

INSTALACIÓN UBUNTU SERVER

Ultima actualización 15/11/2024

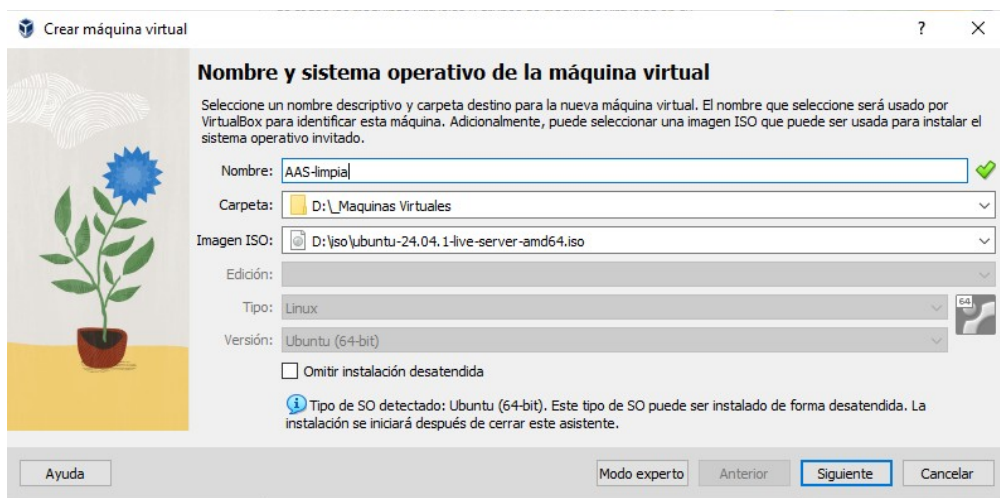
Sumario

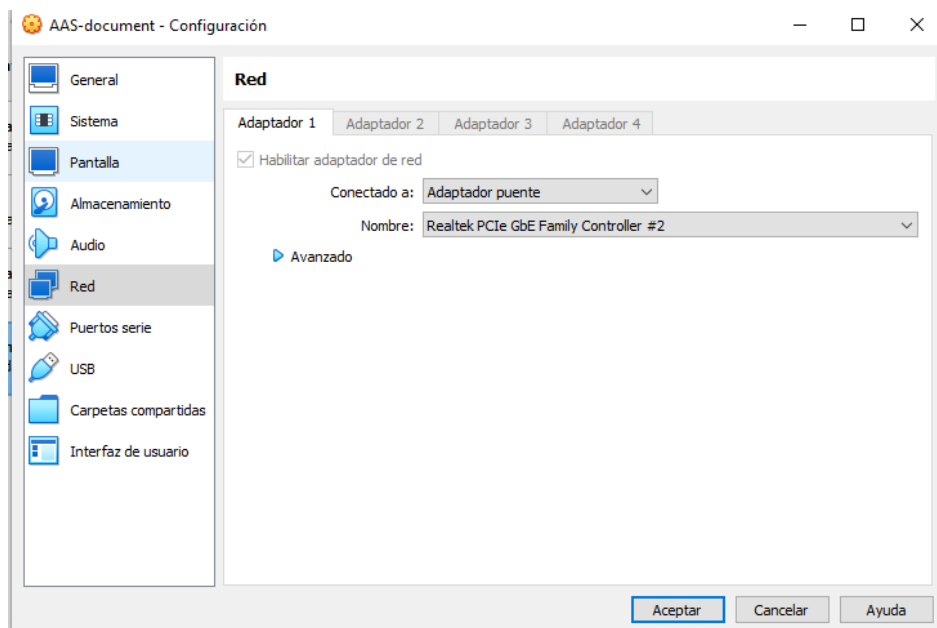
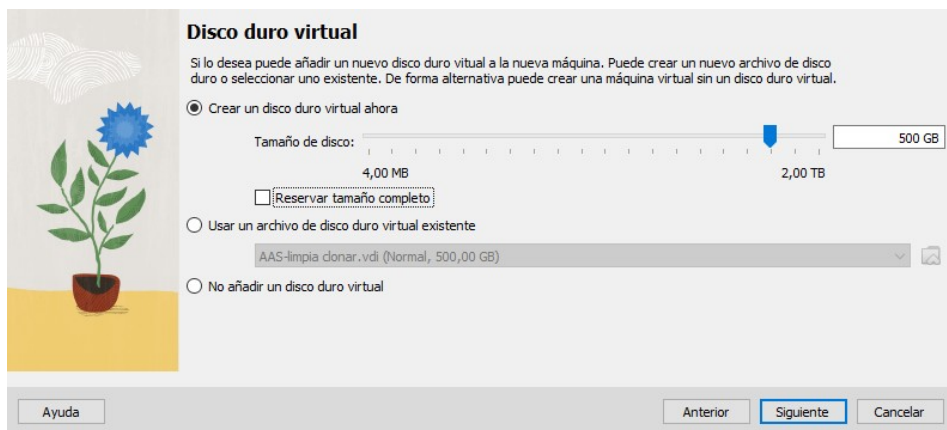
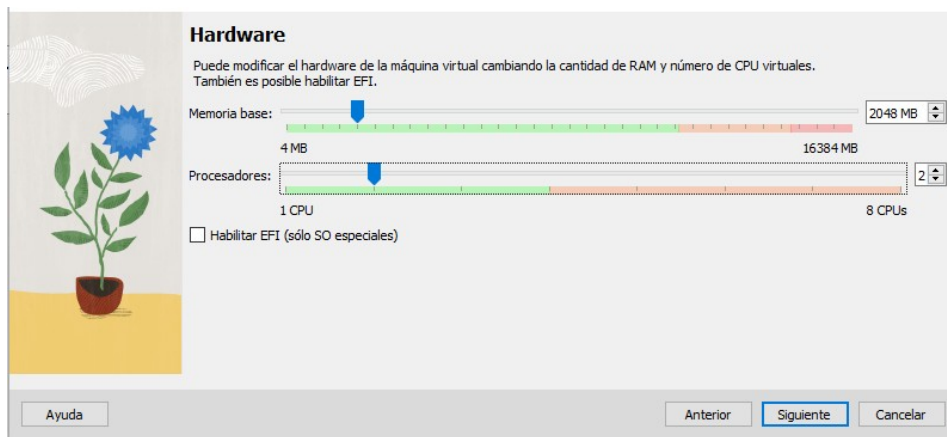
USED-Ubuntu server.....	3
Configuración inicial.....	3
Cuentas de administración.....	16
Apache.....	17
PHP.....	19
MySQL.....	20
XDebug.....	21
Cuentas de desarrollo y hosting virtual.....	22
GITHUB-Internet.....	23
Cuentas de desarrollador.....	23
WXED-Windows X.....	25
Instalación y configuración inicial de la máquina.....	25
Cuentas administradoras y cuenta de desarrollador.....	26
Filezilla.....	27
NetBeans.....	28
Instalación y configuración inicial (plugins).....	28
Creación de proyectos, modificación, borrado, prueba.....	28
Conexión al servidor remoto SFTP. (Almacenamiento local/almacenamiento remoto).....	33
Administración de la base de datos.....	34
Conexión al repositorio – versionado.....	40
Depuración - Configuración de la ejecución para la depuración.....	44
Creación de un proyecto nuevo.....	45
PHP Doc.....	46
CSS / JS / AJAX / XML / JSON.....	46
Paso a explotación-PLESK.....	47
Utilizando GitHub.....	47

USED-Ubuntu server

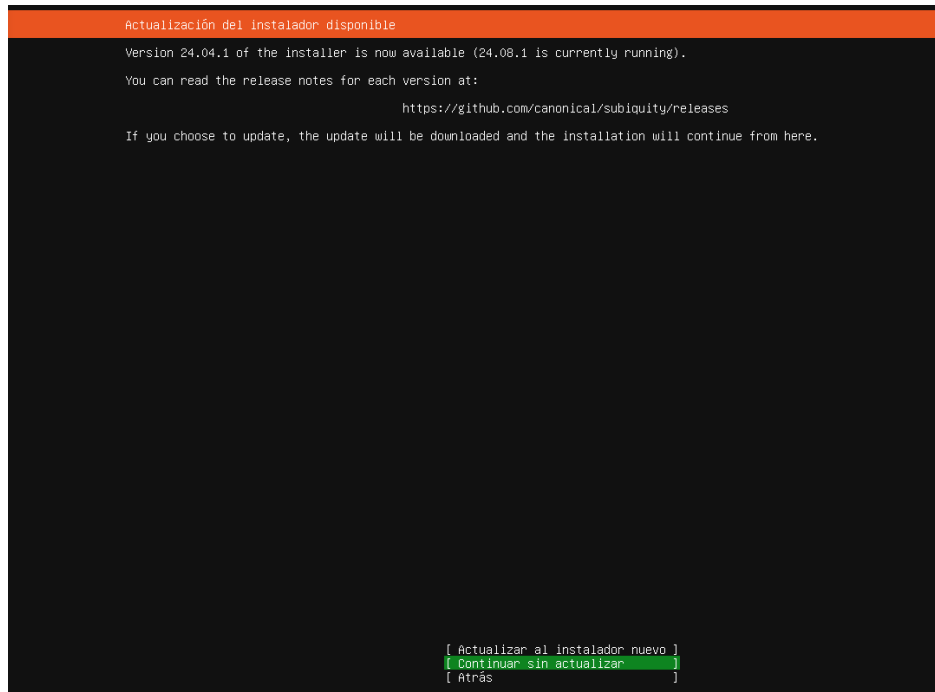
Configuración inicial

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

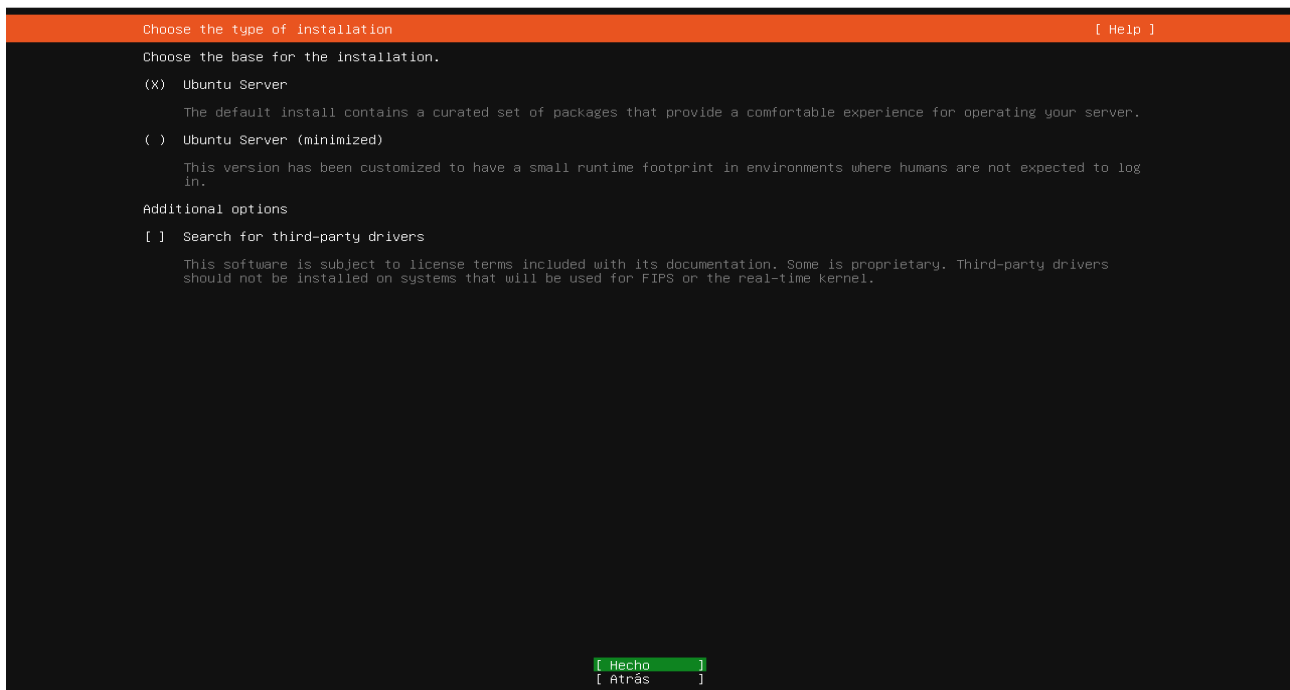




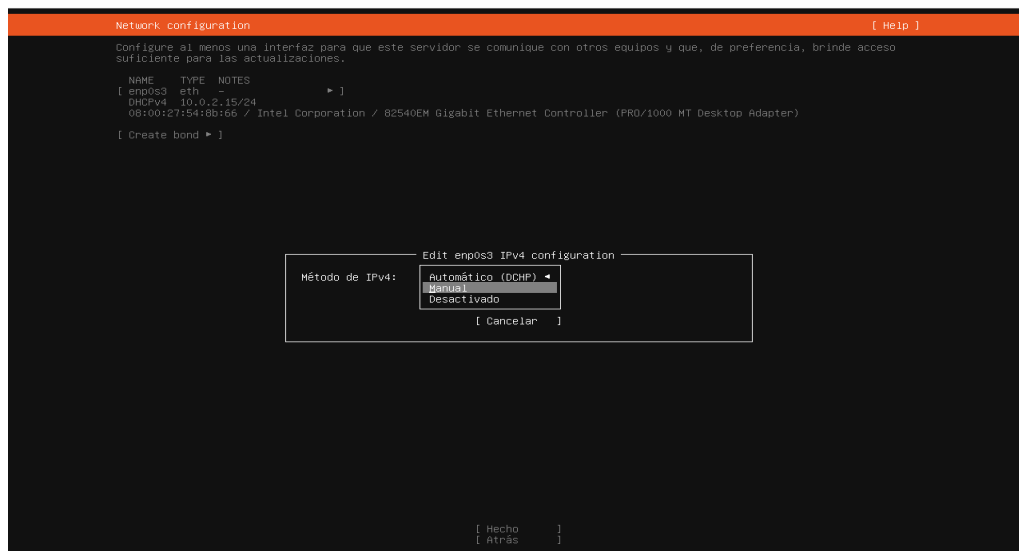
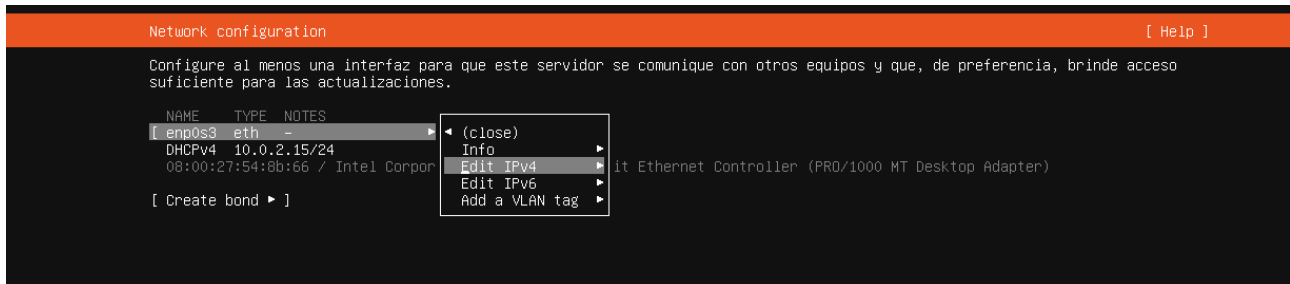
Ya podremos arrancar la maquina, seleccionamos el idioma que queramos. Continuamos sin actualizar



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

Network configuration [Help]

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

NAME	TYPE	NOTES
enp0s3	eth	-

[enp0s3 eth - ▶]

DHCPv4 10.0.2.15/24
08:00:27:54:8b:66 / Intel Corporation / 82540EM Gigabit Ethernet Controller (PRO/1000 MT Desktop Adapter)

[Create bond ▶]

Edit enp0s3 IPv4 configuration

Método de IPv4: [Manual ▼]

Subred: 192.168.3.0/24

Dirección: 192.168.3.204

Puerta de enlace: 192.168.3.1

Servidores de nombres: 192.168.20.30, 8.8.8.8
IP addresses, comma separated

Dominios de búsqueda:
Domains, comma separated

[Guardar]
[Cancelar]

[Hecho]
[Atrás]

No haremos nada con esto

Proxy configuration [Help]

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

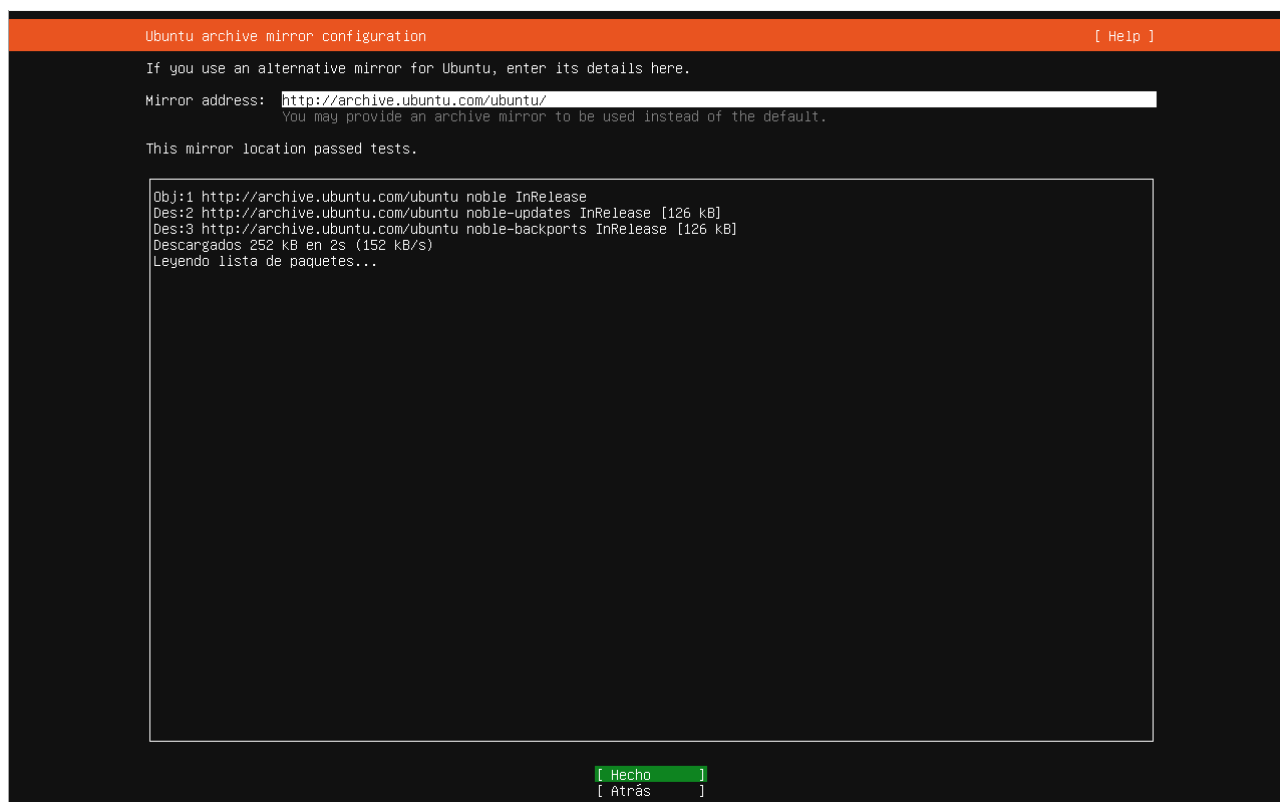
Proxy address:

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

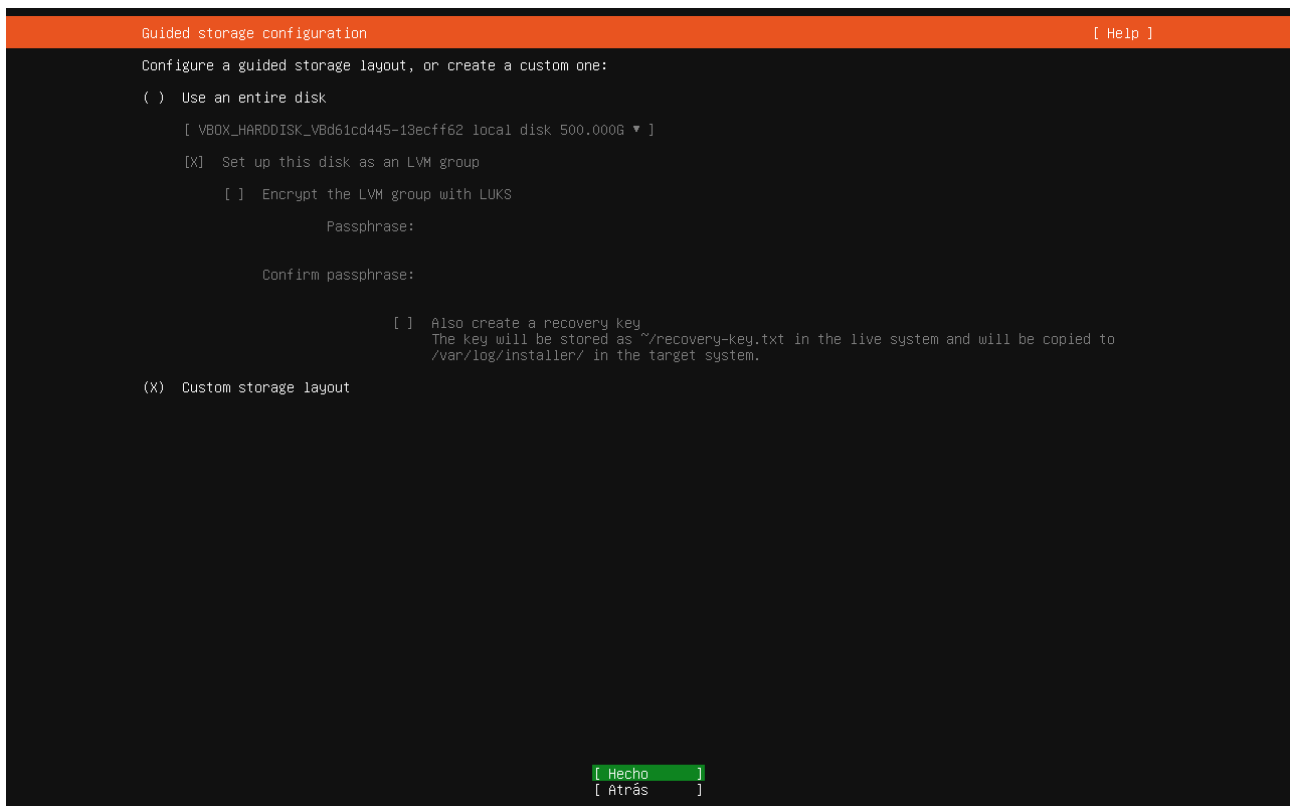
The proxy information should be given in the standard form of "http://[user][:pass]@host[:port]/".

[Hecho]
[Atrás]

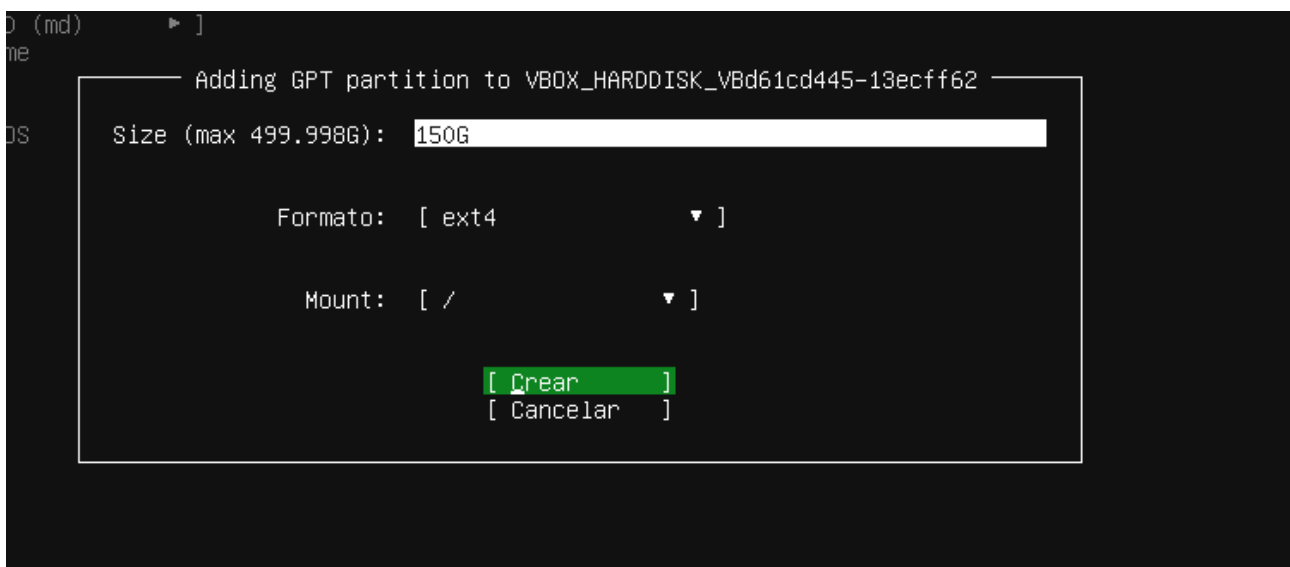
Ni con esto, simplemente continuamos



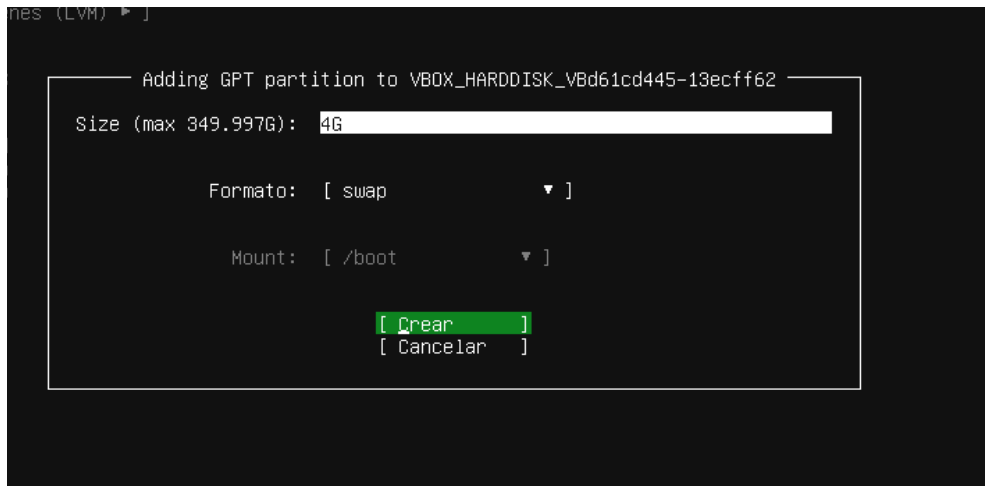
Seleccionaremos la opción “custom store layout”



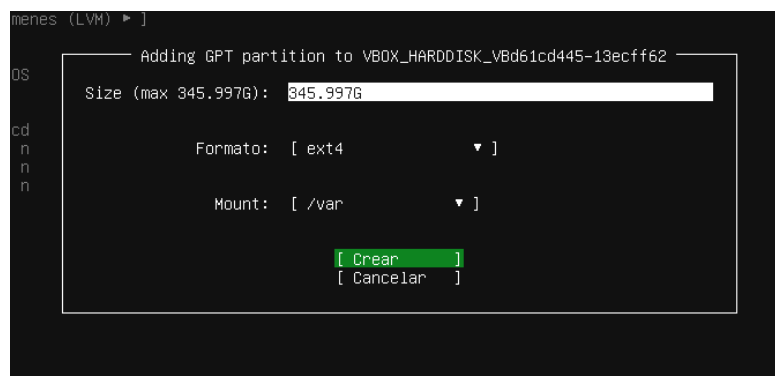
Crearemos 3 particiones, una de 150GB para el sistema



Una de 4GB de tipo swap



Y una con el espacio restante para /var



Introducimos nuestros credenciales

Profile configuration [Help]

Enter the username and password you will use to log in to the system. You can configure SSH access on a later screen, but a password is still needed for sudo.

Su nombre:

Your servers name:
The name it uses when it talks to other computers.

Elija un nombre de usuario:

Elija una contraseña:

Confirme la contraseña:

Nos saltamos la instalacion de ubuntu pro

Upgrade to Ubuntu Pro [Help]

An internet connection is required to enable Ubuntu Pro.

[About Ubuntu Pro ►]

☐ Enable Ubuntu Pro

☒ Skip Ubuntu Pro setup for now

Once you are connected to the Internet, you can enable Ubuntu Pro using the 'pro attach' command.

[Continuar]
[Atrás]

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

SSH configuration

You can choose to install the OpenSSH server package to enable secure remote access to your server.

☒ Instalar servidor OpenSSH

Dejamos que se instale la maquina y la reiniciaremos cuando termine

```
Installation complete! [ Help ]

configuring mount: mount-0
executing curtin install extract step
curtin command install
writing install sources to disk
running 'curtin extract'
curtin command extract
acquiring and extracting image from cp:///tmp/tmpmvvyvnea6/mount
configuring keyboard
curtin command in-target
executing curtin install curthooks step
curtin command install
configuring installed system
running 'curtin curthooks'
curtin command curthooks
configuring apt configuring apt
installing missing packages
Installing packages on target system: ['grub-pc']
configuring iscsi service
configuring raid (mdadm) service
configuring NVMe over TCP
installing kernel
Setting up swap
apply networking config
writing etc/fstab
configuring multipath
updating packages on target system
configuring pollinate user-agent on target
updating initramfs configuration
configuring target system bootloader
installing grub to target devices
copying metadata from /cdrom
final system configuration
calculating extra packages to install
installing openssh-server
retrieving openssh-server
curtin command system-install
unpacking openssh-server
curtin command system-install
configuring cloud-init
restoring apt configuration
subiquity/Late/run:

[ View full log ]
[ Reiniciar ahora ]
```

Ya reiniciada, comprobaremos que tenemos acceso a internet haciendo ping

```
miadmin@daw-limpia:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=12.5 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=13.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=113 time=13.5 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=113 time=13.5 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=113 time=14.2 ms
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4107ms
rtt min/avg/max/mdev = 12.531/13.458/14.182/0.531 ms
```

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

```
miadmin@daw-limpia:~$ sudo nano /etc/hosts
```

Cambiamos el nombre

```
127.0.0.1 localhost
127.0.1.1 daw-limpia_
# The following lines are de
```

```
127.0.0.1 localhost
127.0.1.1 aas-used_
```

Introducimos el siguiente comando y reiniciamos la maquina

```
miadmin@daw-limpia:~$ sudo hostnamectl set-hostname aas-used
```

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

```
miadmin@aas-used:~$ timedatectl
          Local time: lun 2024-10-14 06:43:21 UTC
          Universal time: lun 2024-10-14 06:43:21 UTC
              RTC time: lun 2024-10-14 06:43:21
          Time zone: Etc/UTC (UTC, +0000)
System clock synchronized: no
              NTP service: active
          RTC in local TZ: no
miadmin@aas-used:~$
```

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

```
miadmin@aas-used:~$ sudo timedatectl set-timezone Europe/Madrid
[sudo] password for miadmin:
miadmin@aas-used:~$ timedatectl
          Local time: lun 2024-10-14 08:44:04 CEST
        Universal time: lun 2024-10-14 06:44:04 UTC
           RTC time: lun 2024-10-14 06:44:04
          Time zone: Europe/Madrid (CEST, +0200)
System clock synchronized: yes
            NTP service: active
          RTC in local TZ: no
miadmin@aas-used:~$
```

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

```
miadmin@aas-used:~$ sudo adduser miadmin2
info: Adding user `miadmin2' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `miadmin2' (1001) ...
info: Adding new user `miadmin2' (1001) with group `miadmin2 (1001)' ...
info: Creating home directory `/home/miadmin2' ...
info: 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 []: miadmin2
   Room Number []:
    Work Phone []:
    Home Phone []:
      Other []:
Is the information correct? [Y/n] y
info: Adding new user `miadmin2' to supplemental / extra groups `users' ...
info: Adding user `miadmin2' to group `users' ...
```

Le concedemos permisos

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

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
miadmin@daw-limpia:~$
```

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:

192.168.3.204



Apache2 Default Page



It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
|
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.

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 específico podríamos indicárselo 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”

```
; Production value: Off
; https://php.net/display_errors
display_errors = Off
```

```
; https://php.net/display_errors
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
```

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

(Si 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  
Leyendo lista de paquetes... Hecho
```

```
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/memo
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.
```

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

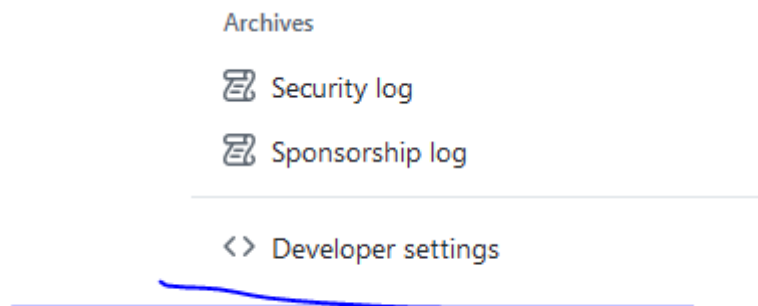
GITHUB-Internet

Cuentas de desarrollador

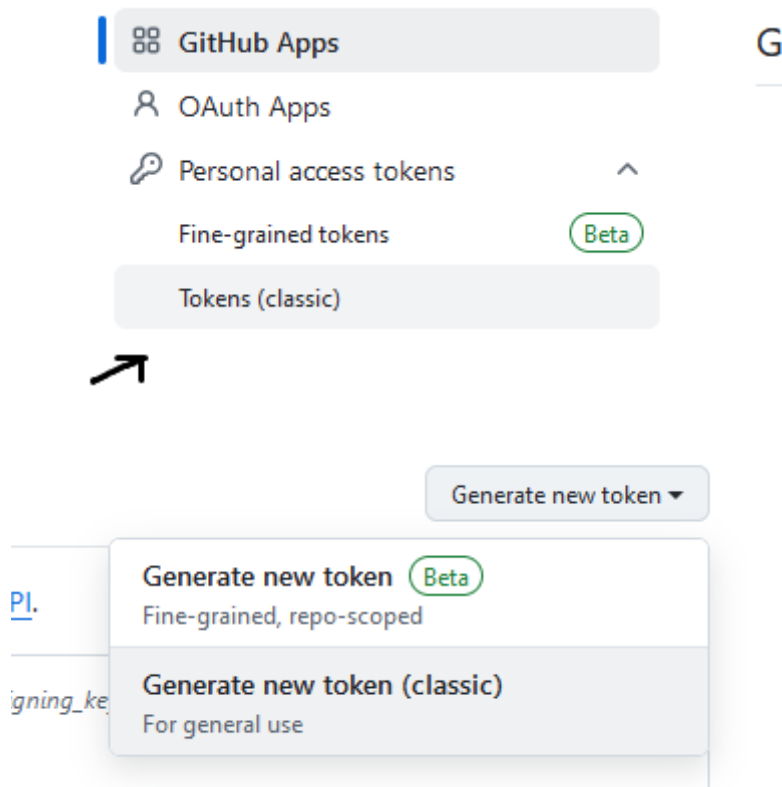
Necesitaremos una cuenta de github, nos registraremos (no tiene perdida) y obtendremos un token personal

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 classic



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

Note

prueba

What's this token for?

Expiration *

7 days ▼

The token will expire on Fri, Oct 11 2024

☐ write:ssh_signing_key

☐ read:ssh_signing_key

Generate token

Cancel

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_ [redacted]

[Delete](#)

[tokenDeAlexGit](#) — *admin:pgp_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

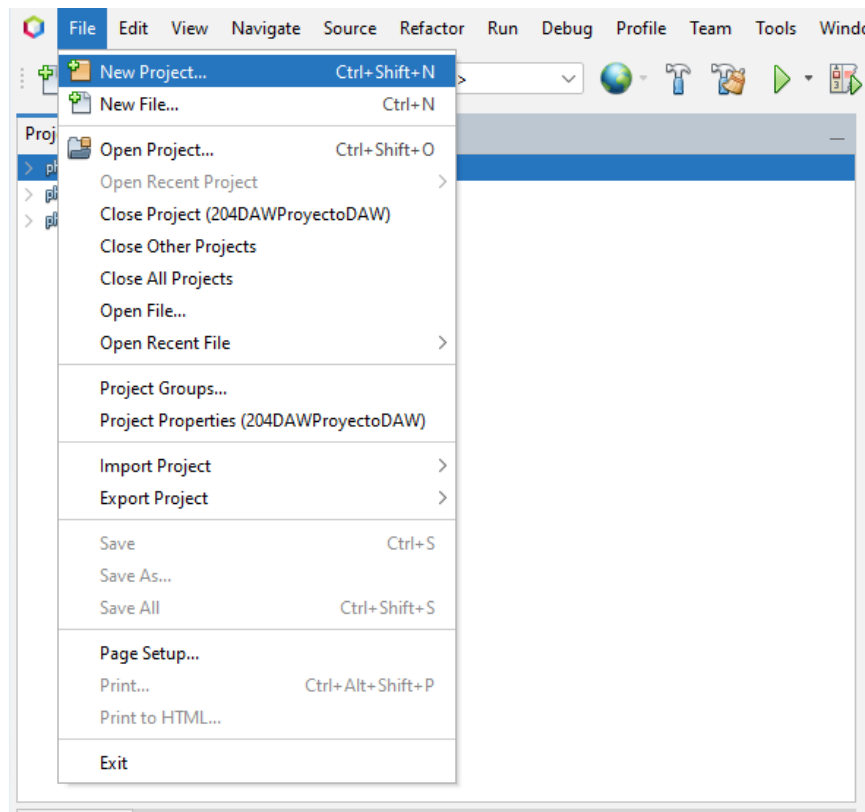
Filezilla

NetBeans

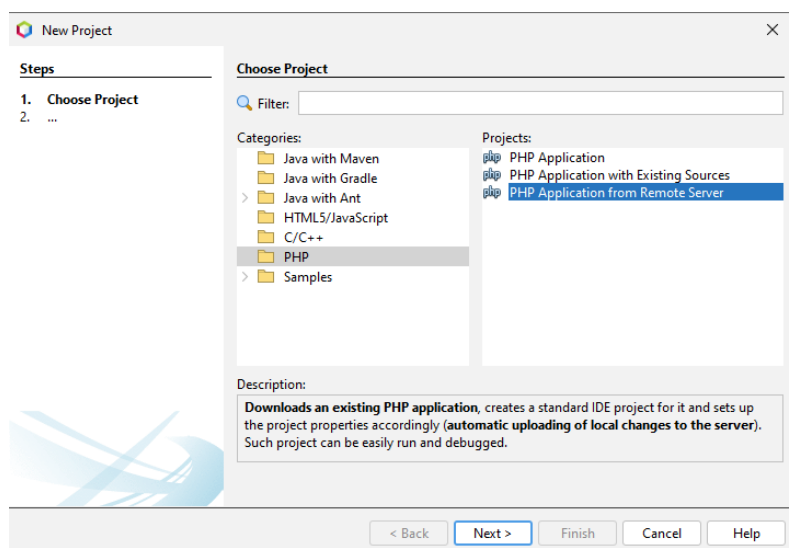
Instalación y configuración inicial (plugins)

Creación de proyectos, modificación, borrado, prueba.

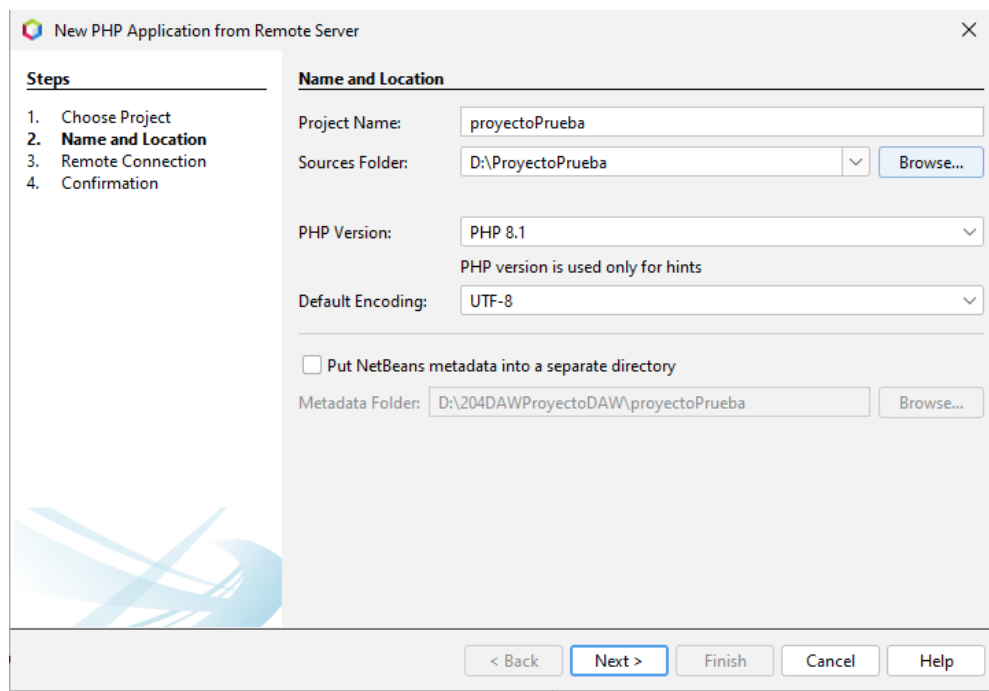
Lo primero sera crear un proyecto de php



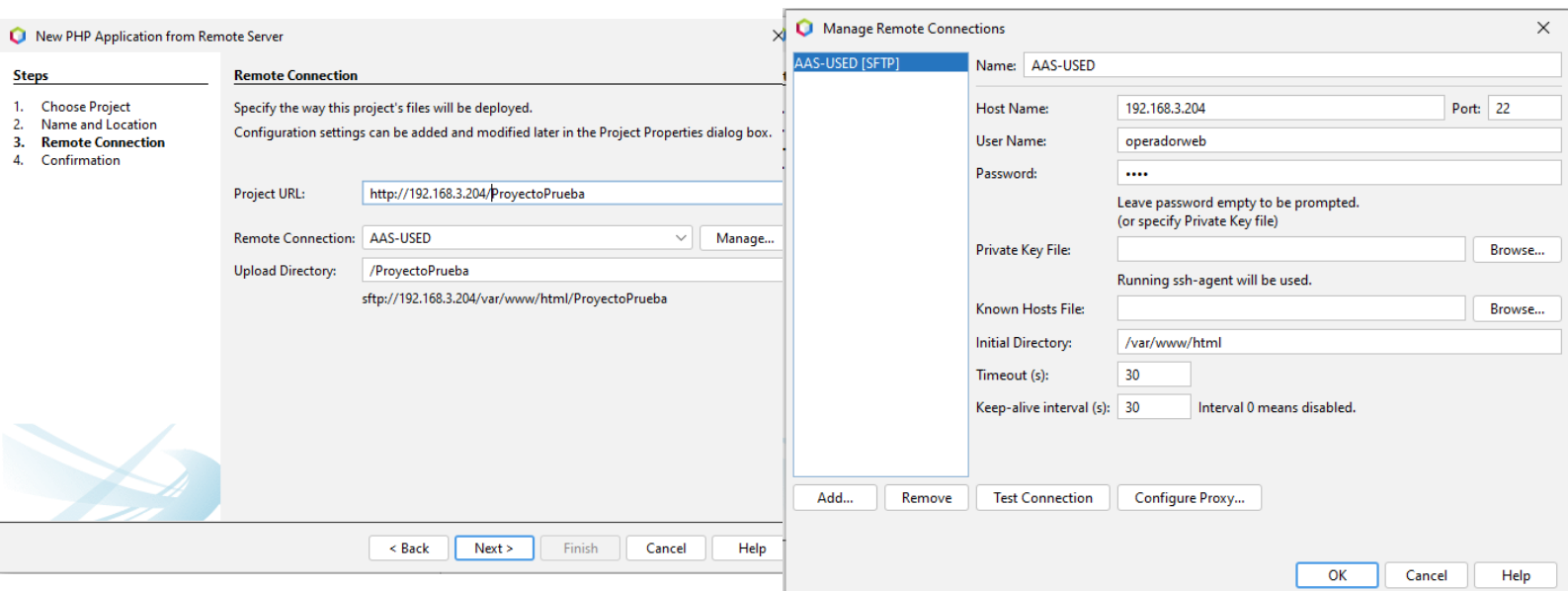
Para crear un proyecto desde 0 seleccionaremos la opción “aplicación desde servidor remoto”



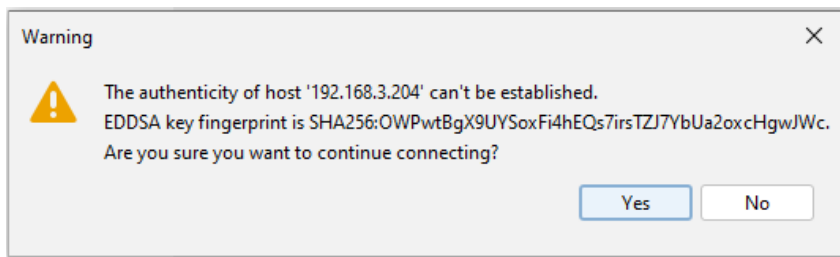
Elegimos el nombre y la ruta del proyecto



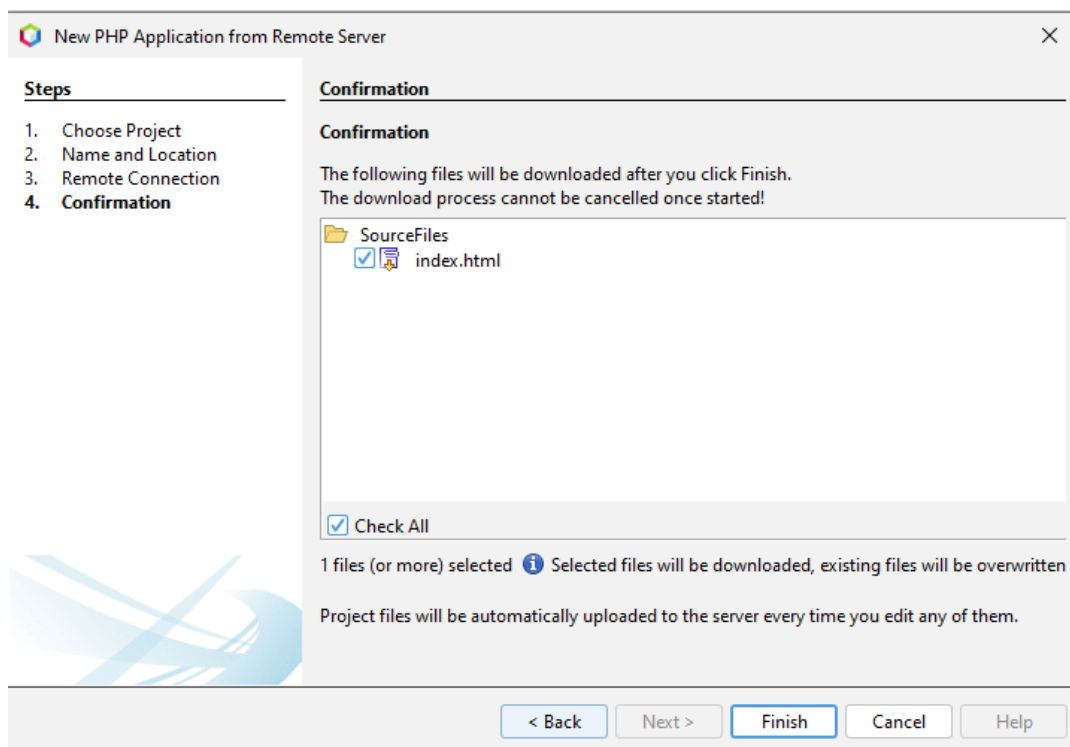
Indicamos la url del directorio del servidor en el que deseamos subir los archivos, también añadimos una conexión remota al servidor por sftp



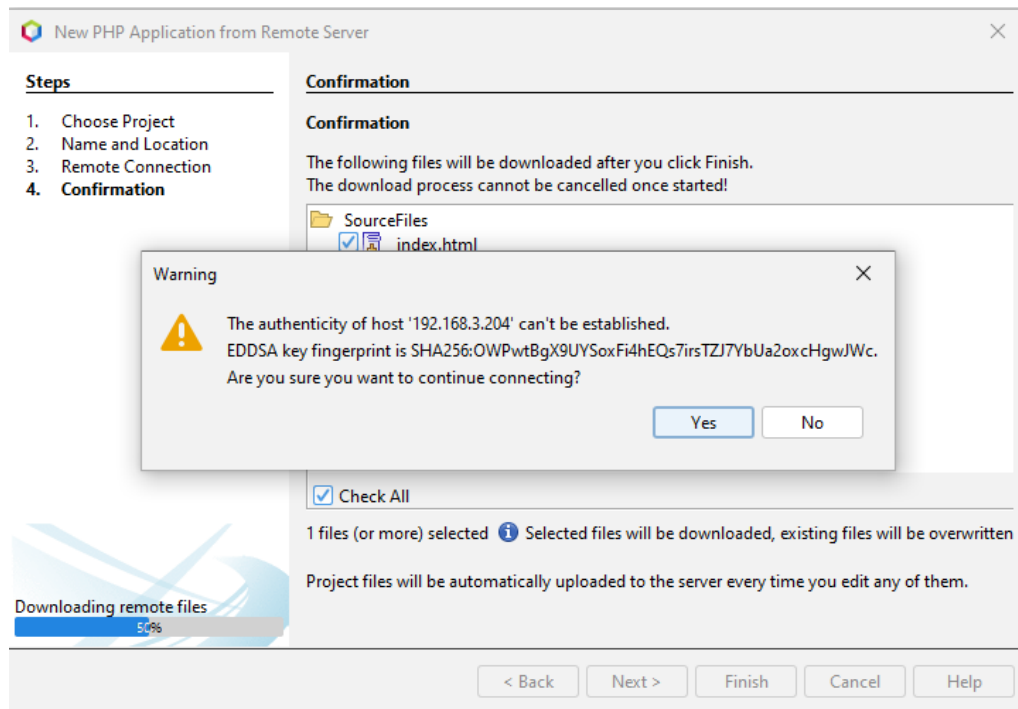
Tras esto seleccionamos que “si”



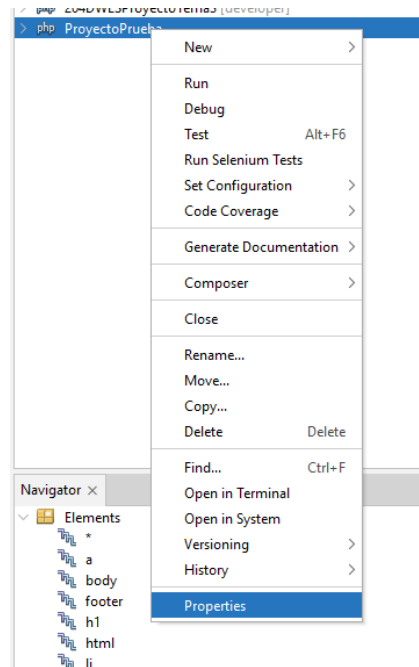
Seleccionamos los archivos que queremos descargar del servidor



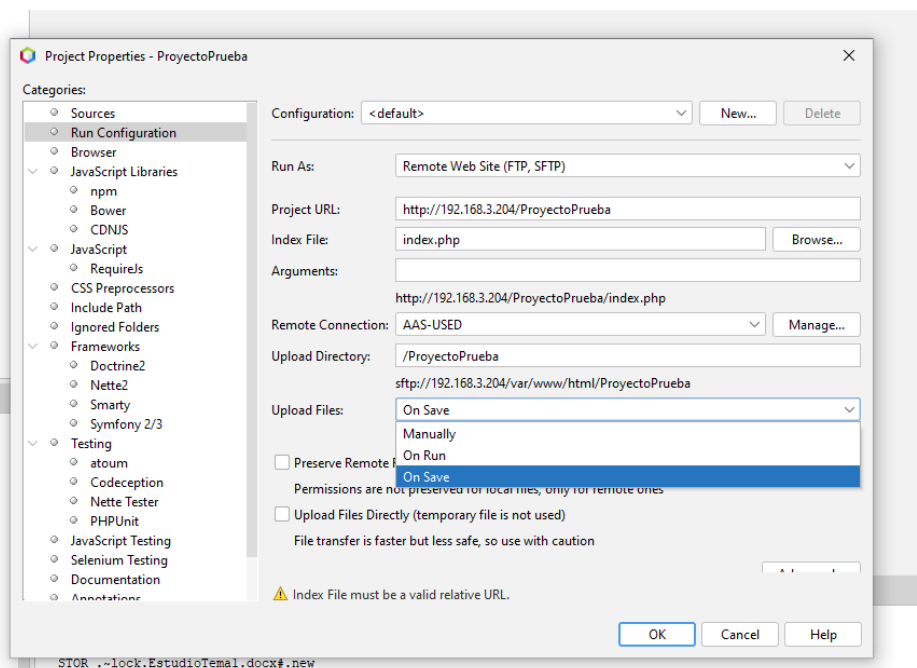
Seleccionamos que si de nuevo



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”



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


Administración de la base de datos

Lo primero que haremos será descargar el driver conector de mysql desde la página de maven:

<https://mvnrepository.com/artifact/com.mysql/mysql-connector-j>

en esta ocasión, descargaremos la versión 9.1

Home » com.mysql » mysql-connector-j



MySQL Connector/J


MySQL Connector/J is a JDBC Type 4 driver, which means that it is pure Java with the Driver Manager, standardized validity checks, categorized SQLExceptions, JDBC-4.x XML processing, support for per connection client information and

Categories	JDBC Drivers
Tags	database sql jdbc driver connector rdbms mys
Ranking	#607 in MvnRepository (See Top Artifacts) #8 in JDBC Drivers
Used By	880 artifacts

Central (9) Redhat GA (1) Redhat EA (1)

Version	
9.1.x	9.1.0

Home » com.mysql » mysql-connector-j » 9.1.0

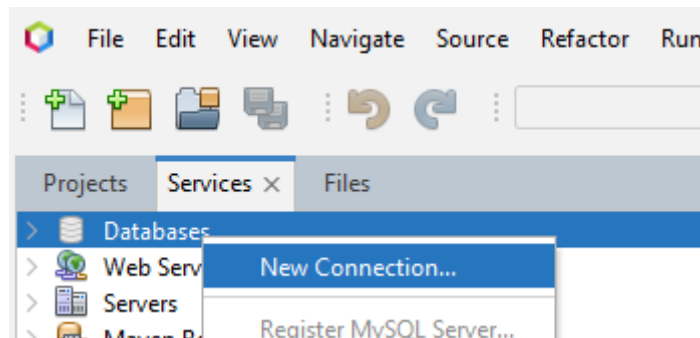


MySQL Connector/J » 9.1.0

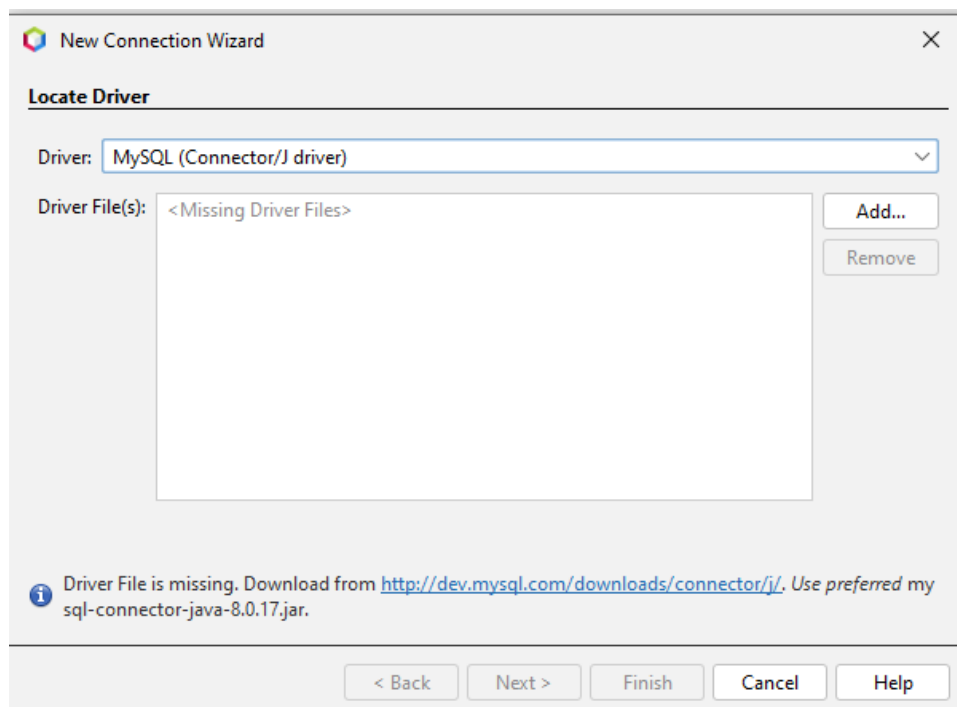
MySQL Connector/J is a JDBC Type 4 driver, which means that it is pure Java implementation of the MySQL with the Driver Manager, standardized validity checks, categorized SQLExceptions, support for large updates, JDBC-4.x XML processing, support for per connection client information and support for the NCHAR, NVARCHAR and CLOB data types.

Categories	JDBC Drivers
Tags	database sql jdbc driver connector rdbms mysql connection
Organization	Oracle Corporation
HomePage	http://dev.mysql.com/doc/connector-j/en/
Date	Oct 15, 2024
Files	pom (3 KB) jar (2.5 MB) View All
Repositories	Central

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”



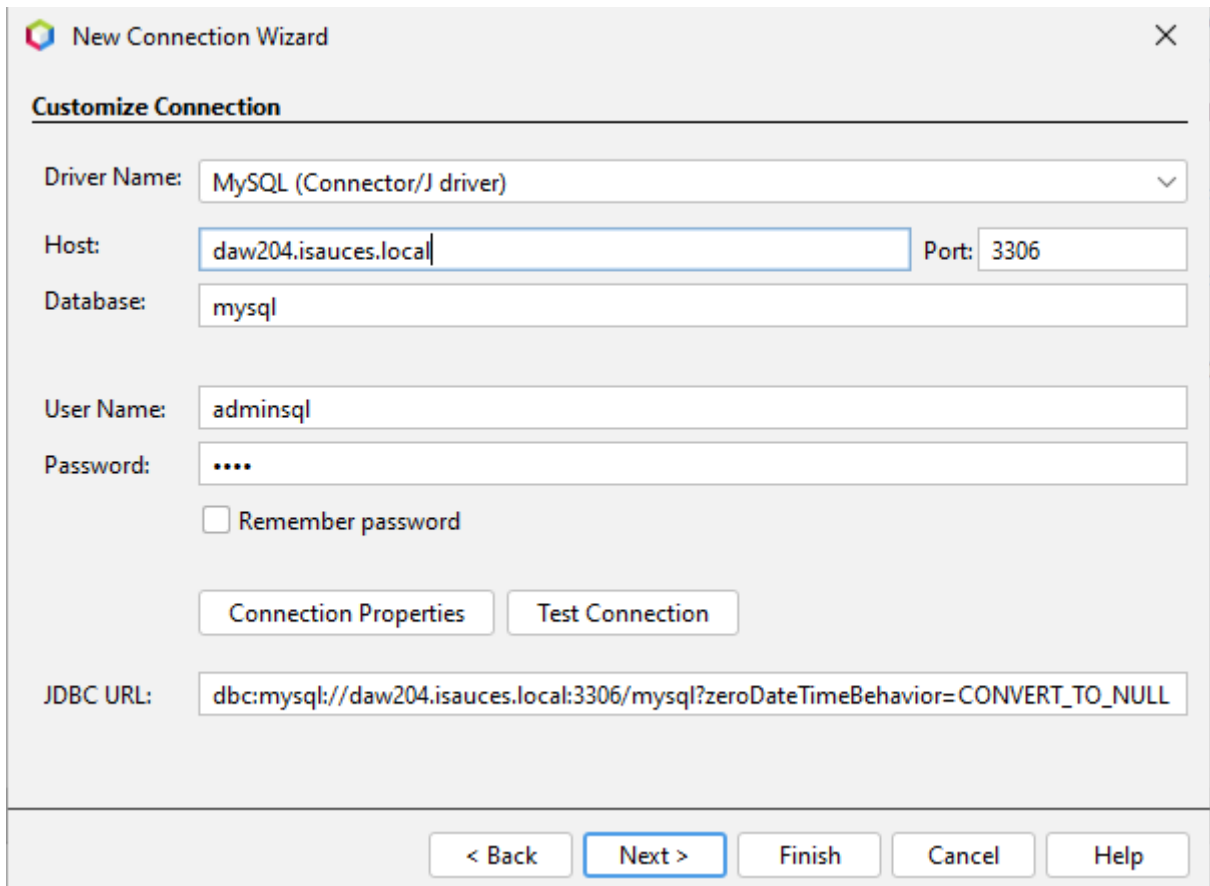
En la siguiente pestaña, seleccionaremos la opción “mysql connector/J driver” y seleccionaremos “add”



Seleccionaremos el driver que descargamos hace un momento y continuaremos.

En esta nueva pestaña configuraremos la conexión

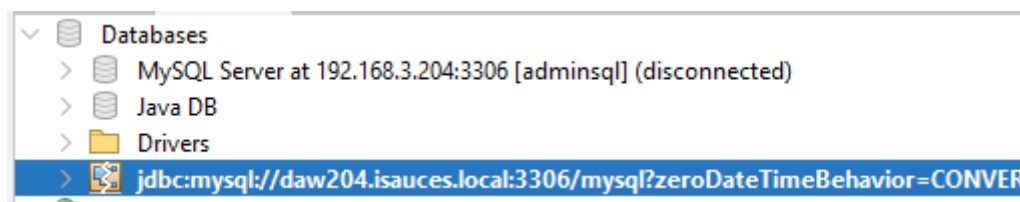
En host indicaremos la ip/alias de la maquina en la que este la base de datos, la base de datos puede ser la generica (mysql) o una ya creada, el puerto sera el predefinido para mysql (3306) y el usuario sera uno ya creado por el administrador de la base de datos



The screenshot shows the 'New Connection Wizard' dialog box with the 'Customize Connection' tab selected. The fields are filled as follows:

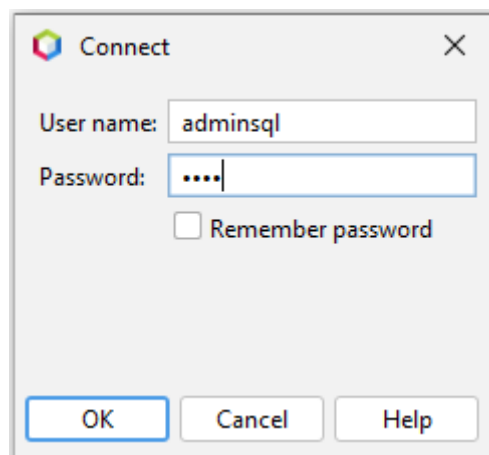
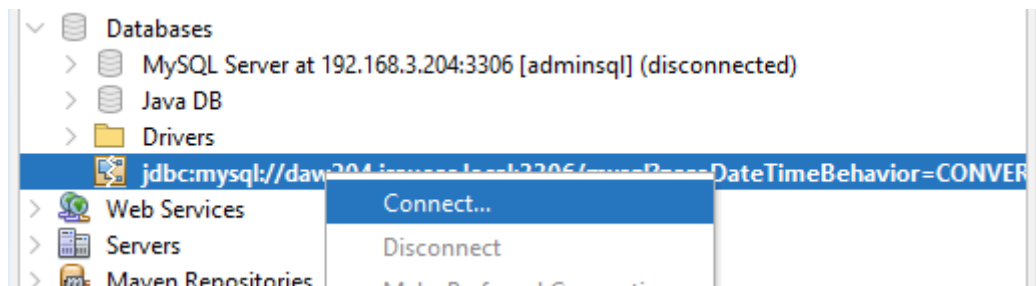
- Driver Name: MySQL (Connector/J driver)
- Host: daw204.isauces.local
- Port: 3306
- Database: mysql
- User Name: adminsql
- Password: (masked with dots)
- ☐ Remember password
- Buttons: Connection Properties, Test Connection
- JDBC URL: jdbc:mysql://daw204.isauces.local:3306/mysql?zeroDateTimeBehavior=CONVERT_TO_NULL
- Navigation buttons: < Back, Next > (highlighted), Finish, Cancel, Help

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:



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

```

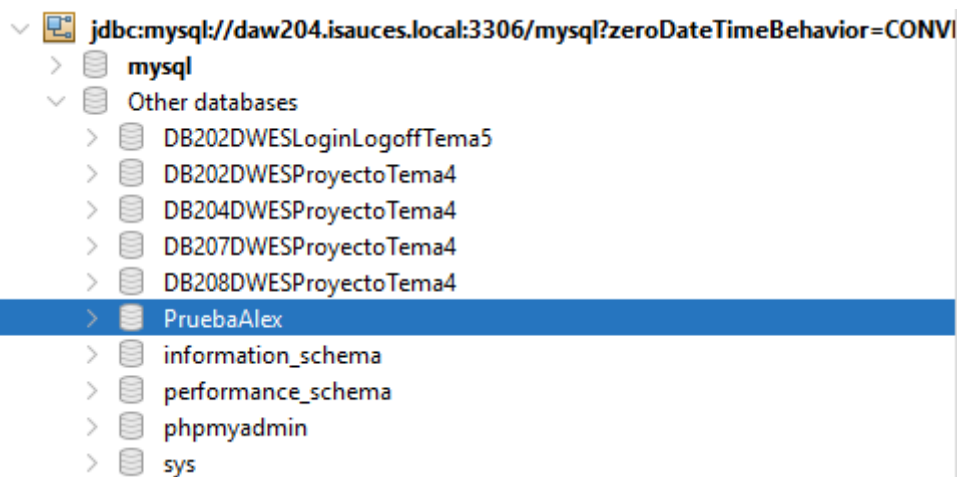
1  create database PruebaAlex;
2
3  use PruebaAlex;
4
5  create user if not exists 'adminPruebaAlex'@'%' identified by 'paso';
6
7  grant all privileges on PruebaAlex.* to 'adminPruebaAlex'@'%;
8
9
10
11 create table tablePrueba(
12     nombre varchar(40) primary key,
13     apellido varchar(60)
14 )engine=innodb;

```

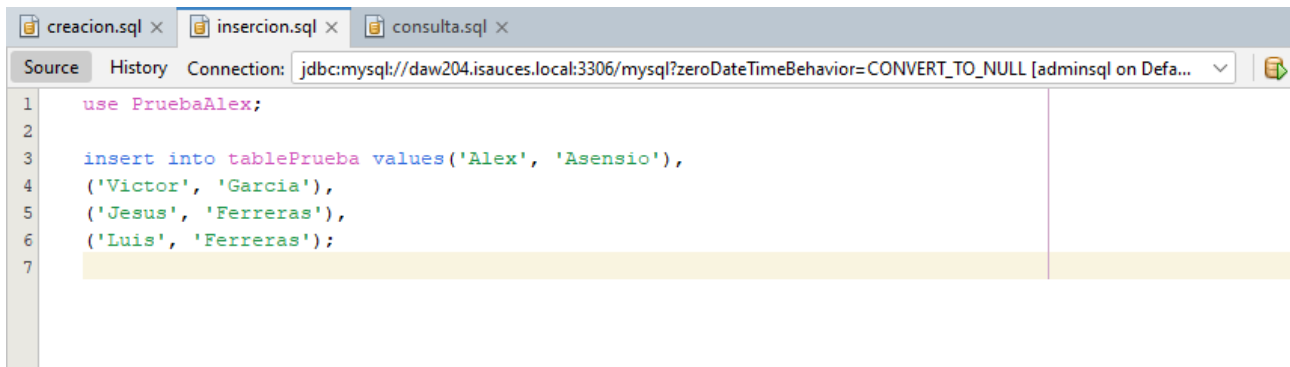
Haremos click en este boton (o pulsaremos control+shift+E) =>



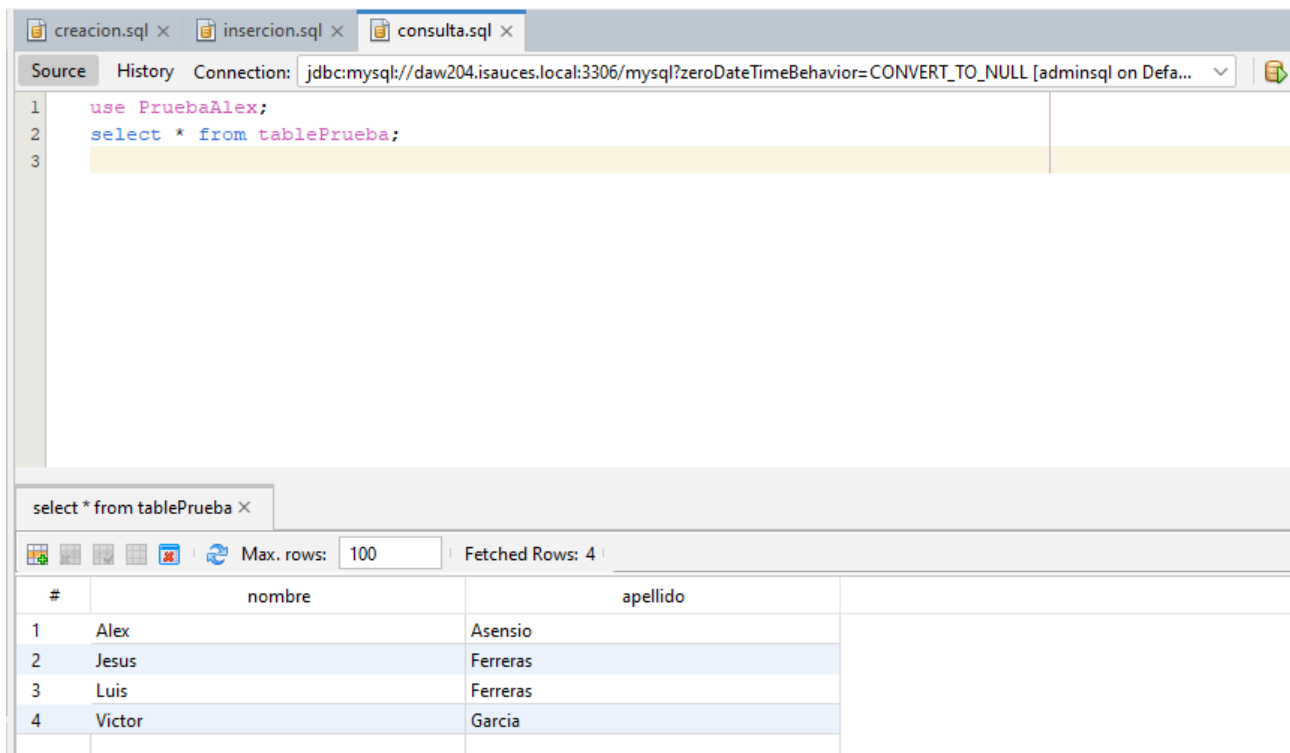
Tras ejecutar, podemos ver que la base de datos se ha creado



Ahora insertaremos datos y haremos una consulta



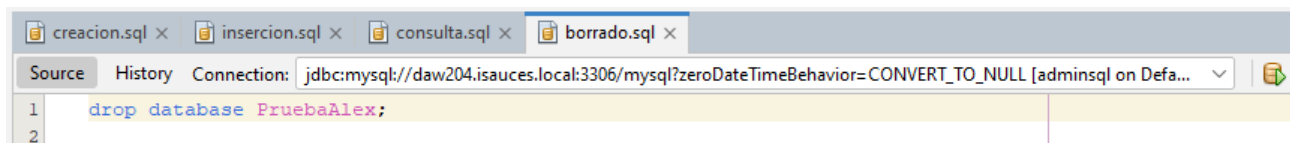
```
1 use PruebaAlex;
2
3 insert into tablePrueba values('Alex', 'Asensio'),
4 ('Victor', 'Garcia'),
5 ('Jesus', 'Ferrereras'),
6 ('Luis', 'Ferrereras');
7
```



```
1 use PruebaAlex;
2 select * from tablePrueba;
3
```

#	nombre	apellido
1	Alex	Asensio
2	Jesus	Ferrereras
3	Luis	Ferrereras
4	Victor	Garcia

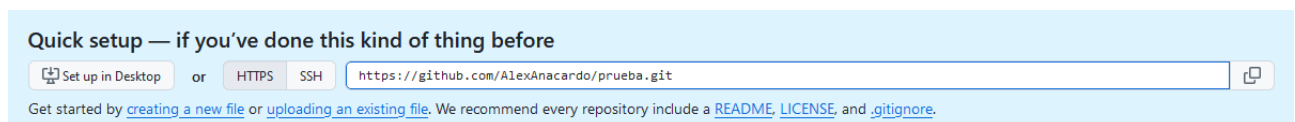
Para terminar, borramos la base de datos de prueba



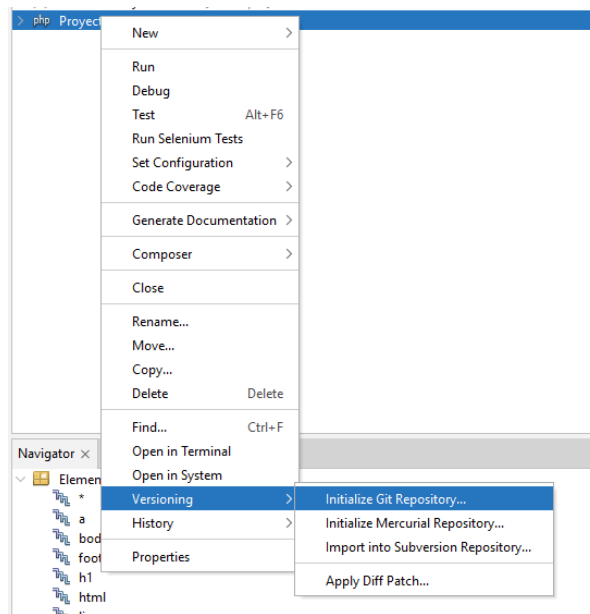
```
1 drop database PruebaAlex;
2
```

Conexión al repositorio – versionado

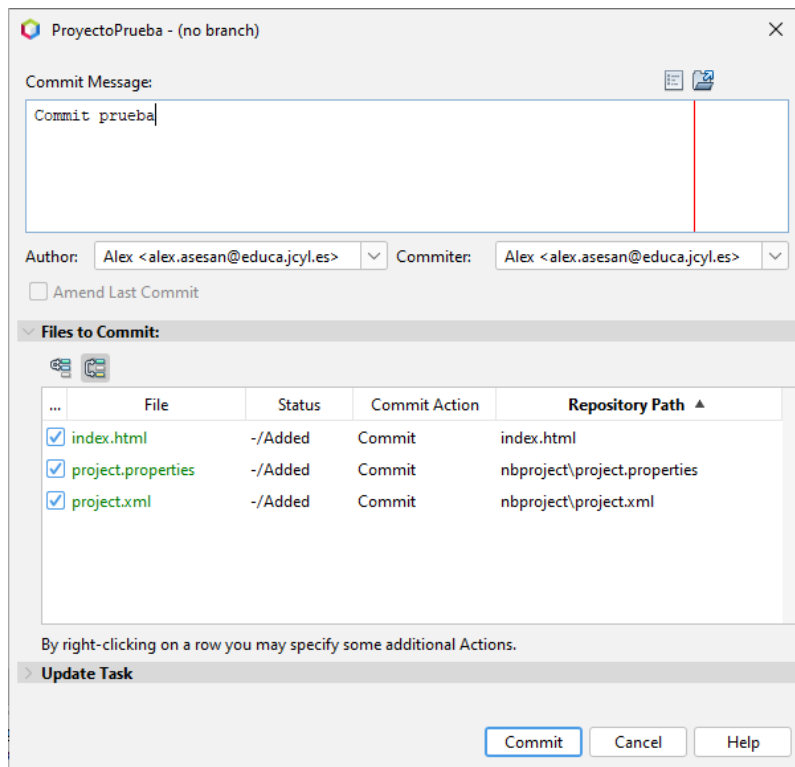
Creamos un repositorio (no obligatorio, puede ser uno ya creado) y copiamos su url



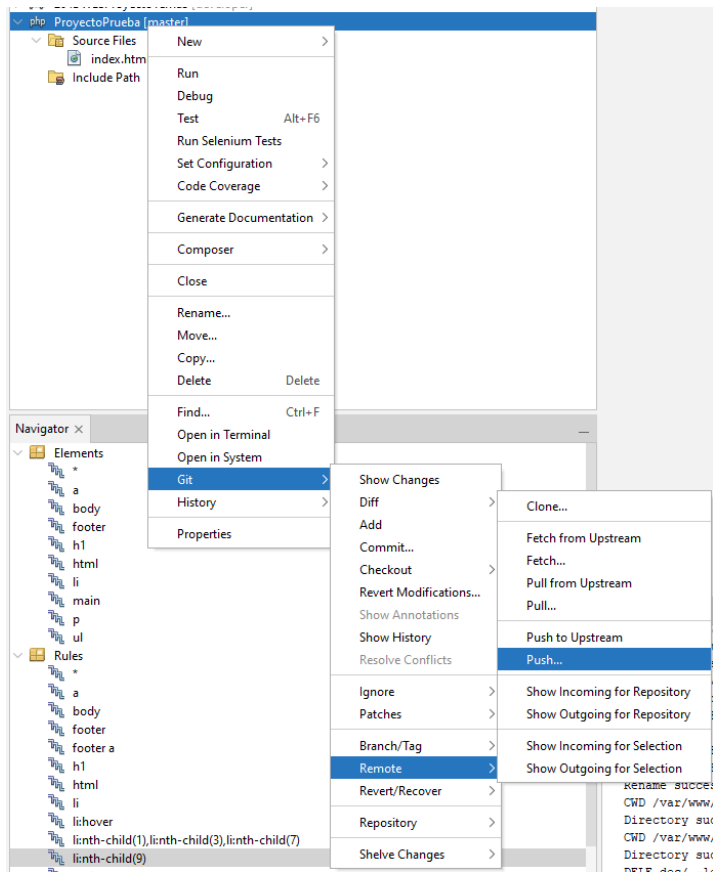
En el proyecto que creamos anteriormente, inicializamos un repositorio de git



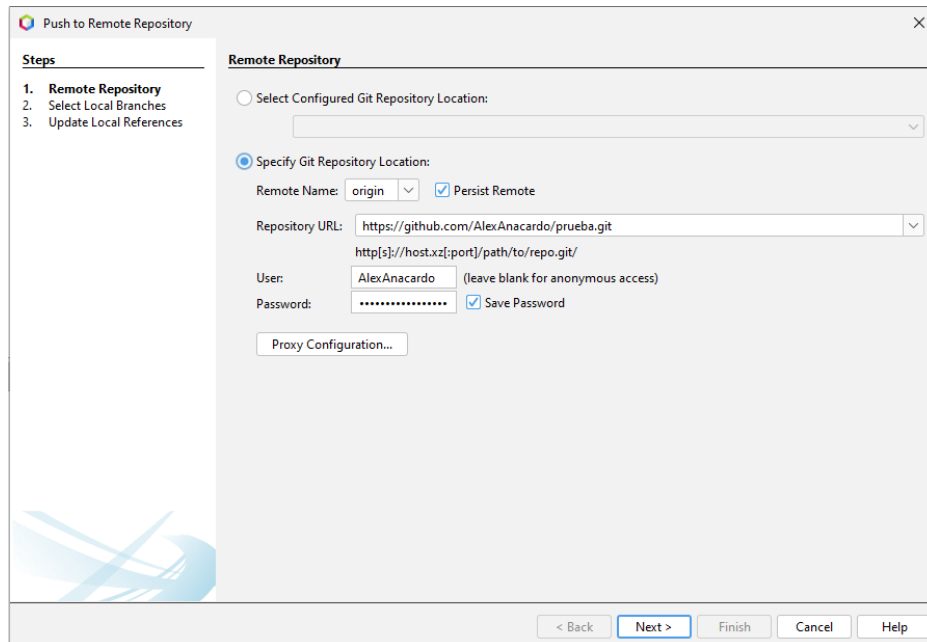
Haremos un commit de los archivos del proyecto



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

Creación de un proyecto nuevo

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



Añadir un subdominio

Los subdominios son direcciones de Internet para las distintas secciones de su sitio web. Estos utilizan su nombre de dominio. También puede crear un subdominio wildcard introduciendo el símbolo * en vez del nombre. En este caso, los visitantes del sitio se verán dirigidos a la carpeta correspondiente.

Nombre del subdominio * .

Introduzca * para crear un subdominio wildcard.

Configuración de hosting

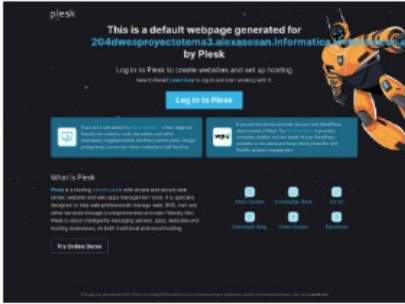
Raíz del documento * /

Ruta al directorio principal del sitio web.

* Campos obligatorios

Tras esto seleccionaremos la opción ‘desplegar mediante git’

204DWESProyectoTema3.alexasesan.informatica.ieslossau




Estadísticas

Espacio en disco 0 MB
 Tráfico este mes 0 MB

[Panel de información](#)
[Hosting y DNS](#)
[Correo](#)
[Primeros pasos](#)


Crear un sitio web o una aplicación en este dominio





Sitejet Builder

Creador de sitios web de última generación: + de 100 plantillas de diseño, sin código, generador de texto con IA, SEO, E-commerce y mucho más.

[Crear sitio web](#)
[Más info](#)


[Cargar archivos](#)


[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

Cree un repositorio

estructura un push de código al ser mismo.

URL del repositorio *

Se soportan los protocolos HTTP(S) y SSH

Si el acceso al repositorio está protegido mediante credenciales HTTP básicas, indíquelas a continuación

Nombre de usuario

Contraseña

Nombre del repositorio *

Indique un nombre único en un dominio.

Configuración de despliegue

Modo de despliegue *

☒ Automático☐ Manual☐ Desactivado

Los archivos se desplegarán en el sitio de producción cuando estén disponibles en el repositorio de Plesk.

Ruta de acceso al servidor *

Directorio del servidor donde se desplegarán los archivos.

☐ Habilite acciones de despliegue adicionales

Especifique comandos shell a ejecutar cada vez que se produzca un despliegue.


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

```
1 DirectoryIndex indexProyectoTema3.php
```

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

204DWESProyectoTema3


URL 


Rama 

Desplegar clave

Commits más recientes


- 2024-11-07 13:16 Merge developer
- 2024-11-07 13:15 htaccess

 [mostrar más](#)

 Pull ahora

Despliegue

master rama automáticamente en [/httpdocs/204DWESProyectoTema3](http://localhost:8080/204DWESProyectoTema3)

 Desplegar ahora



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

No es seguro

https://204dwesproyectotema3.alexasesan.informatica.ieslossauces.es

Proyecto tema 3

nº	Enunciado
0	Hola mundo y phpinfo()

Subiendo los archivos manualmente

Creamos un subdominio tal y como en el anterior apartado



Añadir un subdominio

Los subdominios son direcciones de Internet para las distintas secciones de su sitio web. Estos utilizan su nombre de dominio. También puede crear un subdominio wildcard introduciendo el símbolo * en vez del nombre. En este caso, los visitantes del sitio :

Nombre del subdominio * .

Introduzca * para crear un subdominio wildcard.

Configuración de hosting

Raíz del documento * /

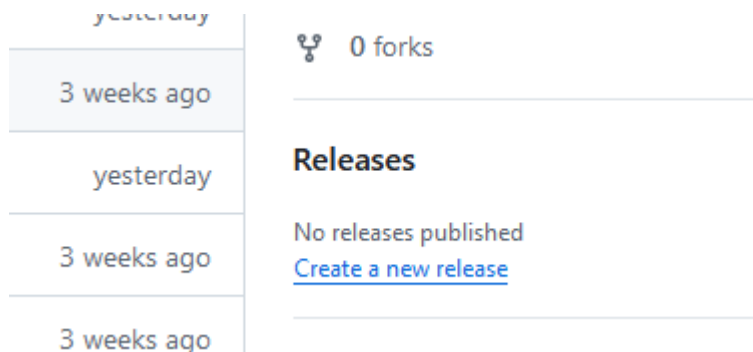
Ruta al directorio principal del sitio web.

* Campos obligatorios

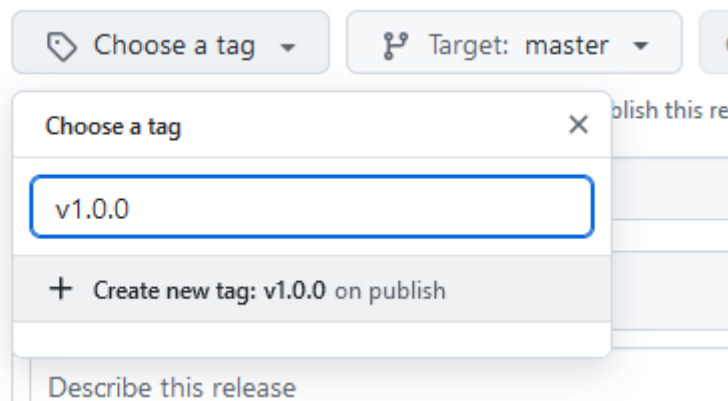
ACEPTAR

Cancelar

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

v1.0.0

Target: master

Previous tag: auto

Generate release notes

Excellent! This tag will be created from the target when you publish this release.

v1.0.0

Write

Preview

H B I

Indice del proyecto DAW

Markdown is supported Paste, drop, or click to add files

↓ Attach binaries by dropping them here or selecting them.

☐ Set as a pre-release
This release will be labeled as non-production ready

Publish release

Save draft

Ahora tenemos la opcion de descargar nuestros archivos como un zip

v1.0.0

Latest

AlexAnacardo released this now v1.0.0 1d4f90d

Indice del proyecto DAW

▼ Assets

2

Source code (zip)

Source code (tar.gz)

Descargaremos dicho zip y lo descomprimos

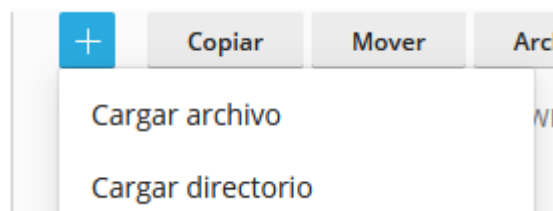
De vuelta a Plesk, entraremos en nuestro subdirectorio y en la pestaña “Primeros pasos” seleccionaremos “cargar archivos”












Aquí borraremos el index generado por defecto y dejaremos los otros dos archivos ahí



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



Seleccionamos los archivos a subir

 desarrollo	07/11/2024 11:54	Carpeta de archivos	
 en	07/11/2024 11:54	Carpeta de archivos	
 error	07/11/2024 11:54	Carpeta de archivos	
 es	07/11/2024 11:54	Carpeta de archivos	
 produccion	07/11/2024 11:54	Carpeta de archivos	
 pruebas	07/11/2024 11:54	Carpeta de archivos	
 pt	07/11/2024 11:54	Carpeta de archivos	
 .gitignore	07/11/2024 11:54	Documento de te...	1 KB
 .htaccess	07/11/2024 11:54	Archivo HTACCESS	1 KB