



24 DE NOVIEMBRE DE 2023

**TEMA 2 – ENTORNO DESARROLLO
Y ENTORNO DE EXPLOTACION**
DESARROLLO WEB EN ENTORNO SERVIDOR

KOKE

INDICE DE CONTENIDO

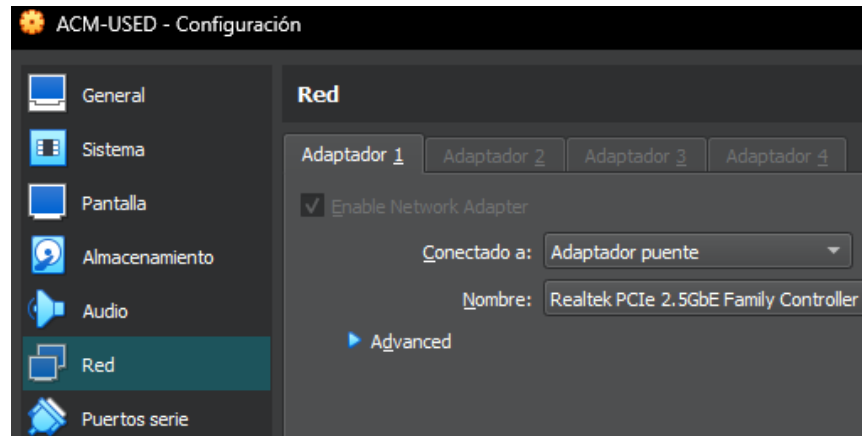
1. USED - UBUNTU SERVER	2
• CONFIGURACION INICIAL	2
• CUENTAS DE ADMINISTRACION	4
• APACHE	5
• PHP	22
• MYSQL	26
• XDEBUJ	32
• CUENTAS DE DESARROLLO Y HOSTING VIRTUAL	33
2. GIT – INTERNET	36
• CUENTAS DE DESARROLLO	36
• OBTENIENDO UN RESPOSITORIO EN GIT	43
• GUARDANDO CAMBIOS EN UN REPOSITORIO	45
3. WXED – WINDOWS X	55
• FILEZILLA	55
• NETBEANS	56
• Instalación y configuración inicial	56
• Creación y modificación de proyectos	57
• Conexión al servidor remoto SFTP	59
• Clonar un repositorio de Github	61
• Uso de Github en NetBeans	65
• Depuracion	66
• VISUAL STUDIO CODE	68

1. USED - UBUNTU SERVER

• CONFIGURACION INICIAL

a. Configuración de red

- Para configurar la red de nuestro Ubuntu server lo primero que hay que hacer es poner la maquina virtual en adaptador puente.



- Lo siguiente que hay que hacer es acceder a la siguiente ruta y editar el archivo 00-installer-config.yaml.

```
miadmin@acm:~$ cd /etc/netplan/
miadmin@acm:/etc/netplan$ ll
total 12
drwxr-xr-x  2 root root 4096 sep 28 12:29 ./
drwxr-xr-x  97 root root 4096 sep 28 12:38 ../
-rw-r--r--  1 root root 117 sep 28 12:29 00-installer-config.yaml
miadmin@acm:/etc/netplan$
```

- Ahora editamos el archivo de la siguiente manera y aplicamos la configuración mediante el siguiente comando

```
# This is the network config written in
network:
  ethernet:
    enp0s3:
      addresses:
        - 192.168.0.206/24
      dhcp4: false
      routes:
        - to: default
          via: 192.168.0.1
      nameservers:
        addresses:
          - 8.8.8.8
      version: 2
```

```
miadmin@acm:/etc/netplan$ sudo netplan apply
miadmin@acm:/etc/netplan$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    link/ether 08:00:27:39:12:37 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.206/24 brd 192.168.0.255 scope glob
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe39:1237/64 scope link
        valid_lft forever preferred_lft forever
miadmin@acm:/etc/netplan$
```

b. Cambiar el nombre del equipo

- Para cambiar el nombre del equipo hay que editar dos archivos (etc/hosts), (/etc/hostname)

```
GNU nano 6.2
127.0.0.1 localhost
127.0.1.1 acm-used
```

```
GNU nano 6.2
acm-used
```

- Para comprobar si se ha cambiado correctamente hay que reiniciar la maquina mediante el comando “reboot” y comprobar el nombre del equipo mediante el comando “hostname”.

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

c. Cambiar la fecha y la hora del sistema

- Para cambiar la hora del sistema hay que realizar el siguiente comando sudo **timedatectl set-timezone Europe/Madrid**.
- Para comprobar la hora del sistema hay que realizar el siguiente comando **timedatectl**.

```
miadmin@acm-used:~$ timedatectl
          Local time: jue 2023-10-26 09:08:05 CEST
          Universal time: jue 2023-10-26 07:08:05 UTC
                RTC time: jue 2023-10-26 07:08:05
                Time zone: Europe/Madrid (CEST, +0200)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
miadmin@acm-used:~$ |
```

d. Actualizar el sistema

- Para actualizar el sistema simplemente tendremos que aplicar el siguiente comando “sudo apt update && sudo apt upgrade”

```
miadmin@acm-used:~$ sudo apt update && sudo apt upgrade
Obj:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
miadmin@acm-used:~$ _
```

• CUENTAS DE ADMINISTRACION

e. Creación del usuario

- Para crear un usuario hay que realizar el comando “adduser”.

```
miadmin@acm-used:~$ sudo adduser miadmin2
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] y
miadmin@acm-used:~$
```

f. Añadirlo al grupo de super usuarios

- Para añadir al usuario miadmin2 al grupo sudo hay que realizar el siguiente comando “sudo usermod -aG sudo miadmin2”.

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

- **APACHE**

g. Instalación del servicio

- Para instalar el servicio de apache hay que realizar el siguiente comando “sudo apt-get install apache2”

```
miadmin@acm-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).
Los paquetes indicados a continuación se instalaron de forma automática. Si no desea algunos de ellos, puede utilizar el control
libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
miadmin@acm-used:~$ _
```

- Para comprobar el estado del servicio usaremos el comando “sudo service apache2 status”

```
miadmin@acm-used:~$ sudo service apache2 status
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-09-28 13:06:29 CEST; 1min 45s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 18102 (apache2)
    Tasks: 55 (limit: 4558)
   Memory: 5.0M
      CPU: 33ms
   CGroup: /system.slice/apache2.service
           └─18102 /usr/sbin/apache2 -k start
             └─18103 /usr/sbin/apache2 -k start
               └─18104 /usr/sbin/apache2 -k start
```

h. Apertura de puertos

- Para que nuestra maquina local se pueda comunicar con el servidor hay que activar el cortafuegos en nuestro servidor y para ello usaremos el siguiente comando (sufo ufw enable).

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

- Después hay que abrir el puerto 80 que es el puerto que usa apache por defecto, para ello podemos usar el siguiente comando (sudo ufw allow 'Apache').

```
miadmin@acm-used:~$ sudo ufw allow 'Apache'
Skipping adding existing rule
Skipping adding existing rule (v6)
miadmin@acm-used:~$ sudo ufw status
Status: active

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

i. Comprobar la conectividad en tu maquina cliente

- Lo único que nos queda es comprobar la conectividad en nuestro cliente para ello nos iremos a un navegador y el url pondremos lo siguiente (<http://192.168.0.206>) y nos tiene que salir lo siguiente.



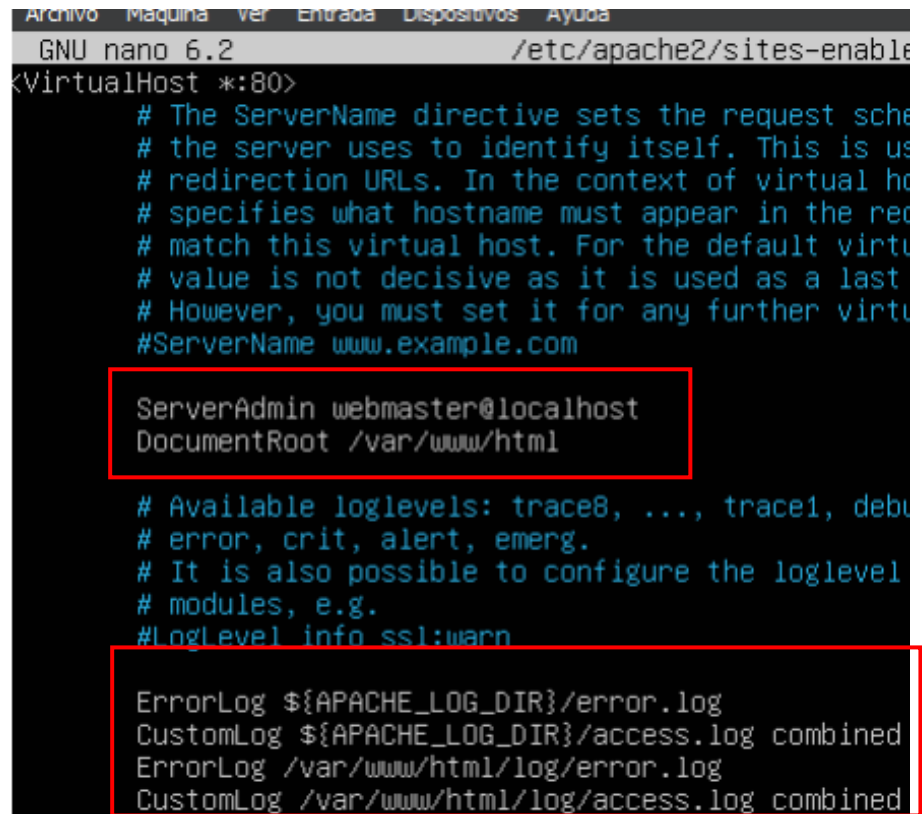
a. Habilitar el uso del fichero .htaccess

- Por defecto viene, AllowOverride None para el directorio /var/www y para ello usamos el siguiente comando
- Y con “Ctrl+W” podemos usar el buscador del archivo y escribimos “/var/www” y luego intro.
- (En este paso también dejamos configurado los indexes.)

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

b. Configuración de los logs

- Configurar que los logs de error y de acceso sean accesibles desde el navegador. Editamos el fichero **sudo nano /etc/apache2/sites-enabled/000-default.conf**.



```
GNU nano 6.2 /etc/apache2/sites-enabled/000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme,
    # the server uses to identify itself. This is used
    # for redirection URLs. In the context of virtual hosts,
    # specifies what hostname must appear in the request
    # to match this virtual host. For the default virtual host,
    # value is not decisive as it is used as a last resort
    # hostname. You must set it for any further virtual
    # hosts.
    #ServerName www.example.com

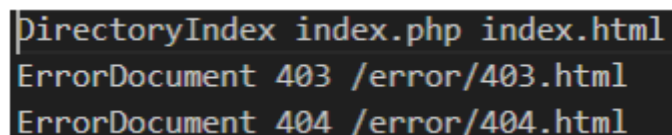
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

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

c. Control de paginas de errores

- Una vez tengamos el archivo .htaccess en funcionamiento, usando la directiva ErrorDocument añadiremos la ruta o mensaje de los errores , en mi caso tengo personalizado el error 403 y 404:



```
DirectoryIndex index.php index.html
ErrorDocument 403 /error/403.html
ErrorDocument 404 /error/404.html
```


j. Configuración del hosting virtual

- Hay que ir al siguiente directorio `cd /etc/apache2/sites-available` y crear el siguiente archivo `sudo nano DAW206.conf` y en el pondremos lo siguiente:
 - **ServerName**: Nombre de dominio de nuestra web.
 - **ServerAdmin**: Usuario administrador de apache
 - **DocumentRoot**: Path del proyecto en el lado del servidor
 - **ErrorLog**: Path donde se van a guardar los logs
 - **CustomLogs**: Path donde se van a guardar los Access.log

```
<VirtualHost *:80>
    ServerName daw214.isauces.local
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/DAW214/public_html
    ErrorLog /var/log/apache2/error-daw214.log
    CustomLog /var/log/apache2/access-daw214.log combined
</VirtualHost>
```

- Ahora comprobamos que el fichero esta correctamente escrito realizando este comando `sudo apache2ctl configtest`.

```
miadmin@DAW-USED:/etc/apache2/sites-available$ sudo apache2ctl configtest
AH00112: Warning: DocumentRoot [/var/www/daw208/public_html] does not exist
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, please add
'DNSLookup' to the list of modules to be sure you have the module loaded, which you should if you are
reconfiguring or adding a new site to a previously unconfigured server
Syntax OK
```

- Habilitamos el fichero mediante el comando “`sudo a2ensite DAW206.conf`”

```
miadmin@DAW-USED:/etc/apache2/sites-available$ sudo a2ensite DAW212.conf
Enabling site DAW212.
To activate the new configuration, you need to run:
    systemctl reload apache2
```

- Y reiniciamos el servicio como nos indica la captura anterior `systemctl reload apache2`

k. Instalación y configuración del bind9

- Para instalar bind9 lo primero que hacer es actualizar el sistema mediante el comando **'sudo apt update'**, una vez actualizado el sistema instalamos el servicio mediante el comando **'sudo apt install bind9'**.

```
miadmin@acm-used:~$ sudo apt install bind9
Leyendo lista de paquetes ... Hecho
Creando árbol de dependencias ... Hecho
Leyendo la información de estado ... Hecho
Se instalarán los siguientes paquetes adicionales:
  bind9-utils dns-root-data
Paquetes sugeridos:
```

- Lo primero que hay que hacer es abrir el puerto 53 del cortafuegos mediante el comando **'sudo ufw allow 53'**

```
miadmin@acm-used:/etc/netplan$ sudo ufw allow 53
Rule added
Rule added (v6)
```

- Lo siguiente que tenemos que hacer será cambiar la configuración de red, para ello nos vamos a la ruta /etc/netplan y editamos el fichero de red e la siguiente manera

```
GNU nano 6.2
# This is the network config written in
network:
  ethernets:
    enp0s3:
      addresses:
        - 192.168.3.206/24
      dhcp4: false
      routes:
        - to: default
          via: 192.168.3.1
      nameservers:
        addresses:
          - 192.168.3.206|
        search: [alvaro.local]
version: 2
```

- Aplicamos los cambios de la configuración de red mediante el comando **'sudo netplan apply'** y comprobamos que funcione haciendo el comando **'ping www.google.es'**

```
miadmin@acm-used:/etc/netplan$ ping www.google.es
PING www.google.es (142.250.201.67) 56(84) bytes of data.
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=1 ttl=114 time=13.8 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=2 ttl=114 time=13.2 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=3 ttl=114 time=13.3 ms
^C
--- www.google.es ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2264ms
rtt min/avg/max/mdev = 13.153/13.407/13.759/0.256 ms
```

a. Declaración de la zona directa

- Zona directa se encarga de registrar los registros necesarios para que resuelve de nombres de dominio de un equipo o servicio y nos devuelva la IP.
- Ahora crearemos la zona `alvaro.local` para ello editamos el fichero `"named.conf.local"` que esta en la ruta `/etc/bind/named.conf.local`.

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

- Comprobamos que los ficheros modificados no contienen ningún error sintactico mediante el comando **'sudo named-checkconf'**.

```
miadmin@acm-used:/etc/bind$ sudo named-checkconf
miadmin@acm-used:/etc/bind$ |
```

- Ahora vamos a crear la zona directa, para ello nos iremos a la ruta (etc/bind/db.alvaro.local) y copiaremos el fichero mediante el comando **'sudo cp /etc/bind/db.local /etc/bind/db.alvaro.local /etc/bind/db.tunombre.local'**

```
miadmin@acm-used:~$ sudo cp /etc/bind/db.local /etc/bind/db.alvaro.local
miadmin@acm-used:~$ |
```

- A continuación editamos el fichero **/etc/bind/db.alvaro.local** de la siguiente manera.

```
GNU nano 6.2 /etc/bind/db.alvaro.local
;
; BIND data file for alvaro.local
;
$TTL      604800
@         IN      SOA      acm-used.alvaro.local. alvaro.cormi.educa.jcyl.es (
                        1          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
@         IN      NS       acm-used.alvaro.local.
; Registros Host
@         IN      A        192.168.3.206
acm-used  IN      A        192.168.3.206
; Registros Alias
daw206    IN      CNAME    acm-used.alvaro.local.
DAW206    IN      CNAME    acm-used.alvaro.local.
|
```

- A continuación chequeamos el fichero que contiene la zona alvaro.local mediante el comando **'sudo named-checkzone'**.

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

- Reiniciamos el servicio mediante el comando **'sudo service bind9 restart'**.

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

- Comprobamos que todo funciona correctamente realizando los siguientes comandos.

```
miadmin@acm-used:/etc/bind$ host daw206.alvaro.local
daw206.alvaro.local is an alias for acm-used.alvaro.local.
acm-used.alvaro.local has address 192.168.3.206
miadmin@acm-used:/etc/bind$ host acm-used.alvaro.local
acm-used.alvaro.local has address 192.168.3.206
miadmin@acm-used:/etc/bind$ host daw206
daw206.alvaro.local is an alias for acm-used.alvaro.local.
acm-used.alvaro.local has address 192.168.3.206
miadmin@acm-used:/etc/bind$ |
```

b. Declaración de la zona inversa

- La resolución inversa, consiste en preguntar al servidor DNS por una IP y nos devuelve el nombre del equipo o del servicio.
- Para declarar la zona editaremos el fichero ‘/etc/bind/named.conf.local’ y añadiremos al final del fichero lo siguiente.

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

- A continuación chequeamos el fichero.

```
miadmin@acm-used:~$ sudo named-checkconf
miadmin@acm-used:~$ |
```

c. Declaración de la RR en zona inversa

- Para crear la zona de resolución inversa, realizamos una copia del fichero de la zona directa. En este caso, es el mismo servidor por lo tanto los registros SOA, NS, son comunes.
- Editamos el fichero `/etc/bind/db.3.168.192.in-addr.arpa`

```
GNU nano 6.2 db.3.168.192.in-addr.arpa
;
; BIND data for 3.168.192.in-addr.arpa
;
$TTL      604800
@         IN      SOA     acm-used.alvaro.local. root.localhost. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;
@         IN      NS      acm-used.alvaro.local.
; Registros Pointer
20x       IN      PTR     acm-used.alvaro.local.
20x       IN      PTR     alvaro.local.
```

- Chequeamos la zona mediante el comando `'sudo named-checkzone 10x.168.192.in-addr.arpa /etc/bind/db.3.168.192.in-addr.arpa'`.

```
miadmin@acm-used:/etc/bind$ sudo named-checkzone 106.168.192.in-addr.arpa /etc/bind/db.3.168.192.in-addr.arpa
zone 106.168.192.in-addr.arpa/IN: loaded serial 1
OK
```

• HTTPS

El protocolo de transferencia de hipertexto seguro (HTTPS) es la versión segura de HTTP, que es el principal protocolo utilizado para enviar datos entre un navegador web y un sitio web. El HTTPS está encriptado para aumentar la seguridad de las transferencias de datos.

a. Generar certificado

Para generar un certificado lo primero que hay que hacer es generar la clave privada mediante el comando `sudo openssl genrsa 2048 > claveprivada.key`

```
miadmin@acm-used:~$ sudo openssl genrsa 2048 > claveprivada.key
[sudo] password for miadmin:
miadmin@acm-used:~$ |
```

Comprobamos que esta la clave en el directorio haciendo listado de archivos de ocultos en **/home/miadmin** mediante el comando **ls -a**.

```
miadmin@acm-used:~$ ls -a
.      206DAWProyectoDAW  .bash_logout  .cache  doc
..     .bash_history         .bashrc       claveprivada.key  .gitconfig
miadmin@acm-used:~$
```

Ahora mediante esa clave podremos generar un certificado CSR para ello realizaremos el siguiente comando **openssl req -new -key claveprivada.key > certificado.csr**.

Al realizar el comando nos pedirá responder distintas preguntas las cuales en mi caso voy a responder lo siguiente.

- **Country Name** → Siglas de tu país
- **State of Province** → Nombre de tu provincia
- **Locality Name** → Nombre de tu localidad
- **Organization Name** → Nombre de tu organización
- **Organizational Unit Name** → Nombre de tu sección
- **Common Name** → Tu dominio
- **Email address** → Tu correo electrónico
- **A Challenge Password** → Posibilidad de poner una contraseña
- **An optional Company Name** → Nombre opcional de comany

```
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) []:daw206.alvaro.local
Email Address []:
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
```

```
A challenge password []:
```

```
An optional company name []:
```

b. Crear el certificado digital autofirmado

Para crear el certificado digital autofirmado hay que usar el siguiente comando
Openssl x509 -req -days 365 -in certificado.csr -signkey claveprivada.key > certificado.crt

```
miadmin@acm-used:~$ sudo openssl x509 -req -days 365 -in certificado.csr -signkey claveprivada.key > certificado.crt
[sudo] password for miadmin:
Certificate request self-signature ok
subject=C = ES, ST = Zamora, L = Benavente, O = IES Los Sauces, OU = Informatica, CN = daw206.alvaro.local
```

Comprobamos que todo esta correcto listando el directorio mediante **ls -a**

```
miadmin@acm-used:~$ ls -a
.          .cache          .lessht
..         certificado.crt .local
206DAWProyectoDAW certificado.csr  .mysql_history
.bash_history claveprivada.key .profile
.bash_logout doc             .ssh
.bashrc     .gitconfig     .sudo_as_admin_successful
```

a. Configurar apache

El método de cifrada SSL / TLS utiliza un método de cifrado de clave publica para la autenticación del servidor.

Lo primero que hay que hacer es activar el modulo ssl mediante el comando **sudo a2enmod ssl**

```
miadmin@acm-used:~$ sudo a2enmod ssl
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 self-signed certificates.
To activate the new configuration, you need to run:
systemctl restart apache2
```


Hay que mover la clave privada (claveprivada.key) al directorio /etc/ssl/private mediante el comando **sudo mv claveprivada.key /etc/ssl/private**

```
miadmin@acm-used:~$ sudo mv claveprivada.key /etc/ssl/private/
miadmin@acm-used:~$ ls -a /etc/ssl/private/
ls: cannot open directory '/etc/ssl/private/': Permission denied
miadmin@acm-used:~$ sudo ls -a /etc/ssl/private/
.  ..  claveprivada.key  ssl-cert-snakeoil.key
miadmin@acm-used:~$ |
```

Cambiar el propietario de la clave privada para que sea **root:ssl-cert** mediante el siguiente comando **sudo chown root:ssl-cert /etc/ssl/private/claveprivada.key**

```
miadmin@acm-used:~$ sudo chown root:ssl-cert /etc/ssl/private/claveprivada.key
miadmin@acm-used:~$ |

miadmin@acm-used:~$ sudo ls -l /etc/ssl/private/
total 8
-rw-rw-r-- 1 root ssl-cert 1704 nov 24 08:48 claveprivada.key
-rw-r----- 1 root ssl-cert 1708 sep 28 08:36 ssl-cert-snakeoil.key
miadmin@acm-used:~$ |
```

Cambiamos los permisos de la clave privada mediante el siguiente comando **sudo chmod 640 /etc/ssl/private/claveprivada.key**

```
miadmin@acm-used:~$ sudo chmod 640 /etc/ssl/private/claveprivada.key
miadmin@acm-used:~$ sudo ls -l /etc/ssl/private/
total 8
-rw-r----- 1 root ssl-cert 1704 nov 24 08:48 claveprivada.key
-rw-r----- 1 root ssl-cert 1708 sep 28 08:36 ssl-cert-snakeoil.key
miadmin@acm-used:~$ |
```

Ahora movemos el certificado a los directorios que utiliza apache mediante el siguiente comando **sudo mv certificado.crt /etc/ssl/certs**

```
miadmin@acm-used:~$ sudo mv certificado.crt /etc/ssl/certs|
```

Ahora le cambiamos el propietario para que sea **root:root**
/etc/ssl/certs/certificado.crt mediante el comando **sudo chown root:root**
/etc/ssl/certs/certificado.crt

```
miadmin@acm-used:~$ sudo ls -l /etc/ssl/certs/certificado.crt
-rw-rw-r-- 1 root root 1281 nov 24 09:07 /etc/ssl/certs/certificado.crt
miadmin@acm-used:~$
```

Ahora configuraremos el sitio virtual y lo primero que hay que hacer es copiar el archivo **/etc/apache2/sites-available/default-ssl.conf** en un archivo nuevo, para ello realizamos el siguiente comando **sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/tunombre-ssl.conf**

```
miadmin@acm-used:~$ sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/alvaro-ssl.conf
```

Editamos el archivo de la siguiente manera.

```
# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on
SSLCertificateFile /etc/ssl/certs/certificado.crt
SSLCertificateKeyFile /etc/ssl/private/claveprivada.key
```

```
<IfModule mod_ssl.c>
    <VirtualHost *:443>
        ServerAdmin webmaster@localhost
        ServerName daw206.alvaro.local
        DocumentRoot /var/www/html
```

Comprobamos que esta todo bien mediante el comando **sudo apache2ctl configtest**.

```
miadmin@acm-used:~$ sudo apache2ctl configtest
Syntax OK
miadmin@acm-used:~$
```

Activamos el sitio virtual mediante el comando **sudo a2ensite alvaro-ssl.conf**

```
miadmin@acm-used:~$ sudo a2ensite alvaro-ssl.conf
Enabling site alvaro-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
miadmin@acm-used:~$ |
```

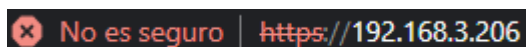
Abrimos el puerto 443 mediante el comando **sudo ufw allow 443**

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

Reiniciamos el servicio de apache mediante **sudo service apache2 restart**

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

Ahora podremos acceder a nuestra pagina mediante https

A warning message displayed in a dark box with a red 'x' icon. The text reads "No es seguro" followed by a vertical bar and the URL "https://192.168.3.206".

Ahora mostraremos una lista de los Virtual Hosts configurados en un servidor web Apache con el siguiente comando **sudo apache2ctl -S**

```
miadmin@acm-used:~$ sudo apache2ctl -S
[sudo] password for miadmin:
VirtualHost configuration:
*:80                                acm-used (/etc/apache2/sites-enabled/000-default.conf:1)
*:443                               daw206.alvaro.local (/etc/apache2/sites-enabled/alvaro-ssl.conf:2)
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"
Mutex watchdog-callback: using_defaults
Mutex ssl-stapling-refresh: using_defaults
Mutex ssl-stapling: using_defaults
Mutex ssl-cache: using_defaults
Mutex default: dir="/var/run/apache2/" mechanism=default
Mutex mpm-accept: using_defaults
PidFile: "/var/run/apache2/apache2.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="www-data" id=33
Group: name="www-data" id=33
```

Como vemos en la captura anterior tenemos el Virtual Host configurado para escuchar en el puerto 443 (el puerto estándar para conexiones seguras a través de HTTPS) con el nombre de host (ServerName) daw206.alvaro.local. Además, indica que la configuración para este Virtual Host se encuentra en el archivo de configuración /etc/apache2/sites-enabled/default-ssl.conf.

Y ahora pasaremos a configurar el del puerto 80, editando la configuración que nos aparece en la ruta del puerto **sudo nano /etc/apache2/sites-enabled/000-default.conf**

Des comentamos la línea del 'ServerName' añadimos el dominio de la MV

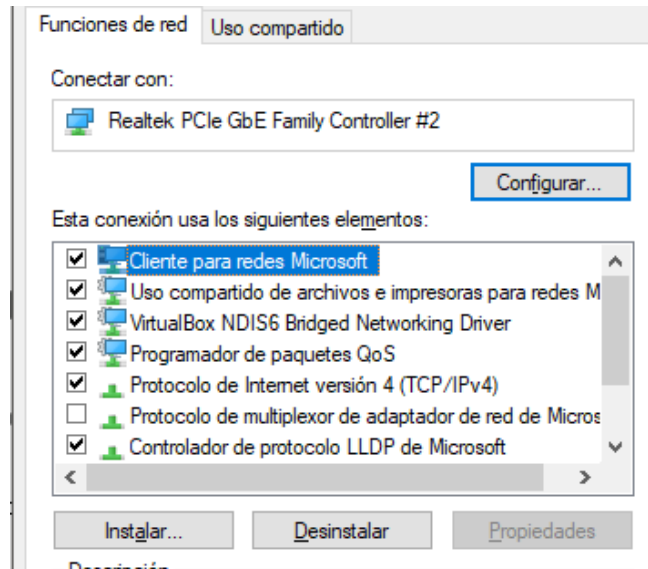
```
ServerName daw206.alvaro.local|
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
Alias /doc "/home/miadmin/doc"
<Directory /home/miadmin/doc>
    Options +FollowSymLinks +Indexes
    AllowOverride None
    Require all granted
</Directory>
```

Ahora verificamos que esta todo correcto con el comando **sudo apache2ctl configtest**

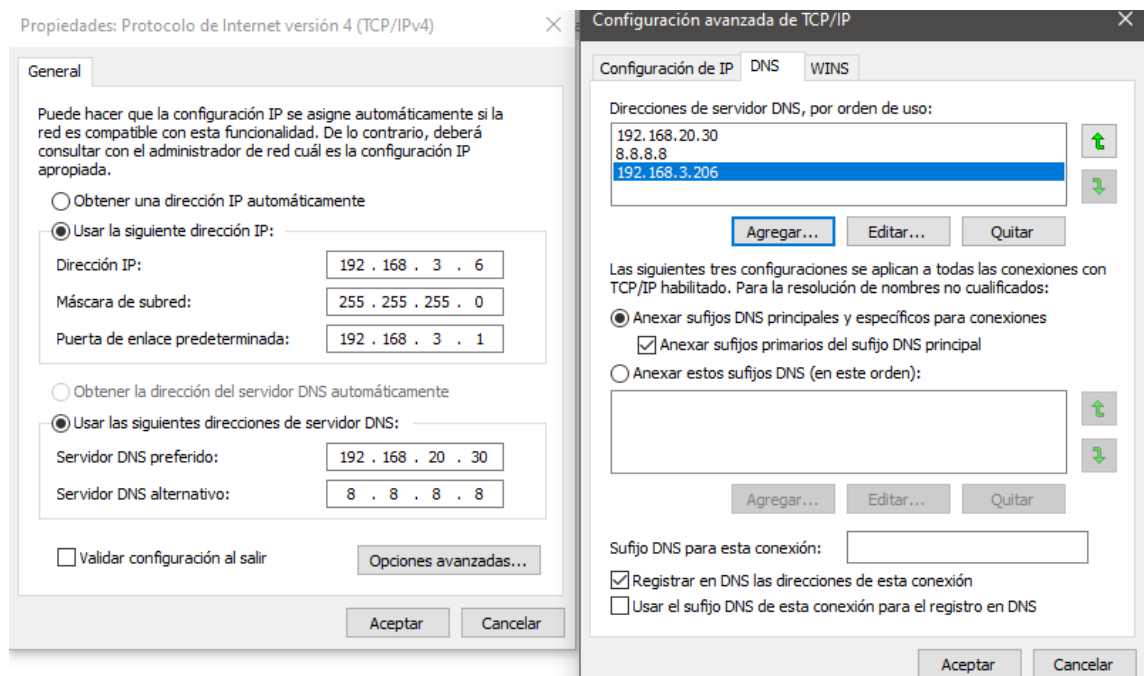
```
miadmin@acm-used:~$ sudo apache2ctl configtest
Syntax OK
```

Después nos surgirá un problema, y es que por nuestra configuración de red en la máquina real, no interpretará el dominio de nuestra MV

Para ello vamos a la siguiente ruta 'Panel de control\Redes e Internet\Conexiones de red' y hacemos click derecho sobre nuestra tarjeta de red, 'Propiedades'



y buscamos esa opción y volvemos a hacer click en 'Propiedades' / 'Opciones Avanzadas...' / 'DNS' y añadimos la IP de nuestra MV y la posicionamos la primera



- **Redireccionar HTTP a HTTPS**

Después de este paso debemos editar el archivo ‘.htaccess’ para redireccionar de ‘HTTP’ a ‘HTTPS’, incluimos lo siguiente.

```
RewriteEngine On
RewriteCond %{SERVER_PORT} 80
RewriteRule ^(.*)$ https://daw206.alvaro.local/$1 [R,L]
```

Ahora activaremos el modulo rewrite mediante el comando **sudo a2enmod rewrite** y reiniciaremos apache

```
miadmin@acm-used:/var/www/html$ sudo a2enmod rewrite
Module rewrite already enabled
miadmin@acm-used:/var/www/html$ sudo service apache2 restart
```

Ahora podemos acceder tanto a nuestra IP como nuestro dominio sin problemas de que no resuelva, ‘HTTP’ y ‘HTTPS’.

General

Detalles

Enviado a

Nombre común (CN)

Organización (O)

Unidad organizativa (OU)

daw206.alvaro.local

IES Los Sauces

Informatica

Emitido por

Nombre común (CN)

Organización (O)

Unidad organizativa (OU)

daw206.alvaro.local

IES Los Sauces

Informatica

Período de validez

Emitido el

Vencimiento el

viernes, 24 de noviembre de 2023, 9:07:22

sábado, 23 de noviembre de 2024, 9:07:22

Huellas digitales SHA-256

Certificado

Clave pública

b0ec2bef9bfbe1efd1392887f41757ea0173abc7351dc7e0472bb7f864ef5ce0

bcd39d035f0648f8a818e5ac8f855ed367e21a49c9fb547c13092f6852f19542

- **PHP**

- **Instalar el módulo del PHP**

- Antes instalar, es recomendable tener actualizado el sistema mediante apt-get update, y apt-get upgrade..
- Después hay que realizar lo siguiente comando (sudo apt-get install php)

```
miadmin@acm-used:~$ sudo apt-get install php
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
libapache2-mod-php8.1 php-common php8.1 php8.1-cgi php8.1-common php8.1-openssl php8.1-readline
Paquetes sugeridos:
php-pear
Se instalarán los siguientes paquetes NUEVOS:
libapache2-mod-php8.1 php-common php8.1 php8.1-cgi php8.1-common php8.1-openssl
php8.1-readline
0 actualizados, 8 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesitan descargar 5.150 kB de archivos.
Se utilizarán 21,3 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? (S/n)
```

- Necesitaremos también algo que conecte nuestro servidores Apache con nuestro motor PHP. Necesitaremos el módulo PHP para Apache. Lo instalaremos de la siguiente manera: (sudo apt-get install libapache2-mod-php).

```
miadmin@acm-used:~$ sudo apt-get install libapache2-mod-php
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes NUEVOS:
libapache2-mod-php
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

- **Configuración php.ini**

- Vamos al directorio y realizamos una copia de seguridad “cd /etc/php/8.1/apache2” y sudo cp php.ini php.ini.backup”

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

- Vamos al directorio y realizamos una copia de seguridad “cd /etc/php/8.1/apache2” y sudo cp php.ini php.ini.backup”.

```

; Development Value: On
; Production Value: Off
; https://php.net/display-errors
display_errors = On

; The display of errors which occur durin
; separately from display_errors. We stro
; for production servers to avoid leaking
; Default Value: On
; Development Value: On
; Production Value: Off
; https://php.net/display-startup-errors
display_startup_errors = On

; https://php.net/memory-limit
memory_limit = 256M

```

- Para comprbar que los cambios se han realizado con éxito, debemos de ir al php.info y verificar si esos campos concuerdan con las modificaciones que hemos realizado en el servidor.

display_errors	On
display_startup_errors	On
memory_limit	256M

a. Instalación de phpMyAdmin

- Para instalar phpMyAdmin hay que instalar los siguientes módulos de php:
 - **php-mbstring**: módulo para administrar cadenas no-ASCII y convertir cadenas a diferentes codificaciones.
 - **php-zip**: esta extensión admite la carga de archivos .zipen phpMyAdmin.
 - **php-gd**: habilita la obtención de ayuda de la biblioteca GD Graphics .
 - **php-json**: proporciona PHP con compatibilidad para serialización JSON.
 - **php-curl**: permite que PHP interactúe con diferentes tipos de servidores utilizando diferentes protocolos.

- Y realizamos la instalación de estos módulos mediante el siguiente comando (sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl).

```
miadmin@acm-used:~$ sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
linux-headers-5.15.0-86 linux-headers-5.15.0-86-generic linux-image-5.15.0-86-generic
```

- Estas son las opciones que debes elegir cuando se le solicita para configurar correctamente su instalación:
 - Para la selección del servidor, elija apache2 Advertencia: Cuando aparece la línea de comandos, “apache2” está resaltado, pero no está seleccionado. Si no pulsa SPACE para seleccionar Apache, el instalador no moverá los archivos necesarios durante la instalación. Pulse ESPACIO, TAB y luego ENTER para seleccionar Apache.

```
Configuración de phpmyadmin
Por favor, elija el servidor web que se debería configurar automáticamente.

Servidor web que desea reconfigurar automáticamente:

[ ] apache2
[ ] lighttpd

<Ok>
```

- Cuando se le pregunte si utiliza dbconfig-common para configurar la base de datos, seleccione Yes.

```
Configuración de phpmyadmin

El paquete phpmyadmin debe tener una base de datos instalada y configurada antes de poder ser utilizado. Puede gestionar esto opcionalmente a través «dbconfig-common».

Debería rechazar esta opción si es Vd. un administrador de bases de datos avanzado y desea realizar esta configuración manualmente, o si la base de datos ya está instalada y configurada. Probablemente podrá encontrar los detalles de las operaciones que debe realizar en «/usr/share/doc/phpmyadmin».

Debería escoger esta opción en cualquier otro caso.

¿Desea configurar la base de datos para phpmyadmin con «dbconfig-common»?

<Yes> <No>
```

- Luego, se le solicitará elegir y confirmar una contraseña para la aplicación de MySQL para phpMyAdmin.

- En el momento en que se redactó este documento, habilitar este componente generará un error cuando intente establecer una contraseña para el usuario phpmyadmin :

- habilitó la autenticación de contraseña para el usuario root de MySQL, ejecute este comando y luego ingrese su contraseña cuando se le solicite:

```
miadmin@acm-used:~$ sudo mysql -u adminsql -p
Enter password:
```

- Desde la línea de comandos, ejecute el siguiente comando para deshabilitar el componente Validar contraseña. Tenga en cuenta que con esto en realidad no se desinstalará, sino solo se evitará que el componente se cargue en su servidor MySQL:

```
mysql> UNINSTALL COMPONENT "file://component_validate_password";
Query OK, 0 rows affected (0,03 sec)
```

- Luego, vuelva a instalar el paquete phpmyadmin, que funcionará según lo previsto:
- Una vez que phpMyAdmin esté instalado, puede abrir la línea de comandos de MySQL una vez más con `sudo mysql` y luego ejecutar el siguiente comando para volver a habilitar el componente “Validar contraseña”:

```
mysql> INSTALL COMPONENT "file://component_validate_password";  
Query OK, 0 rows affected (0,06 sec)
```

- El proceso de instalación añade el archivo de configuración de phpMyAdmin de Apache al directorio `/etc/apache2/conf-enabled/`, donde se lee de forma automática.
- Para terminar de configurar Apache y PHP y fin de que funcionen con phpMyAdmin, la única tarea que queda a continuación en esta sección del tutorial es habilitar explícitamente la extensión PHP `mbstring`. Esto se puede hacer escribiendo lo siguiente:

```
miadmin@acm-used:~$ sudo phpenmod mbstring
```

- Y ahora reiniciamos el servicio de apache mediante el siguiente comando (`sudo systemctl restart apache2`)

```
miadmin@acm-used:~$ sudo systemctl restart apache2  
miadmin@acm-used:~$
```

- **MYSQL**

- a. **Instalar el servicio**

- Para instalar el servicio de MySQL, lo primero es actualizar el sistema mediante `update` y `upgrade`. Después continuaremos con la instalación mediante el siguiente comando.

```
miadmin@acm-used:~$ sudo apt install mysql-server  
Leyendo lista de paquetes ... Hecho  
Creando árbol de dependencias ... Hecho  
Leyendo la información de estado ... Hecho  
Se instalarán los siguientes paquetes adicionales:
```

- Para comprobar que versión de mysql hemos instalado en nuestro sistema, deberemos realizar el siguiente comando.

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

b. Configuración del servicio

- Comprobar los servicios que están escuchando los puertos mediante el comando ss -punta y comprobar que esta el puerto 3306.

```
miadmin@acm-used:~$ ss -punta
```

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
udp	UNCONN	0	0	127.0.0.53%lo:53	0.0.0.0:*
tcp	LISTEN	0	70	127.0.0.1:33060	0.0.0.0:*
tcp	LISTEN	0	151	127.0.0.1:3306	0.0.0.0:*
tcp	LISTEN	0	4096	127.0.0.53%lo:53	0.0.0.0:*
tcp	LISTEN	0	128	0.0.0.0:22	0.0.0.0:*
tcp	ESTAB	0	0	192.168.3.206:22	192.168.3.6:65308
tcp	LISTEN	0	511	*:80	*:*
tcp	LISTEN	0	128	:::22	:::*

```
miadmin@acm-used:~$ |
```

- Después hay que abrir el puerto de Mysql de la siguiente manera mediante ufw.

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

```
miadmin@acm-used:~$ |
```

- Para ello deberemos ir al siguiente fichero de configuración /etc/mysql/mysql.conf/mysqlld.cnf. Una vez dentro del archivo debemos comentar las líneas **bind-address = 127.0.0.1** y **mysqlx-bind-address=127.0.0.1**.

```
#bind-address            = 127.0.0.1
#mysqlx-bind-address     = 127.0.0.1
..
```

- Ahora hay que comprobar el estado del servicio mediante el comando `sudo service mysql status`.

```
miadmin@acm-used:~$ sudo service mysql status
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor
   Active: active (running) since Tue 2023-10-24 08:38:27 CEST; 11min
   Process: 34504 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre
   Main PID: 34512 (mysqld)
   Status: "Server is operational"
   Tasks: 37 (limit: 2221)
   Memory: 362.6M
   CPU: 2.811s
   CGroup: /system.slice/mysql.service
           └─34512 /usr/sbin/mysqld

oct 24 08:38:26 acm-used systemd[1]: Starting MySQL Community Server ...
oct 24 08:38:27 acm-used systemd[1]: Started MySQL Community Server.
miadmin@acm-used:~$
```

- Una vez realizada la instalación se recomienda instalar un script de seguridad para mysql mediante el siguiente comando.

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

- Esto le preguntará si desea configurar el archivo VALIDATE PASSWORD PLUGIN. Responde Y si, o cualquier otra cosa para continuar sin habilitar.

```
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
```

- Para el resto de las preguntas, presione Y y presione la ENTER tecla en cada mensaje. Esto eliminará algunos usuarios anónimos y la base de datos de prueba, deshabilitará los inicios de sesión raíz

remotos y cargará estas nuevas reglas para que MySQL respete inmediatamente los cambios que ha realizado.

```
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) : 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@acm-used:~$ |
```

c. Instalamos los módulos necesarios

Tiene instalado Apache para servir su contenido y MySQL instalado para almacenar y administrar sus datos. PHP es el componente de nuestra configuración que procesará el código para mostrar contenido dinámico al usuario final.

Además del phppaquete, necesitará php-mysqlun módulo PHP que permita a PHP comunicarse con bases de datos basadas en MySQL. También necesitarás libapache2-mod-phphabilitar Apache para manejar archivos PHP. Los paquetes principales de PHP se instalarán automáticamente como dependencias.

- Para instalar estos paquetes, ejecute el siguiente comando:

```
miadmin@acm-used:~$ sudo apt install libapache2-mod-php8.1 php8.1-mysql
Leyendo lista de paquetes ... Hecho
Creando árbol de dependencias ... Hecho
Leyendo la información de estado ... Hecho
```

- Una vez finalizada la instalación, ejecute el siguiente comando para confirmar su versión de PHP:

```
miadmin@acm-used:~$ 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
    with Xdebug v3.1.2, Copyright (c) 2002-2021, by Derick Rethans
miadmin@acm-used:~$ |
```

d. Crear un usuario virtual

- Para ello hay que entrar en la consola de Mysql mediante el comando sudo mysql. Una vez dentro tendremos que crear el usuario mediante el siguiente comando.

```
mysql> CREATE USER 'admindb'@'%' IDENTIFIED BY 'P@ssw0rd';
Query OK, 0 rows affected (0,02 sec)
```

- Una vez creado le asignaremos permisos mediante el siguiente comando.

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'admindb'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0,01 sec)
```

- Comprobamos la conexión del nuevo usuario mediante el siguiente comando.

```
miadmin@acm-used:~$ mysql -u admindb -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.34-0ubuntu0.22.04.1 (Ubuntu)
```

- Una vez hemos hecho la conexión correcta con el nuevo usuario, veremos los métodos de autenticación con el siguiente comando

```
mysql> SELECT user,plugin,host FROM mysql.user;
+-----+-----+-----+
| user          | plugin                | host          |
+-----+-----+-----+
| admindb       | caching_sha2_password | %             |
```

e. Cambiar la contraseña del usuario

- Primero hay que observar las directivas de contraseñas mediante el comando `SHOW VARIABLE LIKE 'validate_password%';` y cambiar el `.length` para que la longitud máxima sea 4 y el `.policy` que sea en `LOW`

```
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 (0,02 sec)

- Para cambiarlo hay que realizar el siguiente comando `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 (0,01 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)

- Ahora crearemos un usuario que se llame adminsql con la contraseña 'paso' mediante el siguiente comando CREATE USER 'adminsql'@'%' IDENTIFIED BY 'paso';

```
mysql> CREATE USER 'adminsql'@'%' IDENTIFIED BY 'paso';
Query OK, 0 rows affected (0,06 sec)
```

- Ahora le daremos privilegios de administrador

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'adminsql'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0,01 sec)
```

• XDEBUJ

b. Instalación del servicio

- Para la instalación de Xdebug en Ubuntu Server hay que realizar el siguiente comando desde la consola.

```
miadmin@acm-used:~$ sudo apt-get install php-xdebug
[sudo] password for miadmin:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma
libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
php8.1-xdebug
Se instalarán los siguientes paquetes NUEVOS:
```

- Entramos en este directorio (/etc/php8.1/apache/conf.d) y comprobamos si existe el archivo "20-xdebug.ini".

```
miadmin@CGC-USED:/etc/php/8.1/apache2/conf.d$ ls
10-opcache.ini  20-exif.ini      20-phar.ini      20-sockets.ini  20-xmlreader.ini
10-pdo.ini      20-ffi.ini       20-posix.ini     20-sysvmsg.ini  20-xmlwriter.ini
15-xml.ini      20-fileinfo.ini  20-readline.ini  20-sysvsem.ini  20-xsl.ini
20-calendar.ini 20-ftp.ini       20-shmop.ini     20-sysvshm.ini
20-ctype.ini    20-gettext.ini   20-simplexml.ini 20-tokenizer.ini
20-dom.ini      20-iconv.ini     20-soap.ini      20-xdebug.ini
```

c. Configurar el archivo xdebug.ini

- Hay que ir a la siguiente ruta /etc/php/8.1/apache2/conf.d/20-xdebug.ini y editar el archivo de la siguiente manera.

```
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
```

• CUENTAS DE DESARROLLO Y HOSTING VIRTUAL

a. Creación del usuario

- Para crear el usuario en este caso “operadorweb” lo crearemos de la siguiente manera (sudo adduser --no-create-home --home /var/www/html/ --ingroup www-data operadorweb)

```
miadmin@acm-used:~$ sudo adduser --no-create-home --home /var/www/html/ --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@acm-used:~$
```

Ahora comprobaremos los usuarios que hay en el sistema haciendo un “cat /etc/passwd” y tienen que estar miadmin2 y operadorweb

```
miadmin2:x:1001:1001:,,,:/home/miadmin2:/bin/bash
operadorweb:x:1002:33:,,,:/var/www/html:/bin/bash
```

b. Modificar los permisos del mismo

- Lo primero que hay que hacer es cambiar el propietario del fichero con este comando. “sudo chown -R operadorweb:www-data /var/www/html/”

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

- Y después le cambiaremos los permisos al fichero para poder editarlo mediante el siguiente comando “Sudo chmod -R 2775 /var/www/html/” El numero 2 es un permiso especial que permite que todos los ficheros que se incluyan en ese grupo tengas los permisos de ese mismo.

```
miadmin@acm-used:~$ sudo chmod -R 2775 /var/www/html/
miadmin@acm-used:~$ ll /var/www/html/
total 20
drwxrwsr-x 2 operadorweb www-data 4096 sep 28 13:06 ./
drwxr-xr-x 3 root         root    4096 sep 28 13:06 ../
-rwxrwsr-x 1 operadorweb www-data 10671 sep 28 13:06 index.html*
miadmin@acm-used:~$ _
```

c. Enjaular al usuario

- Primero vamos a crear un grupo que pertenece a los usuarios que vamos a enjaular (sudo groupadd ftpuser).

```
miadmin@acm-used:~$ sudo groupadd ftpuser
miadmin@acm-used:~$ cat /etc/group | grep ftp
ftpuser:x:1002:
miadmin@acm-used:~$ |
```

- El siguiente paso será crear el usuario utilizando el siguiente comando (sudo useradd -g www-data -G ftpusers -m -d /var/www/DAW212 DAW212)

```
miadmin@DAW-USED:/$ sudo useradd -g www-data -G ftpusers -m -d /var/www/DAW212 DAW212
```

```
miadmin@DAW-USED:/$ passwd DAW212
passwd: You may not view or modify pass
miadmin@DAW-USED:/$ sudo passwd DAW212
New password:
Retype new password:
passwd: password updated successfully
```

- Después vamos a decir que el /home del usuario operador web pertenezca a root (sudo chown root:root /var/www).

```
miadmin@acm-used:~$ sudo chown root:root /var/www/
miadmin@acm-used:~$ ll /var/www/
total 12
drwxr-xr-x  3 root      root      4096 sep 28 13:06 ./
drwxr-xr-x 15 root      root      4096 sep 28 13:06 ../
drwxrwsr-x 13 operadorweb www-data 4096 oct 16 13:46 html/
miadmin@acm-used:~$ |
```

- Eliminamos el permiso de escritura (sudo chmod 555 /var/www)

```
miadmin@acm-used:~$ sudo chmod 555 /var/www/
miadmin@acm-used:~$ ll /var/ | grep www
dr-xr-xr-x  3 root root      4096 sep 28 13:06 www/
miadmin@acm-used:~$ |
```

- Editamos el fichero sshd_config (sudo nano /etc/ssh/sshd_config)

```
# Subsystem sftp /usr/lib/openssh/sftp-server
Subsystem sftp internal-sftp
```

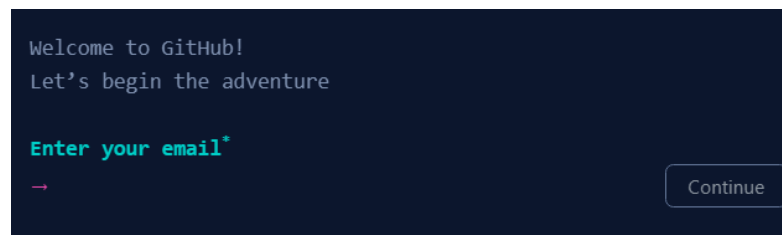
```
Match Group sftputers
ChrootDirectory %h
ForceCommand internal-sftp -u 2
AllowTcpForwarding yes
PermitTunnel no
X11Forwarding no
```

2. GIT – INTERNET

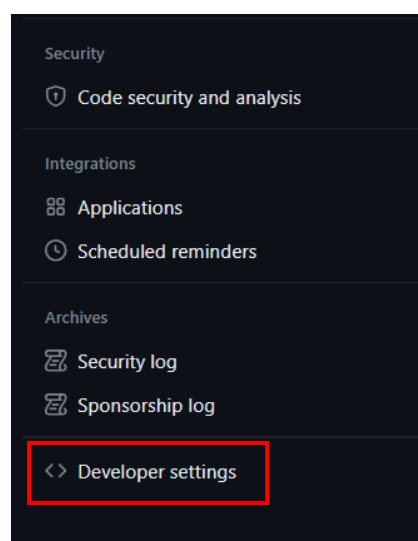
- **CUENTAS DE DESARROLLO**

- **Creación de una cuenta en github**

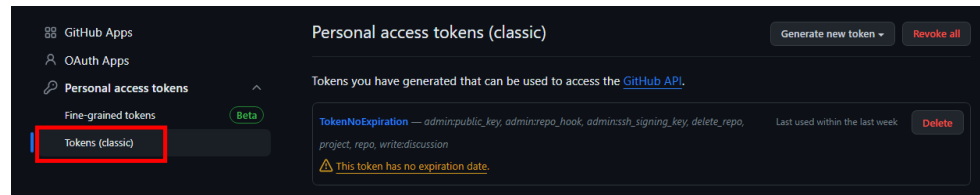
- Lo primero que hay que hacer es crearte una cuenta en la aplicación de github, para ello nos vamos al siguiente enlace <https://github.com/signup?source=login> y te saldrá lo siguiente, solo tienes



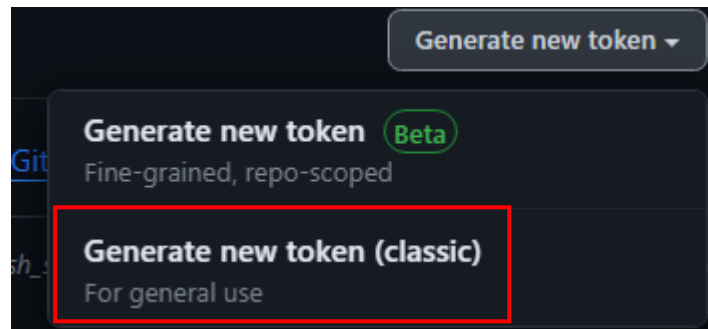
- Ya estando la cuenta creada lo siguiente que tenemos que hacer es crear un token , El Personal access token está diseñado para acceder a los recursos de GitHub en tu nombre.
 - Para crear un nuevo token hay que ir a los ajustes del github. Y darle a developers settings



- Una vez dentro de hay nos saldrán 3 opciones , le tendremos que dar a Personal Access tokens y a token Classic.

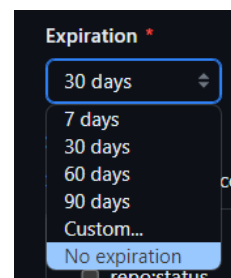


- Para crear un nuevo token le damos a generate new token classic

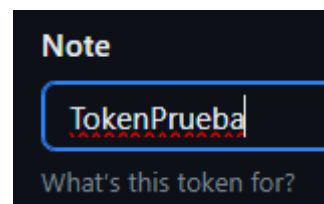


- Una vez le damos nos saldrá distinta información, a nosotros lo que nos interesa es:

- **Expiration** → El tiempo durante el cual el token va estar activo, en mi caso lo voy a poner en no expiration para que nunca expire.



- **Note** → La nota que le vamos a nuestro token , normalmente se suele poner el motivo para lo que va a usar , en mi caso le voy a poner TokenDePrueba .



- Ahora nos toca seleccionar los scopes de nuestro tokens, los scopes sirven para definir el acceso que tiene nuestro token sobre nuestros proyectos, en mi caso lo tengo de la siguiente manera.

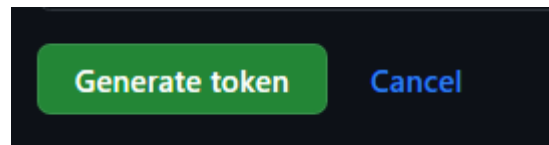
Seleccionar alcances

Los ámbitos definen el acceso a tokens personales. [Lea más sobre los alcances de OAuth.](#)

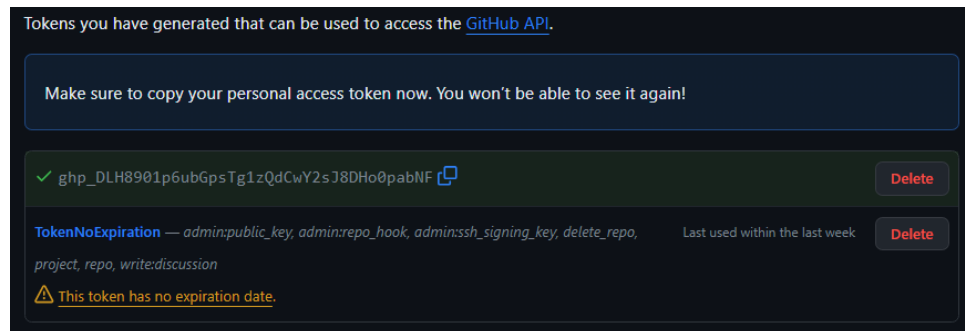
<input checked="" type="checkbox"/> repositorio	Control total de repositorios privados
<input checked="" type="checkbox"/> repositorio: estado	Acceder al estado de confirmación
<input checked="" type="checkbox"/> implementación_repositorio	Acceder al estado de implementación
<input checked="" type="checkbox"/> repositorio_publico	Acceder a repositorios públicos
<input checked="" type="checkbox"/> repositorio: invitar	Acceder a las invitaciones del repositorio
<input checked="" type="checkbox"/> eventos_de_seguridad	Leer y escribir eventos de seguridad.
<input type="checkbox"/> flujo de trabajo	Actualizar los flujos de trabajo de acciones de GitHub
<input checked="" type="checkbox"/> escribir: paquetes	Cargar paquetes en el Registro de paquetes de GitHub
<input checked="" type="checkbox"/> leer: paquetes	Descargar paquetes del Registro de paquetes de GitHub
<input checked="" type="checkbox"/> eliminar: paquetes	Eliminar paquetes del Registro de paquetes de GitHub
<input type="checkbox"/> administrador:org	Control total de organizaciones y equipos, lectura y escritura de proyectos de organización.
<input type="checkbox"/> escribir:org	Leer y escribir membresía de organizaciones y equipos, leer y escribir proyectos de organizaciones
<input type="checkbox"/> leer:org	Leer la membresía de la organización y del equipo, leer los proyectos de la organización
<input type="checkbox"/> gestionar_runners:org	Administrador ejecutores de organizaciones y grupos de ejecutores
<input checked="" type="checkbox"/> administrador: clave_pública	Control total de las claves públicas del usuario.
<input checked="" type="checkbox"/> escribir: clave_pública	Escribir claves públicas de usuario
<input checked="" type="checkbox"/> leer: clave_pública	Leer claves públicas de usuario
<input checked="" type="checkbox"/> administrador:repo_hook	Control total de los enlaces del repositorio
<input checked="" type="checkbox"/> escribir:repo_hook	Escribir ganchos de repositorio
<input checked="" type="checkbox"/> leer:repo_hook	Leer ganchos del repositorio
<input type="checkbox"/> administrador:org_hook	Control total de los ganchos de la organización.
<input type="checkbox"/> esencia	Crear esencias

<input type="checkbox"/> esencia	Crear esencias
<input type="checkbox"/> notificaciones	Acceder a notificaciones
<input type="checkbox"/> usuario	Actualizar TODOS los datos del usuario
<input type="checkbox"/> leer: usuario	Leer TODOS los datos del perfil de usuario
<input type="checkbox"/> usuario: correo electrónico	Acceder a las direcciones de correo electrónico de los usuarios (solo lectura)
<input type="checkbox"/> usuario: seguir	Seguir y dejar de seguir a los usuarios
<input checked="" type="checkbox"/> eliminar_repo	eliminar repositorios
<input checked="" type="checkbox"/> escribir: discusión	Leer y escribir discusiones en equipo.
<input type="checkbox"/> leer: discusión	Leer discusiones del equipo
<input type="checkbox"/> administrador:empresa	Control total de las empresas.
<input type="checkbox"/> gestionar_runners: empresa	Gestionar ejecutores empresariales y grupos de ejecutores
<input type="checkbox"/> gestionar_facturación: empresa	Leer y escribir datos de facturación empresarial
<input type="checkbox"/> leer:empresa	Leer datos del perfil empresarial
<input type="checkbox"/> registro de auditoria	Control total del registro de auditoría
<input type="checkbox"/> leer:audit_log	Acceso de lectura al registro de auditoría
<input type="checkbox"/> espacio de código	Control total de los espacios de código.
<input type="checkbox"/> espacio de código: secretos	Capacidad para crear, leer, actualizar y eliminar secretos de codespace
<input type="checkbox"/> copiloto	Control total de la configuración de GitHub Copilot y las asignaciones de asientos
<input type="checkbox"/> gestionar_facturación: copiloto	Ver y editar asignaciones de asientos de Copilot for Business
<input checked="" type="checkbox"/> proyecto	Control total de los proyectos.
<input type="checkbox"/> leer: proyecto	Acceso de lectura de proyectos
<input type="checkbox"/> administrador:gpg_key	Control total de las claves GPG de los usuarios públicos.
<input type="checkbox"/> escribir:gpg_key	Escribir claves GPG de usuario público
<input type="checkbox"/> leer:gpg_key	Leer claves GPG de usuarios públicos
<input checked="" type="checkbox"/> administrador:ssh_signing_key	Control total de las claves de firma SSH de los usuarios públicos
<input type="checkbox"/> escribir:ssh_signing_key	Escribir claves de firma SSH de usuario público
<input type="checkbox"/> leer:ssh_signing_key	Leer claves de firma SSH de usuarios públicos

- Una vez realizados los anteriores pasos le podremos dar a generar token para que nos genere el token.

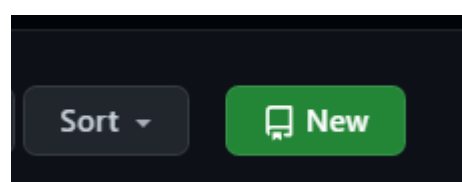
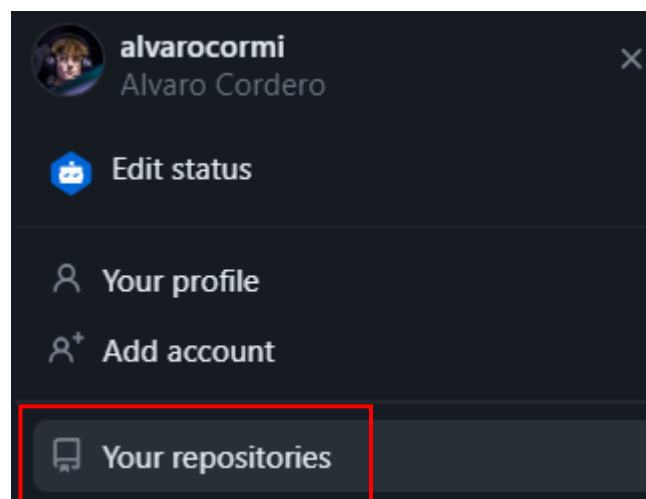


- Esto te dará un numero de identificación hash el cual tienes que guardar para usarlos en tu proyecto .

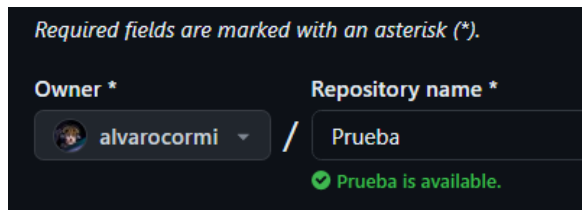


- **Creación de un proyecto**

- Para crear un proyecto en github hay que irse al apartado de tus repositorios. Y una vez allí le damos a New.




- Al darle a new , nos saldrá distinta información que tendremos que tener en cuenta a la hora de crear nuestro proyecto.
 - **Owner** → Eliges la persona que va a ser la dueña del proyecto, en este caso voy a ser yo.
 - **Repository Name** → Este es el nombre que le vas a dar tu repositorio.



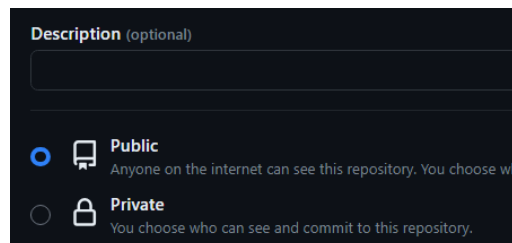
Required fields are marked with an asterisk (*).

Owner * Repository name *


 alvarocormi / Prueba


✓ Prueba is available.

- **Description** → Pones una breve description sobre lo que va a ir tu proyecto o que lenguajes has utilizado para ello.

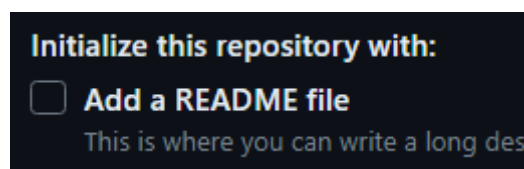


Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can push to it.

☐  **Private**
You choose who can see and commit to this repository.

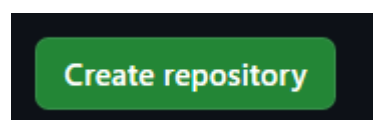
- **README.md** → Este es un archivo que nos servirá para mostrar una breve descripción de nuestro proyecto y mediante esta opción podremos elegir si queremos que nos lo cree directamente o no.



Initialize this repository with:

☐ **Add a README file**
This is where you can write a long description of your project.

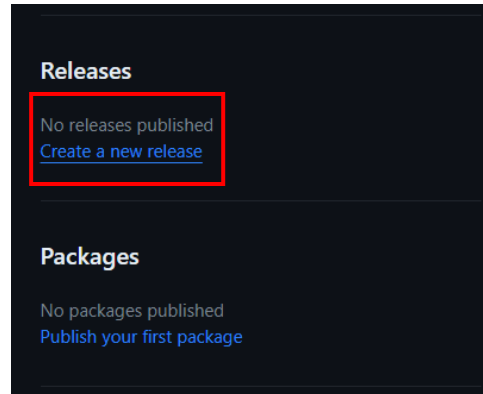
- Una vez comprobado eso, le podremos dar a create repository y se nos creará un **repositorio en remoto** el cual podremos clonar a local.



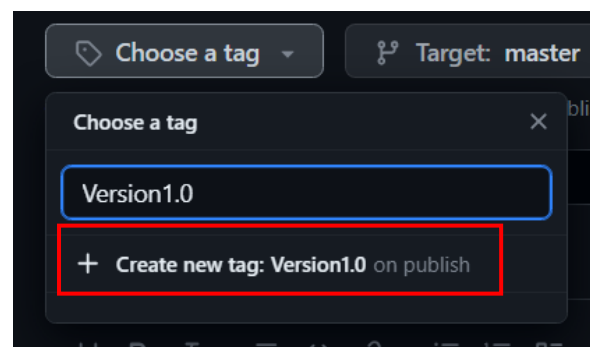
Create repository

- **Creación de un reléase**

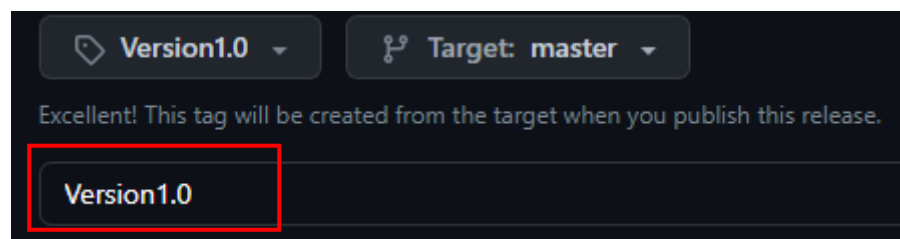
- Lo primero que hay que hacer es dirigirnos al proyecto al proyecto al que queramos hacer la reléase , una vez allí veremos el siguiente apartado a la derecha y le damos a créate new release.



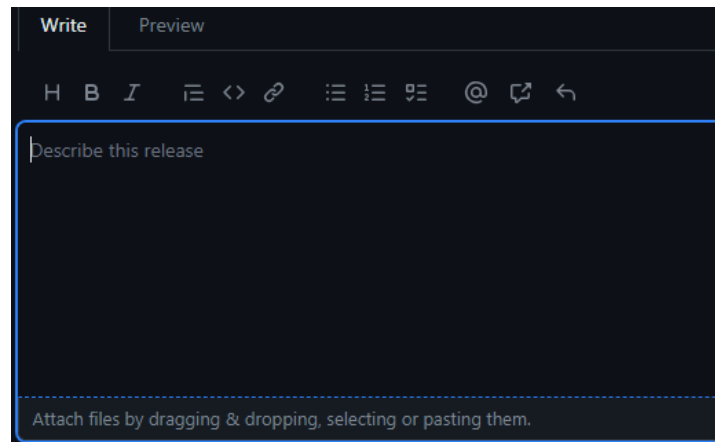
- Una vez le damos ahí, lo primero que tenemos que hacer es crear una tag, para ello nos vamos al apartado que pone Choose a tag, le damos un nombre a la tag y le damos a créate new tag.



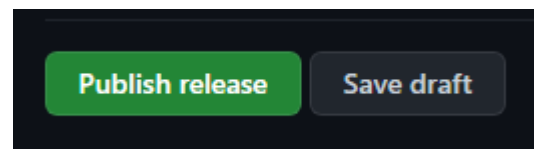
- Ahora le pondremos un nombre a la reléase, para realizar esto tendremos que escribir el nombre en el apartado que pone 'Release title', en este caso le voy a llamar Version1.0



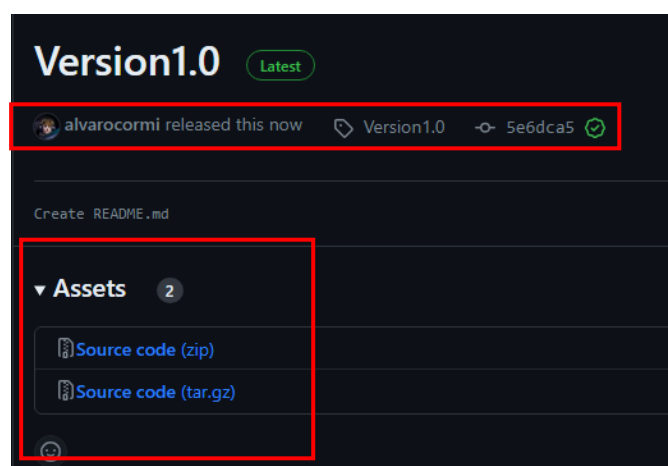
- En el apartado de write podremos poner una breve descripción sobre nuestro proyecto en formato Markdown , por ejemplo podríamos poner las funcionalidades del mismo y los lenguajes que hemos usado.



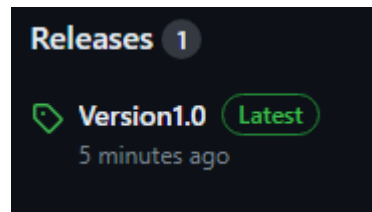
- Si hemos comprobado y realizado los apartados anteriores podremos publicar la reléase en nuestro dando al botón Publis Release.



- Una vez dado al botón nos saldrá una pantalla que nos muestra, quien ha creado la reléase, a versión de la misma y el commit. Y mas abajo nos ha creado unos archivos comprimidos los cuales contienen el código de nuestro proyecto y podremos descargar



- Si ahora entramos en nuestro proyecto, nos podremos fijar que a la derecha en el apartado reléase nos aparece una tag con el nombre que le hayamos dado a la misma.



- **OBTENIENDO UN RESPOSITORIO EN GIT**

Puedes obtener un proyecto Git de dos maneras. La primera es tomar un proyecto o directorio existente e importarlo en Git. La segunda es clonar un repositorio existente en Git desde otro servidor.

- **Inicializar un repositorio en un directorio existente**

Si estás empezando a seguir un proyecto existente en Git, debes ir al directorio del proyecto y usar el siguiente comando. `git init`

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit
$ git init
Initialized empty Git repository in C:/Users/daw2/
```

Esto crea un subdirectorio nuevo llamado `.git`, el cual contiene todos los archivos necesarios del repositorio – un esqueleto de un repositorio de Git. Todavía no hay nada en tu proyecto que esté bajo seguimiento.

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ ls -a
./  ../  .git/

daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ ls -a ./ .git/
./  ../  HEAD  config  description  hooks/  info/  objects/  refs/
```

Si deseas empezar a controlar versiones de archivos existentes (a diferencia de un directorio vacío), probablemente deberías comenzar el seguimiento de esos archivos y hacer una confirmación inicial. Puedes conseguirlo con unos pocos **comandos git add** para especificar qué archivos quieres controlar, seguidos de un **git commit** para confirmar los cambios:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git add .

daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git commit -m "Primer commit de ejemplo"
[master (root-commit) c74d57b] Primer commit de ejemplo
4 files changed, 31 insertions(+)
create mode 100644 cosasQueHacer.txt
create mode 100644 funcionesPHP.txt
create mode 100644 mysql-connector-java-8.0.30.jar
create mode 100644 tokenGH.txt
```

- **Clonando un repositorio existente**

Si deseas obtener una copia de un repositorio Git existente — por ejemplo, un proyecto en el que te gustaría contribuir — el comando que necesitas es **git clone**, cada versión de cada archivo de la historia del proyecto es descargada por defecto cuando ejecutas git clone.

Puedes clonar un repositorio con **git clone [url]**. Por ejemplo, si quieres clonar la librería de Git llamada libgit2 puedes hacer algo así:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploClone
$ git clone https://github.com/libgit2/libgit2 .
Cloning into '.' ...
remote: Enumerating objects: 124993, done.
remote: Counting objects: 100% (124993/124993), done.
remote: Compressing objects: 100% (34451/34451), done
remote: Total 124993 (delta 89139), reused 124200 (de
Receiving objects: 100% (124993/124993), 64.82 MiB |
Resolving deltas: 100% (89139/89139), done.
Updating files: 100% (11810/11810), done.
```

Esto crea un directorio llamado libgit2, inicializa un directorio .git en su interior, descarga toda la información de ese repositorio y saca una copia de trabajo de la última versión.

```
$ ls
AUTHORS      README.md   ci/         docs/       git.git-authors  script/
CMakeLists.txt SECURITY.md  cmake/      examples/   include/         src/
COPYING      api.docurium  deps/      fuzzers/    package.json     tests/
```

Git te permite usar distintos protocolos de transferencia. El ejemplo anterior usa el protocolo `https://`, pero también puedes utilizar `git://` o `usuario@servidor:ruta/del/repositorio.git` que utiliza el protocolo de transferencia SSH

- **GUARDANDO CAMBIOS EN UN REPOSITORIO**

Cada archivo de tu repositorio puede tener dos estados: rastreados y sin rastrear. Los archivos rastreados (**tracked files en inglés**) son todos aquellos archivos que estaban en la última instantánea del proyecto.

Cualquier otro archivo en tu directorio de trabajo que no estaba en tu última instantánea y que no está en el área de preparación (**staging area**). Cuando clonas por primera vez un repositorio, todos tus archivos estarán rastreados y sin modificar pues acabas de sacarlos y aun no han sido editados.

- **Revisando el estado de tus archivos**

La herramienta principal para determinar qué archivos están en qué estado es el comando **git status**. Si ejecutas este comando inmediatamente después de clonar un repositorio, deberías ver algo como esto:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploClone (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Esto significa que tienes un directorio de trabajo limpio - en otras palabras, que no hay archivos rastreados y modificados. Además, Git no encuentra archivos sin rastrear, de lo contrario aparecerían listados aquí.

Supongamos que añades un nuevo archivo a tu proyecto, un simple `hola.txt`. Si el archivo no existía antes y ejecutas **git status**, verás el archivo sin rastrear de la siguiente manera:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploClone (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file> ..." to include in what will be committed)
  hola.txt

nothing added to commit but untracked files present (use "git add" to track)
```

- **Rastrear archivos nuevos**

Para comenzar a rastrear un archivo debes usar el comando **git add**. Para comenzar a rastrear el archivo hola.txt puedes ejecutar lo siguiente:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploClone (main)
$ git add hola.txt
warning: in the working copy of 'hola.txt', LF will l
time Git touches it
```

Ahora si vuelves a ver el estado del proyecto, verás que el archivo README está siendo rastreado y está preparado para ser confirmado:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploClone (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file> ..." to unstage)
    new file:   hola.txt
```

- **Ignorar archivos**

A veces, tendrás algún tipo de archivo que no quieres que Git añada automáticamente o más aun, que ni siquiera quieras que aparezca como no rastreado. En estos casos, puedes crear un archivo llamado .gitignore que liste patrones a considerar.

Este es un ejemplo de un archivo .gitignore:

```
daw2@IS32WX06 MINGW64 ~/Des
$ cat .gitignore
/build/
.DS_Store
*~
*.swp
/tags
CMakeSettings.json
.vs
.idea
```

- **Confirmar tus Cambios**

Ahora que tu área de preparación está como quieres, puedes confirmar tus cambios. la última vez que ejecutaste git status verificaste que todo estaba preparado y que estás listo para confirmar tus cambios.

La forma más sencilla de confirmar es escribiendo **git commit**:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file> ..." to update what will be committed)
  (use "git restore <file> ..." to discard changes in working directory)
        deleted:    mysql-connector-java-8.0.30.jar

no changes added to commit (use "git add" and/or "git commit -a")

daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git add .

daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git commit -m "Mensaje del commit"
[master 70832ef] Mensaje del commit
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 mysql-connector-java-8.0.30.jar
```

- **Ver historial de confirmaciones**

Después de haber hecho varias confirmaciones, o si has clonado un repositorio que ya tenía un histórico de confirmaciones, probablemente quieras mirar atrás para ver qué modificaciones se han llevado a cabo.

La herramienta más básica y potente para hacer esto es el comando **git log**.

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git log
commit 70832ef64fc182f5158e9cb64f2828fde63accb9 (HEAD -> master)
Author: alvarocormi <alvarocormin@gmail.com>
Date:   Tue Nov 21 09:25:12 2023 +0100

    Mensaje del commit

commit c74d57b695b21a2b85db3ad8bcb56867e214398a
Author: alvarocormi <alvarocormin@gmail.com>
Date:   Fri Nov 17 10:25:38 2023 +0100

    Primer commit de ejemplo
```


- **Deshacer cosas**

Uno de las acciones más comunes a deshacer es cuando confirmas un cambio antes de tiempo y olvidas agregar algún archivo, o te equivocas en el mensaje de confirmación. Si quieres rehacer la confirmación, puedes reconfirmar con la opción `--amend`: **git commit --amend**.

Por ejemplo, si confirmas y luego te das cuenta que olvidaste preparar los cambios de un archivo que querías incluir en esta confirmación, puedes hacer lo siguiente:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git commit -m "commit inicial"
On branch master
Untracked files:
  (use "git add <file> ..." to include in what will be committed)
        hola.txt

nothing added to commit but untracked files present (use "git add" to track)

daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git add hola.txt
Warning: in the working copy of 'hola.txt', LF will be replaced by CRLF the next
time Git touches it

daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git commit --amend
hint: Waiting for your editor to close the file ...
unix2dos: converting file C:/Users/daw2/Desktop/ejemploUsoGit/.git/COMMIT_EDITMSG to DOS format ...
dos2unix: converting file C:/Users/daw2/Desktop/ejemploUsoGit/.git/COMMIT_EDITMSG to Unix format ...
[master b41db0b] Mensaje del commit
Date: Tue Nov 21 09:25:12 2023 +0100
2 files changed, 1 insertion(+)
create mode 100644 hola.txt
delete mode 100644 mysql-connector-java-8.0.30.jar
```

- **TRABAJAR CON REMOTOS**

Para poder colaborar en cualquier proyecto Git, necesitas saber cómo gestionar repositorios remotos. Los repositorios remotos son versiones de tu proyecto que están hospedadas en Internet o en cualquier otra red.

- **Ver tus remotos**

Para ver los remotos que tienes configurados, debes ejecutar el comando **git remote**. Mostrará los nombres de cada uno de los remotos que tienes especificados.

También puedes pasar la opción `-v`, la cual muestra las URLs que Git ha asociado al nombre y que serán usadas al leer y escribir en ese remoto:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploClone (main)
$ git remote
origin

daw2@IS32WX06 MINGW64 ~/Desktop/ejemploClone (main)
$ git remote -v
origin https://github.com/libgit2/libgit2 (fetch)
origin https://github.com/libgit2/libgit2 (push)
```

- **Añadir repositorios remotos**

Para añadir un remoto nuevo y asociarlo a un nombre que puedas referenciar fácilmente, ejecuta **git remote add [nombre] [url]**.

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploClone (main)
$ git remote add pb https://github.com/paulboone/ticgit
```

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploClone (main)
$ git remote -v
origin https://github.com/libgit2/libgit2 (fetch)
origin https://github.com/libgit2/libgit2 (push)
pb      https://github.com/paulboone/ticgit (fetch)
pb      https://github.com/paulboone/ticgit (push)
```

A partir de ahora puedes usar el nombre pb en la línea de comandos en lugar de la URL entera. Por ejemplo, si quieres traer toda la información que tiene Paul pero tú aún no tienes en tu repositorio, puedes ejecutar **git fetch pb**:

```
$ git fetch pb
remote: Enumerating objects: 634, done.
remote: Total 634 (delta 0), reused 0 (delta 0), pack-reused 634
Receiving objects: 100% (634/634), 88.93 KiB | 1.46 MiB/s, done.
Resolving deltas: 100% (261/261), done.
From https://github.com/paulboone/ticgit
* [new branch]      master    -> pb/master
* [new branch]      ticgit     -> pb/ticgit
```

- **Enviar tus remotos**

Cuando tienes un proyecto que quieres compartir, debes enviarlo a un servidor. El comando para hacerlo es simple: **git push [nombre-remoto] [nombre-rama]** se enviarán todos los commits que hayas hecho al servidor:

```
daw2@IS32WX06 MINGW64 ~/Desktop/
$ git push origin master
```

Este comando solo funciona si clonaste de un servidor sobre el que tienes permisos de escritura y si nadie más ha enviado datos por el medio. Si alguien más clona el mismo repositorio que tú y envía información antes que tú, tu envío será rechazado.

- **Eliminar y renombrar remotos**

Si quieres cambiar el nombre de la referencia de un remoto puedes ejecutar `git remote rename`. Por ejemplo, si quieres cambiar el nombre de `pb` a `paul`, puedes hacerlo con **git remote rename**:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploClone (
$ git remote rename pb paul
Renaming remote references: 100% (2/2), done.
```

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploClone (
$ git remote
origin
paul
```

Es importante destacar que al hacer esto también cambias el nombre de las ramas remotas. Por lo tanto, lo que antes estaba referenciado como `pb/master` ahora lo está como `paul/master`.

Si por alguna razón quieres eliminar un remoto - has cambiado de servidor o no quieres seguir utilizando un mirror o quizás un colaborador ha dejado de trabajar en el proyecto - puedes usar **git remote rm**:

```
daw2@IS32WX06 MINGW64 ~/D€
$ git remote rm paul
```

```
daw2@IS32WX06 MINGW64 ~/D€
$ git remote
origin
```

- **RAMIFICACION**

Cuando hablamos de ramificaciones, significa que tú has tomado la rama principal de desarrollo (`master`) y a partir de ahí has continuado trabajando sin seguir la rama principal de desarrollo.

- **Crear una rama**

Por ejemplo, supongamos que quieres crear una rama nueva denominada `"testing"`. Para ello, usarás el comando **git branch**:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git branch testing
```

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git branch
* master
  testing
```

- **Cambiar de rama**

Para saltar de una rama a otra, tienes que utilizar el comando **git checkout**. Hagamos una prueba, saltando a la rama testing recién creada:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git checkout testing
Switched to branch 'testing'
M      cosasQueHacer.txt

daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (testing)
$ |
```

- **Procedimientos básicos de ramificación**

Para crear una nueva rama y saltar a ella, en un solo paso, puedes utilizar el comando **git checkout con la opción -b**:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (testing)
$ git checkout -b rama2
Switched to a new branch 'rama2'

daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (rama2)
$ |
```

Para borrar una rama podremos usar el comando **git branch -d [nombreRama]**, en mi caso voy a borrar la rama testing.

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git branch -d testing
Deleted branch testing (was b41db0b).
```

- **Procedimientos básicos de fusion**

Vas a fusionar la rama rama2. Simplemente, activa (checkout) la rama donde deseas fusionar y lanza el comando **git merge**:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git checkout master
Already on 'master'
M      cosasQueHacer.txt

daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git merge rama2
Already up to date.
```

- **Gestion de ramas**

Ahora que ya has creado, fusionado y borrado algunas ramas, vamos a dar un vistazo a algunas herramientas de gestión muy útiles cuando comienzas a utilizar ramas de manera avanzada.

El comando **git branch** tiene más funciones que las de crear y borrar ramas. Si lo lanzas sin parámetros, obtienes una lista de las ramas presentes en tu proyecto:

```
daw2@IS32WX06 MINGW64 ~/Desk
$ git branch
* master
  rama2
```

Fijate en el carácter * delante de la rama master: nos indica la rama activa en este momento (la rama a la que apunta HEAD). Si hacemos una confirmación de cambios (commit), esa será la rama que avance.

Para ver la última confirmación de cambios en cada rama, puedes usar el comando **git branch -v**:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git branch -v
* master b41db0b Mensaje del commit
  rama2 b41db0b Mensaje del commit
```

Otra opción útil para averiguar el estado de las ramas, es filtrarlas y mostrar solo aquellas que han sido fusionadas (o que no lo han sido) con la rama actualmente activa. Para ello,

Git dispone de las opciones **--merged** y **--no-merged**. Si deseas ver las ramas que han sido fusionadas con la rama activa, puedes lanzar el comando **git branch --merged**:

```
daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git branch --merged
* master
  rama2

daw2@IS32WX06 MINGW64 ~/Desktop/ejemploUsoGit (master)
$ git branch --no-merged
```

• GENERANDO CLAVE SSH

Puedes acceder a datos y escribirlos en repositorios en GitHub.com mediante SSH (protocolo Secure Shell). Al conectarse a través de SSH, se realiza la autenticación mediante un archivo de clave privada en el equipo local.

Para configurarlo nos iremos al siguiente link <https://github.com/settings/key>

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication Keys



key-home

SHA256:a3FWKYd09Gjw9FqinHd7bgTFNaPgMQKjtYFfcN+ktLU

Added on Feb 22, 2023

Last used within the last 10 months — Read/write

Delete

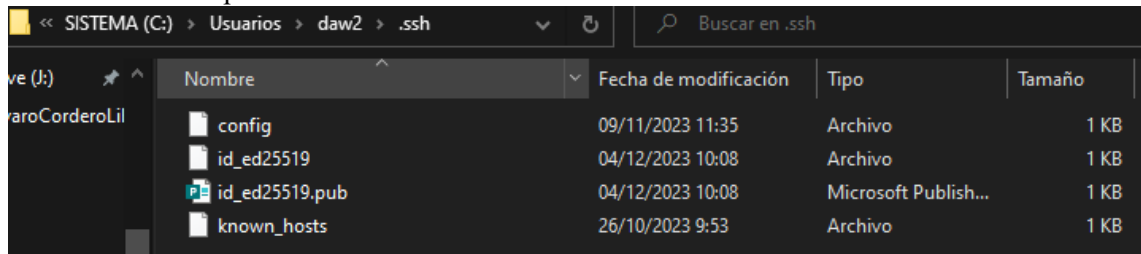
Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

Para completar los campos que nos aparecen, debemos abrir un git Bash en nuestro equipo, en cualquier sitio, eh introducir el siguiente comando **ssh-keygen -t ed25519 -C "EMAIL_ASOCIADO_A_GITHUB"**.

- **Enter File** → En esta opción nos pide que elijamos una ruta por defecto para guardar los archivos que va a generar, si no indicamos ninguno se quedará el que marca por defecto. En nuestro caso la dejamos por defecto.
- **Enter passphrase** → Aquí nos pide una “frase como contraseña” y nos dice que por defecto la deja vacía. En nuestro caso ponemos ‘paso’.

```
daw2@IS32WX06 MINGW64 ~
$ ssh-keygen -t ed25519 -C "alvarocormin@gmail.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://uY3c6tTv6PDEzvLCVHw8PCG1f1CvbysigWAG9ztWI alvarocormin@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|      . . . o o |
|    o . .o+ *o |
|   = E .. o=oo |
|  . =S. o.+ o |
|    .. o + = |
|    .. + * |
|    o o & +. |
|    . . *+X+O |
+-----[SHA256]-----
```

Si nos vamos a la ruta **C:/Usuarios/daw2/.ssh** veremos que se han guardado los siguientes archivos en esa carpeta.



Nombre	Fecha de modificación	Tipo	Tamaño
config	09/11/2023 11:35	Archivo	1 KB
id_ed25519	04/12/2023 10:08	Archivo	1 KB
id_ed25519.pub	04/12/2023 10:08	Microsoft Publish...	1 KB
known_hosts	26/10/2023 9:53	Archivo	1 KB

Aquí vemos que se han generado 3 archivos, el primero es la clave privada, el segundo la pública (de ahí su extensión '.pub') y el último se utiliza para autenticar servidores remotos. (Que muestra unas líneas que se dividen en 3 partes, 'IP/HostName', Algoritmo de clave pública y clave pública del servidor).

Copiamos el contenido de la clave pública y la pegamos en el campo 'key'
También le damos un nombre y elegimos el tipo de clave 'Authentication Key'.

Title



Key type

Authentication Key

Key

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOiA85ac44V4jUzwVoBcn5Im6eh67FF1DT9uEu4UesKc alvarocormin@gmail.com
```

Una vez completado, hacemos click en 'Add SSH key' y nos aparecerá lo siguiente



ClavePersonal

SHA256://uY3c6tTv6PDEzv1CVHw8PCG1f1CvbysigWAG9ztWI

Added on Dec 4, 2023

Never used — Read/write

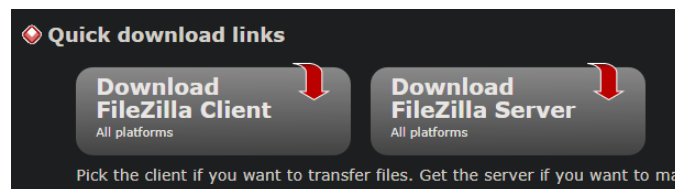
3. WXED – WINDOWS X

• FILEZILLA

FileZilla es una aplicación FTP libre y de código abierto que consta de un cliente y un servidor. Soporta los protocolos FTP, SFTP y FTP sobre SSL/TLS.

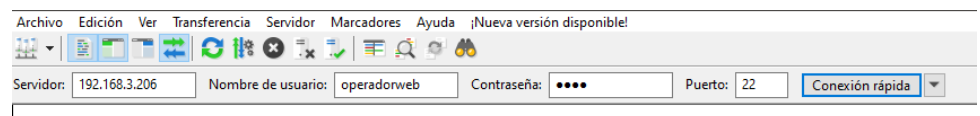
• Instalación y configuración inicial

- Para instalar Filezilla hay que ir a la pagina oficial de Filezilla y descargar la ultima versión del mismo (<https://filezilla-project.org/>), y descargamos el filezilla client para todas las paltaformas.

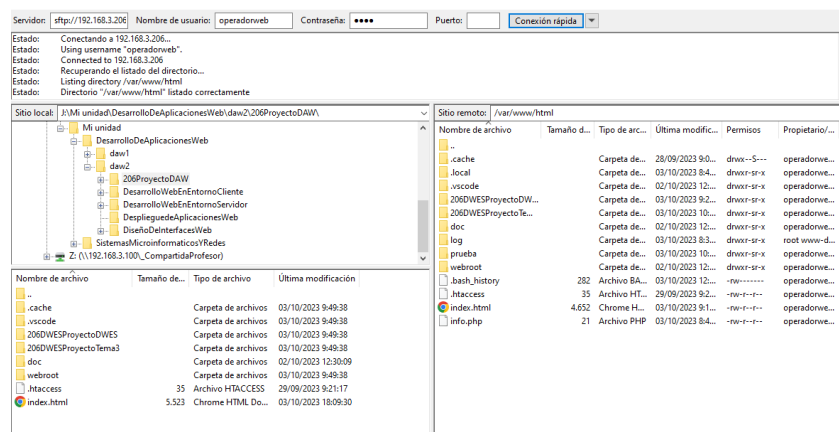


• Como conectarse a un servidor mediante SFTP

- Para conectarse a un servidor SFTP mediante filezilla hay que rellenar los campos que aparecen en la aplicación de Filezilla Client con los datos correspondientes de dicho servidor.



- Una vez dentro podremos traspasar archivos simplemente arrastrando documentos de una ventana a otra

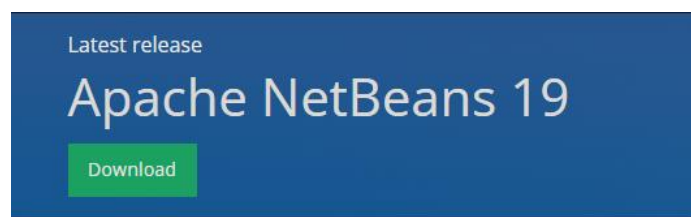


- **NETBEANS**

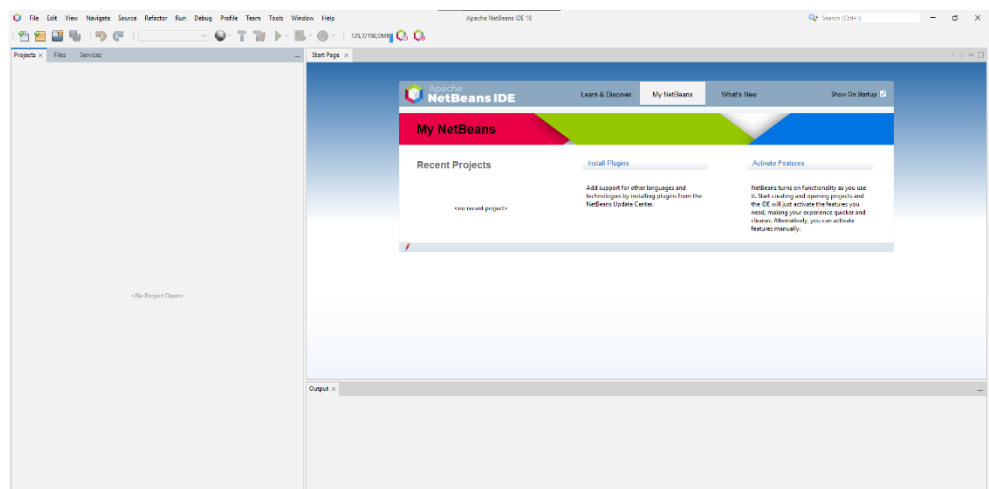
NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

- **Instalación y configuración inicial**

- Para instalar netbeans hay que ir a la página oficial de apache netbeans mediante el siguiente enlace (<https://netbeans.apache.org/download/index.html>).

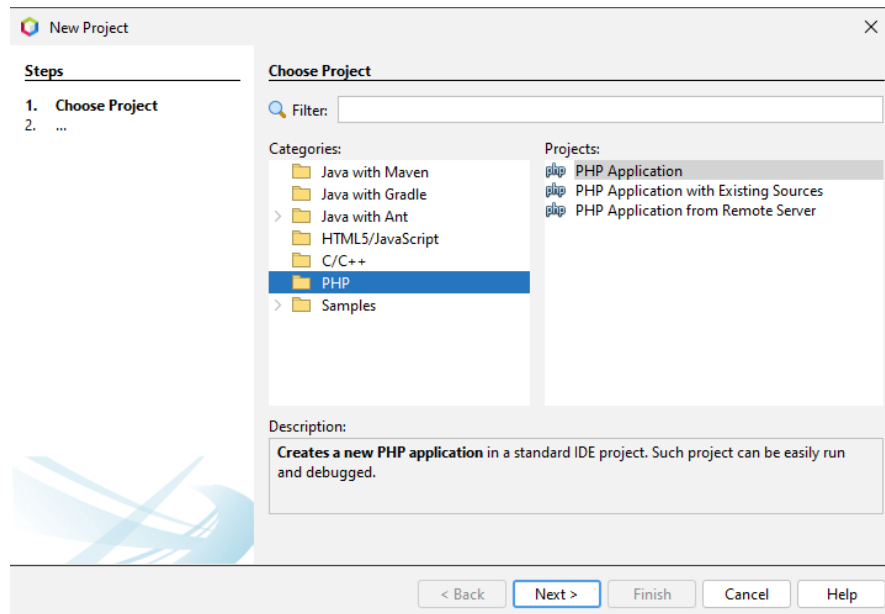


- En mi caso no tengo ningún plugin instalado en Apache NetBeans, me gusta la configuración que viene por defecto

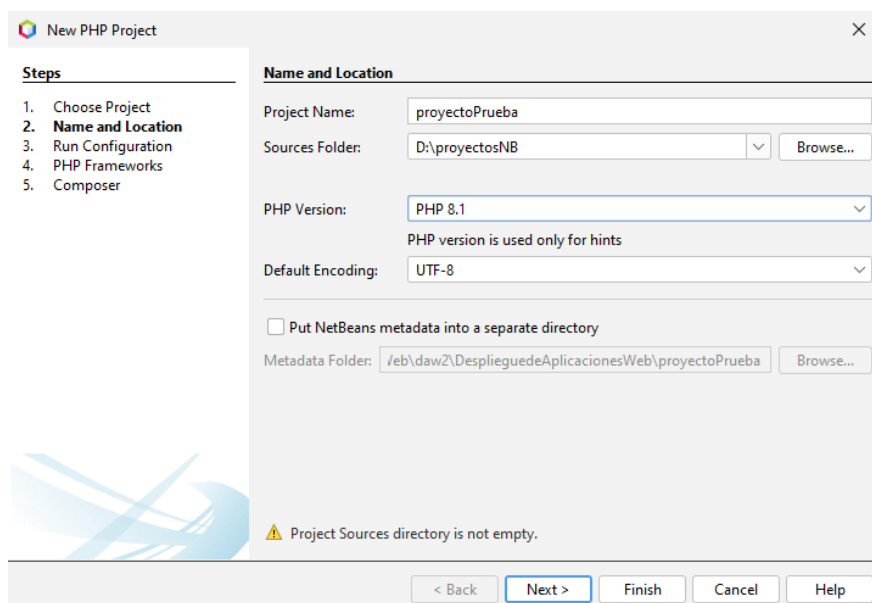


- **Creación y modificación de proyectos**

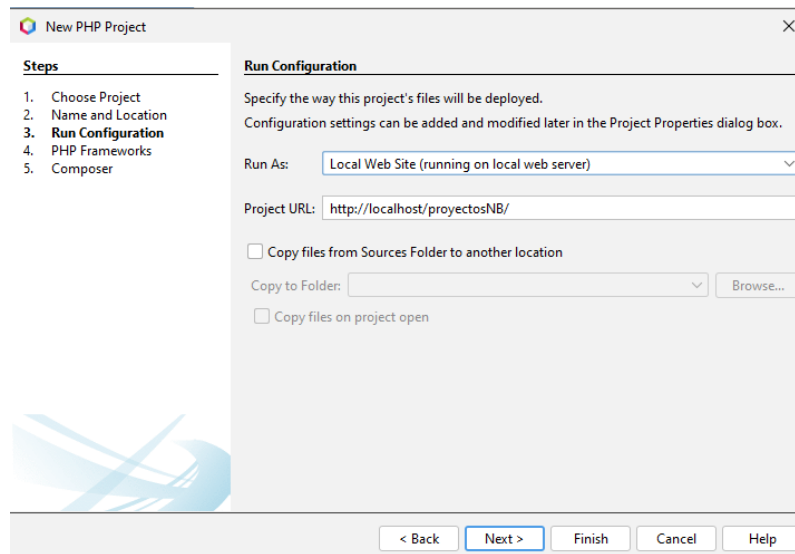
- Para crear un proyecto en ApacheNetBeans hay que irse a File/New Project y cuando le demos saldrá lo siguiente.
- En la sección de categorías podremos elegir el lenguaje de programación en el que crearemos nuestro proyecto, para este ejemplo voy a usar PHP



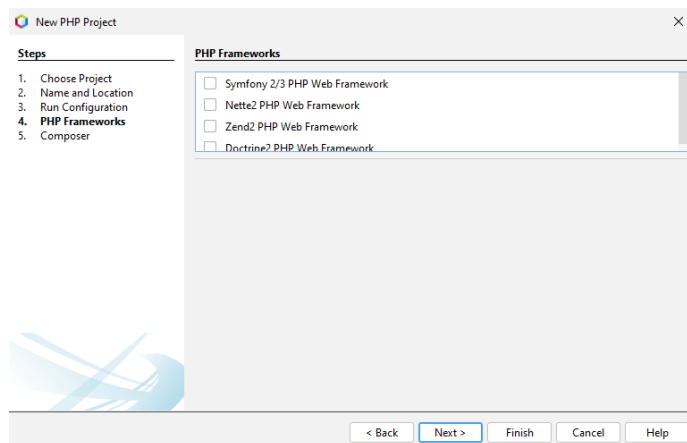
- En el siguiente apartado tendremos que elegir el nombre que le queremos dar al proyecto, el lugar donde lo quieres guardar y la versión de PHP que queremos usar en mi caso es lo siguiente.



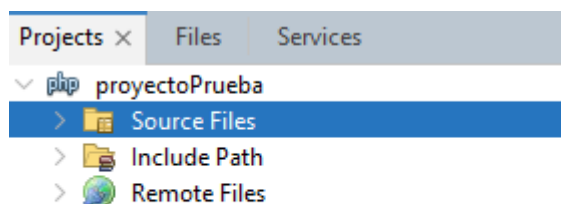
- Ahora nos pedirá elegir si queremos guardar el proyecto en local o en remoto, en este primer ejemplo lo voy a guardar en local.



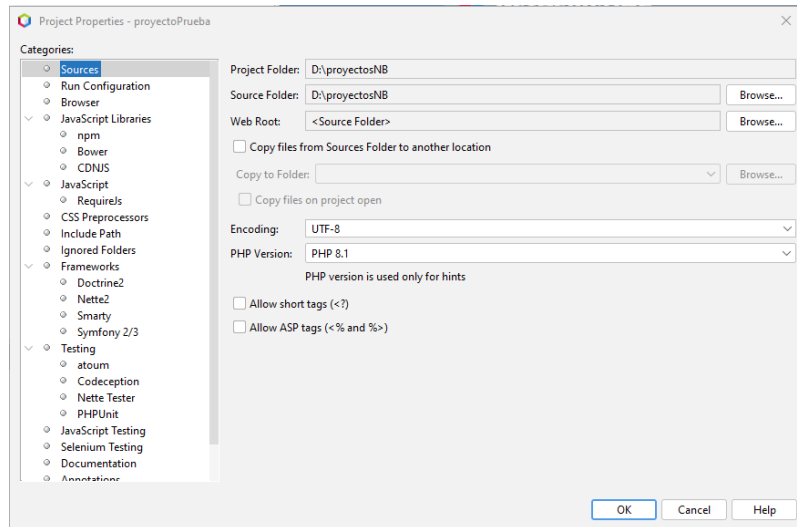
- Podemos elegir la opción de escoger un framework de PHP para poder utilizarlo en nuestro proyecto en mi caso no voy a coger ninguno. Y después de esto le damos a finish o Finalizar.



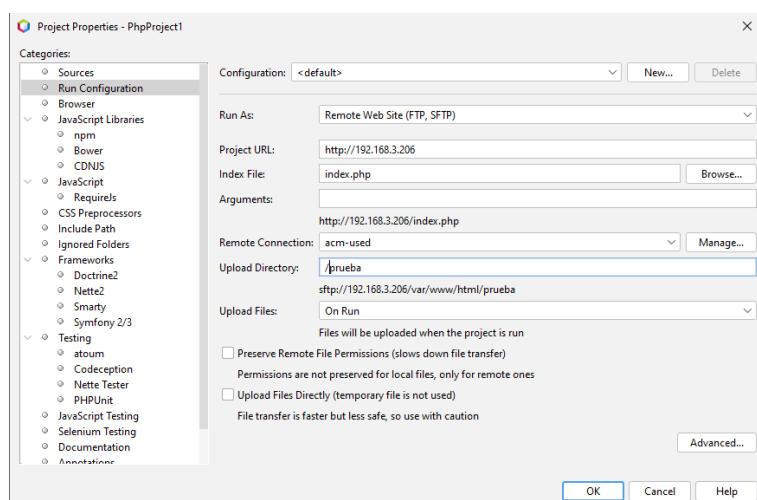
- En el apartado de Projects se nos habrá creado lo siguiente



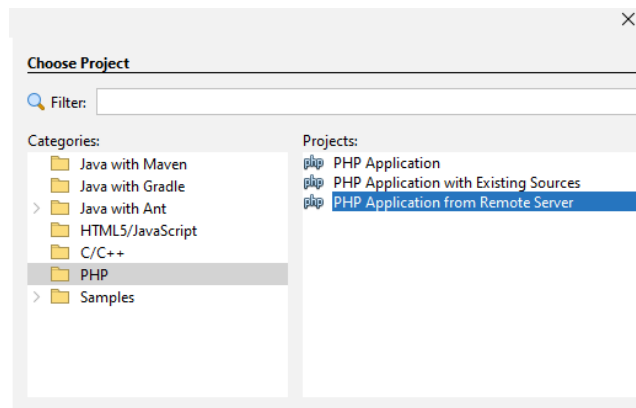
- Si le damos click derecho y nos vamos a propiedades, podremos modificar muchos apartados de nuestro proyecto.



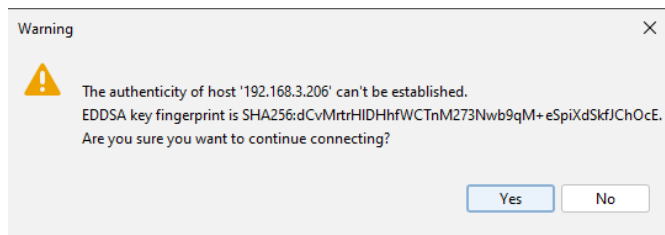
- Y para borrar el proyecto es muy sencillo simplemente tenemos que dar click derecho al proyecto y darle a delete. Y de esta manera se nos habrá borrado el proyecto en local de nuestro equipo
- **Conexión al servidor remoto SFTP**
- Para poder vincular un proyecto tuyo en local a un servidor remoto mediante el protocolo SFTP hay que ir a tu proyecto, propiedades e ir al apartado de configuración de arranque y cambiar los siguiente.



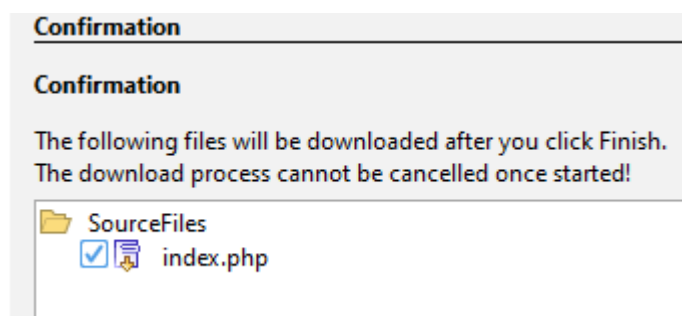
- Otra opción es crear un nuevo proyecto en remoto de la siguiente manera. Lo primero que tienes que hacer es escoger la opción de PHP remote.



- Después tienes que introducir la Dirección Ip de tu servidor y el directorio donde quieres que se guarde tu proyecto en remoto.
- Te saldrá el siguiente aviso , le tienes que decir que si

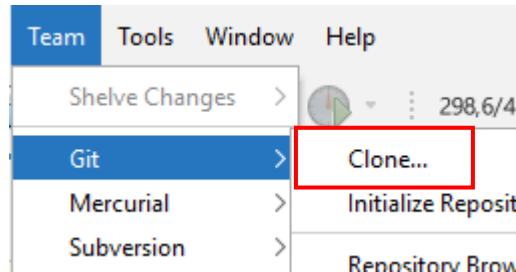


- Ahora te sale los ficheros que tienes en tu servidor para poder vincularlos y trabajar con ellos en remoto, así que le damos a finish.

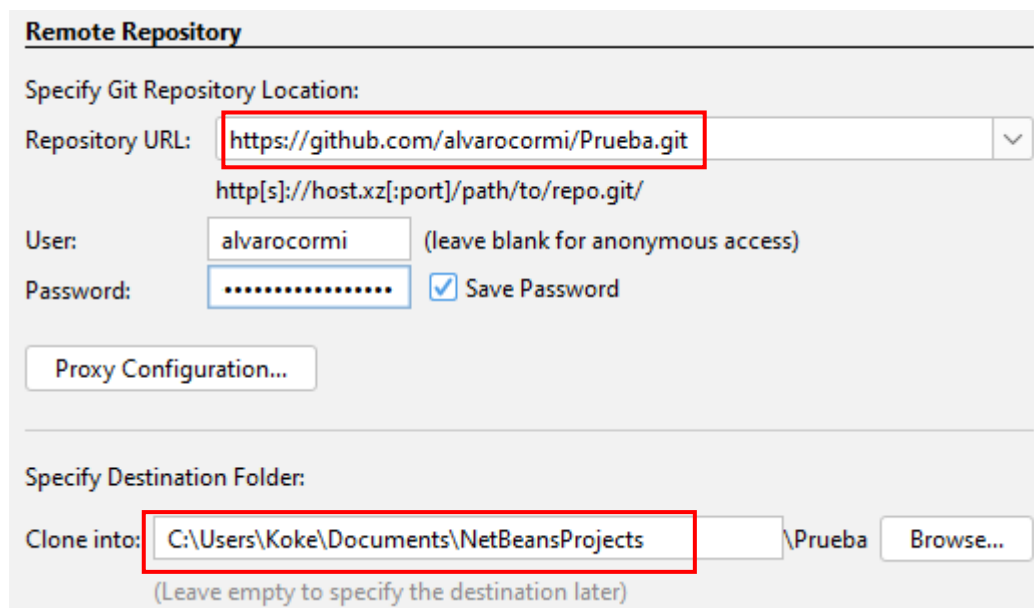


- **Clonar un repositorio de Github**

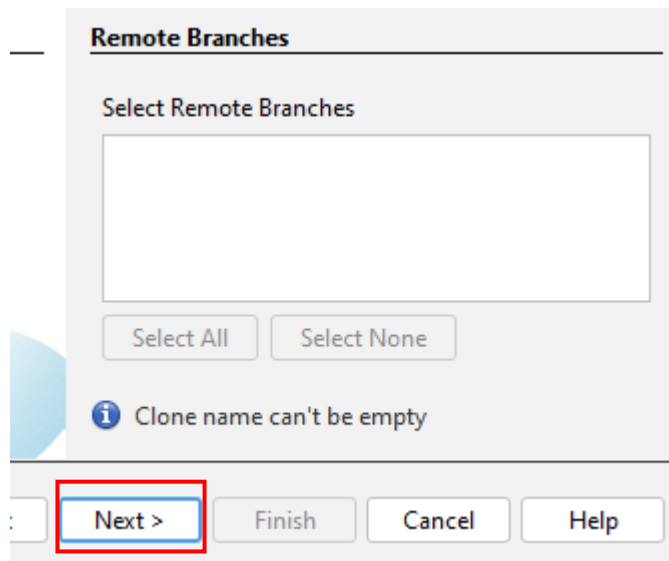
- Para clonar un repositorio de GitHub , lo que tenemos que hacer es ir al apartado Team / Git / Clone



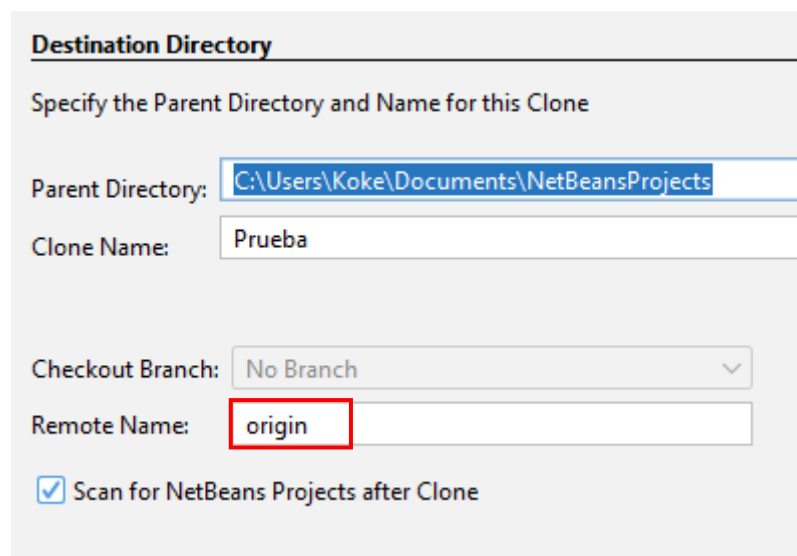
- Una vez allí, tendremos que indicarle la url de nuestro repositorio y el directorio donde queremos que se nos guarde el proyecto de forma local.
 - **User** → Nombre de usuario de tu cuenta de Github
 - **Password** → Token personal



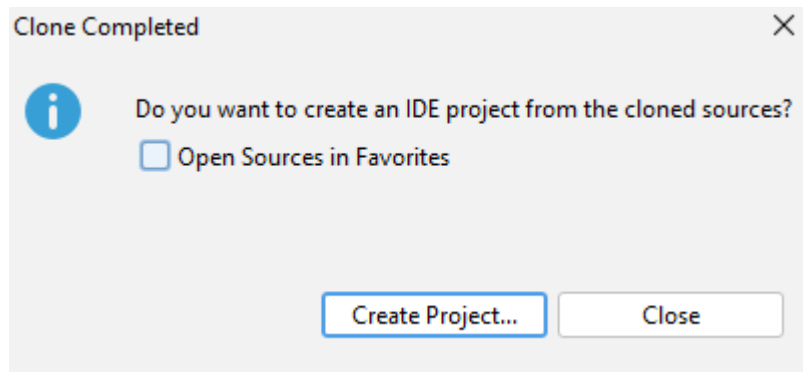
- El siguiente paso que nos pide es elegir la rama del proyecto que queremos clonar, en mi caso no tengo ninguna rama entonces le doy anext



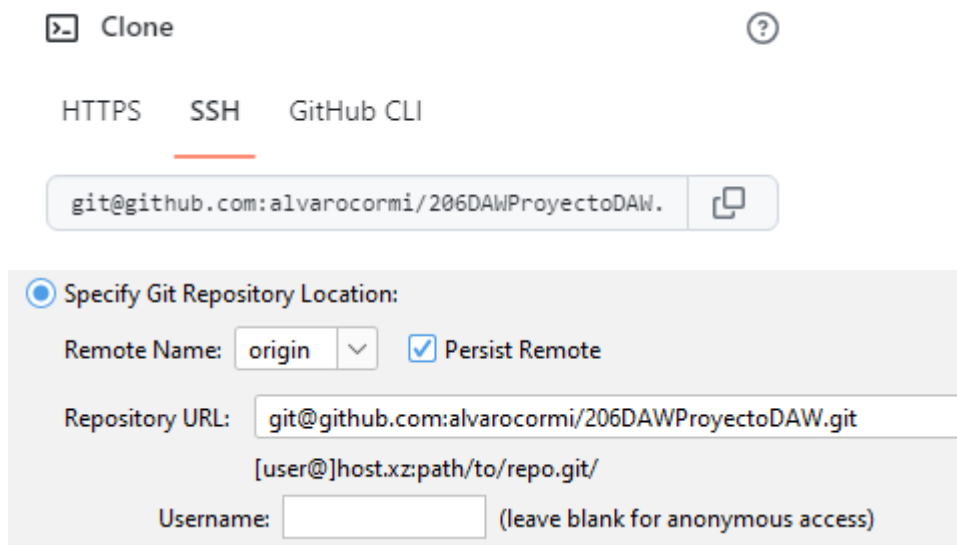
- Ahora nos pide el nombre que le vamos a dar al **proyecto en local** y nos vuelve a preguntar el destino de **nuestro proyecto en local** junto con el nombre que le queremos a dar a la rama principal que va a crear automáticamente.



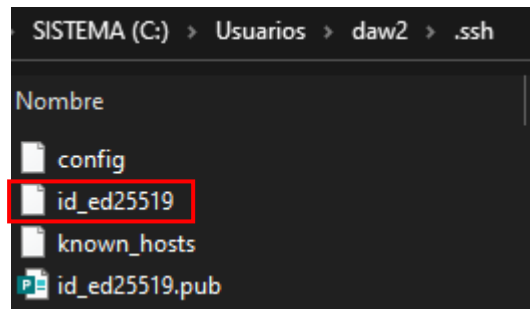
- Una vez echo eso le damos a finish y nos preguntara si queremos crear un proyecto de NetBeans con el proyecto que hemos clonado, en mi caso no lo voy a hacer.



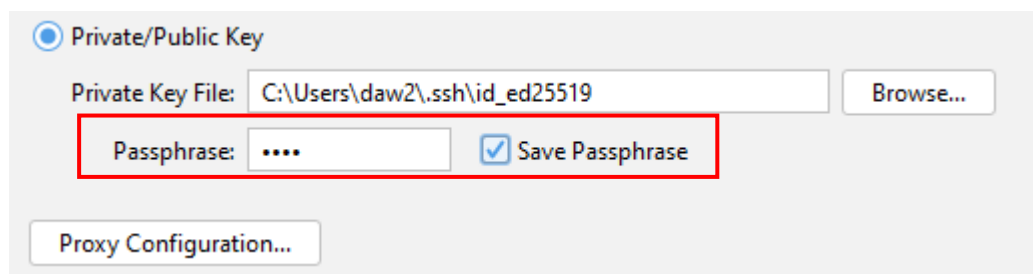
- Para hacer una pull request desde netbeans usando claves SSH tendremos que poner como url del proyecto la url ssh de github. **Es importante no poner tu Username ya que esta elegido por defecto en la url.**



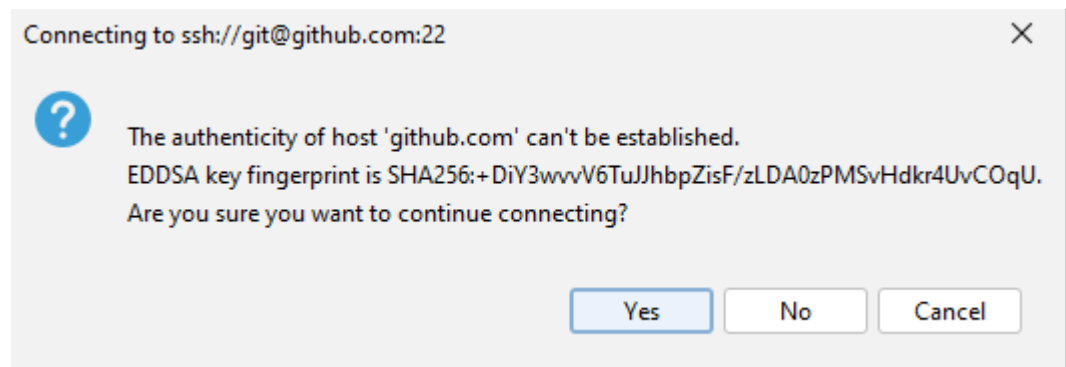
- El segundo paso es pasarle a nuestro ID nuestra clave privada , para ello nos tendremos que ir a la ruta C:/Usuarios/daw2/.ssh/ y alli encontraremos lo siguiente.



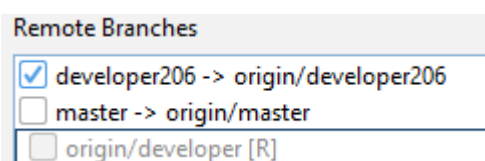
- Elegimos esa clave y escribimos la contraseña de la clave SSH



- Al darle a siguiente nos mostrar esta ventana la cual nos indica si queremos seguir la conexión, en mi caso le voy a decir que si

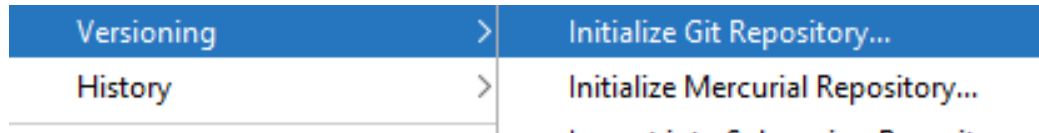


- La siguiente pantalla que nos saldrá es para elegir la rama que queremos hacer un pull en mi caso voy a elegir la rama developer.

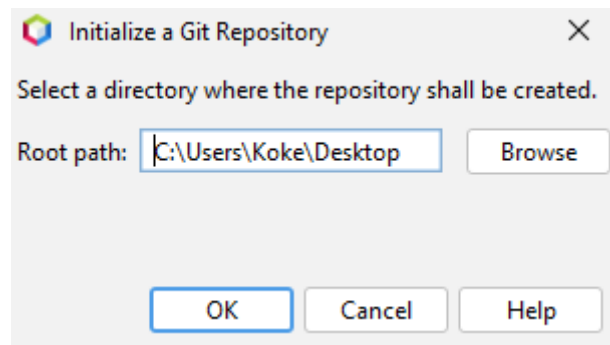


- **Uso de Github en NetBeans**

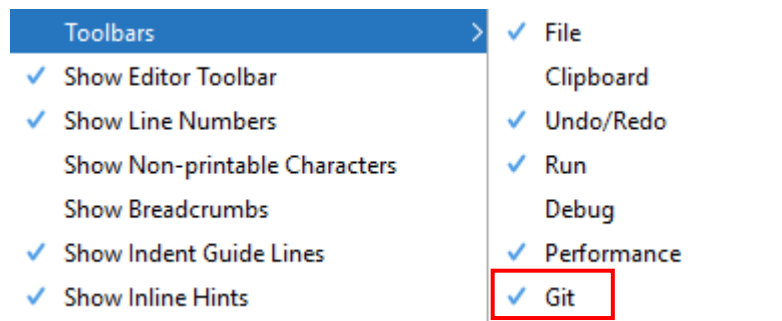
- Lo primero que hay que saber hacer con Git es inicializar un proyecto en NetBeans es muy sencillo, para ello lo primero que tenemos que tener es un proyecto vinculado a github, para inicializarlo es muy sencillo, le damos click derecho al proyecto Versioning / Inicializate Git Respository



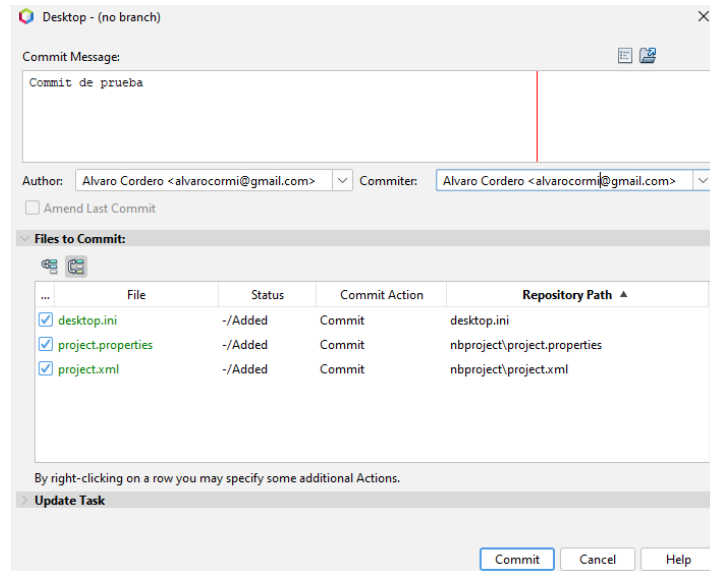
- Ahora nos pedirá la ruta donde queremos guardar nuestro proyecto en local, y una vez le damos a OK nuestro proyecto ya esta inicializado



- Para facilitar el uso de Git en NetBeans es recomendable activar la toolbar de acciones de GIT, mediante esta opción podremos tener mas a mano las opciones principales de Git, para ello nos a View / ToolBars / Git

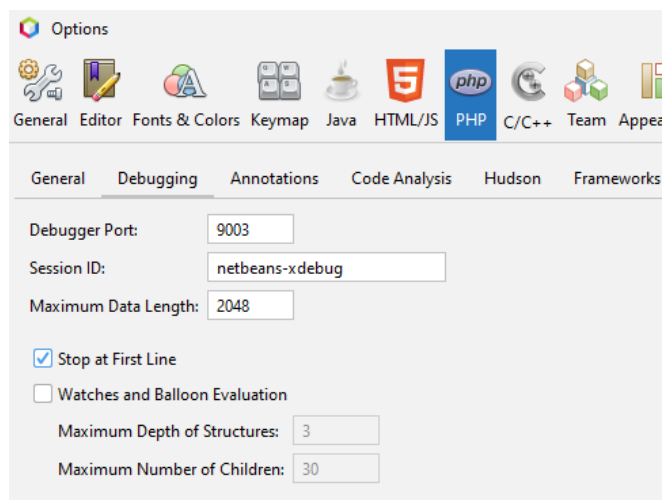


- Para hacer un commit, tenemos que darle click derecho al proyecto y le damos a Git / Commit y nos saldrá lo siguiente.
 - **Commit Message** → Mensaje que le queremos dar al commit.
 - **Author** → Autor del commit, | Nombre <correodegithub>
 - **Committer** → Autor del commit, | Nombre <correodegithub>
 - **Files** → Ficheros en los cuales vas a hacer los commits

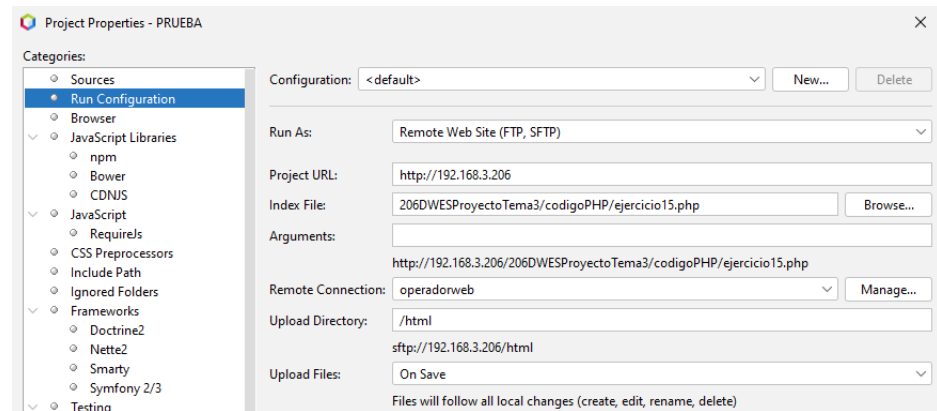


• Depuracion

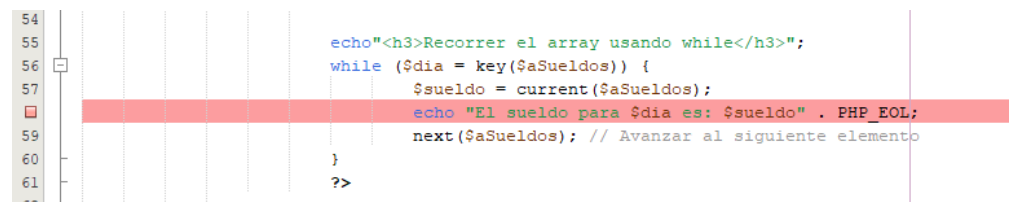
- Abrimos la configuración de NetBeans mediante los menús Tools -> Options. Accedemos a la configuración de PHP y en la pestaña Debugging nos aseguramos de que hemos especificado el mismo puerto que en la configuración de XDebug para PHP, en nuestro caso el puerto 9003.



- La última configuración necesaria para poder debugar se realiza en la configuración de cada proyecto de Netbeans. Para acceder a ella debemos tener visible la ventana de proyectos, botón derecho sobre el propio proyecto y seleccionamos la opción Properties el menú.



- Para usar xdebug lo primero que hay que hacer es marcar un punto de parada o “breackpoint”. Para ello damos click donde queramos que sea ese punto de parada

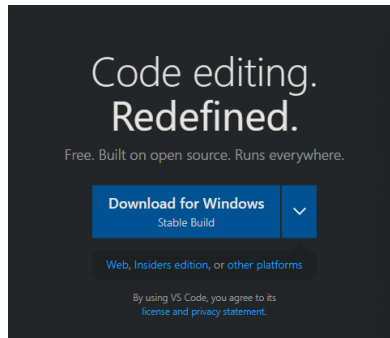


- **VISUAL STUDIO CODE**

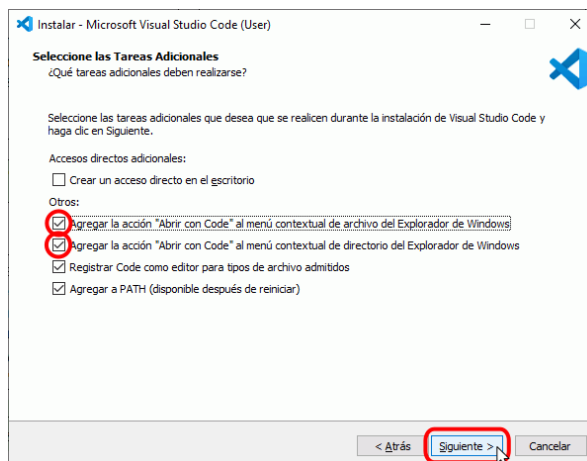
Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

- **Instalación y configuración**

- Para instalar es IDE hay que ir a la pagina oficial de visual studio mediante este enlace (<https://code.visualstudio.com/>) y darle click a lo siguiente.



- La instalación es muy sencilla, pero hay un momento en el que te pide clickar o no unos ticks yo os recomiendo tenerlo de la siguiente manera para que podáis abrir cualquier archivo o carpeta en visual studio code con el click derecho.

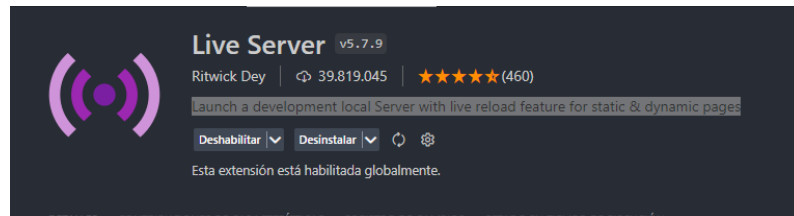


a. Extensiones

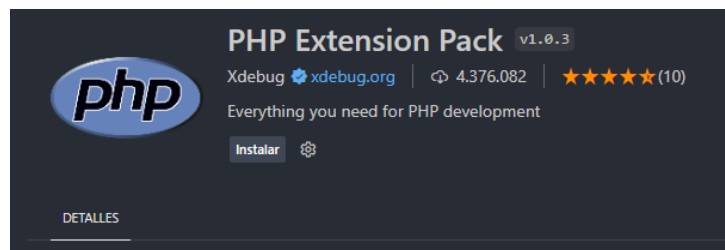
- Visual Studio destaca por la gran cantidad de posibilidades que tienes para personalizar este editor gracias a las extensiones del mismo, una de las primeras configuraciones que recomiendo hacer es cambiar el idioma del IDE, para ello vamos al apartado de extensiones y buscamos ‘Español’.



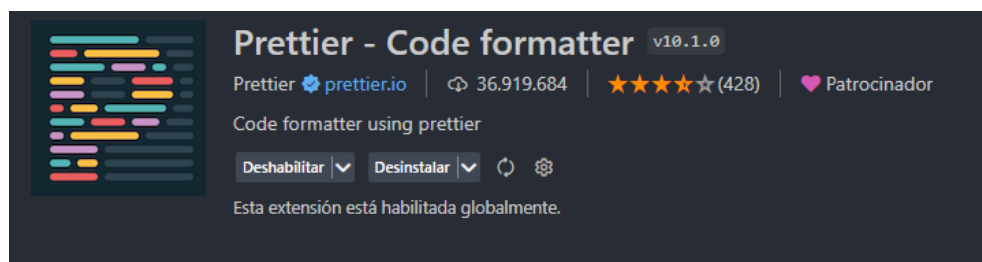
- Otra extensión que recomiendo mucho instalar es ‘Live Server’ esta extensión nos permite lanzar un servidor local de desarrollo con función de recarga en vivo para páginas estáticas y dinámicas.



- Para desarrollar en PHP hay un pack de extensiones que puedes descargar que te ayudara mucho a la hora de autocompletar, colorado de sintaxis, debug..., esta se llama ‘PHP Extension Pack’

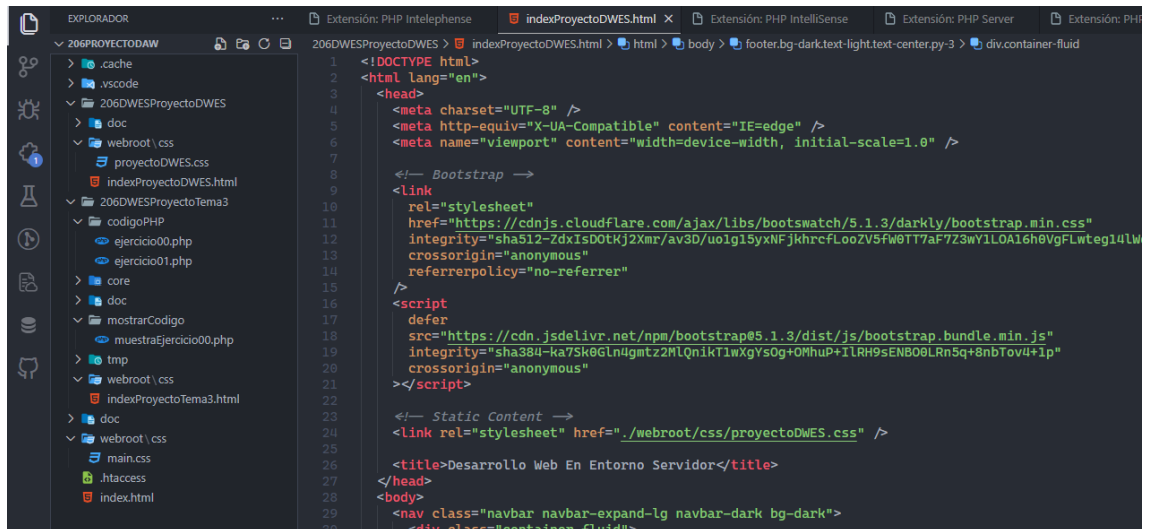


- Prettier es una extensión que te permitira formatear tu código de una manera muy sencilla.



b. Crear o abrir un proyecto

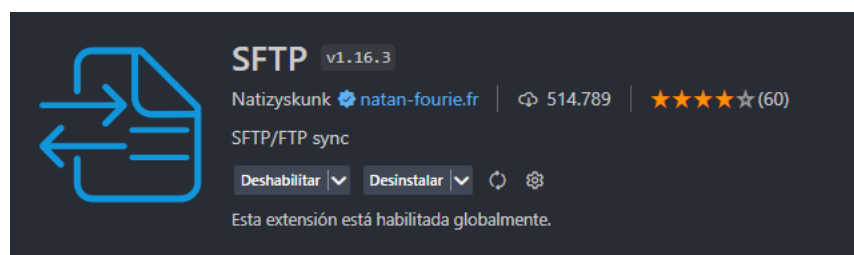
- En este IDE no hay un apartado específico para crear un proyecto normalmente los proyectos que se suelen realizar suelen crearse mediante el gestor de paquetes de node que es npm, o dependiendo del lenguaje tu puedes descargar las extensiones del mismo para crear proyectos de ese tipo.
- En java puedes descargar el pack de extensiones en java para poder crear proyectos en Maven.
- Si tu quieres empezar un proyecto te servirá con abrir una carpeta en el IDE y este ya lo detectará como un proyecto que puedes comprimir.



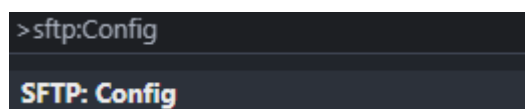
The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a file tree for a project named '206DWESProyectoDWES'. The tree includes folders like '.cache', '.vscode', 'doc', 'webroot', and 'css', along with various files such as 'ejercicio00.php', 'ejercicio01.php', 'core', 'doc', 'mostrarCodigo', 'muestraEjercicio00.php', 'tmp', 'webroot', 'css', 'indexProyectoTema3.html', 'main.css', '.htaccess', and 'index.html'. The main editor area shows the content of 'indexProyectoDWES.html', which is an HTML document using Bootstrap 5.1.3. The code includes a DOCTYPE declaration, meta tags for charset, viewport, and compatibility, and links to Bootstrap CSS and jQuery. The body contains a navigation bar and a container fluid.

c. Conectarse a un servidor mediante SFTP

- Para conectarte a un servidor mediante SFTP en Visual Studio Code, lo primero que hay que hacer es instalar una extensión que se llama SFTP.



- Ya instalada la extensión usaremos las teclas de (Ctrl+shift+p) para abrir el buscador de Visual, en ese buscador escribiremos SFTP config y le damos a enter.



- Ya dentro nos saldrá un documento con distinta información que tenemos que poner para poder conectarnos a ese servidor, en mi caso lo tengo configurado de la siguiente manera.
- <https://oscarnovas.com/blog/configurar-el-modulo-sftp-en-vscode>

```
Coincide con varios esquemas cuando solo se debe validar uno.
{
  "name": "ProyectoDAW",
  "defaultProfile": "desarrollo",

  "profiles": {
    "desarrollo": {
      "host": "192.168.20.19",
      "protocol": "sftp",
      "port": 22,
      "username": "DAW206",
      "password": "paso",
      "remotePath": "./public_html/206DWESProyectoDWES"
    },

    "explotacion": {
      "host": "home496960781.1and1-data.host",
      "protocol": "sftp",
      "port": 22,
      "username": "u74942863-daw206",
      "password": "daw2_Sauces",
      "remotePath": "[ruta remota]"
    },

    "casa": {
      "host": "192.168.0.206",
      "protocol": "sftp",
      "port": 22,
      "username": "operadorweb",
      "password": "paso",
      "remotePath": "[ruta remota]"
    }
  },

  "ignore": [".vscode/", ".vscode/**", ".nbproject/**", ".gitignore"],

  "watcher": {
    "files": "css/*.{map,css}",
    "autoUpload": true,
    "autoDelete": true
  },

  "downloadOnOpen": true,
  "syncMode": "full",
  "uploadOnSave": true,
  "useTempFile": false,
  "openSsh": true
}
```

You, hace 9 minutos • Uncommitted changes

- Al instalar esta extensión automáticamente en la barra de tareas izquierda nos va aparecer un nuevo icono en el cual si pinchamos nos lleva a los archivos del servidor que estamos vinculados.

