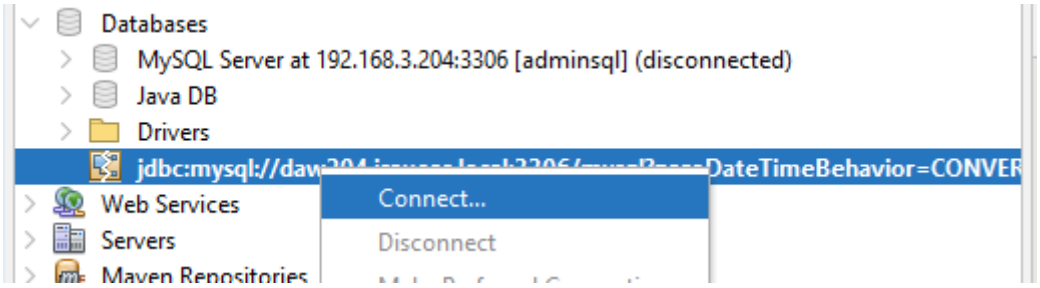


todo ha ido bien, en el desplegable “databases” debería aparecer nuestra conexión

Si desplegamos esta conexión, podremos ver las bases de datos disponibles (solo podremos hacerlo si estamos conectados)

Como conectarse:

Click derecho sobre la conexión que acabamos de crear



Para probar esta conexión, crearemos una base de datos y jugaremos con ella usando scripts, crearemos 4 archivos .sql

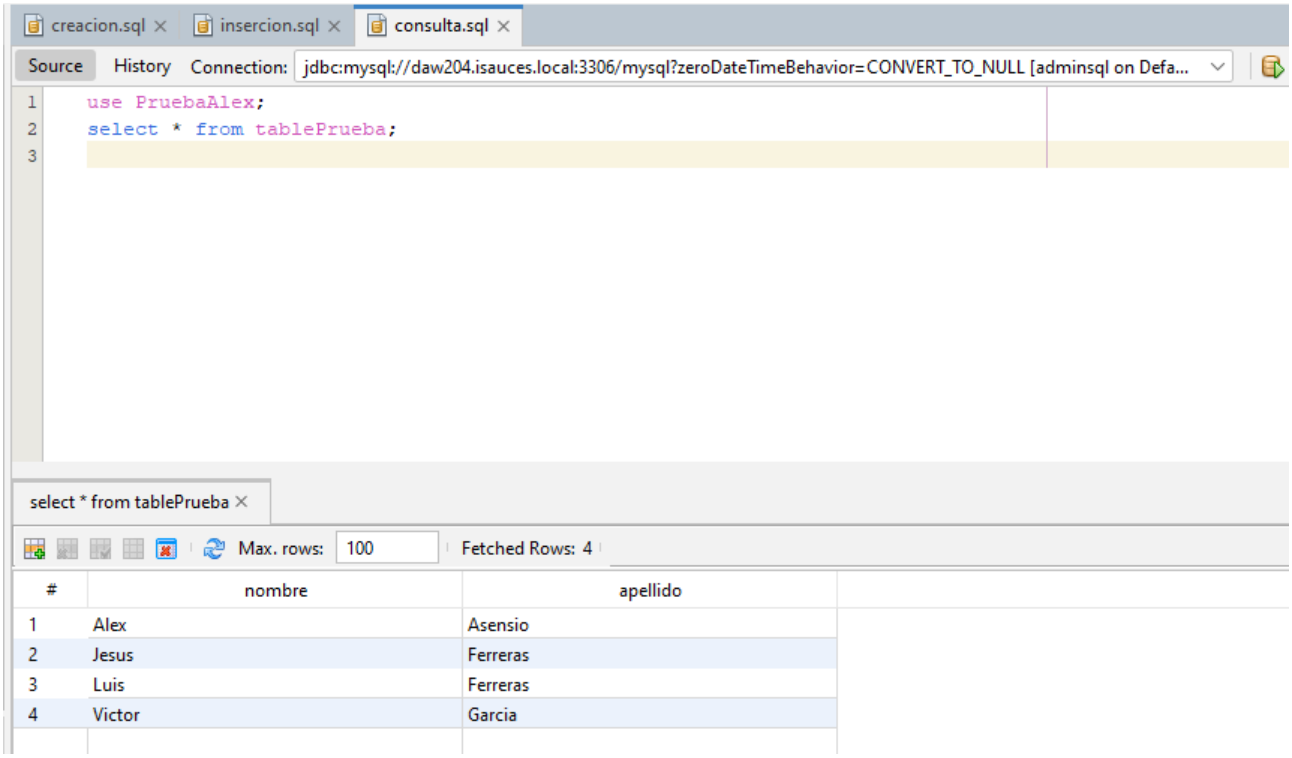
Escribimos el script de creación de una base de datos y un administrador para la misma,

en la parte de arriba seleccionaremos la conexión que queramos usar

Haremos click en este boton (o pulsaremos control+hift+e) =>

Tras ejecutar, podemos ver que la base de datos se ha creado

Ahora insertaremos datos y haremos una consulta

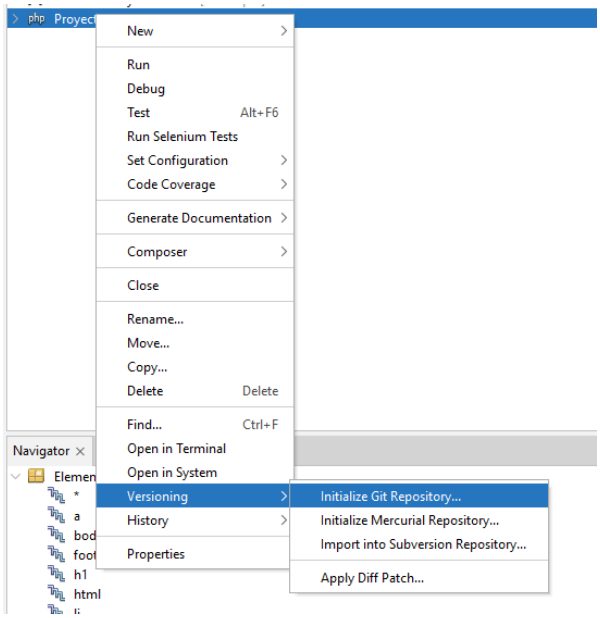


Para terminar, borramos la base de datos de prueba

Conexión al repositorio – versionado

Creamos un repositorio (no obligatorio, puede ser uno ya creado) y copiamos su url

En el proyecto que creamos anteriormente, inicializamos un repositorio de git



Haremos un commit de los archivos del proyecto

Hacemos un push (ojo, esto lo hago por que el repositorio remoto esta vacío, si el repositorio tiene archivos que no están en local o si ha sido actualizado por una fuente externa, es importante hacer un “pull” para descargar todo lo anterior mencionado en nuestro repositorio local)

Aquí introducimos la URL del repositorio, el nombre de nuestra cuenta, y en el apartado “password” introduciremos nuestro **token**

Y ya habríamos subido nuestros archivos a remoto

Depuración - Configuración de la ejecución para la depuración

PHP Doc

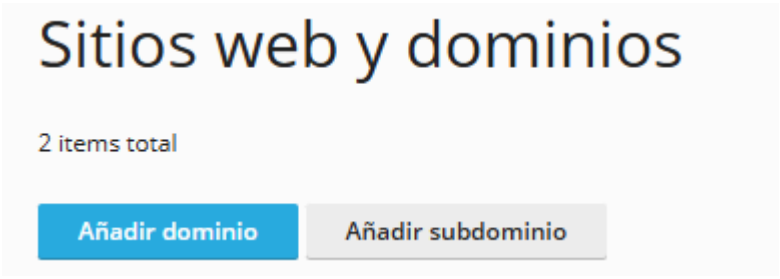
CSS / JS / AJAX / XML / JSON

Paso a explotación-PLESK

Utilizando GitHub

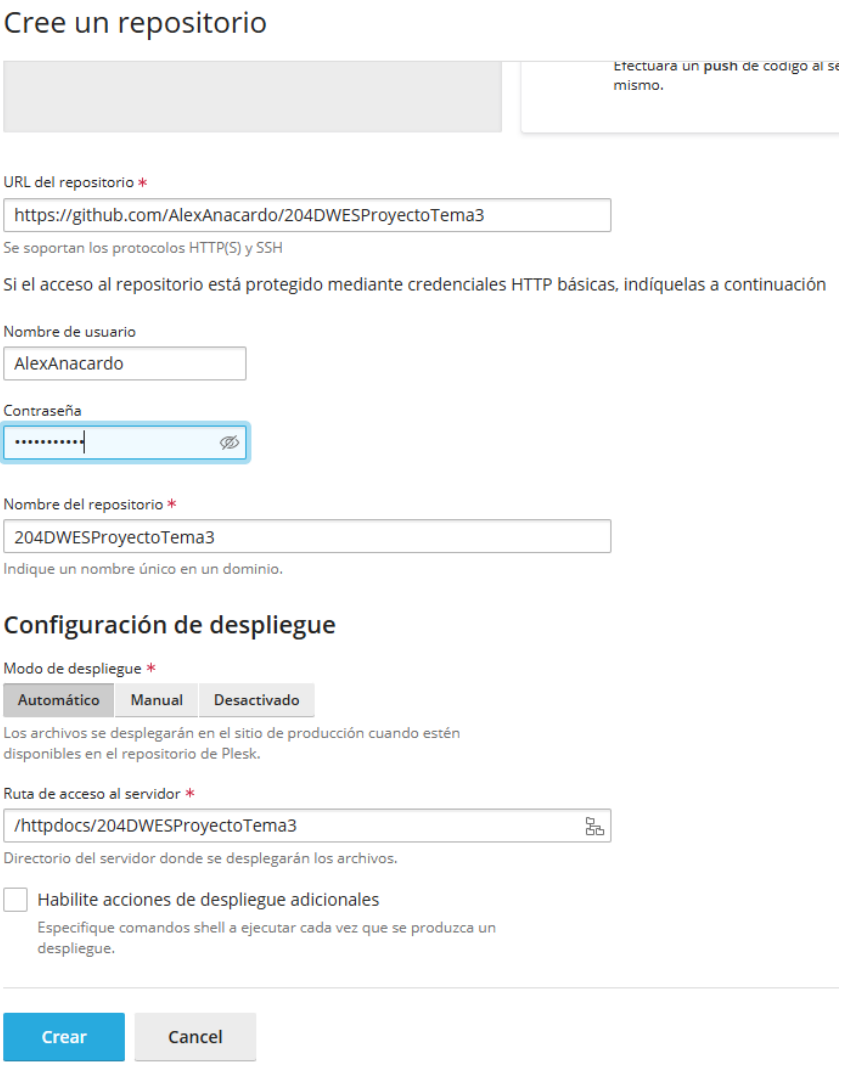
Primero deberemos crear un nuevo subdirectorio, en este caso su nombre será el nombre del proyecto + la extensión del dominio

La carpeta en la que se guardaran los archivos estará dentro de la carpeta httpdocs y su nombre será el mismo que el del proyecto



Tras esto seleccionaremos la opción ‘desplegar mediante git’

Seleccionaremos la opción ‘repositorio remoto’, introduciremos la url del repositorio, nuestro usuario y contraseña (que no token, ojo) de git. Dejaremos lo demás tal y como esta y le daremos a crear



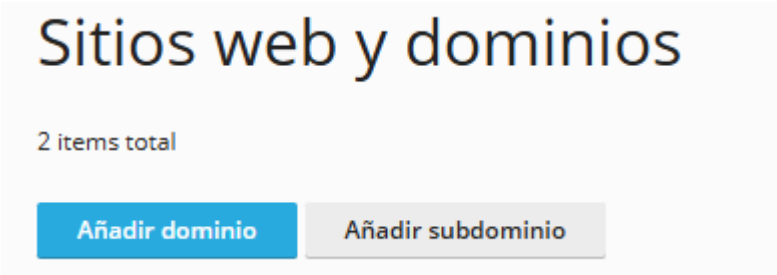
Importante, para poder entrar a esa subpágina a través del subdominio, deberemos incluir un archivo htaccess indicándole cual es el índice del proyecto
Ejemplo:

Nos aseguraremos de que la rama que se usara para subir los archivos es la rama master y seleccionamos “desplegar ahora”

Ya hecho esto, nuestros archivos habrán sido subidos al entorno de explotación

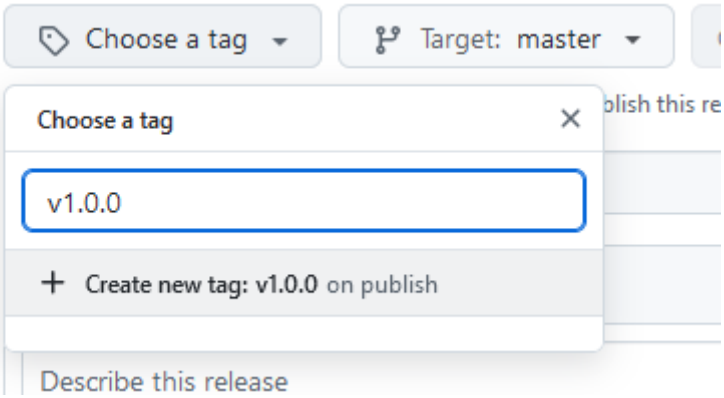
Subiendo los archivos manualmente

Creamos un subdominio tal y como en el anterior apartado

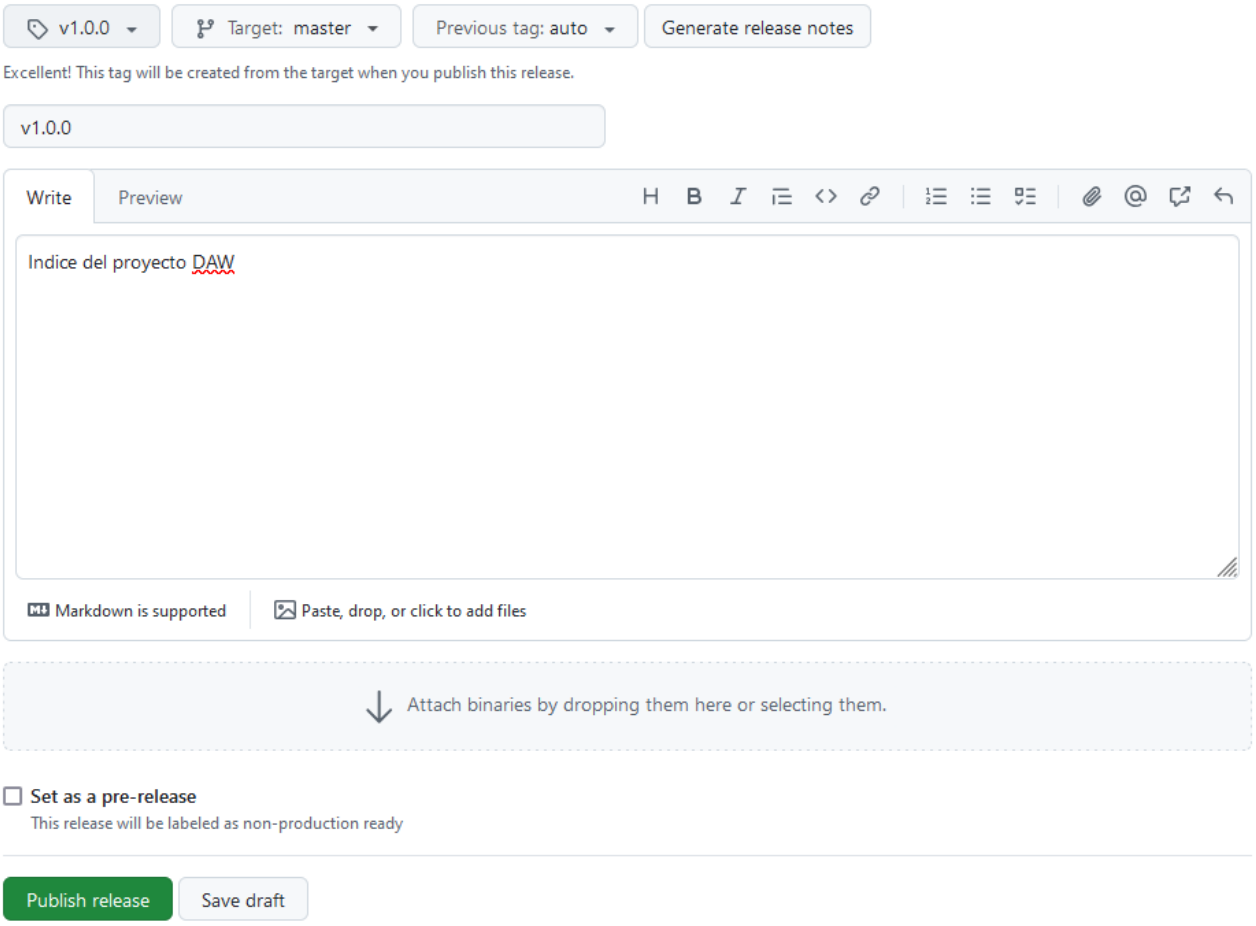


Vamos a nuestro repositorio de github, entraremos en la pestaña de “releases”

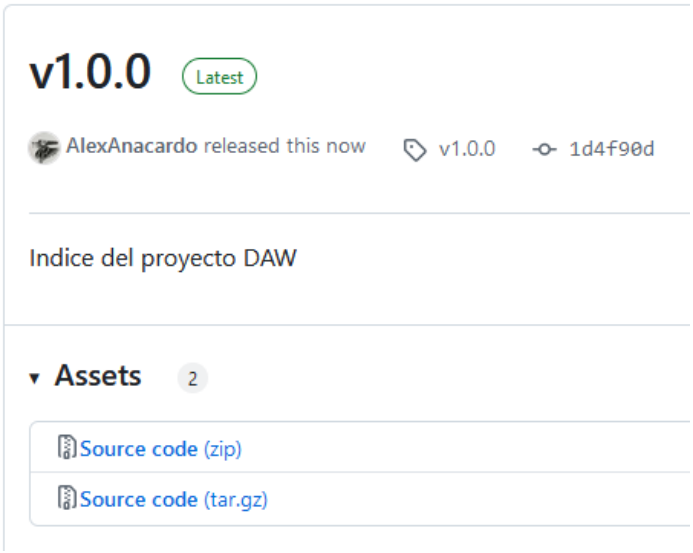
Crearemos una nueva etiqueta, como esta es la primera release sera la version 1.0.0



Elegiremos la rama master como la rama a usar y como titulo pondremos el nombre de la version, publicamos la release



Ahora tenemos la opcion de descargar nuestros archivos como un zip



Descargaremos dicho zip y lo descomprimimos

De vuelta a Plesk, entraremos en nuestro subdirectorio y en la pestaña “Primeros pasos” selecionaremos “cargar archivos”

Aqui borraremos el index generado por defecto y dejaremos los otros dos archivos ahi

Haremos click en el + azul y seleccionaremos “cargar archivo” (puede que para subir carpetas sea obligatorio usar la opcion “cargar directorio”)

Seleccionamos los archivos a subir