

DESARROLLO DE APLICACIONES WEB 2

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



4 DE DICIEMBRE DE 2023

DESARROLLO WEB EN ENTORNO SERVIDOR - DESPLIEGUE DE APLICACIONES WEB
CARLOS GARCIA CACHON

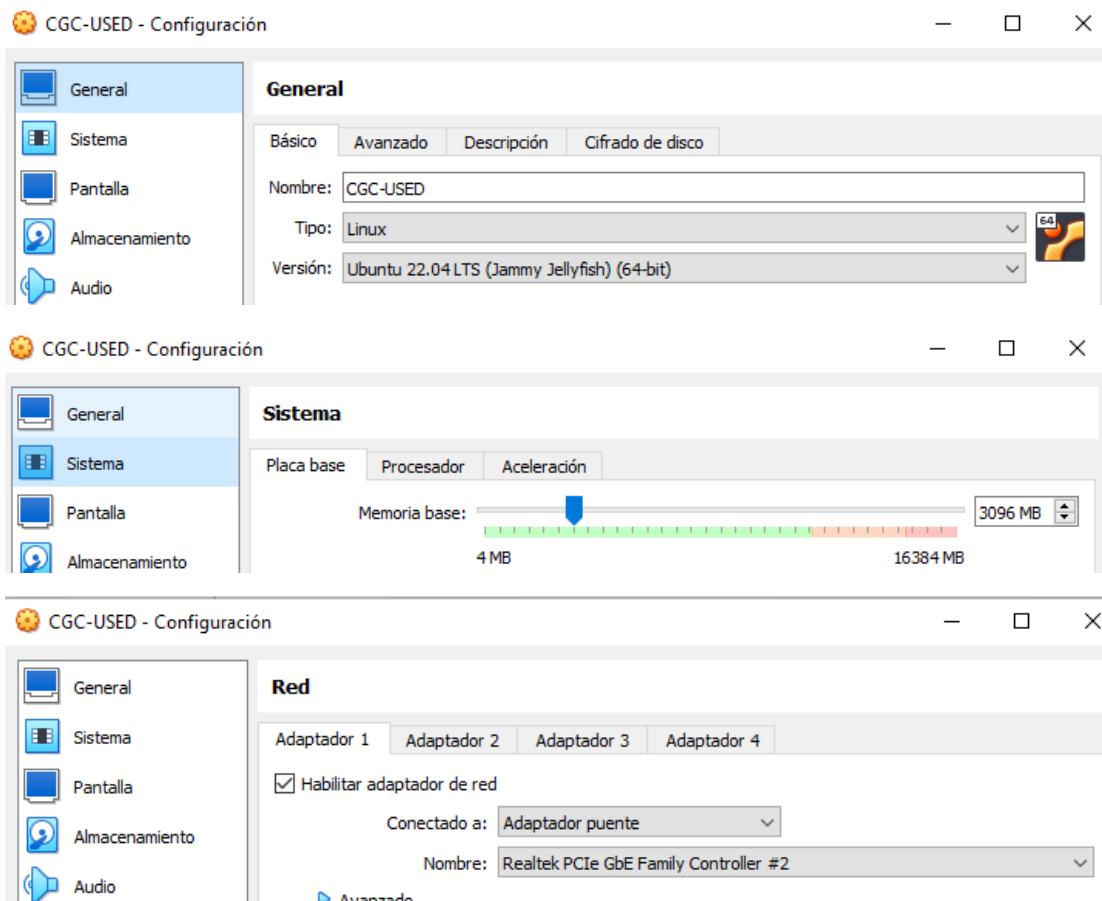
DOCUMENTACIÓN DE LA INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR (USED):.....	4
Configuración inicial:.....	4
Configuración de red:.....	4
Cambiar nombre del equipo:.....	5
Actualizar sistema:	5
Cuentas administración:	5
Creación de usuario:	5
Para añadirlo al grado de super usuarios	5
APACHE	5
Instalar el servicio:	5
Apertura de puertos.....	6
Comprobar conectividad con tu maquina cliente:	6
Habilitar el uso del fichero .htaccess	7
Configurar que los logs de error y de acceso sean accesibles desde el navegador.....	8
Personalizar páginas de error de cada proyecto	8
Directiva Apache2	8
Informe de estado del servicio de apache	9
Directorios Virtuales	9
Redirigir en Apache	12
SERVIDOR DNS	13
Instalación del servidor DNS	13
Declaración de zona directa.....	15
Chequear ficheros de configuración	15
RR en zona directa	15
RR en zona inversa	17
Comprobación de funcionamiento	18
Reenviador	18
HTTPS	20
PHP	29
Instalación de PHP:	29
Instalación del módulo interprete:	29
Comprobación de PHP:	29
Configuración php.ini.....	30
XDEBUG.....	31
Instalar directivas Xdebug.....	31

MYSQL.....	33
Instalación MySQL.....	33
Script de Seguridad	34
Logs y Errores.....	37
Conexión con NetBeans a MySQL.....	37
PHPMYADMIN.....	39
Verificación de los módulos PHP	39
Instalación phpMyAdmin.....	39
CUENTAS DE DESARROLLO Y HOSTING VIRTUAL.....	44
Creación del usuario:	44
Modificar los permisos ‘operadorweb’:.....	44
Enjaulado de usuarios	45
Configuración fichero de hosting.....	46
DOCUMENTACIÓN DE LA INSTALACIÓN Y CONFIGURACIÓN DEL CLIENTE (WXED):.....	47
Instalación y configuración inicial de la máquina	47
Cuentas administradoras y cuenta de desarrollador.....	47
Navegadores	48
Filezilla.....	50
Versión	50
Conexión con el servidor.....	50
Subir y Bajar archivos del servidor.....	51
NetBeans.....	52
Instalación y configuración inicial (pluggings)	52
Creación de un proyecto PHP	52
Administración del almacenamiento local y remoto	56
Configuración de la ejecución.....	57
Modificación de proyectos.....	57
Borrado de proyectos	58
Traslado de proyectos.....	58
Depuración de código PHP.....	60
Conexión NetBeans con el servidor isauces.....	61
Administración de la base de datos.....	63
Conexión al repositorio – versionado	65
Instalación de Git	65
Configuración de Git	69

Crear repositorio	69
Iniciar repositorio.....	71
Commit.....	76
Push y generar token	78
Create Branch.....	85
Pull	88
Tag.....	92
Release	93
Merge.....	94
Rebase.....	98
SSH Key (Clave SSH)	100
Git Cheat Sheet	103
Paso a explotación	104
PHP Doc.....	106

DOCUMENTACIÓN DE LA INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR (USED):

Configuración inicial:



Configuración de red:

Una vez instalada configuramos la red desde el archivo 00-installer-config.yaml:

```
cd /etc/netplan/
ls
ip a
# This is the network config
network:
  ethernets:
    enp0s3:
      addresses:
        - 192.168.3.214/24
      nameservers:
        addresses:
          - 8.8.8.8
        search: []
      routes:
        - to: default
          via: 192.168.3.1
      version: 2
```

miadmin@CGC-USED:/\$ cd /etc/netplan/
miadmin@CGC-USED:/etc/netplan\$ ls
00-installer-config.yaml

miadmin@CGC-USED:/\$ sudo netplan apply

miadmin@CGC-USED:/\$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP>
 link/loopback 00:00:00:00:00:00
 inet 127.0.0.1/8 scope host
 valid_lft forever prefer
 inet6 ::1/128 scope host
 valid_lft forever prefer
2: enp0s3: <BROADCAST,MULTICAST>
 link/ether 08:00:27:7a:66:1
 inet 192.168.3.214/24 brd
 valid_lft forever prefer
 inet6 fe80::a00:27ff:fe7a:66%2
 valid_lft forever prefer

La estructura anterior, normalmente viene por defecto, en la parte donde dice ‘ethernets’ y luego ‘addresses’ pondríamos la IP de la MV (EL ‘/24’ se refiere a la máscara de subred, donde hacer referencia a que los primeros 24 bits se utilizan para identificar la red y los últimos 8 bits para la red, en nuestro caso 214). Donde dice ‘nameservers’ , en el apartado de ‘addresses’ pondremos el DNS (Sistema de Nombres de Dominio) que es como la guía telefónica de Internet, en este caso usamos una dirección que corresponde a uno de los servidores DNS de Google. Y por último en ‘routes’ donde dice ‘via’ va la puerta de enlace, es como la “salida” de la red local hacia otras redes.

Cambiar nombre del equipo:

Para ello debemos editar 2 archivos, (/etc/hosts), (/etc/hostname), y reiniciar la MV para que se apliquen los cambios:

```
sudo nano /etc/host
```

```
sudo nano /etc/hostname
```

Comando para reiniciar la MV

```
sudo reboot
```

Y comprobamos los cambios con el siguiente comando `hostname`

```
miadmin@CGC-US ED:/$ hostname  
CGC-US ED
```

Actualizar sistema:

Para actualizar el sistema tendremos que aplicar el siguiente comando

```
sudo apt update && sudo apt upgrade
```

```
miadmin@CGC-US ED:/$ sudo apt update && sudo apt upgrade
```

Cuentas administración:

Creación de usuario:

Para crear un usuario se utiliza el comando adduser y seguimos los pasos que nos indica el propio comando: `sudo adduser miadmin2`

```
miadmin@CGC-US ED:/$ sudo adduser miadmin2
```

Para añadirlo al grado de super usuarios

Para añadir al usuario miadmin2 al grupo sudo hay que realizar el siguiente comando

```
sudo usermod -aG sudo miadmin2
```

```
miadmin@CGC-US ED:/$ sudo usermod -aG sudo miadmin2
```

APACHE

Instalar el servicio:

Para instalar el servicio de apache usamos el siguiente comando

```
sudo apt install apache2
```

```
miadmin@CGC-US ED:/$ sudo apt install apache2
```

Y comprobamos el estado de este para saber si este encendido o apagado

```
sudo service apache2 status
```

```
miadmin@CGC-USDED:/$ sudo service apache2 status
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2
  Active: active (running) since Thu 2023-10-
```

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: `sudo ufw enable`

```
miadmin@CGC-USDED:/$ sudo ufw enable
```

Después hay que abrir el puerto 80 , que es el puerto que usa apache por defecto, con:

```
sudo ufw allow apache
```

```
miadmin@CGC-USDED:/$ sudo ufw allow apache
Rule added
Rule added (v6)
```

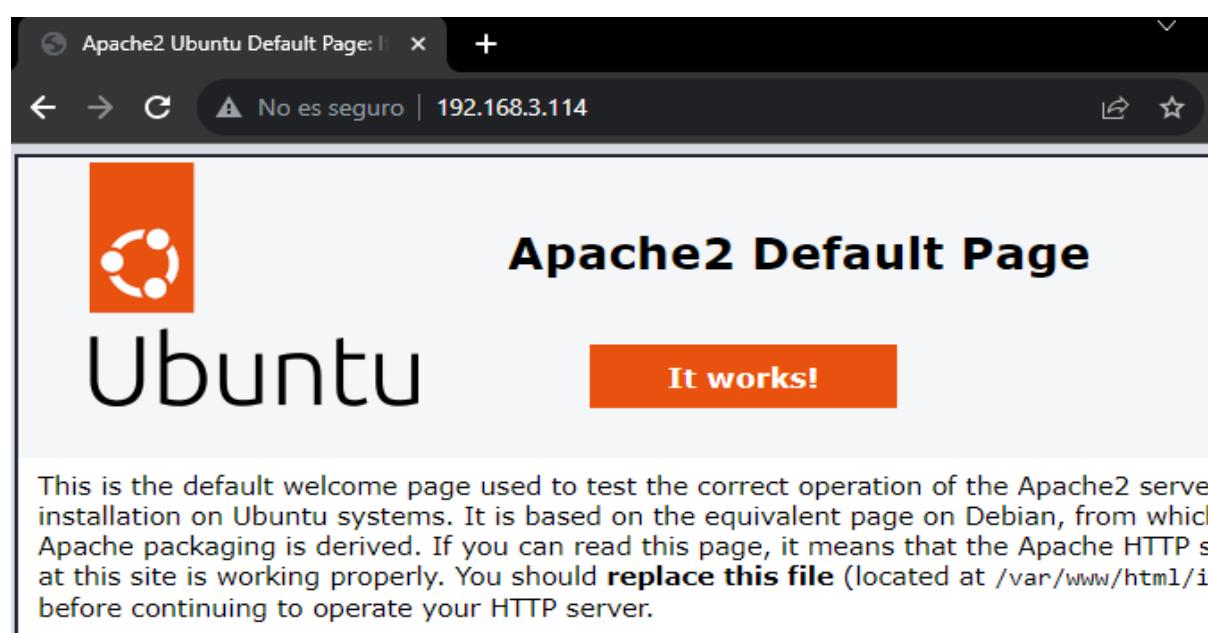
Y comprobamos si están abiertos los puertos en el cortafuegos

```
sudo ufw status
```

```
miadmin@CGC-USDED:/$ sudo ufw status
80/tcp                  ALLOW
Apache                 ALLOW
80/tcp (v6)             ALLOW
Apache (v6)             ALLOW
```

Comprobar conectividad con 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://192.168.3.214`) y nos tiene que salir lo siguiente.



Habilitar el uso del fichero .htaccess

Accedemos a la edición del siguiente archivo por medio del comando

sudo nano /etc/apache2/apache2.conf

```
miadmin@CGC-USDED:/$ sudo nano /etc/apache2/apache2.conf
```

Por defecto viene, AllowOverride None para el directorio /var/www

Y con “Ctrl+W” podemos usar el buscador del archivo y escribimos “/var/www” y luego pulsamos intro. (En este paso también dejamos configurado los indexes.)

Y deberemos copiarlo exactamente igual que en la captura

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

<Directory /var/www/>: Esto establece un bloque de configuración que se aplica al directorio /var/www/ en el servidor web. Todas las configuraciones entre <Directory /var/www/> y </Directory> se aplicarán a ese directorio específico.

Options -Indexes +FollowSymLinks: Define las opciones para el directorio. Significa que se desactiva la función de mostrar automáticamente el listado de archivos si no hay un archivo de índice en el directorio (-Indexes) y se permite el seguimiento de enlaces simbólicos (+FollowSymLinks).

AllowOverride All: Permite la anulación de configuraciones mediante archivos .htaccess en ese directorio. Esto permite a los usuarios locales del directorio personalizar algunas configuraciones para sus propios fines.

Require all granted: Esto establece que se permite el acceso a este directorio a cualquier usuario o cliente que intente acceder. Es una configuración de permisos que permite a cualquiera ver los contenidos en el directorio.

Luego deberemos crear un archivo en el directorio raíz de nuestro servidor que se llame '.htaccess'

sudo nano /var/www/html/.htaccess

```
miadmin@CGC-USDED:/$ sudo nano /var/www/html/.htaccess
```

Y añadimos esta línea

```
DirectoryIndex index.php index.html
```

La directiva 'DirectoryIndex' se utiliza en la configuración de un servidor web, como Apache, para especificar el nombre del archivo que se debe servir como página de inicio cuando se accede a un directorio en el servidor web.

Configurar que los logs de error y de acceso sean accesibles desde el navegador.

Entramos a editar el siguiente archivo con este comando

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

```
miadmin@CGC-USDED:/$ sudo nano /etc/apache2/sites-enabled/000-default.conf
```

Debajo de las 2 primeras líneas que viene por defecto añadimos las nuestras

```
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
```

Con estas líneas estamos indicando el nuevo directorio donde queremos guardar los mensajes de error 'ErrorLog /var/www/html/log/error.log' y en la siguiente indicamos donde queremos guardar el archivo que almacena los registros de acceso 'CustomLog /var/www/html/log/access.log'

Personalizar páginas de error de cada proyecto

Una vez tengamos el archivo .htaccess en funcionamiento, usando la directiva 'ErrorDocument' añadiendo la ruta o mensaje de los errores en ella, mi caso tengo personalizado el error 403 y 404, y voy a indicarle la ruta donde alojo los index de error:

```
sudo nano /var/www/html/.htaccess
```

```
miadmin@CGC-USDED:/$ sudo nano /var/www/html/.htaccess
```

```
ErrorDocument 403 /error/403.html
ErrorDocument 404 /error/404.html
```

Entonces cuando solicito una página que no existe me muestra la siguiente página:



Directiva Apache2

El primer paso que debemos tomar es comprobamos si esta la directiva configurada, para ello usamos: `sudo apache2ctl configtest`

```
miadmin@CGC-USDED:/$ sudo apache2ctl configtest
```

Si no lo tuviéramos configurado debemos editar el siguiente archivo

```
sudo nano /etc/apache2/apache2.conf
```

```
miadmin@CGC-USDED:/$ sudo nano /etc/apache2/apache2.conf
```

Y en la última línea deberemos añadir el 'hostname' (Comando que hemos usado anteriormente, al principio de la página 5.)

```
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
ServerName CGC-USDED
```

La directiva 'ServerName' se utiliza en la configuración de un servidor web, como Apache, para especificar el nombre de host o el dominio al que el servidor web debe responder.

Reiniciamos el servicio y volvemos a usar el comando del principio

```
sudo service apache2 restart
```

```
sudo apache2ctl configtest
```

```
miadmin@CGC-USDED:/ $ sudo service apache2 restart  
miadmin@CGC-USDED:/ $ sudo apache2ctl configtest  
Syntax OK
```

Informe de estado del servicio de apache

Instalamos un navegador de texto, en este caso lynx

```
sudo apt install lynx
```

```
miadmin@CGC-USDED:/ $ sudo apt install lynx
```

Ahora comprobamos que el módulo mod_status este activo

```
sudo a2enmod status
```

```
miadmin@CGC-USDED:/ $ sudo a2enmod status  
Module status already enabled
```

Y reiniciamos el servicio

```
sudo service apache2 restart
```

```
miadmin@CGC-USDED:/ $ sudo service apache2 restart  
.....
```

Con los siguientes comandos podemos pedir un informe completo o reducido

```
sudo apache2ctl fullstatus
```

```
sudo apache2ctl status
```

Directorios Virtuales

Comprobamos que podemos configurar directorios virtuales con el siguiente comando

```
ls /etc/apache2/mods-enabled/ | grep alias
```

```
miadmin@CGC-USDED:/ $ ls /etc/apache2/mods-enabled/ | grep alias  
alias.conf  
alias.load
```

(Si aparecen estos archivos estamos en buen camino.) Ahora editamos el siguiente archivo y añadimos lo que aparece en la captura

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

```
miadmin@CGC-USDED:/ $ sudo nano /etc/apache2/sites-enabled/000-default.conf
```

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

ServerAdmin webmaster@localhost: Esta línea especifica la dirección de correo electrónico del administrador del servidor web. Se utiliza para enviar notificaciones y mensajes de error al administrador del servidor en caso de problemas.

DocumentRoot /var/www/html: Esta línea establece la ruta del directorio raíz del sitio web. Todos los archivos y carpetas del sitio web se encuentran en el directorio "/var/www/html". Cuando alguien accede al servidor web, verá los archivos desde este directorio.

Alias /documentos /home/miadmin/doc: Aquí se crea un alias llamado "/documentos" que apunta al directorio "/home/miadmin/doc". Esto significa que cuando alguien accede a "/documentos" en el servidor web, en realidad está viendo el contenido del directorio "/home/miadmin/doc".

<Directory /home/miadmin/doc>: Esto establece las directivas de configuración específicas para el directorio "/home/miadmin/doc". Las siguientes líneas dentro de <Directory> se aplicarán solo a este directorio en particular.

Options +FollowSymLinks +Indexes: Aquí se establecen algunas opciones para el directorio "/home/miadmin/doc". "+FollowSymLinks" permite seguir enlaces simbólicos en el directorio, y "+Indexes" habilita la generación de listados de directorios si no hay un archivo de índice.

AllowOverride None: Esto impide la anulación de configuraciones desde archivos de configuración de nivel inferior (.htaccess) en este directorio. "None" significa que no se permiten anulaciones.

Require all granted: Aquí se especifica que todas las solicitudes de acceso a este directorio se les concede automáticamente. En otras palabras, cualquier persona que tenga acceso a este directorio tendrá permiso para ver su contenido.

Creamos un directorio en /home/miadmin/doc y dentro un archivo llamado 'file.txt'.

Primero debemos colocarnos en el 'home' y crear la carpeta desde dentro

```
cd /home
miadmin@CGC-US ED:/home$ sudo mkdir miadmin/doc
sudo mkdir miadmin/doc
```

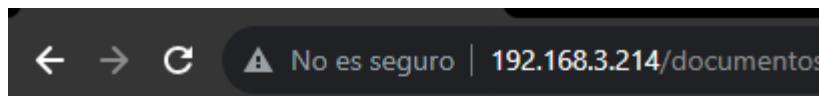
Nos colocamos en dentro del directorio '/doc' y con 'nano' creamos el archivo

```
cd miadmin/doc
miadmin@CGC-US ED:/home$ cd miadmin/doc
sudo nano file1.txt
miadmin@CGC-US ED:~/doc$ sudo nano file1.txt
miadmin@CGC-US ED:~/doc$ ls
file1.txt
```

Luego cambiamos los permisos del directorio '/home/miadmin'

```
sudo chmod 755 /home/miadmin miadmin@CGC-US-ED:/ $ sudo chmod 755 /home/miadmin
```

Y probamos la conexión desde un navegador



Index of /documentos

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
Parent Directory		-	
file1.txt	2023-10-06 09:23	8	

Ahora vamos a añadir 'Indexes' en este archivo para poder visualizar los iconos que vienen por defecto en apache, para tener acceso a ellos independientemente del directorio donde estemos usando el alias 'icons' :

```
sudo nano /etc/apache2/mods-available/alias.conf
```

Dentro de la etiqueta por defecto añadimos las siguientes líneas

```
Alias /icons/ "/usr/share/apache2/icons/"

<Directory "/usr/share/apache2/icons">
    Options FollowSymlinks
    AllowOverride None
    Require all granted
</Directory>
```

Alias /icons/ "/usr/share/apache2/icons/": Esta línea establece un alias que permite acceder a recursos ubicados en el directorio "/usr/share/apache2/icons/" utilizando la ruta "/icons/". En otras palabras, cuando alguien accede a "/icons/", en realidad está accediendo a los archivos en "/usr/share/apache2/icons/".

<Directory "/usr/share/apache2/icons">: Este bloque de configuración se aplica al directorio "/usr/share/apache2/icons". Dentro de este bloque, se establecen las directivas de configuración específicas para ese directorio.

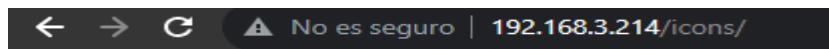
Options FollowSymlinks: Esto permite que se sigan los enlaces simbólicos dentro de este directorio.

AllowOverride None: Esto significa que no se permite la anulación de configuración (.htaccess) en este directorio.

Require all granted: Esto especifica que se permite el acceso a todos los usuarios (permisos "granted") a los recursos dentro de este directorio.

Reiniciamos el servicio y en la máquina cliente accedemos a su ip'/icons'

```
sudo service apache2 restart miadmin@CGC-US-ED:/ $ sudo service apache2 restart
```



Index of /icons

Name	Last modified	Size	Description
Parent Directory		-	
a.gif	2004-11-20 21:16	246	
a.png	2023-05-03 22:02	189	
..	2023-05-03 22:02	210	

Redirigir en Apache

Habilitamos el módulo con la siguiente línea de comando

`sudo a2enmod rewrite`

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

Y reiniciamos el servicio

`systemctl restart apache2`

```
miadmin@CGC-USDED:/$ systemctl restart apache2
```

Y con este comando mostramos los módulos cargados

`apache2ctl -M`

```
miadmin@CGC-USDED:/$ apache2ctl -M
```

Ahora en el archivo .htaccess añadimos esta línea (Es un ejemplo.)

`sudo nano /var/www/html/.htaccess`

```
miadmin@CGC-USDED:/$ sudo nano /var/www/html/.htaccess
```

```
|Redirect 301 /google https://www.google.es
```

La directiva ‘Redirect 301’ se utiliza para redirigir de manera permanente una URL a otra

Ahora si vamos al navegador de la máquina cliente y seguido de la ip ponemos ‘/google’, automáticamente nos redirigirá a la ruta indicada

SERVIDOR DNS

Instalación del servidor DNS

Comprobamos si el paquete está instalado

`aptitude show bind9`

```
miadmin@CGC-US ED:/$ aptitude show bind9
Command 'aptitude' not found, but can be installed with:
sudo apt install aptitude
miadmin@CGC-US ED:~$ aptitude search bind9
```

Ahora visualizamos el nombre del paquete con el siguiente comando

`dpkg -l | grep bind9`

ii bind9-dnsutils	1:9.18.12-0ubuntu0.22.04.3	amd64	Clients provided with BIN
ii bind9-host	1:9.18.12-0ubuntu0.22.04.3	amd64	DNS Lookup Utility
ii bind9-libs:amd64	1:9.18.12-0ubuntu0.22.04.3	amd64	Shared Libraries used by
BIND 9			

Una vez conocemos el nombre del paquete actualizamos el sistema

`sudo apt update -y`

```
miadmin@CGC-US ED:/$ sudo apt update -y
```

E instalamos el paquete 'bind9' y comprobamos el estado

`sudo apt install bind9 -y`

```
miadmin@CGC-US ED:/$ sudo apt install bind9 -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
bind9 ya está en su versión más reciente (1:9.18.18-0ubuntu0.22.04.1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 38 no actualizados.
```

`sudo service bind9 status`

```
miadmin@CGC-US ED:/etc/netplan$ sudo service bind9 status
● named.service - BIND Domain Name Server
  Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor
  Active: active (running) since Tue 2023-11-07 08:41:
    Docs: man:named(8)
   Main PID: 2063 (named)
      Tasks: 6 (limit: 3453)
     Memory: 6.5M
        CPU: 262ms
       CGroup: /system.slice/named.service
                 └─2063 /usr/sbin/named -u bind
```

Ahora debemos editar la configuración de red, que se aloja en la siguiente ruta, el archivo que tenga como extensión '.yaml' , en mi caso se llama '00-installer-config.yaml'

```
sudo nano /etc/netplan/00-installer-config.yaml
```

```
# This is the network config written by 'subiquity'  
network:  
  ethernets:  
    enp0s3:  
      addresses:  
        - 192.168.3.214/24  
      nameservers:  
        addresses:  
          - 192.168.3.214  
        search: [carlos.local]  
      routes:  
        - to: default  
          via: 192.168.3.1  
  version: 2
```

Le sustituimos el DNS anterior por nuestra propia 'IP' y en el apartado 'search: []' añadimos 'tuNombre.local', en mi caso 'carlos.local'.

Aplicamos los cambios

```
sudo netplan apply
```

```
miadmin@CGC-US ED:/etc/netplan$ sudo netplan apply
```

Y abrimos el puerto del bind9 que es el 53

```
sudo ufw allow 53
```

```
miadmin@CGC-US ED:/etc/netplan$ sudo ufw allow 53  
Rule added  
Rule added (v6)
```

Y hacemos ping para comprobar si funciona

```
miadmin@CGC-US ED:/etc/netplan$ ping www.google.es  
PING www.google.es (142.250.201.67) 56(84) bytes of data.  
64 bytes from 142.250.201.67: icmp_seq=1 ttl=114 time=13.6 ms  
64 bytes from 142.250.201.67: icmp_seq=2 ttl=114 time=13.8 ms  
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=3  
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=4  
^C  
--- www.google.es ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3971ms  
rtt min/avg/max/mdev = 13.579/13.699/13.849/0.118 ms
```

Declaración de zona directa

Ahora tendremos que modificar un archivo del bind9 añadiendo estas líneas

sudo nano /etc/bind/named.conf.local

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

/*
// Do any local configuration here
//

// Consider adding the 1918 zones here, if t
// organization
//include "/etc/bind/zones.rfc1918";
zone "carlos.local" {
    type master;
    file "/etc/bind/db.carlos.local";
};

type master;

file "/etc/bind/db.carlos.local";

}
```

Chequear ficheros de configuración

sudo named-checkconf

```
miadmin@CGC-USDED:~$ sudo named-checkconf
```

RR en zona directa

Hacemos una copia de seguridad de este archivo antes de modificarlo

sudo cp /etc/bind/db.local /etc/bind/db.carlos.local

```
miadmin@CGC-USDED:~$ sudo cp /etc/bind/db.local /etc/bind/db.carlos.local
```

Y editamos el archivo

sudo nano /etc/bind/db.carlos.local

```
miadmin@CGC-USDED:~$ sudo nano /etc/bind/db.carlos.local
```

Dentro de este archivo deberemos añadir ciertas líneas y modificar otras, nos tiene que quedar como la la captura

```
GNU nano 6.2                                     /etc/bind/db.carlos.local *
; BIND data for carlos.local
;
$TTL      604800
@       IN      SOA      cgc-used.carlos.local. marodriguez.educa.jcyl.es. (
                          1           ; Serial
                          604800      ; Refresh
                          86400       ; Retry
                          2419200     ; Expire
                          3600 )      ; Negative Cache TTL
;
@       IN      NS       cgc-used.carlos.local.
; Registros Host
@               IN A   192.168.3.214
cgc-used          IN A   192.168.3.214
; Registros Alias
daw201           IN CNAME   cgc-used.carlos.local.
daw202           IN CNAME   cgc-used.carlos.local.
daw214           IN CNAME   cgc-used.carlos.local.
```

```
;BIND data for carlos.local
;
$TTL 604800
@ IN SOA cgc-used.carlos.local. marodriguez.educa.jcyl.es. (
    1      ; Serial
    604800  ; Refresh
    86400   ; Retry
    2419200 ; Expire
    3600 )  ; Negative cache TTL
;
@           IN NS  cgc-used.carlos.local.
; Registros Host
@           IN A 192.168.43.214

cgc-used      IN A 192.168.43.214
; Registros Alias
daw201        IN CNAME cgc-used.carlos.local.
daw202        IN CNAME cgc-used.carlos.local.
daw214        IN CNAME cgc-used.carlos.local.
```

Y comprobamos que la sintaxis esta correcta

```
sudo named-checkzone carlos.local /etc/bind/db.carlos.local
```

```
miadmin@CGC-USDED:~$ sudo named-checkzone carlos.local /etc/bind/db.carlos.local
zone carlos.local/IN: loaded serial 1
OK
```

```
sudo service bind9 restart
```

Y reiniciamos el servicio

```
| miadmin@CGC-USDED:~$ sudo service bind9 restart
```

RR en zona inversa

Como medida de seguridad hacemos una copia de este archivo

```
sudo cp /etc/bind/db.carlos.local /etc/bind/db.3.168.192.in-addr.arpa
```

```
miadmin@CGC-USDED:~$ sudo cp /etc/bind/db.carlos.local /etc/bind/db.3.168.192.in-addr.arpa
```

Lo editamos y añadimos las siguientes líneas para que quede como en la siguiente captura

```
sudo nano /etc/bind/db.3.168.192.in-addr.arpa
```

```
;BIND data for 3.168.192.in-addr.arpa
;
$TTL 604800
@ IN SOA cgc-used.carlos.local. root.localhost. (
```

```
    1 ; Serial
        604800      ; Refresh
            86400      ; Retry
                2419200   ; Expire
                    3600 )   ; Negative cache TTL
                        ;
                            @ IN NS cgc-used.carlos.local.
;
                            ; registros Pointer o punteros
                            214      IN PTR cgc-used.carlos.local.
                            214      IN PTR carlos.local.
;
@ IN NS cgc-used.carlos.local.
; registros Pointer o punteros
214      IN PTR cgc-used.carlos.local.
214      IN PTR carlos.local.
```

Comprobamos la zona con el siguiente comando

```
sudo named-checkzone 214.168.192.in-addr.arpa /etc/bind/db.3.168.192.in-addr.arpa
```

```
miadmin@CGC-USDED:/etc/netplan$ sudo named-checkzone 214.168.192.in-addr.arpa /etc/bind/db.3.168.192.in-addr.arpa
zone 214.168.192.in-addr.arpa/IN: loaded serial 1
OK
```

Y volvemos a reiniciar el servicio

```
sudo service bind9 restart
```

```
miadmin@CGC-USDED:/etc/netplan$ sudo service bind9 restart
```

Comprobación de funcionamiento

Con el siguiente comando comprobamos el funcionamiento: **resolvectl status**

Con el siguiente comando comprobamos la configuración

```
miadmin@CGC-USDED:~$ resolvectl status

    Protocols: -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
resolv.conf mode: stub

    Current Scopes: DNS
    Protocols: +DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
Current DNS Server: 192.168.3.214
    DNS Servers: 192.168.3.214
    DNS Domain: carlos.local
```

Reenviador

Editamos el siguiente archivo para indicarle la IP de otros servidores DNS para que reenvíe las consultas que no es capaz de resolver.

sudo nano /etc/bind/named.conf.options

```
miadmin@CGC-USDED:~$ sudo nano /etc/bind/named.conf.options

GNU nano 6.2                                     /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you wan-
    // to talk to, you may need to fix the firewall to allow mult-
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses re-
    // the all-0's placeholder.

    forwarders {
        8.8.8.8;
        8.8.4.4;
    };

    //=====
    // If BIND logs error messages about the root key being expir-
    // you will need to update your keys. See https://www.isc.or
    //=====
    dnssec-validation auto;

    listen-on-v6 { any; };
};
```

Y chequeamos el archivo con el siguiente comando :

sudo named-checkconf

```
miadmin@CGC-USDED:~$ sudo named-checkconf
```

Luego comprobamos si resuelve los nombres con el comando ping, por ejemplo, a www.google.es

```
miadmin@CGC-USDED:~$ ping www.google.es
PING www.google.es (142.250.200.131) 56(84) bytes of data.
64 bytes from 142.250.200.131: icmp_seq=1 ttl=114 time=272 ms
64 bytes from 142.250.200.131: icmp_seq=2 ttl=114 time=271 ms
64 bytes from mad4ls14-in-f3.1e100.net (142.250.200.131): icmp_seq=3 ttl=114 time=313 ms
64 bytes from mad4ls14-in-f3.1e100.net (142.250.200.131): icmp_seq=4 ttl=114 time=299 ms
64 bytes from mad4ls14-in-f3.1e100.net (142.250.200.131): icmp_seq=5 ttl=114 time=286 ms
^C
--- www.google.es ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 10440ms
rtt min/avg/max/mdev = 270.580/287.908/312.721/16.035 ms
```

HTTPS

Para trabajar con un protocolo seguro de transferencia de hipertexto (HTTPS), deberemos configurar el protocolo SSL/TLS.

Como primer paso creamos la clave privada en el home del usuario

`openssl genrsa 2048 > clavePrivada.key`

`| miadmin@CGC-US ED:~$ openssl genrsa 2048 > clavePrivada.key`

Y el certificado en el mismo directorio

`openssl req -new -key clavePrivada.key > certificado.csr`

`| miadmin@CGC-US ED:~$ openssl req -new -key clavePrivada.key > certificado.csr`

Cuando usemos el comando de arriba nos irá pidiendo una serie de datos, unos obligatorios y otros opcionales

```
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.  
----  
Country Name (2 letter code) [AU]:ES  
State or Province Name (full name) [Some-State]:Zamora  
Locality Name (eg, city) []:Benavente  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IES Los Sauces  
Organizational Unit Name (eg, section) []:Informatica  
Common Name (e.g. server FQDN or YOUR name) []:daw214.carlos.local  
Email Address []:  
  
Please enter the following 'extra' attributes  
to be sent with your certificate request  
A challenge password []:  
An optional company name []:  
Nos pide el nombre del país.  
Nombre del estado o provincia.  
Nombre de la localidad.  
Nombre de la organización.  
Nombre de la unidad organizacional.  
Nombre del dominio.  
Correo.  
Contraseña.  
Nombre de la compañía (Opcional)
```

Con el siguiente comando firmaremos el certificado digital, aunque si no estuviéramos en el equipo de desarrollo, nos lo tendría que firmar una persona autorizada a ello.

`openssl x509 -req -days 365 -in certificado.csr -signkey clavePrivada.key > certificado.crt`

```
| miadmin@CGC-US ED:~$ openssl x509 -req -days 365 -in certificado.csr -signkey clavePrivada.key > certificado.crt  
Certificate request self-signature ok  
subject=C = ES, ST = Zamora, L = Benavente, O = IES Los Sauces, OU = Informatica, CN = daw214.carlos.local  
-----
```

'openssl' : Es el comando base de OpenSSL, una herramienta de línea de comandos para trabajar con criptografía y certificados.

'x509' : Indica que se realizará una operación relacionada con certificados X.509.

'-req' : Especifica que el archivo de entrada (certificado.csr) es una solicitud de certificado (CSR).

'-days 365' : Establece la vigencia del certificado en 365 días (1 año). Puedes ajustar este valor según tus necesidades.

'-in certificado.csr' : Especifica el nombre del archivo que contiene la solicitud de certificado.

'-signkey clavePrivada.key' : Indica el archivo que contiene la clave privada con la cual se firmará el certificado. La clave privada debe corresponder a la clave pública asociada con la solicitud de certificado.

'> certificado.crt' : Redirige la salida del comando hacia un archivo llamado certificado.crt. Este será el archivo de certificado resultante.

Ahora debemos instalar el modulo de SSL para poder usarlo.

sudo a2enmod ssl

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

Luego nos pide reiniciar el servicio de apache con el comando

systemctl restart apache2

Ahora debemos de seguir una serie de pasos para configurar de manera correcta los ficheros y archivos para el protocolo.

sudo mv clavePrivada.key /etc/ssl/private

```
miadmin@CGC-USDED:~$ sudo mv clavePrivada.key /etc/ssl/private
```

Este comando mueve el archivo clavePrivada.key al directorio /etc/ssl/private. En sistemas Linux, este es un lugar común para almacenar claves privadas utilizadas por servicios seguros, como el servidor web.

sudo chown root:ssl-cert /etc/ssl/private/clavePrivada.key

```
miadmin@CGC-USDED:~$ sudo chown root:ssl-cert /etc/ssl/private/clavePrivada.key
root@CGC-USDED: ~%
```

Cambia el propietario y el grupo del archivo clavePrivada.key a root y ssl-cert, respectivamente. Esto generalmente se hace para asegurar que solo usuarios específicos tengan acceso a la clave privada.

```
sudo ls -l /etc/ssl/private/clavePrivada.key
```

```
miadmin@CGC-USDED:~$ sudo ls -l /etc/ssl/private/clavePrivada.key
-rw-rw-r-- 1 root ssl-cert 1704 nov 24 08:49 /etc/ssl/private/clavePrivada.key
```

Listamos los detalles del archivo clavePrivada.key, incluyendo el propietario, el grupo y los permisos.

```
sudo chmod 640 /etc/ssl/private/clavePrivada.key
```

```
miadmin@CGC-USDED:~$ sudo chmod 640 /etc/ssl/private/clavePrivada.key
```

Cambia los permisos del archivo clavePrivada.key para que solo el propietario tenga permisos de lectura y escritura, y el grupo tenga permisos de lectura.

```
sudo ls -l /etc/ssl/private/clavePrivada.key
```

```
miadmin@CGC-USDED:~$ sudo ls -l /etc/ssl/private/clavePrivada.key
-rw-r----- 1 root ssl-cert 1704 nov 24 08:49 /etc/ssl/private/clavePrivada.key
```

Lista nuevamente los detalles del archivo clavePrivada.key después de cambiar los permisos.

```
sudo mv certificado.crt /etc/ssl/certs
```

```
miadmin@CGC-USDED:~$ sudo mv certificado.crt /etc/ssl/certs
```

Mueve el archivo certificado.crt al directorio /etc/ssl/certs. Al igual que con las claves privadas, este directorio es comúnmente utilizado para almacenar certificados en sistemas Linux.

```
sudo chown root:root /etc/ssl/certs/certificado.crt
```

```
miadmin@CGC-USDED:~$ sudo chown root:root /etc/ssl/certs/certificado.crt
```

Cambia el propietario y el grupo del archivo certificado.crt a root. Esto se hace para asegurar que el archivo sea accesible solo por usuarios privilegiados.

```
sudo ls -l /etc/ssl/certs/certificado.crt
```

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

Lista los detalles del archivo certificado.crt, incluyendo el propietario, el grupo y los permisos después de los cambios realizados.

Desde la siguiente ruta

```
cd /etc/apache2/sites-available
```

Realizamos una copia de seguridad del siguiente fichero

```
sudo cp default-ssl.conf default-ssl.conf.backup
```

```
miadmin@CGC-USDED:/etc/apache2/sites-available$ sudo cp default-ssl.conf default-ssl.conf.backup
```

Y editamos lo siguiente, haciendo uno de este comando

sudo nano default-ssl.conf

```
| miadmin@CGC-USDED:/etc/apache2/sites-available
|   GNU nano 6.2                                     default-ssl.conf *
| <IfModule mod_ssl.c>
|     <VirtualHost *:443>
|       ServerAdmin webmaster@localhost
|       ServerName daw214.carlos.local
|       DocumentRoot /var/www/html
|
| # SSL Engine Switch:
| # Enable/Disable SSL for this virtual host.
| SSLEngine on
|
| # A self-signed (snakeoil) certificate can be created by installing
| # the ssl-cert package. See
| # /usr/share/doc/apache2/README.Debian.gz for more info.
| # If both key and certificate are stored in the same file, only the
| # SSLCertificateFile directive is needed.
| SSLCertificateFile      /etc/ssl/certs/certificado.crt
| SSLCertificateKeyFile  /etc/ssl/private/clavePrivada.key
|
| ErrorLog ${APACHE_LOG_DIR}/errordaw214-ssl.log
| CustomLog ${APACHE_LOG_DIR}/accessdaw214-ssl.log combined
```

Luego para comprobar que lo hemos editado de manera correcta utilizamos el comando

sudo apache2ctl configtest

```
| miadmin@CGC-USDED:/etc/apache2/sites-available$ sudo apache2ctl configtest
| Syntax OK
| . . . . .
```

Una vez confirmado que esta la sintaxis correcta, y no nos aparece ningún error, habilitamos el uso del archivo

sudo a2ensite default-ssl.conf

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

Y nos pedirá que recarguemos el servicio de apache

sudo systemctl reload apache2

```
| miadmin@CGC-USDED:/etc/apache2/sites-available$ sudo systemctl reload apache2
| miadmin@CGC-USDED:/etc/apache2/sites-available$
```

Ahora debemos habilitar el puerto 443 que es el que usa 'HTTPS', para ello comprobamos si está habilitado en el cortafuegos

sudo ufw status

```
miadmin@CGC-US ED:/etc/apache2/sites-available$ sudo ufw status
Status: active

To                         Action      From
--                         --          --
22                         ALLOW       Anywhere
80/tcp                      ALLOW       Anywhere
20/tcp                      ALLOW       Anywhere
21/tcp                      ALLOW       Anywhere
9003                        ALLOW       Anywhere
3306                        ALLOW       Anywhere
53                          ALLOW       Anywhere
22 (v6)                     ALLOW       Anywhere (v6)
80/tcp (v6)                 ALLOW       Anywhere (v6)
20/tcp (v6)                 ALLOW       Anywhere (v6)
21/tcp (v6)                 ALLOW       Anywhere (v6)
9003 (v6)                   ALLOW       Anywhere (v6)
3306 (v6)                   ALLOW       Anywhere (v6)
53 (v6)                     ALLOW       Anywhere (v6)
```

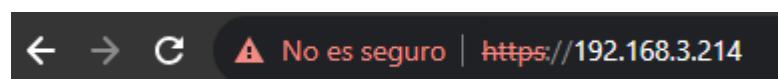
Como no aparece lo habilitaremos usando
sudo ufw allow 443

```
miadmin@CGC-US ED:/etc/apache2/sites-available$ sudo ufw allow 443
Rule added
Rule added (v6)
```

Y reiniciamos el servicio de apache
sudo service apache2 restart

```
miadmin@CGC-US ED:/etc/apache2/sites-available$ sudo service apache2 restart
```

Ahora cuando accedemos a nuestra IP como aún no nos han certificado no aparece de esta manera



Ahora mostraremos una lista de los Virtual Hosts configurados en un servidor web Apache con el siguiente comando

sudo apache2ctl -S

```
miadmin@CGC-US ED:/etc/apache2/sites-available$ sudo apache2ctl -S
VirtualHost configuration:
*:80                  CGC-US ED (/etc/apache2/sites-enabled/000-default.conf:1)
*:443                 daw214.carlos.local (/etc/apache2/sites-enabled/default-ssl.conf:2)
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"
Mutex watchdog-callback: using_defaults
Mutex rewrite-map: 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) daw214.carlos.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 domino de la MV

```
miadmin@CGC-USDED:/etc/apache2/sites-available
GNU nano 6.2
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.

    ServerName daw214.carlos.local
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    Alias /documentos /home/miadmin/doc
    <Directory /home/miadmin/doc>
        Options +FollowSymLinks +Indexes
        AllowOverride None
        Require all granted
    </Directory>

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

    ErrorLog ${APACHE_LOG_DIR}/errordaw214.log
    ErrorLog /var/www/html/log/error.log
    CustomLog ${APACHE_LOG_DIR}/accessdaw214.log combined
    CustomLog /var/www/html/log/access.log combined
```

Ahora verificaremos el archivo con uno de los comando que hemos usado anteriormente

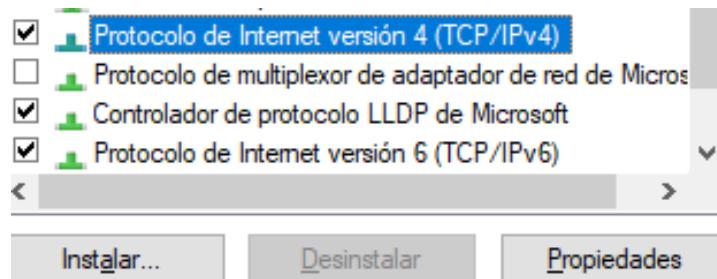
sudo apache2ctl configtest

Y volvemos a mostrar la lista de los Virtual Hosts

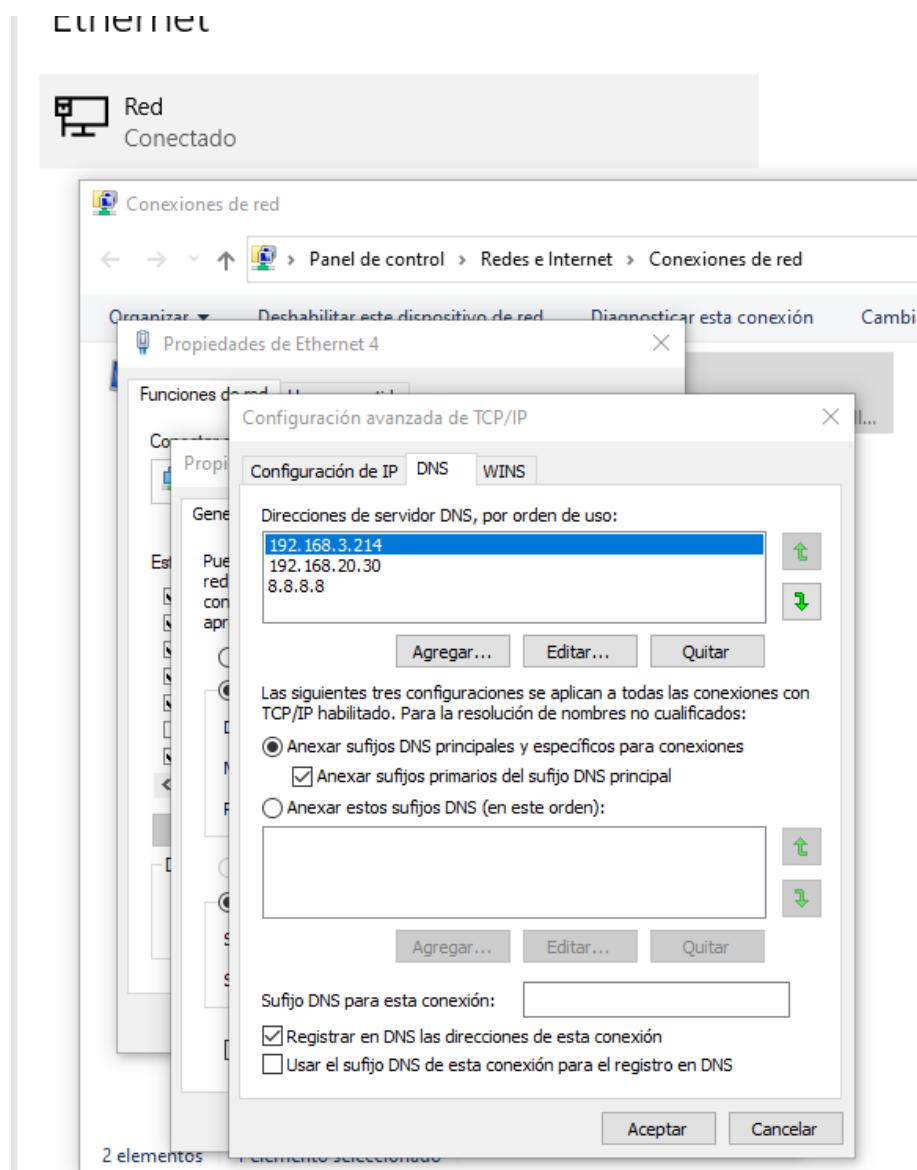
sudo apache2ctl -S

```
miadmin@CGC-USDED:/etc/apache2/sites-available$ sudo apache2ctl configtest
Syntax OK
miadmin@CGC-USDED:/etc/apache2/sites-available$ sudo apache2ctl -S
VirtualHost configuration:
*:80          daw214.carlos.local (/etc/apache2/sites-enabled/000-default.conf:1)
*:443         daw214.carlos.local (/etc/apache2/sites-enabled/default-ssl.conf:2)
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"
Mutex rewrite-map: 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
Mutex watchdog-callback: 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
```

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

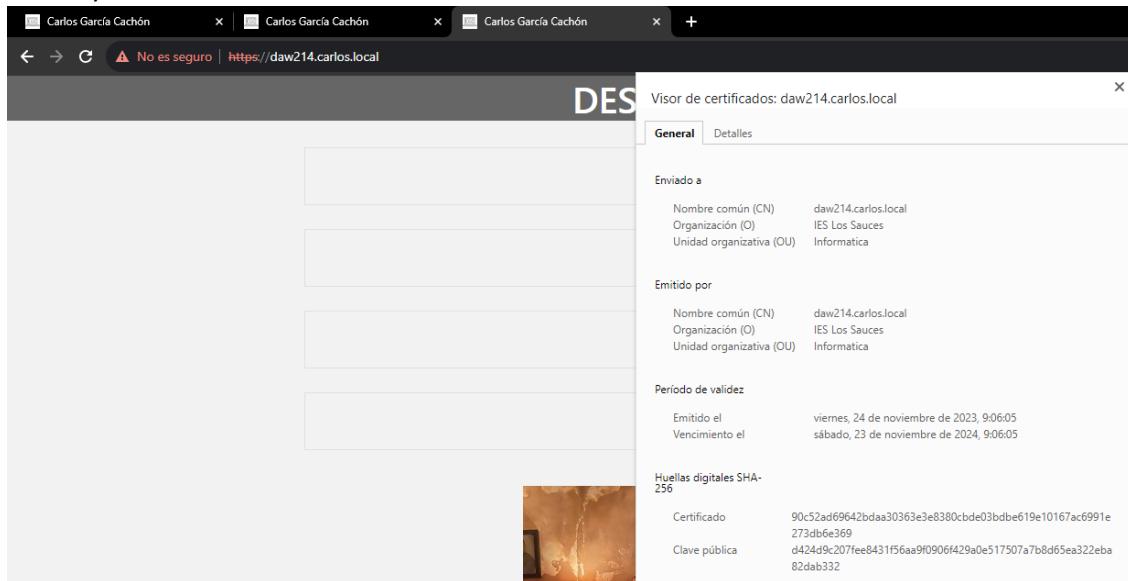


Después de este paso debemos editar el archivo ‘.htaccess’ para redireccionar de ‘HTTP’ a ‘HTTPS’ , incluimos lo siguiente

```
RewriteEngine On  
RewriteCond %{SERVER_PORT} 80  
RewriteRule ^(.*)$ https://daw214.carlos.local/\$1 [R,L]
```

1. **RewriteEngine On:** Esta línea activa el motor de reescritura de Apache. Indica que se va a utilizar la funcionalidad de reescritura de URL para procesar las reglas que siguen a continuación.
2. **RewriteCond %{SERVER_PORT} 80:** Esta línea establece una condición para la regla de reescritura que sigue. En este caso, la condición es que el puerto del servidor sea 80.
3. **RewriteRule ^(.*)\$ [https://daw214.carlos.local/\\$1](https://daw214.carlos.local/$1) [R,L]:** Esta es la regla de reescritura en sí. Si la condición anterior se cumple (es decir, si el puerto del servidor es 80), entonces esta regla reescribirá la URL.
 - **^(.*)\$:** Esto es una expresión regular que coincide con cualquier cadena (cualquier URL en este contexto). La parte entre paréntesis captura esta cadena para ser usada más adelante.
 - **[https://daw214.carlos.local/\\$1](https://daw214.carlos.local/$1):** Esta es la URL de destino a la que se redirige. **\$1** es la variable que contiene la cadena capturada por la expresión regular en la parte izquierda. En otras palabras, redirige a la misma URL pero utilizando el protocolo HTTPS en lugar de HTTP.
 - **[R,L]:** Estos son los indicadores de la regla de reescritura:
 - **[R]:** Indica que la redirección debe realizarse (es un redireccionamiento externo).
 - **[L]:** Indica que esta es la última regla que se debe aplicar si la condición se cumple.

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



En la captura anterior podemos ver el certificado de nuestra página 'HTTPS' y distinta información:

1. **Nombre común (CN): daw214.carlos.local**
 - Esto es el nombre del dominio para el cual se emitió el certificado. En este caso, el certificado es válido para el dominio "daw214.carlos.local".
2. **Organización (O): IES Los Sauces**
 - Indica la organización a la que pertenece el propietario del certificado. En este caso, pertenece a la organización "IES Los Sauces".
3. **Unidad organizativa (OU): Informatica**
 - Especifica la unidad organizativa a la que pertenece el propietario del certificado. En este caso, está asociado con la unidad organizativa "Informatica".
4. **Emitido el (Issued on): viernes, 24 de noviembre de 2023, 9:06:05**
 - La fecha y hora en que se emitió el certificado.
5. **Vencimiento el (Expires on): sábado, 23 de noviembre de 2024, 9:06:05**
 - La fecha y hora en que el certificado expirará. Después de esta fecha, el certificado ya no será válido y se deberá renovar.
6. **Certificado: 90c52ad69642bdaa30363e3e8380cbde03bdb619e10167ac6991e273db6e369**
 - Es el identificador único (hash) del certificado. Cada certificado tiene su propio hash único.
7. **Clave pública (Public Key):**
d424d9c207fee8431f56aa9f0906f429a0e517507a7b8d65ea322eba82dab332
 - Es la clave pública asociada con el certificado. La clave pública se utiliza en el proceso de establecimiento de conexión segura para cifrar la comunicación.

PHP

Instalación de PHP:

Vamos a instalar la versión 8.1

```
sudo apt -y install php8.1
```

```
miadmin@CGC-US ED:/$ sudo apt -y install php8.1
```

Y para comprobar la versión instalada usamos

```
php -version
```

```
miadmin@CGC-US ED:/$ php -version  
PHP 8.1.2-1ubuntu2.14 (cli) (buil
```

Instalación del módulo interprete:

Para instalar el intérprete utilizaremos

```
sudo apt -y install libapache2-mod-php
```

```
miadmin@CGC-US ED:/$ sudo apt -y install libapache2-mod-php
```

Comprobación de PHP:

Iremos al directorio /var/www/html y crearemos un archivo que se llame "index.php" y dentro pondremos esto:

Crearemos un archivo con extensión php para comprobar que funciona correctamente

```
sudo nano /var/www/html/index.php
```

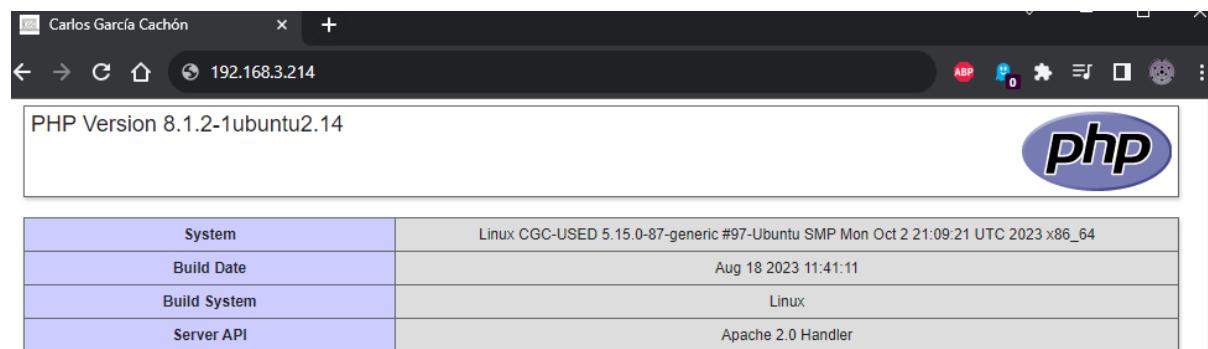
```
miadmin@CGC-US ED:/$ sudo nano /var/www/html/index.php
```

y añadimos el contenido de la siguiente captura

```
GNU nano 6.2                               /var/www/html/index.php
<?php
phinfo();
?>
```

Este código PHP es una función que muestra información detallada sobre la configuración de PHP y el entorno del servidor web

Como con anterioridad al crear el archivo '.htaccess' le pusimos como prioridad 'index.php' aunque tengamos un archivo con extensión '.html' no cogerá el nuevo archivo que hemos creado, entonces desde la máquina cliente, escribiremos la IP del servidor y se nos abrirá el archivo:



Configuración php.ini

Antes de modificar el archivo creamos una copia de seguridad

```
sudo cp /etc/php/8.1/apache2/php.ini php.ini.backup
```

```
miadmin@CGC-USDED:/$ sudo cp /etc/php/8.1/apache2/php.ini php.ini.backup
```

Luego entramos en el archivo, editamos esas variables y reiniciamos el servidor de apache

```
sudo nano /etc/php/8.1/apache2/php.ini
```

```
; Development Value: On  
; Production Value: Off  
; https://php.net/display-er  
display_errors = On  
  
; The display of errors which  
; separately from display_errors  
; for production servers to  
; Default Value: On  
; Development Value: On  
; Production Value: Off  
; https://php.net/display-st  
display_startup_errors = On  
  
; Maximum amount of memory a script  
; https://php.net/memory-limit  
memory_limit = 256M
```

display_errors = On : Es para que PHP muestre los errores directamente en el navegador cuando ocurran.

display_startup_errors = On : Determina si los errores ocurridos durante el proceso de inicio de PHP (como cuando se inicia un script) deben mostrarse en la pantalla o no, en este caso si.

memory_limit = 256M : Establece el límite de memoria disponible para los scripts PHP que se ejecutan en el servidor. En este caso, se está configurando un límite de 256 MB

Y reiniciamos el servicio

```
systemctl restart apache2
```

```
miadmin@CGC-USDED:/$ systemctl restart apache2
```

XDEBUG

Instalar directivas Xdebug

Comprobamos si lo tenemos

```
php -m | grep xdebug
```

```
miadmin@CGC-USDED:~$ php -m | grep xdebug
```

Si no devuelve nada buscamos el nombre del paquete

```
apt search php8.1 | grep xdebug
```

```
miadmin@CGC-USDED:~$ apt search php8.1 | grep xdebug
```

WARNING: apt does not have a stable CLI interface. Use at your own risk.

```
php8.1-xdebug/jammy, now 3.1.2+2.9.8+2.8.1+2.5.5-4 arm64 [installed]
```

Una vez encontrado el nombre lo instalamos

```
sudo apt install php8.1-xdebug
```

```
miadmin@CGC-USDED:~$ sudo apt install php8.1-xdebug
```

Entramos en el siguiente directorio y comprobamos si existe el siguiente archivo '20-xdebug.ini'

```
cd /etc/php/8.1/apache2/conf.d
```

```
ls
```

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

Accedemos al archivo y escribimos las líneas de la captura

```
sudo nano 20-xdebug.ini
```

```
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
```

zend_extension=xdebug.so: Esta línea carga el módulo Xdebug en PHP como una extensión (mediante el archivo compartido xdebug.so).

xdebug.discover_client_host=1: Habilita la detección automática de la dirección IP del cliente que solicita la depuración. Esto puede ser útil para conectar la depuración desde un entorno de desarrollo remoto.

xdebug.mode=debug: Establece el modo de depuración de Xdebug en "debug", lo que permite la depuración de código PHP.

xdebug.client_host=localhost: Define el host del cliente para la depuración como "localhost". Esto significa que la depuración se realizará desde la misma máquina donde se ejecuta el servidor PHP.

xdebug.client_port=9003: Especifica el puerto en el que Xdebug escuchará las solicitudes de depuración. El valor 9003 es el puerto predeterminado.

xdebug.idekey="netbeans-xdebug": Define una clave de identificación del IDE (Entorno de Desarrollo Integrado) que se utilizará para establecer la conexión entre el cliente de depuración y el servidor PHP. En este caso, se establece la clave en "netbeans-xdebug".

xdebug.show_error_trace=1: Habilita la visualización de la traza de errores (error trace) cuando ocurren errores en PHP.

xdebug.remote_autostart=on: Habilita la activación automática de la depuración en cada solicitud, lo que significa que la depuración se iniciará automáticamente en cada ejecución de script PHP.

xdebug.start_with_request=yes: Esta configuración asegura que Xdebug se inicie automáticamente con cada solicitud.

El siguiente paso sería abrimos el puerto 9003 y comprobar si está abierto

sudo ufw allow 9003

```
miadmin@CGC-USDED:/$ sudo ufw allow 9003
Skipping adding existing rule
Skipping adding existing rule (v6)
```

sudo ufw status

```
miadmin@CGC-USDED:/$ sudo ufw status
9003                         ALLOW
9003 (v6)                     ALLOW
```

MYSQL

Instalación MySQL

Usamos el siguiente comando para descargar e instalar

```
sudo apt install mysql-server
```

```
miadmin@CGC-USDED:~$ sudo apt install mysql-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
mysql-server ya está en su versión más reciente (8.0.34-0ubuntu0.22.04.1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 9 no actualizados.
```

Y comprobamos la versión instalada (Si no especificamos la versión al instalar nos descargará la última versión estable.)

```
sudo mysql --version
```

```
miadmin@CGC-USDED:~$ sudo mysql --version
mysql Ver 8.0.34-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
```

Hacemos una copia de seguridad de este archivo

```
sudo cp /etc/mysql/mysql.conf.d/mysqld.cnf mysqld.cnf.backup
```

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

Luego accedemos al archivo y comentamos estas 2 líneas añadiendo '#' delante de cada una.

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

bind-address se refiere a la dirección IP a la que MySQL escuchará para las conexiones tradicionales de MySQL.

mysqlx-bind-address se refiere a la dirección IP a la que MySQL X Plugin escuchará para las conexiones relacionadas con MySQL X Protocol (un protocolo utilizado para aplicaciones modernas)

Y reiniciamos el servicio

```
sudo service mysql restart
```

```
miadmin@CGC-USDED:/ $ sudo service mysql restart
```

Luego listamos los puertos que están escuchando el servidor y comprobamos que aparezca el '3306'

ss -punta

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
udp	UNCONN	0	0	127.0.0.53%lo:53	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	192.168.43.214:22	192.168.43.16:40998	
tcp	LISTEN	0	70	*:33060	*:*	
tcp	LISTEN	0	151	*:3306	*:*	
tcp	LISTEN	0	511	*:80	*:*	
tcp	LISTEN	0	128	[::]:22	[::]:*	

Y abrimos el puerto en el cortafuegos

sudo ufw allow 3306

```
miadmin@CGC-USDED:/$ sudo ufw allow 3306
Rule added
Rule added (v6)
```

sudo ufw status

```
miadmin@CGC-USDED:/$ sudo ufw status
3306                         ALLOW
3306 (v6)                     ALLOW
```

Script de Seguridad

Ahora configuramos el script de seguridad

sudo mysql_secure_installation

Nos pregunta si queremos validar la contraseña y en qué nivel de seguridad queremos elegir, en este caso le decimos que sí y el nivel 0, luego si queremos eliminar usuarios anónimos, restablecer root de manera remota, eliminar la base de datos test y restablecer los privilegios de las tablas

```
Securing the MySQL server deployment.
Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?
Press y|Y for Yes, any other key for No: Y

There are three levels of password validation policy:
LOW   Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary

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 "#"
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-manage

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!
```

Ahora descargamos el módulo de la librería de apache, de PHP para MYSQL

```
sudo apt install libapache2-mod-php8.1 php8.1-mysql
miadmin@CGC-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
libapache2-mod-php8.1 ya está en su versión más reciente (8.1.2-1ubuntu2.14)
php8.1-mysql ya está en su versión más reciente (8.1.2-1ubuntu2.14).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 9 no actualizados.
```

Antes del siguiente paso vamos a encontrarnos con un problema, y que como queremos que nuestro usuarios tenga una contraseña que no cumple las políticas de contraseña (En nuestro caso LOW, que especifica un mínimo de 8 caracteres), deberemos editar las variables de contraseña de MySQL (volviendo a dejarlas como estaban anteriormente por seguridad.) Para ello iniciamos Mysql

`sudo mysql`

Ahora mostramos las variables de MySQL para saber el nombre del valor que necesitamos modificar (En nuestro caso con modificar la longitud es suficiente)

`SHOW VARIABLES LIKE 'validate_password%';`

```
mysql> SHOW VARIABLES LIKE 'validate_password%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| validate_password.changed_characters_percentage | 0      |
| validate_password.check_user_name | ON     |
| validate_password.dictionary_file |        |
| validate_password.length | 8      |
| validate_password.mixed_case_count | 1      |
| validate_password.number_count | 1      |
| validate_password.policy | LOW    |
| validate_password.special_char_count | 1      |
+-----+-----+
```

`validate_password.changed_characters_percentage`: Esta variable determina cuántos caracteres deben cambiar al definir o cambiar una contraseña con respecto a la contraseña anterior. Un valor de 0 significa que no se requiere un cambio en los caracteres.

`validate_password.check_user_name`: Cuando está habilitado (ON), esta variable requiere que la contraseña no sea igual al nombre de usuario. Es decir, impide que se utilice el nombre de usuario como contraseña.

`validate_password.dictionary_file`: Esta variable especifica el archivo que se utilizará como diccionario para verificar contraseñas. Si está en blanco, no se utiliza un diccionario.

`validate_password.length`: Esta variable establece la longitud mínima que debe tener una contraseña.

`validate_password.mixed_case_count`: Establece el número mínimo de caracteres en mayúscula (letras mayúsculas) que debe contener una contraseña.

`validate_password.number_count`: Establece el número mínimo de dígitos (números) que debe contener una contraseña.

`validate_password.policy`: Define el nivel de la política de contraseñas. Los valores comunes son "LOW," "MEDIUM," o "STRONG," lo que determina la combinación de requisitos para las contraseñas. Un nivel "LOW" permite contraseñas más simples, mientras que un nivel "STRONG" requiere contraseñas más complejas.

`validate_password.special_char_count`: Establece el número mínimo de caracteres especiales (como símbolos) que debe contener una contraseña.

Buscamos el nombre de la variable que indica la longitud de la contraseña y con el siguiente comando cambiamos su valor (En este caso a 4, que es la longitud de la contraseña que queremos.)

```
SET GLOBAL validate_password.length = 4;
```

Y volvemos a comprobar con el comando anterior para verificarlo

```
SHOW VARIABLES LIKE 'validate_password%';
```

```
mysql> SHOW VARIABLES LIKE 'validate_password%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| validate_password.changed_characters_percentage | 0      |
| validate_password.check_user_name | ON     |
| validate_password.dictionary_file |        |
| validate_password.length | 4      |
+-----+-----+
```

Ahora ya podemos creamos el nuevo usuario y le darle permisos de super usuario

```
CREATE USER 'adminsql'@'%' IDENTIFIED BY 'paso';
```

```
GRANT ALL PRIVILEGES ON *.* TO 'adminsql'@'%' WITH GRANT OPTION;
```

```
mysql> CREATE USER 'adminsql'@'%' IDENTIFIED BY 'paso';
Query OK, 0 rows affected (0,01 sec)
```

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

Después de crear el usuario por seguridad volveremos a poner la variable a su valor por defecto (Que en nuestro caso, al ser LOW, sería de 8).

```
SET GLOBAL validate_password.length = 8;
```

```
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      |
+-----+-----+
```

Salimos de MySQL con: `exit`

Y comprobamos si accede el nuevo usuario creado

(-u : Este parámetro sirve para indicarle que queremos acceder con el siguiente usuario indicado.

-p : Para que nos pida la contraseña y podamos verificarla.)

`mysql -u admindb -p`

```
miadmin@CGC-US ED:/$ mysql -u admins q -p
Enter password:
Welcome to the MySQL monitor. Commands ? (for help), 
Your MySQL connection id is 25
```

Como curiosidad podemos ver los métodos de encriptado que tiene cada autentificación

`SELECT user, authentication_string, plugin, host FROM mysql.user;`

```
mysql> SELECT user, authentication_string, plugin, host FROM mysql.user;
+-----+-----+-----+-----+
| user | authentication_string | plugin | host |
+-----+-----+-----+-----+
| adminsql | $A$005$zB@^Rt@Zn@|[ C@gnL55Gx3PeL1HX1QxkBht/w2/ykPrKjADLxmAdE2vDfnh9 | caching_sha2_password | % |
| debian-sys-maint | $A$005${Eio@Bp2^E&j7GUpg86orVmikOexnIVR0bljwFxIhFlDHTK/CqRQZ1 | caching_sha2_password | localhost |
| mysql.infoschema | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBEUSED | caching_sha2_password | localhost |
| mysql.session | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBEUSED | caching_sha2_password | localhost |
| mysql.sys | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBEUSED | caching_sha2_password | localhost |
| root | auth_socket | localhost |
+-----+-----+-----+-----+
```

Logs y Errores

Con este comando podemos comprobar las últimas 200 líneas dentro del archivo '.log'

`tail -n200 /var/log/mysql/error.log | grep ERROR`

```
miadmin@CGC-US ED:/$ tail -n200 /var/log/mysql/error.log | grep ERROR
```

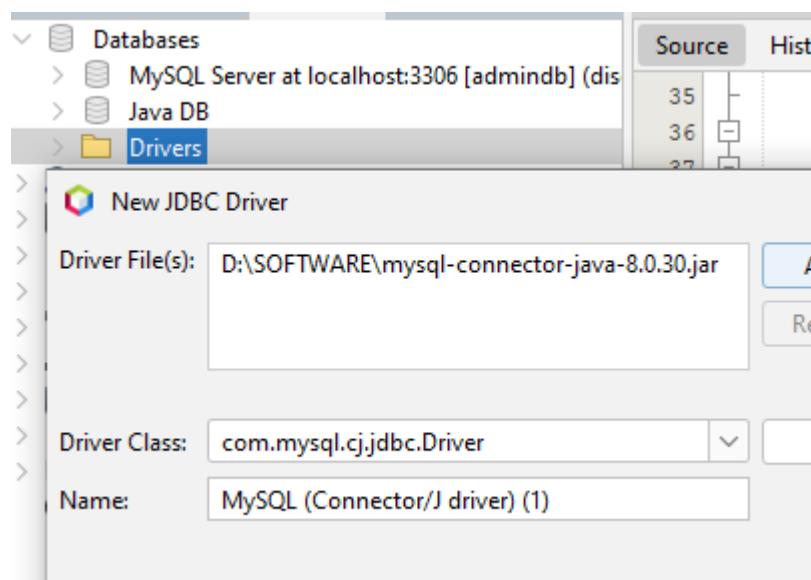
Y con este limpiaríamos el contenido del archivo en caso de que fuese demasiado grande o quisieramos hacer mantenimiento

`mysqladmin -u admindb -p flush-logs error`

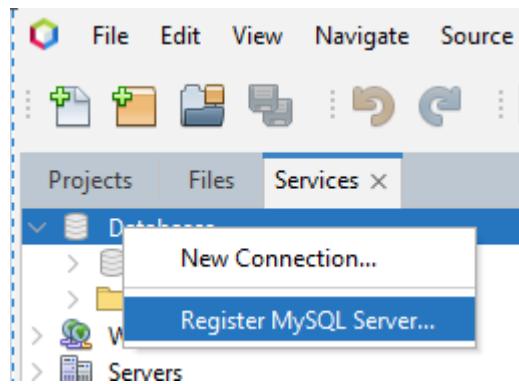
```
miadmin@CGC-US ED:/$ mysqladmin -u admindb -p flush-logs error
```

Conexión con NetBeans a MySQL

Primero añadimos el driver de MySQL en esta carpeta



El siguiente paso, nos vamos a la pestaña 'Services' y con click derecho sobre 'Databases' seleccionamos la última opción



Y nos aparece esta ventana donde introducimos la información de conexión

Basic Properties		Admin Properties
Server Host Name:	192.168.3.214	
Server Port Number:	3306	
Administrator User Name:	admindb	
Administrator Password:	*****	
<input checked="" type="checkbox"/> Remember Password		

A screenshot of the 'Basic Properties' tab in the MySQL Workbench connection configuration dialog. It shows fields for Server Host Name (192.168.3.214), Server Port Number (3306), Administrator User Name (admindb), and Administrator Password (represented by five asterisks). There is also a checked checkbox for 'Remember Password'. An 'Admin Properties' tab is visible but not selected.

PHPMYADMIN

Verificación de los módulos PHP

Primero mostraremos los módulos ya instalados con el siguiente comando

```
dpkg --get-selections | grep php
```

```
miadmin@CGC-USC:~$ dpkg --get-selections | grep php
libapache2-mod-php          install
libapache2-mod-php8.1         install
php-common                   install
php8.1                       install
php8.1-cli                   install
php8.1-common                install
php8.1-mysql                 install
php8.1-opcache               install
php8.1-readline              install
php8.1-soap                  install
php8.1-xdebug                install
php8.1-xml                   install
```

Instalación phpMyAdmin

Utilizamos el siguiente comando para instalar phpMyAdmin y los siguientes módulos que necesitaremos más adelante

```
sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```

```
miadmin@CGC-USC:~$ sudo apt -y install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```

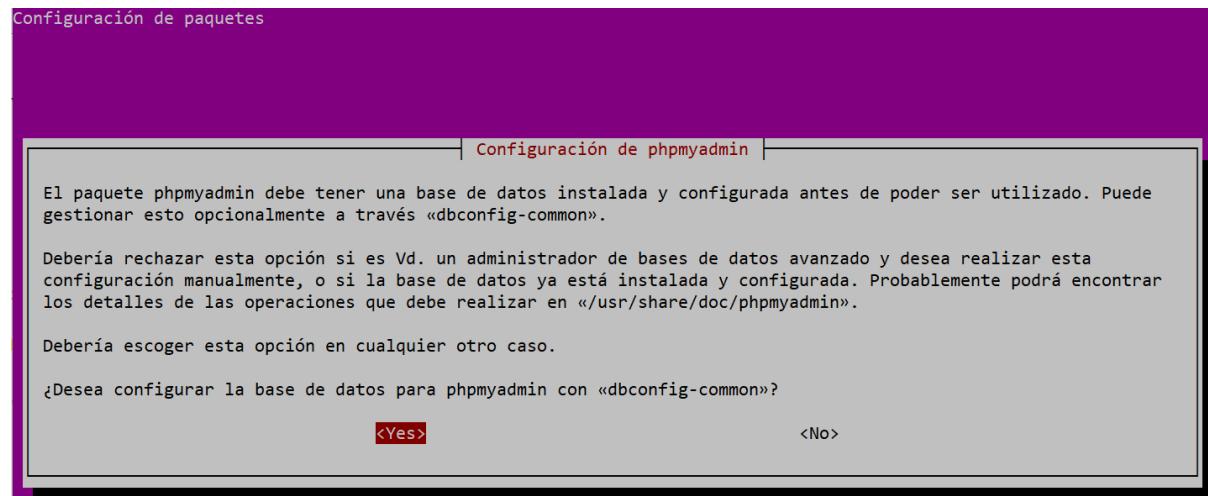
(-y : Sirve para indicar que 'SI' , queremos instalar , para que no nos pregunte más adelante en plena instalación)

Una vez lanzado el comando nos aparecerá la siguiente ventana morada en la que nos preguntará en que servidor queremos que se ejecute phpMyAdmin, nosotros seleccionamos con la barra espaciadora apache2

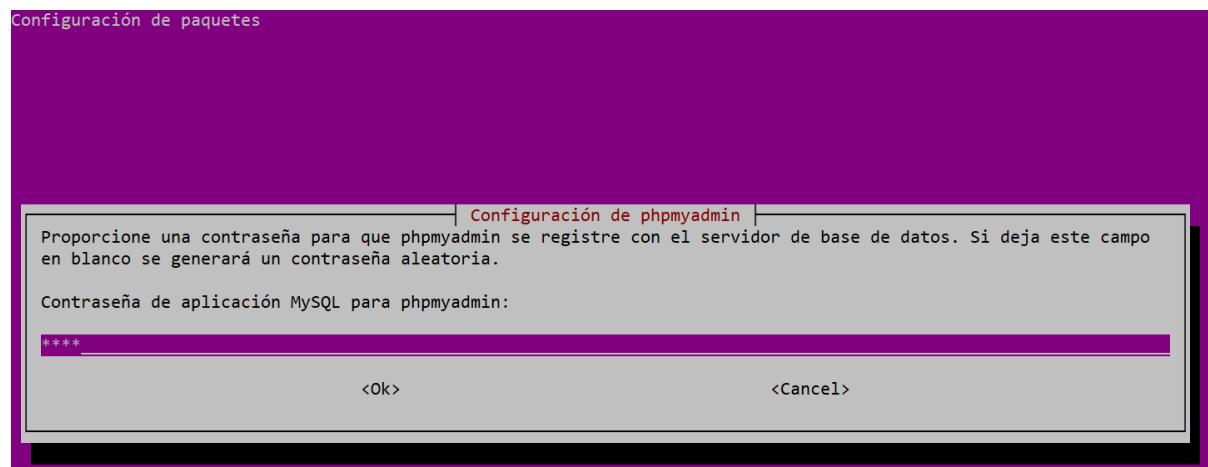
Configuración de paquetes

| Configuración de phpmyadmin |
Por favor, elija el servidor web que se debería configurar automáticamente para que ejecute phpMyAdmin.
Servidor web que desea reconfigurar automáticamente:
[*] apache2
[] lighttpd

En la siguiente nos pregunta si queremos configurar la base de datos , en nuestro caso le decimos '**<Yes>**'



E introducimos la contraseña de nuestro usuario de MySQL que es 'paso'



Al ser una contraseña que no cumple las directivas de seguridad no dará un error, es normal, para solucionarlo, entramos a MySQL

`mysql -u adminsql -p`

`miadmin@CGC-US:/$ mysql -u adminsql -p`

Y desinstalamos el archivo del componente de validación de contraseñas

`uninstall component "file://component_validate_password";`

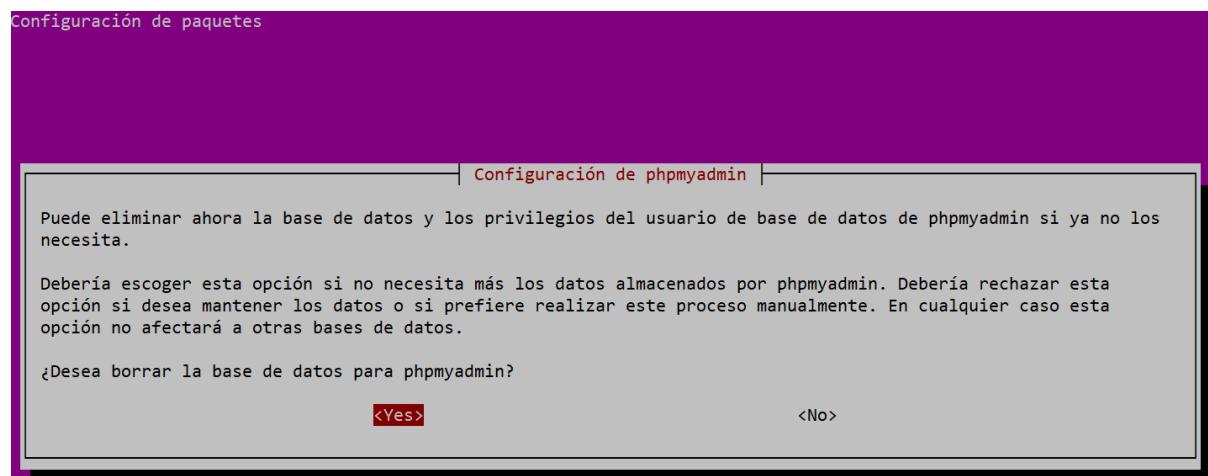
`mysql> uninstall component "file://component_validate_password";
Query OK, 0 rows affected (0,01 sec)`

Ahora si instalamos de nuevo phpMyAdmin, podría saltarnos alguna parte de la instalación por lo tanto 'purgaremos' y así eliminaremos los archivos que no se instalaron correctamente

`sudo apt purge phpmyadmin`

```
miadmin@CGC-US ED:/$ sudo apt purge phpmyadmin
```

Nos aparecerá la siguiente ventana después de lanzar el comando anterior, indicando si queremos eliminar parte de la configuración de phpMyAdmin, la cual indicamos que si queremos

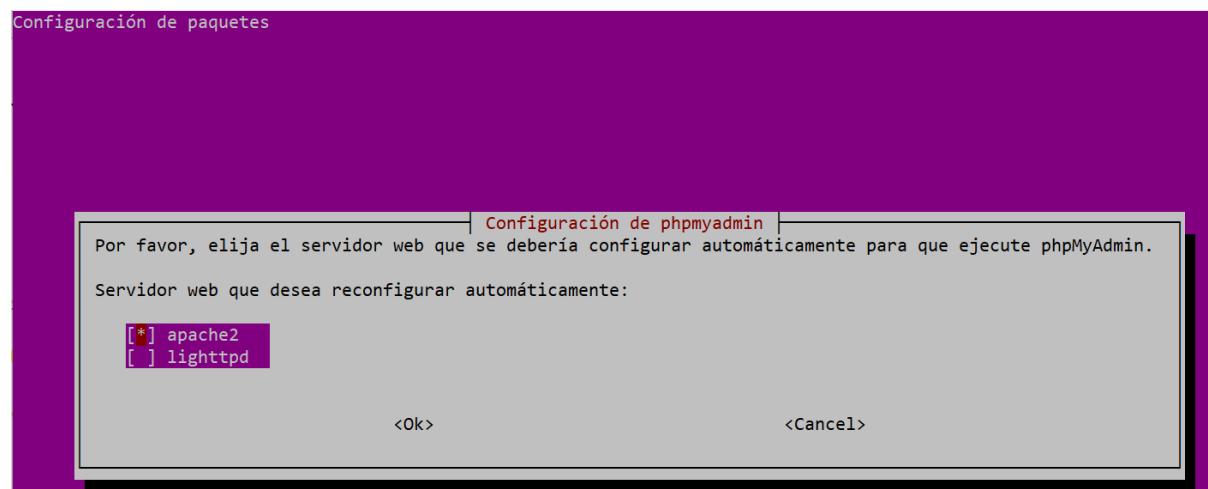


Y volvemos a instalar phpMyAdmin (-y : Sirve para indicar que 'SI' , queremos instalar , para que no nos pregunte más adelante en plena instalación)

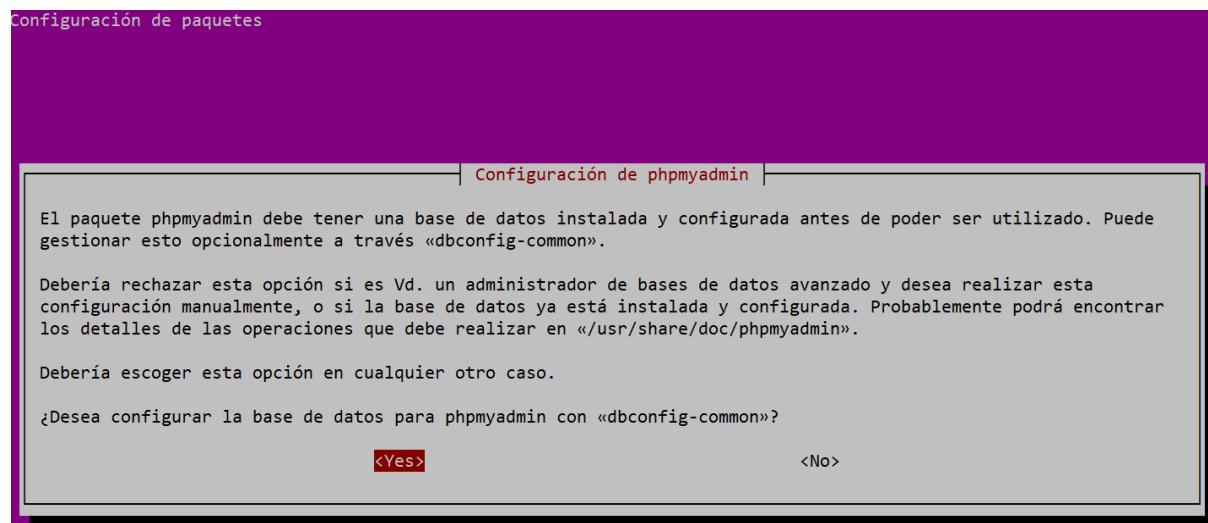
```
sudo apt -y install phpmyadmin
```

```
miadmin@CGC-US ED:/$ sudo apt -y install phpmyadmin
```

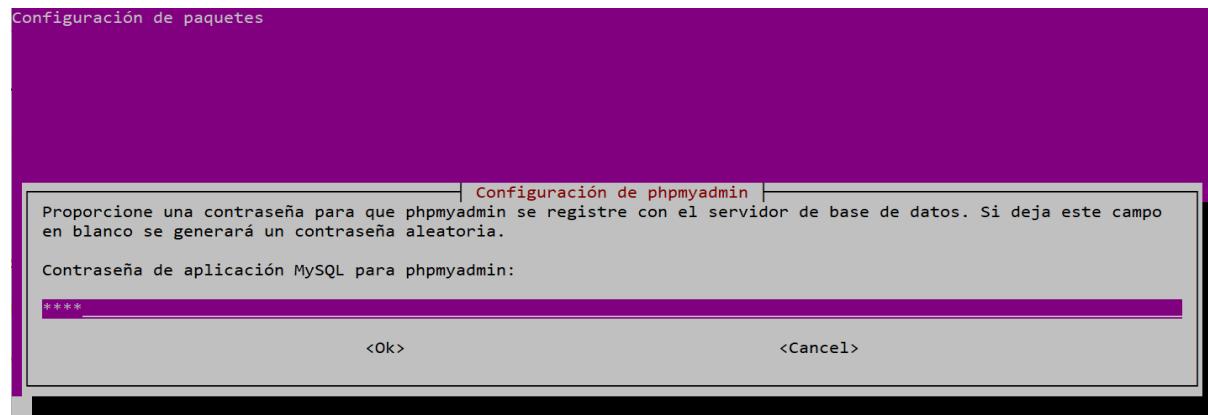
Al iniciar la instalación nos volverá a aparecer la instalación desde el principio, seleccionamos el servidor



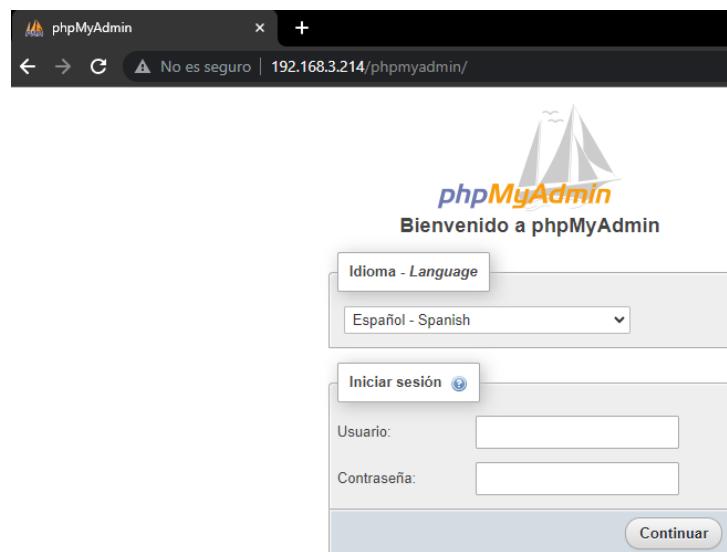
En la siguiente, la configuración, le decimos que si '<Yes>'



Y la siguiente ponemos la contraseña, en este caso 'paso'



Y comprobamos que se instaló correctamente desde un navegador , introduciendo la IP del servidor seguido de barra y phpmyadmin: <http://192.168.3.214/phpmyadmin>



Y en el servidor volvemos a mostrar los módulos instalados y comprobamos los nuevos módulos

`dpkg --get-selections | grep php`

```
miadmin@CGC-USED:/$ dpkg --get-selections | grep -i php
libapache2-mod-php          install
libapache2-mod-php8.1         install
php-bz2                      install
php-common                   install
php-curl                      install
php-gd                       install
php-google-recaptcha         install
php-json                      install
php-mariadb-mysql-kbs        install
php-mbstring                  install
php-mysql                     install
php-nikic-fast-route          install
php-phpmyadmin-motranslator   install
php-phpmyadmin-shapefile     install
php-phpmyadmin-sql-parser    install
php-phpseclib                 install
php-psr-cache                 install
php-psr-container              install
php-psr-log                   install
php-symfony-cache             install
php-symfony-cache-contracts   install
php-symfony-config             install
php-symfony-dependency-injection install
php-symfony-deprecation-contracts install
php-symfony-expression-language install
php-symfony-filesystem         install
php-symfony-polyfill-php80      install
php-symfony-polyfill-php81      install
php-symfony-service-contracts  install
php-symfony-var-exporter       install
php-tcpdf                      install
php-twig                       install
php-twig-i18n-extension        install
php-zip                        install
php8.1                         install
php8.1-bz2                     install
php8.1-cli                     install
php8.1-common                  install
php8.1-curl                    install
php8.1-gd                      install
php8.1-mbstring                install
php8.1-mysql                   install
php8.1-opcache                 install
php8.1-readline                install
php8.1-soap                    install
php8.1-xdebug                  install
php8.1-xml                     install
php8.1-zip                     install
phpmyadmin                     install
```

CUENTAS DE DESARROLLO Y HOSTING VIRTUAL

Creación del usuario:

Para crear el usuario, en este caso 'operadorweb', usamos la siguiente línea de comandos

```
sudo adduser --no-create-home --home /var/www/html --ingroup www-data operadorweb
```

```
miadmin@CGC-US:/$ sudo adduser --no-create-home --home /var/www/html --ingroup www-data operadorweb
```

Y para comprobar que se ha creado correctamente

```
cat /etc/passwd | grep operadorweb
```

```
miadmin@CGC-US:/$ cat /etc/passwd | grep operadorweb  
operadorweb:x:1001:33:operadorweb,,,:/var/www/html:/bin/sh
```

Modificar los permisos 'operadorweb':

Lo primero que hay que hacer es cambiar el propietario del fichero

```
sudo chown -R operadorweb:www-data /var/www/html
```

```
miadmin@CGC-US:/$ sudo chown -R operadorweb:www-data /var/www/html
```

Y comprobamos los permisos de los archivos

```
ll /var/www/html
```

```
miadmin@CGC-US:/$ ll /var/www/html/  
total 56  
drwxrwsr-x 11 operadorweb www-data 4096 oct 26 18:39 ./  
dr-xr-xr-x  4 root      root    4096 oct 24 17:24 ../  
drwxr-sr-x  5 operadorweb www-data 4096 oct 24 18:23 214DAWProyectoDAW/  
drwxr-sr-x  5 operadorweb www-data 4096 oct 24 18:23 214DIWProyectoDIW/  
drwxr-sr-x  6 operadorweb www-data 4096 oct 24 18:24 214DWECProyectoDWEC/  
drwxr-sr-x  6 operadorweb www-data 4096 oct 24 18:24 214DWESProyectoDWES/  
drwxr-sr-x  9 operadorweb www-data 4096 oct 25 18:19 214DWESProyectoTema3/  
-rw-----  1 operadorweb www-data    5 oct 24 17:19 .bash_history  
drwx--S---  2 operadorweb www-data 4096 sep 28 09:08 .cache/  
drwxr-sr-x  2 operadorweb www-data 4096 oct 24 18:25 error/  
-rw-r--r--  1 operadorweb www-data 169 oct 24 18:25 .htaccess  
-rw-r--r--  1 operadorweb www-data 3999 oct 24 18:25 index.html  
drwxr-sr-x  2 operadorweb www-data 4096 oct 24 18:25 log/  
drwxr-sr-x  4 operadorweb www-data 4096 oct  8 12:41 webroot/
```

Después le cambiaremos los permisos al fichero para poder editarlo. (El número 2 es un permiso especial que permite que todos los ficheros que se incluyan en ese grupo tengas los permisos de ese mismo.)

```
sudo chmod -R 2775 /var/www/html
```

```
miadmin@CGC-US:/$ sudo chmod -R 2775 /var/www/html
```

Volvemos a comprobar los permisos

```
ll /var/www/html
```

```
miadmin@CGC-USDED:/$ ll /var/www/html/
total 56
drwxrwsr-x 11 operadorweb www-data 4096 oct 26 18:39 .
dr-xr-xr-x  4 root      root    4096 oct 24 17:24 ..
drwxrwsr-x  5 operadorweb www-data 4096 oct 24 18:23 214DAWProyectoDAW/
drwxrwsr-x  5 operadorweb www-data 4096 oct 24 18:23 214DIWProyectoDIW/
drwxrwsr-x  6 operadorweb www-data 4096 oct 24 18:24 214DWECProyectoDWEC/
drwxrwsr-x  6 operadorweb www-data 4096 oct 24 18:24 214DWESProyectoDWES/
drwxrwsr-x  9 operadorweb www-data 4096 oct 25 18:19 214DWESProyectoTema3/
-rwxrwsr-x  1 operadorweb www-data   5 oct 24 17:19 .bash_history*
drwxrwsr-x  2 operadorweb www-data 4096 sep 28 09:08 .cache/
drwxrwsr-x  2 operadorweb www-data 4096 oct 24 18:25 error/
-rwxrwsr-x  1 operadorweb www-data 169 oct 24 18:25 .htaccess*
-rwxrwsr-x  1 operadorweb www-data 3999 oct 24 18:25 index.html*
drwxrwsr-x  2 operadorweb www-data 4096 oct 24 18:25 log/
drwxrwsr-x  4 operadorweb www-data 4096 oct  8 12:41 webroot/
```

Enjaulado de usuarios

(En este caso vamos a usar un nuevo usuario llamado DAW212, pero puede ser cualquier nombre)

Primero creamos el grupo (A este grupo le añadiremos todos los usuarios que se conecten a nuestro servidor por FTP o SFTP)

```
sudo groupadd ftpusers miadmin@CGC-USDED:~$ sudo groupadd ftpusers
```

Luego el usuario y cambio de contraseña

```
sudo useradd -g www-data -G ftpusers -m -d /var/www/DAW212 DAW212
```

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

```
passwd DAW212
```

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

El propietario del directorio debe ser root , así que aplicaremos los siguientes permisos

```
sudo chown root:root /var/www/DAW212
```

```
miadmin@DAW-USDED:/$ sudo chown root:root /var/www/DAW212
```

Y eliminamos el permiso de escritura

```
sudo chmod 555 /var/www/DAW212
```

```
miadmin@DAW-USDED:/$ sudo chmod 555 /var/www/DAW212
```

Ahora creamos la carpeta donde el usuario puede accedes llamada 'public_html'

```
sudo mkdir /var/www/DAW212/public_html
```

```
miadmin@DAW-USDED:/$ sudo mkdir /var/www/DAW212/public_html
```

Le aplicamos los permisos al nuevo directorio

```
sudo chmod 2775 -R /var/www/DAW212/public_html
```

```
miadmin@DAW-USDED:/$ sudo chmod 2775 -R /var/www/DAW212/public_html
```

Y por último indicamos el propietario de dicha carpeta

```
sudo chown DAW212:www-data -R /var/www/DAW212/public_html
```

```
miadmin@DAW-USDED:/$ sudo chown DAW212:www-data -R /var/www/DAW212/public_html
```

Configuración fichero de hosting

Nos dirigimos al siguiente directorio y creamos un archivo en blanco

```
cd /etc/apache2/sites-available
```

```
sudo nano DAW212.conf
```

```
miadmin@DAW-USDED:/etc/apache2/sites-available$ sudo nano DAW212.conf
```

Y lo rellenamos igual que en la captura

```
<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 con el siguiente comando comprobamos que no exista ningún error en los archivos de ese directorio

```
sudo apache2ctl configtest
```

```
miadmin@DAW-USDED:/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
rName' directive globally to suppress this message
Syntax OK
```

Después habilitamos el fichero

```
sudo a2ensite DAW212.conf
```

```
miadmin@DAW-USDED:/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
```

```
miadmin@DAW-USDED:/etc/apache2/sites-available$ systemctl reload apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to reload 'apache2.service'.
Multiple identities can be used for authentication:
 1. miadmin
 2. miadmin2
Choose identity to authenticate as (1-2): 1
Password:
==== AUTHENTICATION COMPLETE ===
```

(En este caso nos pide que nos autentiquemos estamos conectados al servidor por ssh)

DOCUMENTACIÓN DE LA INSTALACIÓN Y CONFIGURACIÓN DEL CLIENTE (WXED):

Instalación y configuración inicial de la máquina



Cuentas administradoras y cuenta de desarrollador

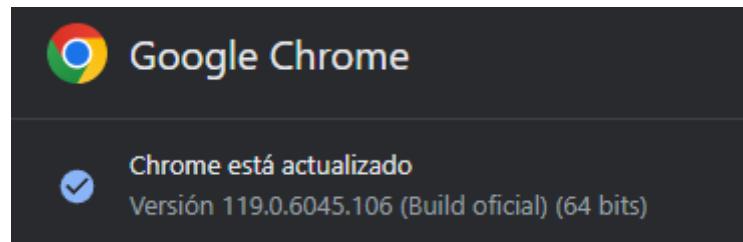
Usamos la cuenta 'daw2' como cuenta de desarrollo y 'Administrador' , como cuenta administradora

```
C:\Users\daw2>net users
Cuentas de usuario de \\IS32WX14
-----
admin           Administrador          bachillerato
daw1           daw2                 DefaultAccount
eso            Invitado              smr1
smr2           WDAGUtilityAccount
Se ha completado el comando correctamente.

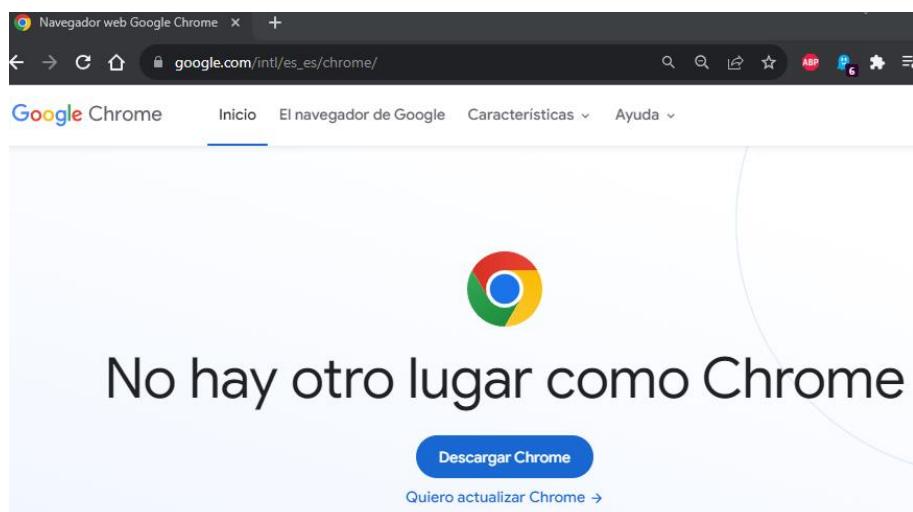
C:\Users\daw2>net localgroup
Alias para \\IS32WX14
-----
*Administradores
*Administradores de Hyper-V
*Duplicadores
*IIS_IUSRS
*Invitados
*Lectores del registro de eventos
*Operadores criptográficos
*Operadores de asistencia de control de acceso
*Operadores de configuración de red
*Operadores de copia de seguridad
*System Managed Accounts Group
*Usuarios
*Usuarios avanzados
*Usuarios COM distribuidos
*Usuarios de administración remota
*Usuarios de escritorio remoto
*Usuarios del monitor de sistema
*Usuarios del registro de rendimiento
Se ha completado el comando correctamente.
```

Navegadores

Ahora descargaremos un navegador para cuando queramos acceder a nuestra página, usaremos Google Chrome, concretamente esta:



Accederemos a esta URL, https://www.google.com/intl/es_es/chrome/, para descargar el nuevo navegador.

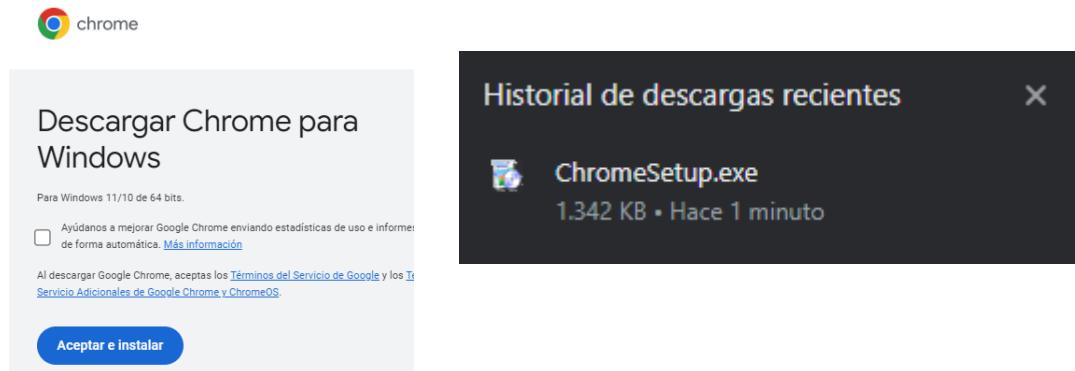


Como curiosidad, si descendemos casi hasta el final de la página, nos aparecerán cuatro opciones de navegador, Enterprise, Dev, Beta y Canary.

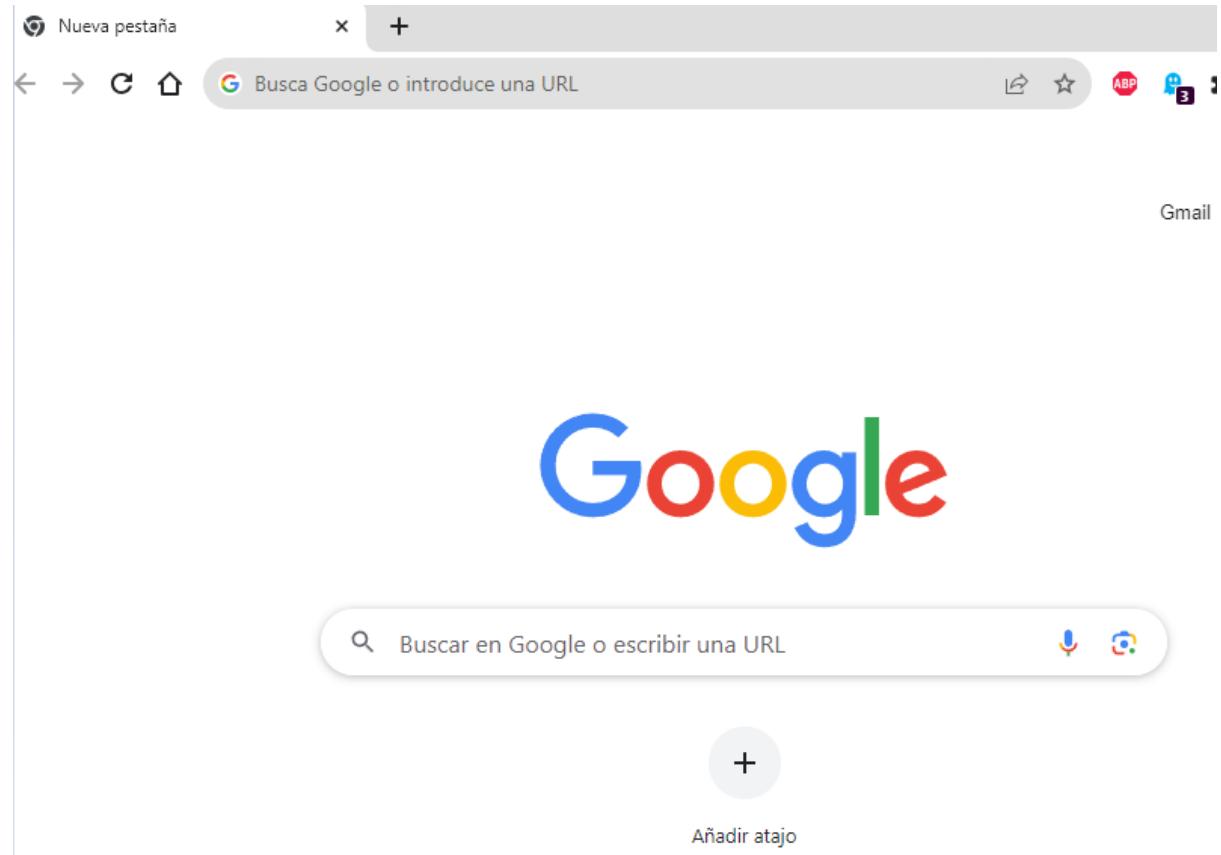
MÁS DE CHROME Descubre más herramientas y recursos

Para empresas Protege la seguridad de las personas y de los datos con actualizaciones sin interrupciones y unas medidas intuitivas para el cumplimiento de políticas. Ir a Chrome Enterprise →	Para desarrolladores Desarrolla sitios web para la próxima versión de la Web abierta con Chrome para desarrolladores. Ir a Chrome Dev →	Para usuarios pioneros Con Chrome Beta, puedes echar un vistazo a las próximas funciones de Chrome antes de que se lancen. Ir a Chrome Beta →	Para exploradores Mantente a la vanguardia de Internet y obtén actualizaciones frecuentes con Chrome Canary. Ir a Chrome Canary →
--	---	---	---

En nuestro caso instalaremos la última versión estable presionando el botón azul ‘Descargar Chrome’ , nos aparecerá una ventana emergente y hacemos click sobre ‘Aceptar e instalar’ y descargará un ejecutable

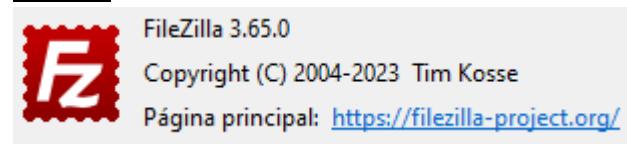


Ahora le daremos click derecho ‘Abrir’ y nos iniciará la instalación, seguiremos las instrucciones del instalador y por último nos abrirá el navegador y nos aparecerá un acceso directo en el escritorio.



Filezilla

Versión



Conexión con el servidor

Al abrir FileZilla debemos introducir los parámetros de conexión, en nuestro caso son los siguientes

Servidor:	192.168.3.214	Nombre de usuario:	operadorweb	Contraseña:	*****	Puerto:	22	Conexión rápida	▼
-----------	---------------	--------------------	-------------	-------------	-------	---------	----	-----------------	---

En el siguiente apartado de la ventana nos aparece el directorio remoto y el contenido de este

The screenshot shows the FileZilla interface with the following details:

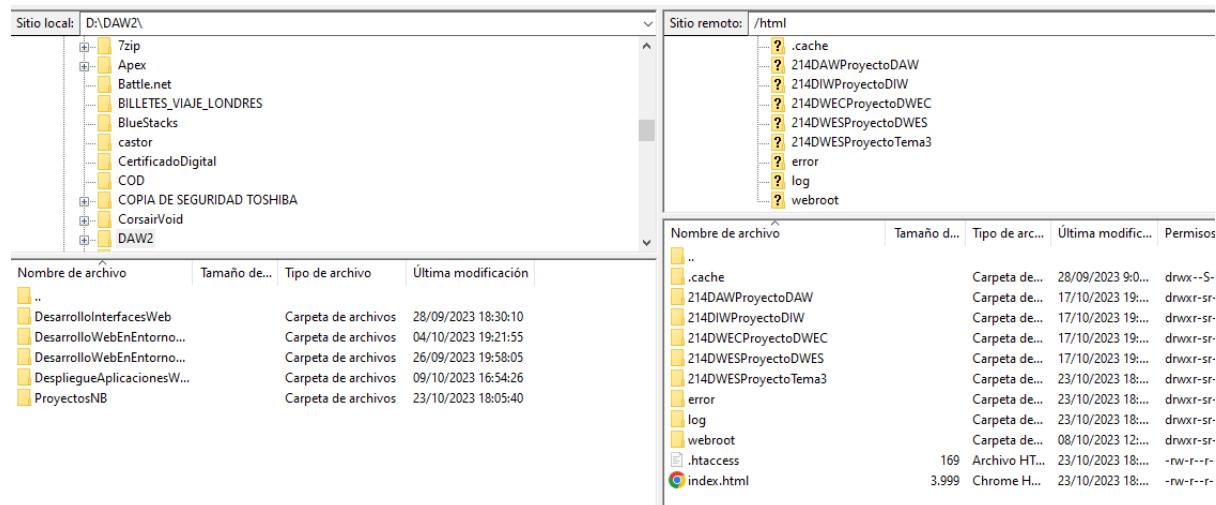
Site Remote Path: /html

File List:

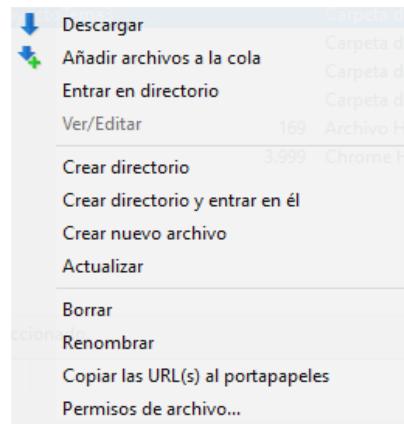
Nombre de archivo	Tamaño d...	Tipo de arc...	Última modific...	Permisos	Propietario/...
..		Carpeta de...	28/09/2023 9:0...	drwx--S---	operadorwe...
.cache		Carpeta de...	17/10/2023 19:...	drwxr-sr-x	operadorwe...
214DAWProyectoDAW		Carpeta de...	17/10/2023 19:...	drwxr-sr-x	operadorwe...
214DIWProyectoDIW		Carpeta de...	17/10/2023 19:...	drwxr-sr-x	operadorwe...
214DWECProyectoDWEC		Carpeta de...	17/10/2023 19:...	drwxr-sr-x	operadorwe...
214DWESProyectoDWES		Carpeta de...	17/10/2023 19:...	drwxr-sr-x	operadorwe...
214DWESProyectoTema3		Carpeta de...	23/10/2023 18:...	drwxr-sr-x	operadorwe...
error		Carpeta de...	23/10/2023 18:...	drwxr-sr-x	operadorwe...
log		Carpeta de...	23/10/2023 18:...	drwxr-sr-x	operadorwe...
webroot		Carpeta de...	08/10/2023 12:...	drwxr-sr-x	operadorwe...
.htaccess	169	Archivo HT...	23/10/2023 18:...	-rw-r--r--	operadorwe...
index.html	3.999	Chrome H...	23/10/2023 18:...	-rw-r--r--	operadorwe...

Subir y Bajar archivos del servidor

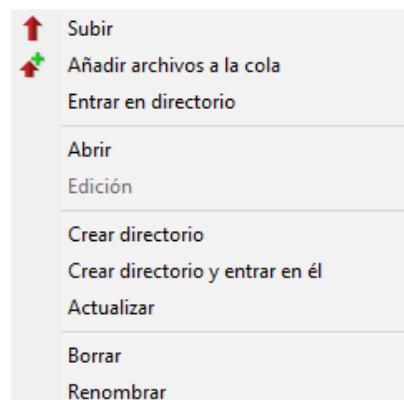
A la hora de mover archivos entre la maquina cliente y el servidor es tan sencillo como arrastrar lo que se necesite subir al servidor o bajar de él



O también podemos hacer 'click' derecho sobre el archivo o directorio e indicarle lo que queremos hacer. Este desplegable saldría en caso seleccionar algo en la parte derecha (SERVIDOR)



Y este en la parte izquierda (ALMACENAMIENTO LOCAL / CLIENTE)



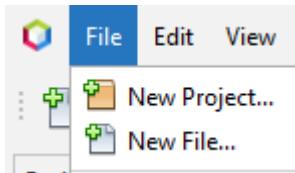
NetBeans

Product Version: Apache NetBeans IDE 19
Java: 17.0.6; Java HotSpot(TM) 64-Bit Server VM 17.0.6+9-LTS-190

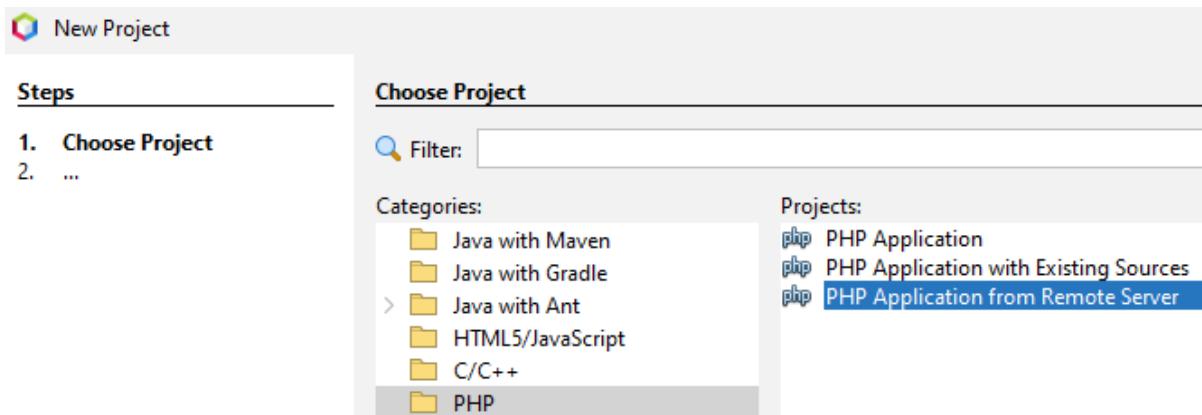
Instalación y configuración inicial (pluggins)

Creación de un proyecto PHP

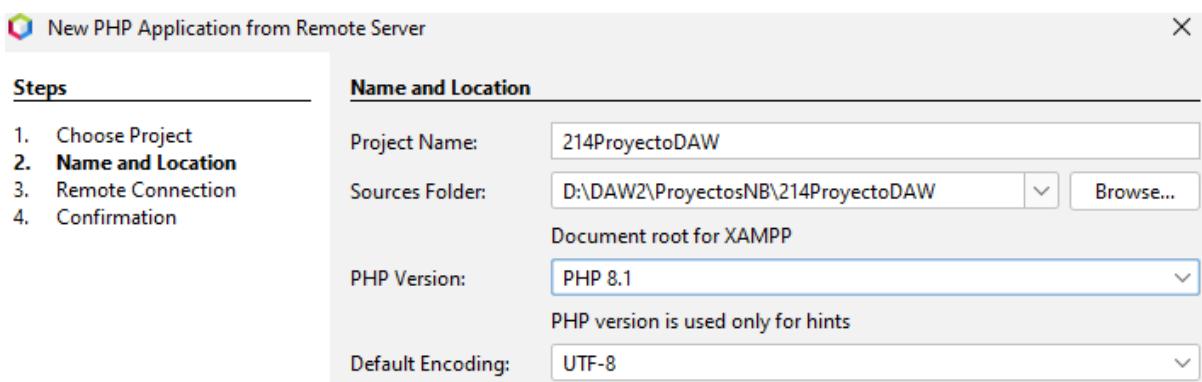
Abrimos NetBeans (NB) y hacemos click el 'File' y luego 'New Project...'



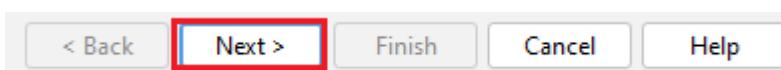
Se nos abrirá la siguiente ventana, donde podemos elegir el tipo, en este caso seleccionamos la opción de 'Aplicación PHP desde servidor remoto':



Lo siguiente es ponerle nombre al proyecto, seleccionar la ruta del almacenamiento local, la versión de PHP y el 'Default Encoding' (Que hace referencia a la configuración de la codificación de caracteres por defecto que se utilizará en el proyecto):

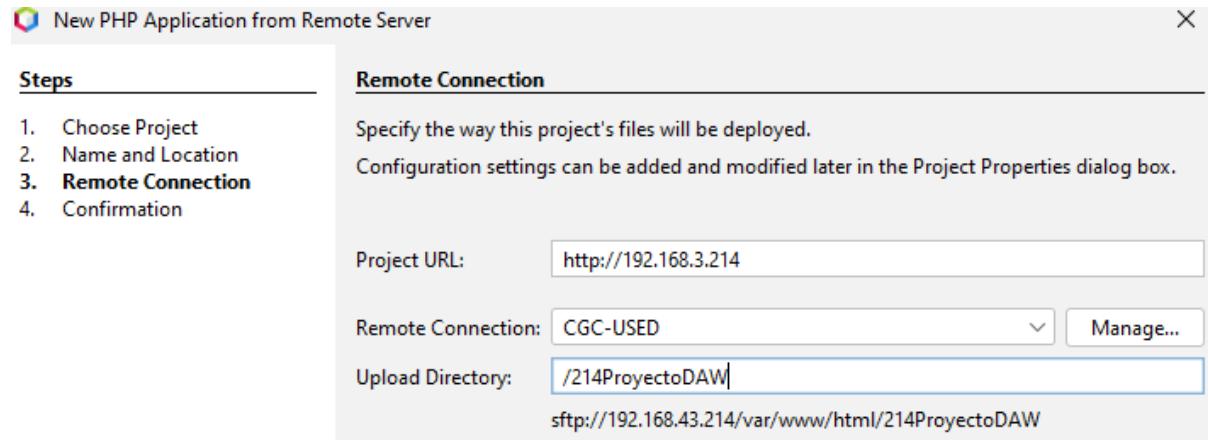


Luego haremos click en 'Next >' en la parte baja de la ventana

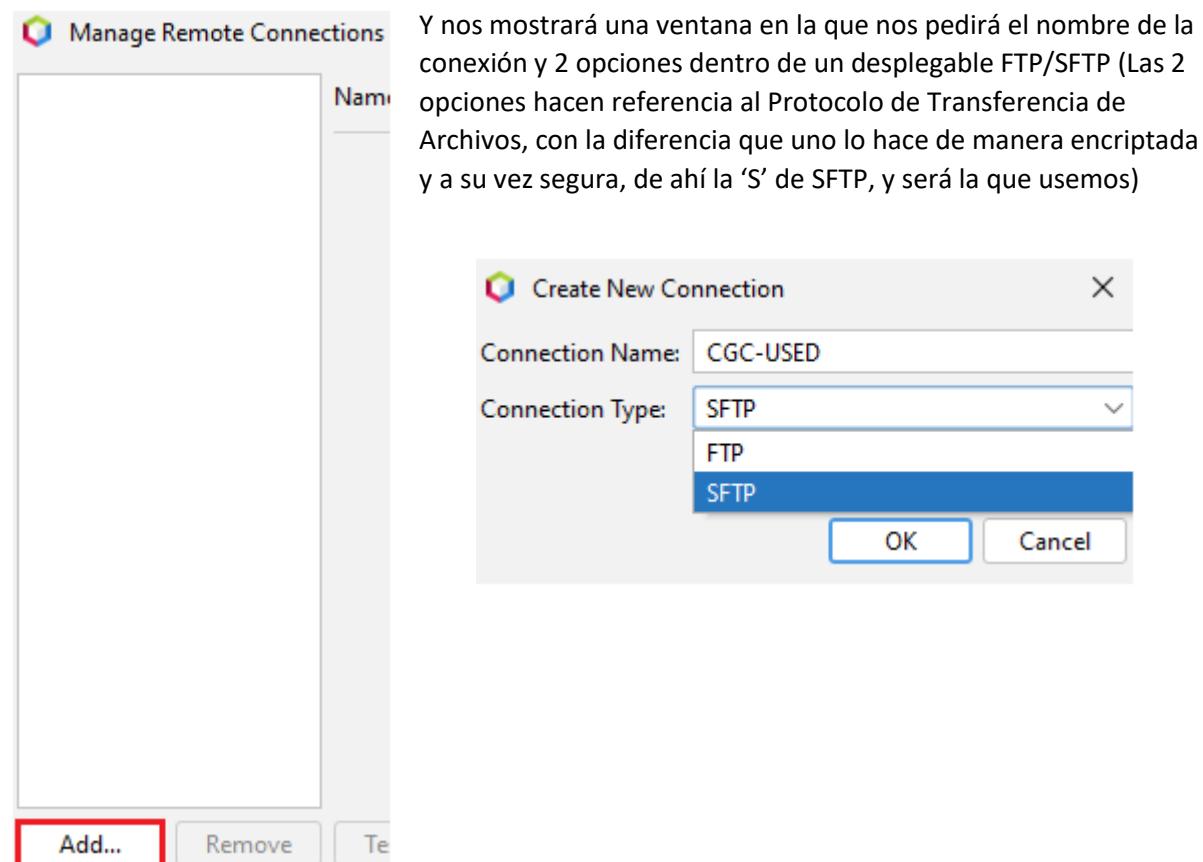


En la siguiente ventana deberemos indicar la URL de nuestro proyecto (en este caso nosotros usamos la IP del servidor) , seleccionar la conexión remota e indicarle el directorio de subida.

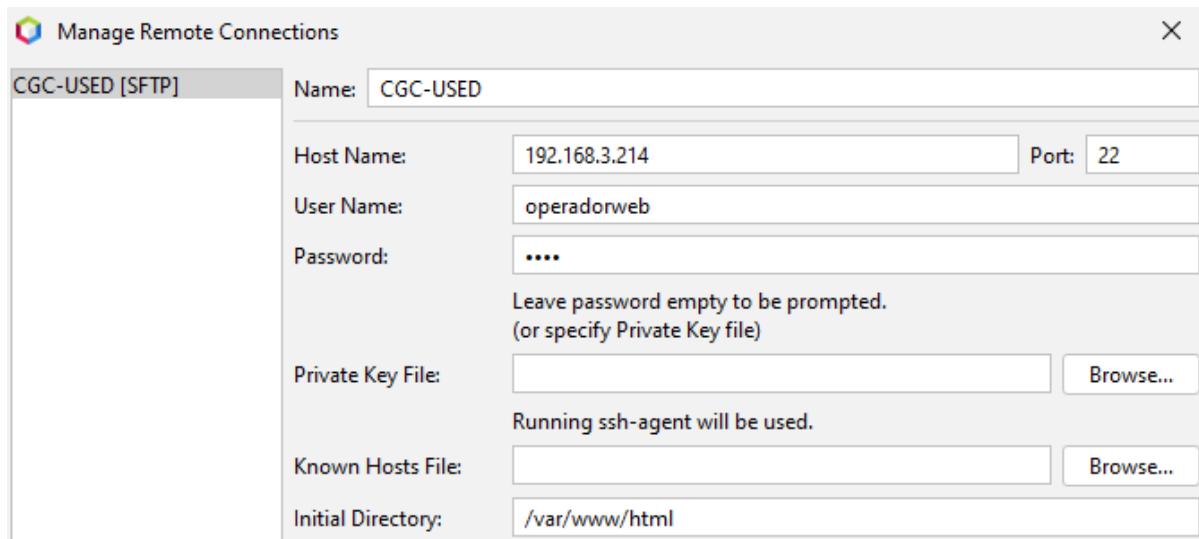
Antes de continuar haremos click en ‘Manage...’



Nos aparecerá la ventana de ‘Manage Remote Connections’ , en esta, deberemos dirigirnos a la parte baja izquierda y hacer click en ‘Add...’



Una vez creada la nueva conexión, pasaremos a configurarla con los siguientes valores



Hostname: Debemos indicar la IP de nuestro servidor

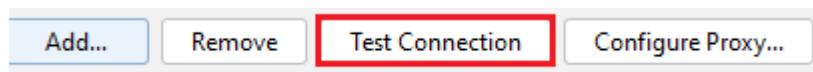
Port: Indica el puerto por el que vamos a acceder al servidor en este caso el 22 porque estamos usando SFTP

User Name: El nombre del usuario con el cual nos conectaremos al servidor

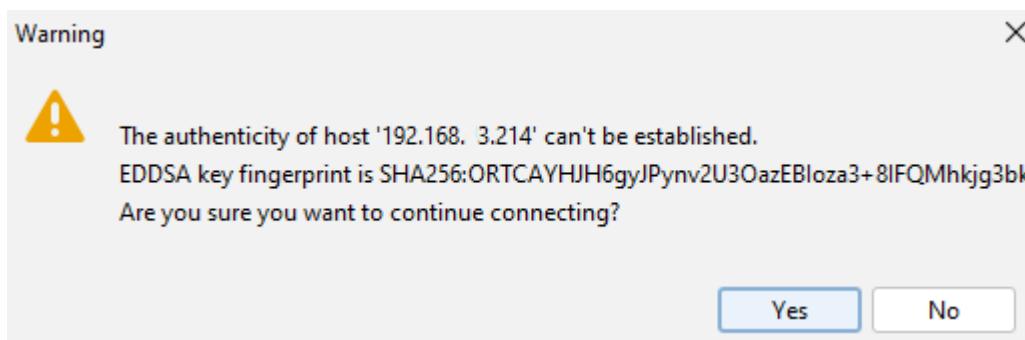
Password: La contraseña del usuario que usamos para acceder al servidor

Initial Directory: Hace referencia a la ruta donde se va a llevar los archivos que subamos

Después de llenar los campos anteriores tenemos que comprobar la conexión, para ello en la parte inferior de la misma ventana, haremos click al botón 'Test Connection'



Nos aparecerá la siguiente ventana en caso de que el test sea favorable, preguntando si queremos establecer conexión



Ahora tendíamos un problema y que NetBeans nos pedirá subir un archivo, para completar la creación del proyecto, para solucionarlo tendremos 2 opciones, usando Filezilla o de manera manual en nuestro servidor.

Usaremos la segunda opción, así que tendremos que dirigirnos al directorio donde se está guardando nuestro proyecto en el servidor, que es /var/www/html/

No dirigiremos a él y por medio de una dirección absoluta usando ‘nano’ crearemos un index con el siguiente comando (Después de ‘html/’ indicaremos el nombre de nuestro proyecto)

```
sudo nano /var/www/html/214ProyectoDAW/index.html
```

```
miadmin@CGC-USED:/ $ sudo nano /var/www/html/214ProyectoDAW/index.html
```

Y añadimos una estructura HTML dentro de nuestro archivo para que pueda interpretarlo el navegador

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Título de la página</title>
</head>
<body>
  <header>
  </header>
  <main>
  </main>
  <footer>
  </footer>
</body>
</html>
```

<!DOCTYPE html>: Declaración que especifica la versión de HTML que se está utilizando. En este caso, se utiliza HTML5.

<html>: Elemento raíz que envuelve todo el contenido de la página. El atributo lang se utiliza para especificar el idioma principal de la página.

<head>: Contiene metadatos, como la codificación de caracteres y el título de la página.

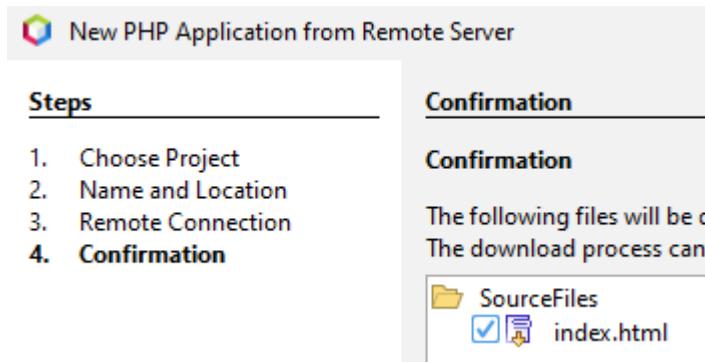
<meta charset="UTF-8">: Especifica la codificación de caracteres UTF-8, que admite la mayoría de los caracteres utilizados en diferentes idiomas.

<meta name="viewport" content="width=device-width, initial-scale=1.0">: Configura la visualización de la página en dispositivos móviles.

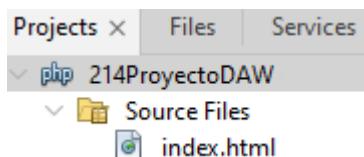
<title>: Define el título de la página que se muestra en la pestaña del navegador.

<body>: Contiene el contenido visible de la página, que incluye encabezado (<header>), contenido principal (<main>), y pie de página (<footer>).

Y al pasar a la siguiente ventana confirmaremos la conexión con el servidor y seleccionaremos que archivos desde la carpeta del servidor



Y ahora ya nos aparecerá en 'Proyectos' de NetBeans



Administración del almacenamiento local y remoto

En nuestro caso, de manera local organizaremos los directorios de los proyectos de esta manera:

Este equipo > HDD (D:) > DAW2 > ProyectosNB

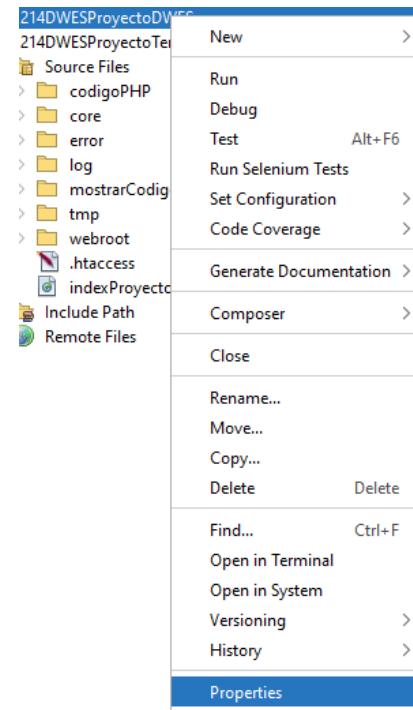
Nombre	Fecha de modificación	Tipo
214DAWProyectoDAW	23/10/2023 16:41	Carpeta de archivos
214DIWProyectoDIW	26/10/2023 20:32	Carpeta de archivos
214DWECProyectoDWEC	23/10/2023 16:41	Carpeta de archivos
214DWESProyectoDWES	23/10/2023 16:41	Carpeta de archivos
214DWESProyectoTema3	26/10/2023 16:37	Carpeta de archivos
214ProyectoDAW	23/10/2023 18:06	Carpeta de archivos

En remoto sufre un pequeño cambio respecto al local, en la raíz almacenaremos el contenido de '214ProyectoDAW' y el resto en sus respectivas carpetas, como muestra la siguiente captura:

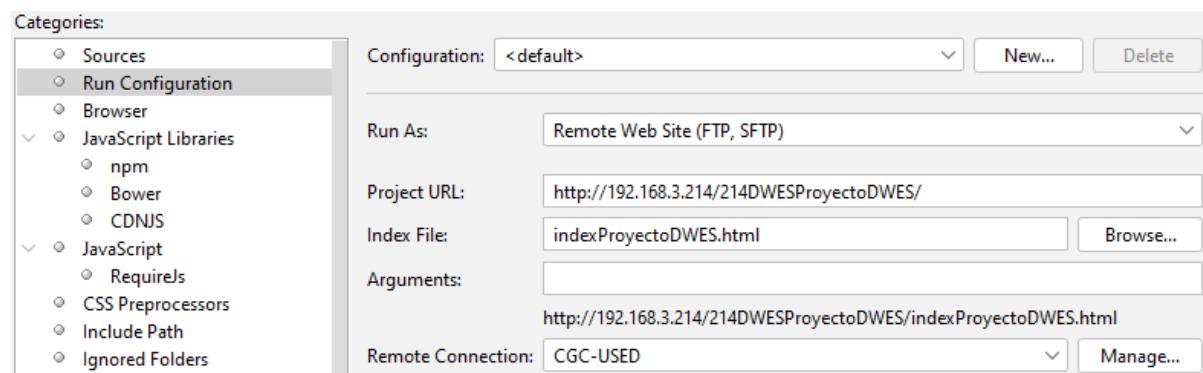
..		
.cache	Carpetas de... 28	
214DAWProyectoDAW	Carpetas de... 17	
214DIWProyectoDIW	Carpetas de... 17	
214DWECProyectoDWEC	Carpetas de... 17	
214DWESProyectoDWES	Carpetas de... 17	
214DWESProyectoTema3	Carpetas de... 18	
error	Carpetas de... 13	
log	Carpetas de... 13	
webroot	Carpetas de... 0	
.htaccess	169 Archivo HT... 13	
index.html	3.983 Chrome H... 13	

Configuración de la ejecución

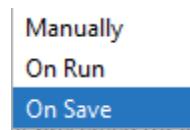
Hacemos click derecho sobre un proyecto y buscamos al final de la pestaña 'Properties'



En la siguiente ventana debería aparecer la configuración de ejecución, pero en caso de que diera algún tipo de problema al conectar con el servidor o que no se subieran los archivos, tendríamos que revisar la ruta del proyecto en el servidor, el index del archivo, la configuración de la conexión remota, el directorio remoto y en la última opción podemos elegir cuando queremos que se suban los archivos



De manera manual, al ejecutarlo y al guardar el proyecto:



Modificación de proyectos

Para modificar el contenido de un proyecto solo tenemos que seleccionar el proyecto que queramos editar navegando por el código fuente, realizando modificaciones en los archivos, agregando nuevos archivos, ejecutando el proyecto y realizar cualquier tarea relacionada con el desarrollo de software.

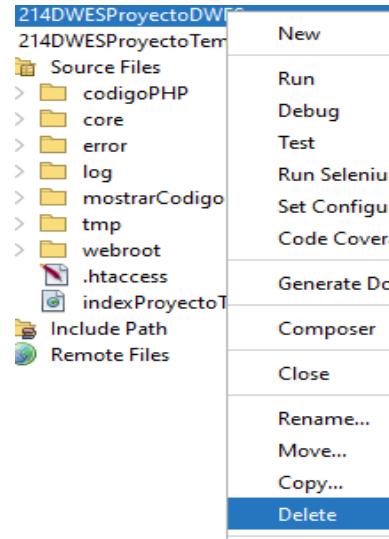
```

10 <meta name="author" content="Carlos García Cachón"/>
11 <meta name="description" content="CodigoEjercicio_1"/>
12 <meta name="keywords" content="CodigoEjercicio, ("/>
13 <meta name="generator" content="Apache NetBeans 1"/>
14 <meta name="generator" content="60"/>
15 <meta name="viewport" content="width=device-width"/>
16 <title>Carlos García Cachón</title>
17 <link rel="icon" type="image/jpg" href="../webroot/icono.jpg"/>
18 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"/>

```

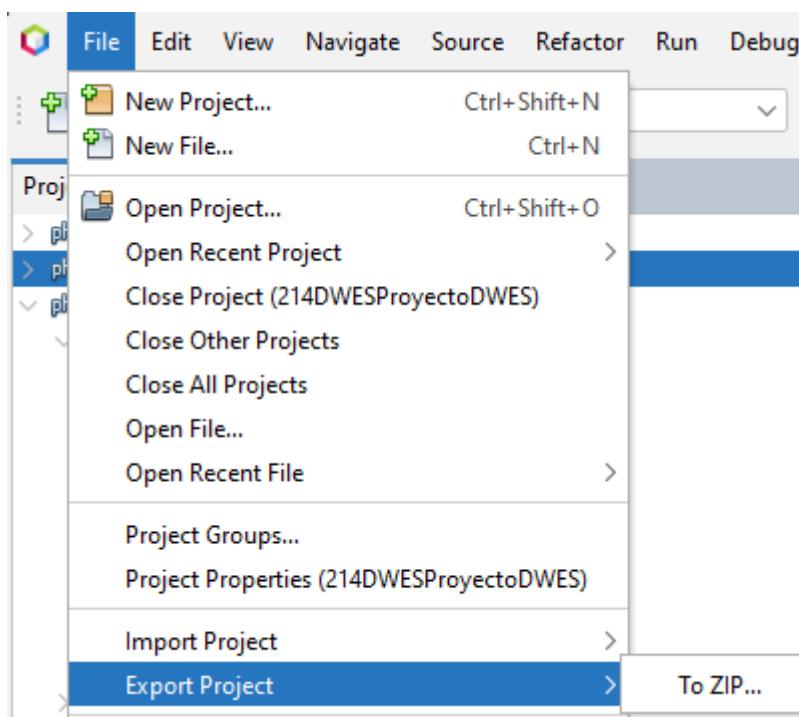
Borrado de proyectos

En el caso de borrar un proyecto es tan sencillo como seleccionarlo con 'click' derecho y darle donde dice 'Delete'

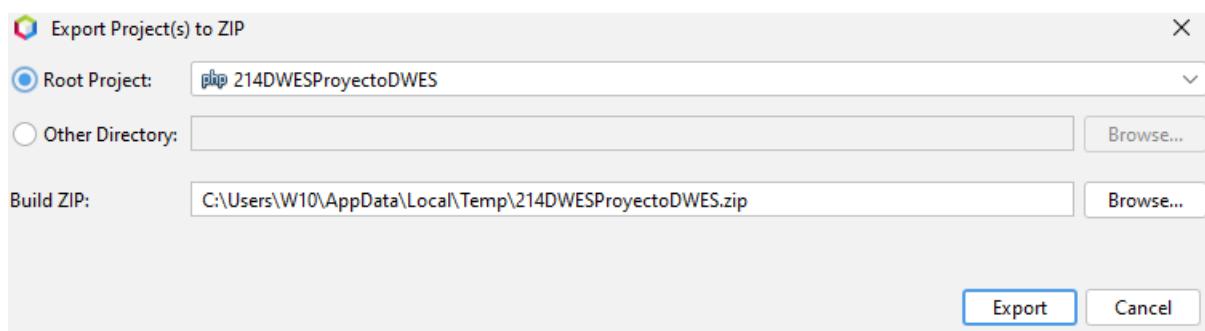


Traslado de proyectos

Nos vamos donde dice 'File' y nos aparecen entre muchas otras opciones, la de 'Export Project' o 'Import Project' un proyecto, primero le daremos a 'Export Project' > 'To ZIP...'

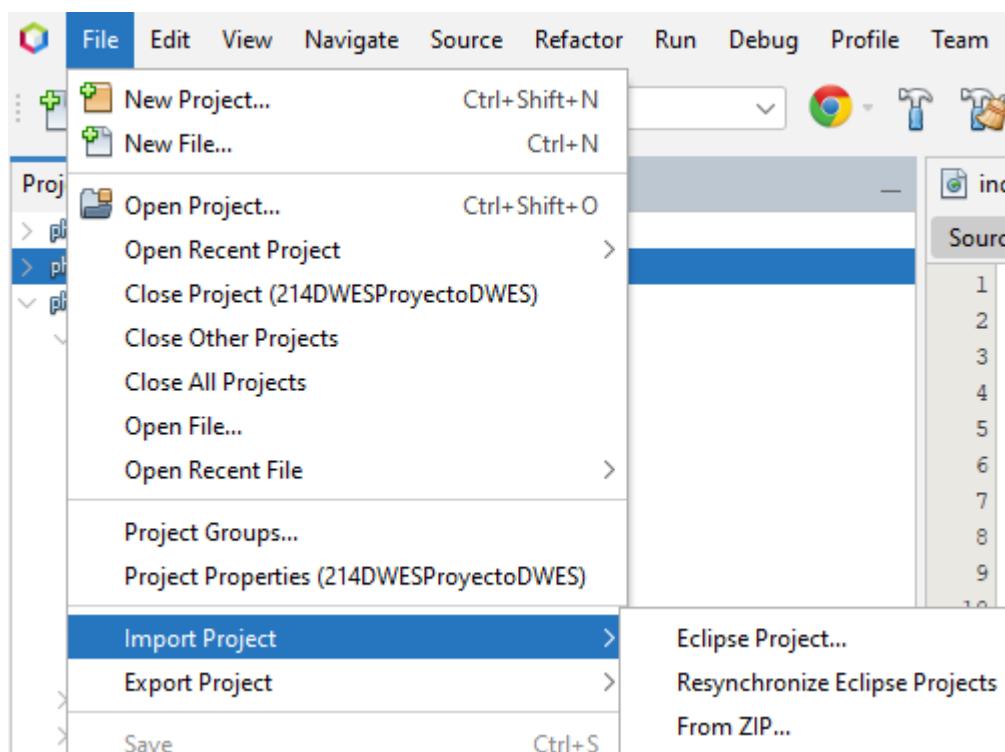


Nos aparecerá una ventana donde elegiremos el proyecto y la ruta donde queremos almacenarlo.

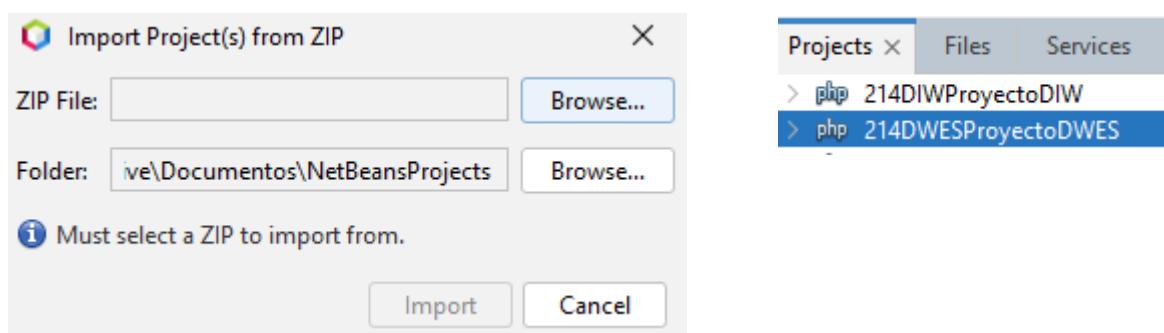


Y hacemos click en 'Export'

Para importarlo es tan sencillo como darle a 'Import Project' y elegir una de las 3 opciones, en nuestro caso lo haremos por ZIP, así que hacemos click en 'From ZIP'



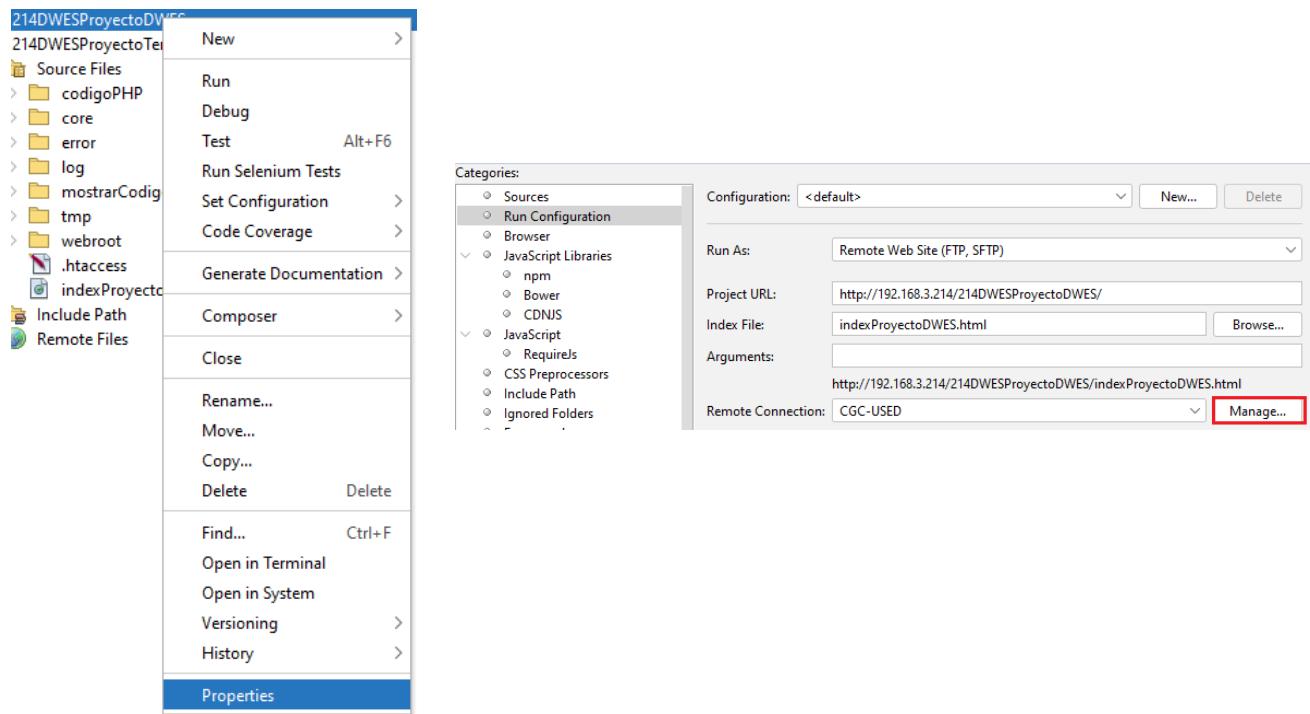
Luego nos aparecerá esta ventana en la que haciendo click en el primero botón 'Browse...', seleccionaremos la ruta nuestro 'proyecto.zip' y por último haremos click en 'Import' y nos aparecerá nuestro proyecto en 'Projects'



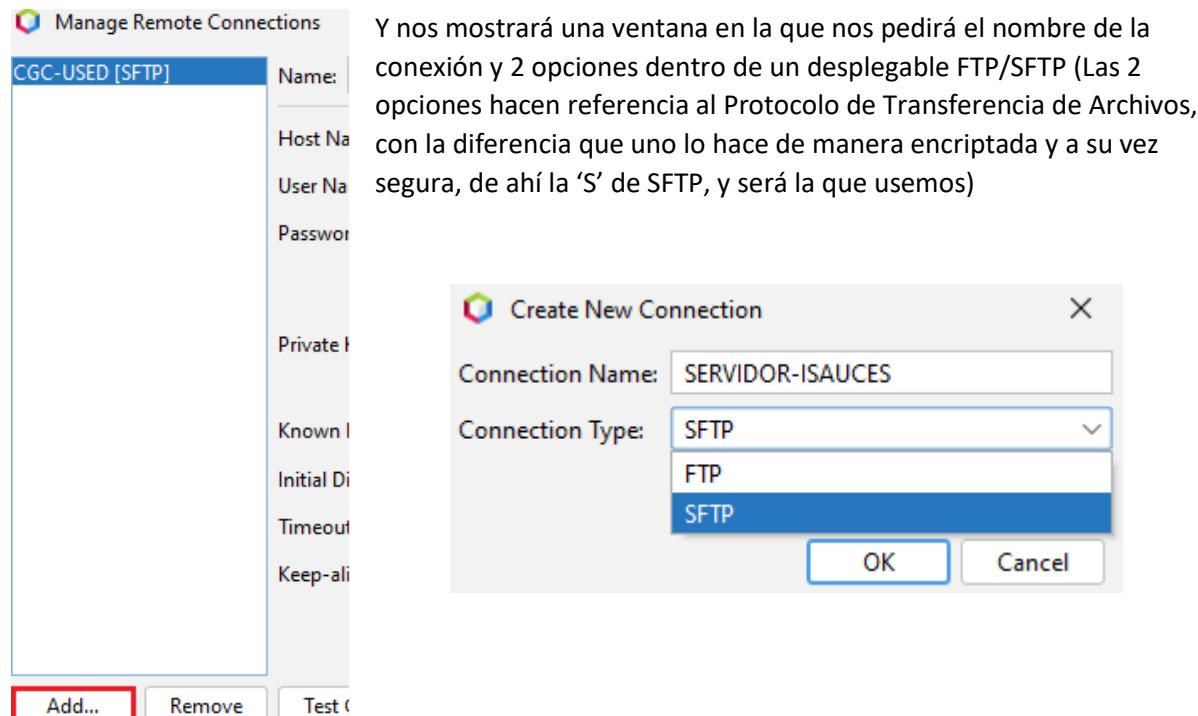
Depuración de código PHP

Conexión NetBeans con el servidor isauces

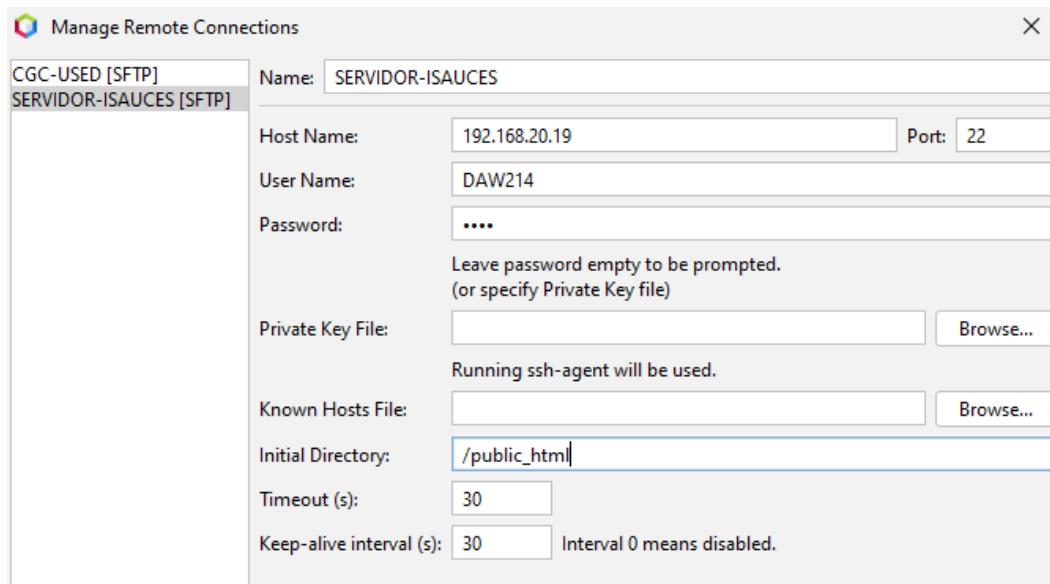
Hacemos click derecho sobre el proyecto que queramos subir al servidor, y seleccionamos 'Properties', en la siguiente ventana que nos aparecerá, (en la parte de la izquierda), seleccionamos 'Run Configuration' y hacemos click en el botón que pone 'Manage...'.



Nos aparecerá la ventana de 'Manage Remote Connections', en esta, deberemos dirigirnos a la parte baja izquierda y hacer click en 'Add...'



Una vez creada la nueva conexión, pasaremos a configurarla con los siguientes valores



Hostname: Debemos indicar la IP de nuestro servidor

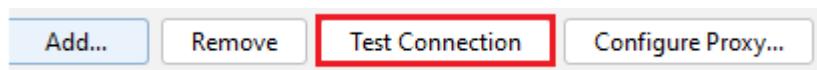
Port: Indica el puerto por el que vamos a acceder al servidor en este caso el 22 porque estamos usando SFTP

User Name: El nombre del usuario con el cual nos conectaremos al servidor

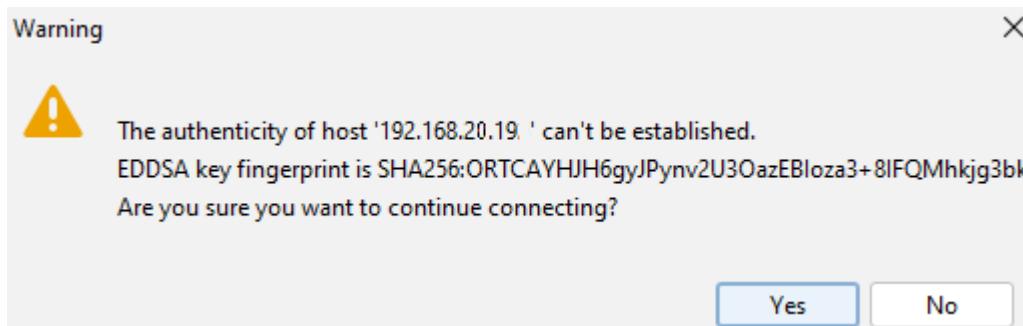
Password: La contraseña del usuario que usamos para acceder al servidor

Initial Directory: Hace referencia a la ruta donde se va a llevar los archivos que subamos

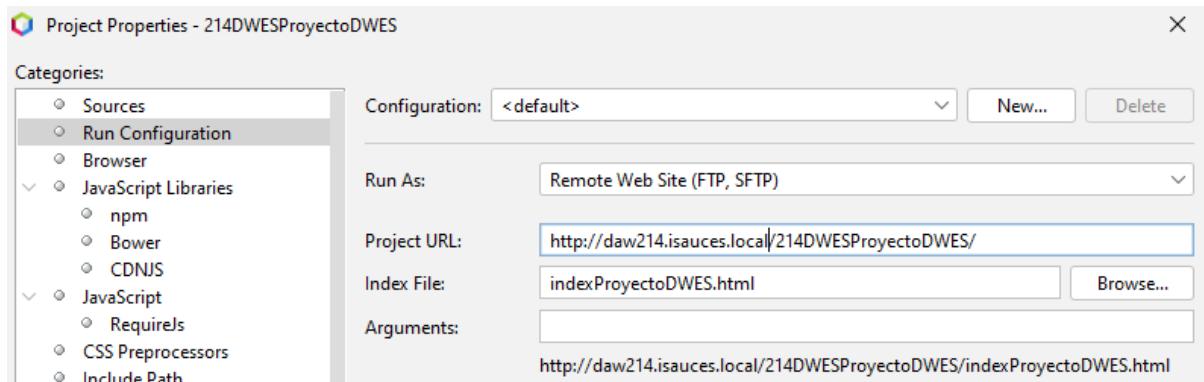
Después de llenar los campos anteriores tenemos que comprobar la conexión, para ello en la parte inferior de la misma ventana, haremos click al botón 'Test Connection'



Nos aparecerá la siguiente ventana en caso de que el test sea favorable, preguntando si queremos establecer conexión



Después desde la ventana de ‘Run Configuration’ cambiamos nuestra ‘IP’ anterior por el nuevo dominio:

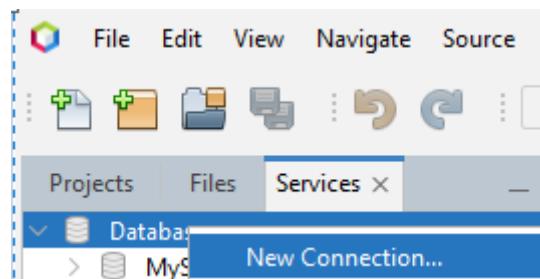


Y por último comprobamos desde un navegador la conexión con el servidor usando el dominio que hemos utilizado en la captura anterior:

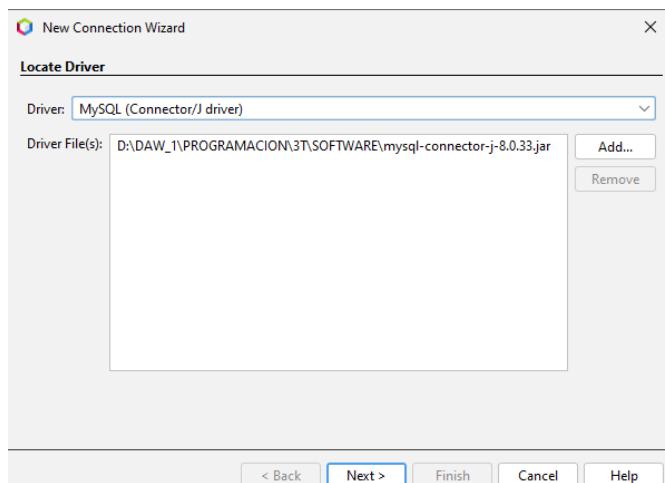


Administración de la base de datos

Tenemos que abrir NetBeans, y una vez dentro seleccionar en la parte superior izquierda ‘Services’, no aparecerán distintos servicios, hacemos click derecho en ‘Databases’ y ‘New Connection...’



Y en nos aparecerá la siguiente ventana, en la que seleccionaremos el ‘Driver’ , (Como en el paso de MySQL de la misma documentación), y le damos a ‘Next >’



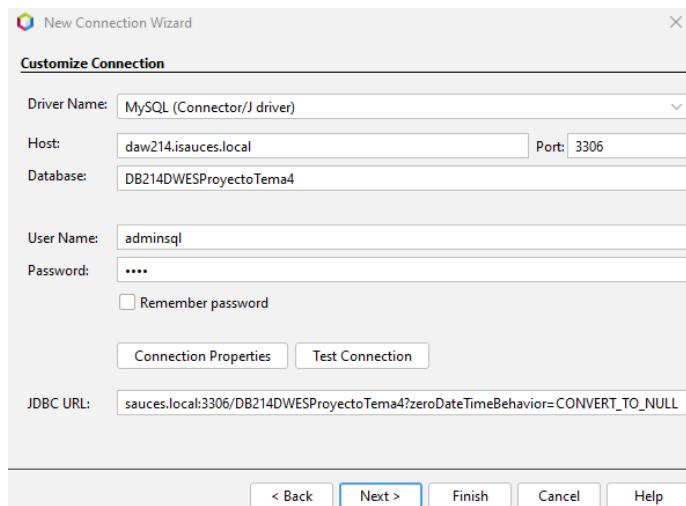
En la siguiente ventana configuraremos la conexión y nos pedirá el:

‘Host’ : Hace referencia a la IP o dominio de nuestro servidor donde tenemos instalado MySQL (En nuestro caso particular.)

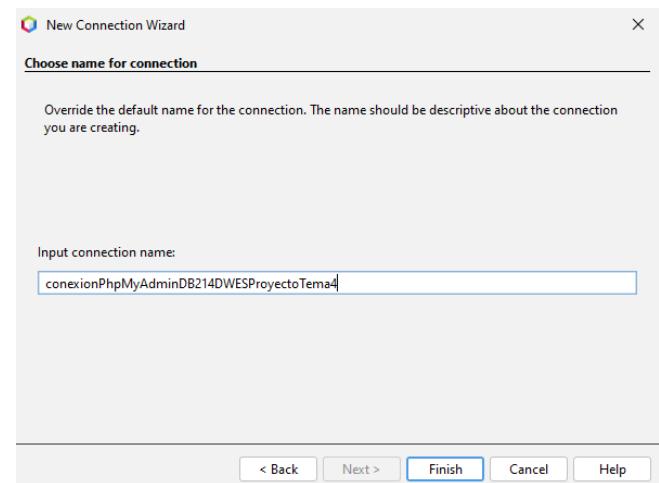
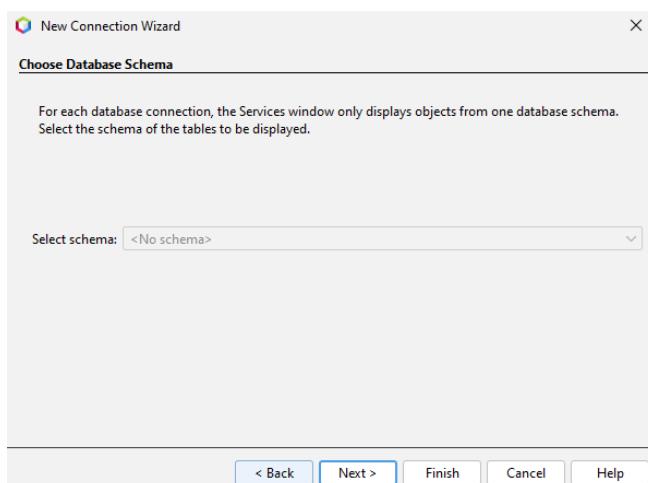
‘Database’ : Hace referencia al nombre de la BD a la que queremos conectarnos.

‘User Name’ : Hace referencia al nombre de usuario con el que accederemos a la BD.

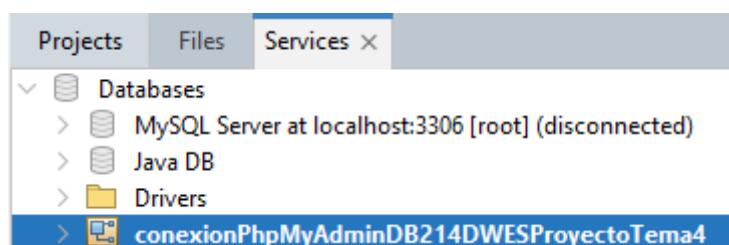
‘Password’ : Hace referencia a la contraseña del usuario con el que accedemos a la BD.



Luego pasamos a la siguiente ventana. Nos pide que seleccionemos el esquema de nuestra Base de Datos, en nuestro caso nos aparece bloqueado, así que pasamos a la última ventana, la cual nos pide que elijamos un nombre para asignar a la conexión



Cuando terminemos no aparecerá así



Conexión al repositorio – versionado

Instalación de Git

Desde el siguiente link descargaremos Git según las propiedades de nuestro equipo, en nuestro caso descargaremos la opción de Windows 64 bits, una vez descargado hacemos doble click sobre el archivo e iniciamos la instalación.

<https://git-scm.com/download/win>

Download for Windows

Click here to download the latest (2.42.0) 64-bit version of **Git for Windows**. This is the most recent **maintained build**. It was released **3 months ago**, on 2023-08-30.

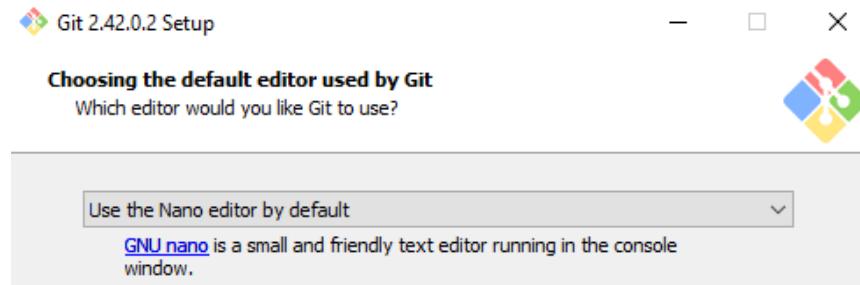
Other Git for Windows downloads

Standalone Installer

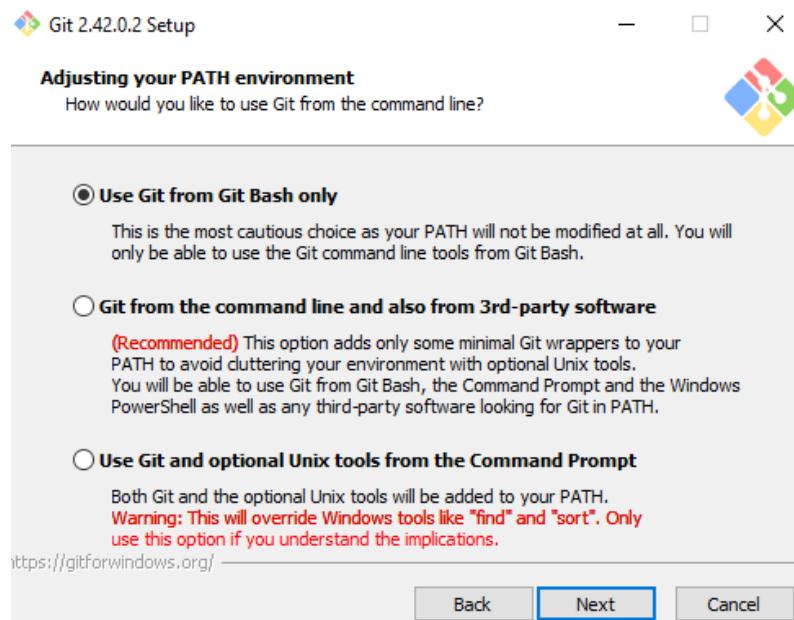
[32-bit Git for Windows Setup.](#)

[64-bit Git for Windows Setup!](#)

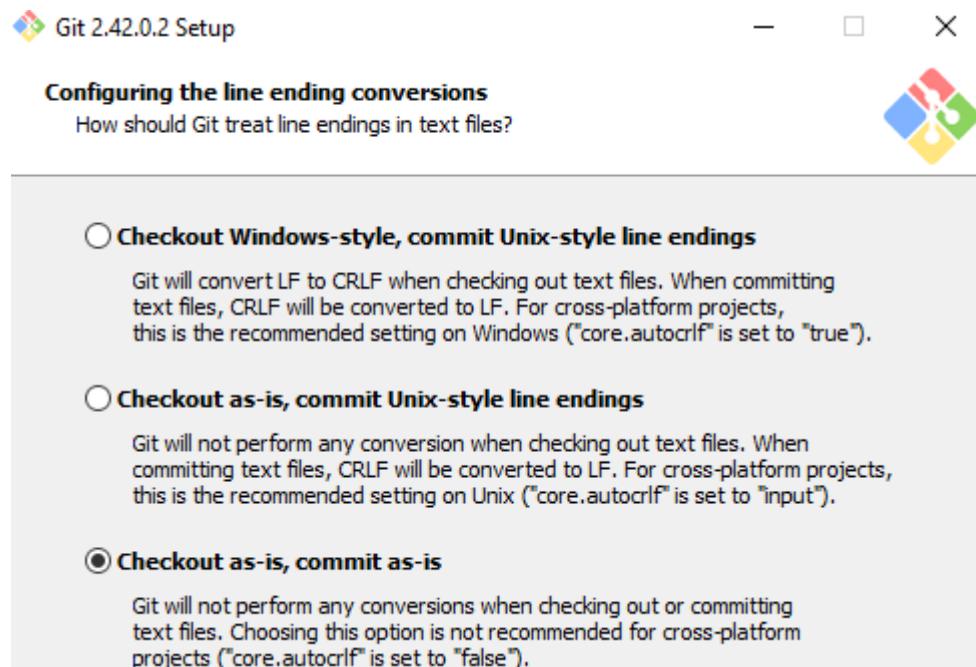
Una vez iniciada la instalación solo necesitamos editar ciertos parámetros, como en esta ventana. La cual nos pregunta que editor de texto queremos usar por defecto y le indicamos que utilice ‘Nano’.



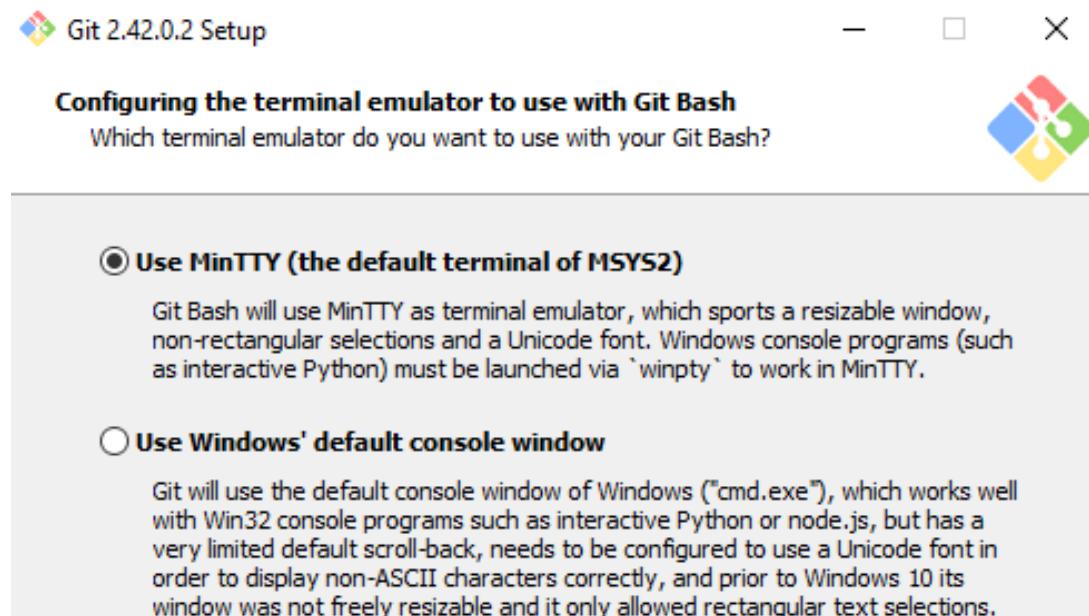
En esta nos pregunta como queremos utilizar la línea de comando, y escogemos la primera que significa que Git solo estará disponible desde la terminal Git Bash, que es una emulación de terminal para Git en sistemas Windows.



En la siguiente ventana con la última opción le decimos que significa que Git no realizará ninguna conversión automática de terminaciones de línea. Los archivos se manejarán tal como están en tu sistema local sin hacer cambios al realizar un checkout y al hacer commits.

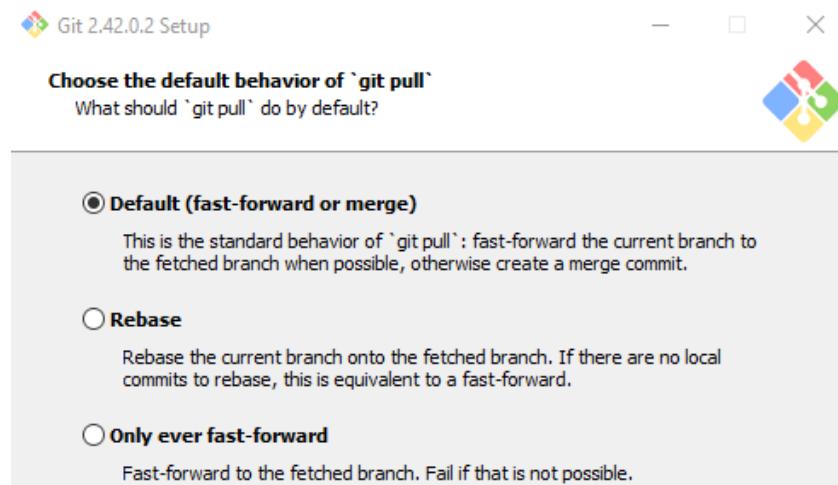


En esta nos pregunta que emulador de terminal queremos usar con Git Bash, seleccionamos la opción que pone 'Use MinTTY (the default terminal of MSYS2)', que es el emulador de terminal predeterminado para MSYS2, que es un sistema de emulación de entorno Unix para Windows que se utiliza junto con Git Bash.

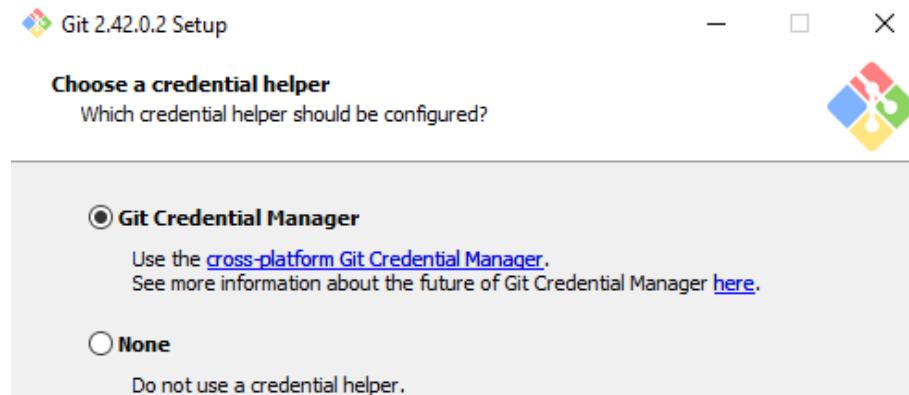


Y en la siguiente ventana nos pregunta que escogemos el comportamiento de 'git pull' y escogemos la que dice 'Default (fast-forward or merge)', esta opción significa que Git intentará hacer un "fast-forward" cuando sea posible. Un "fast-forward" es una operación de combinación simple en la que los cambios realizados en la rama remota se aplican directamente a tu rama local si no has realizado

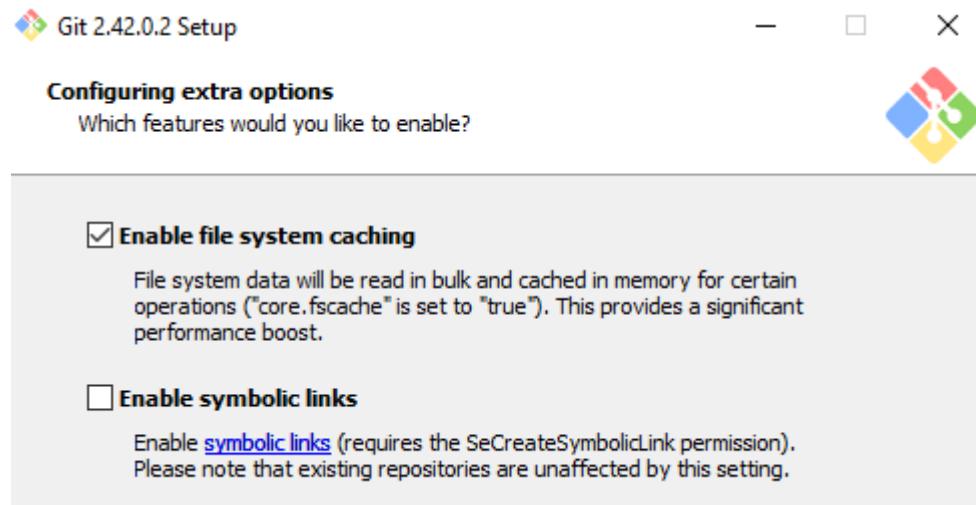
cambios locales que entrarían en conflicto. Si no es posible hacer un "fast-forward", Git realizará una fusión.



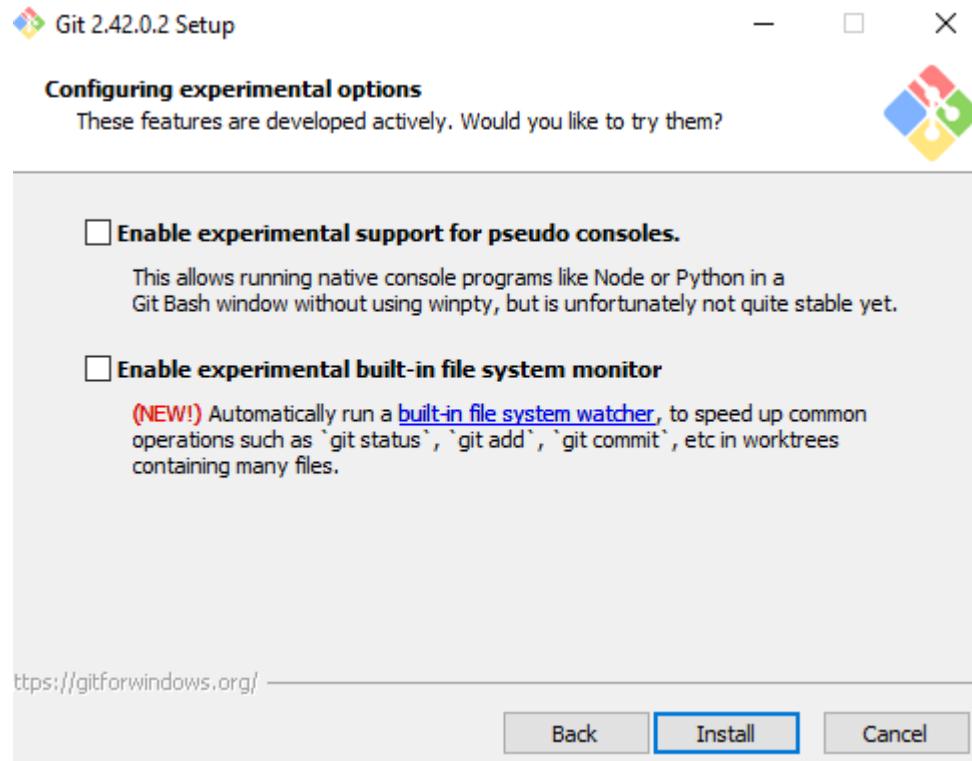
La siguiente nos habla sobre la credenciales de ayuda y viene seleccionada 'Git Credential Manager' por defecto que es una herramienta que ayuda a gestionar las credenciales de autenticación al interactuar con repositorios Git que requieren autenticación.



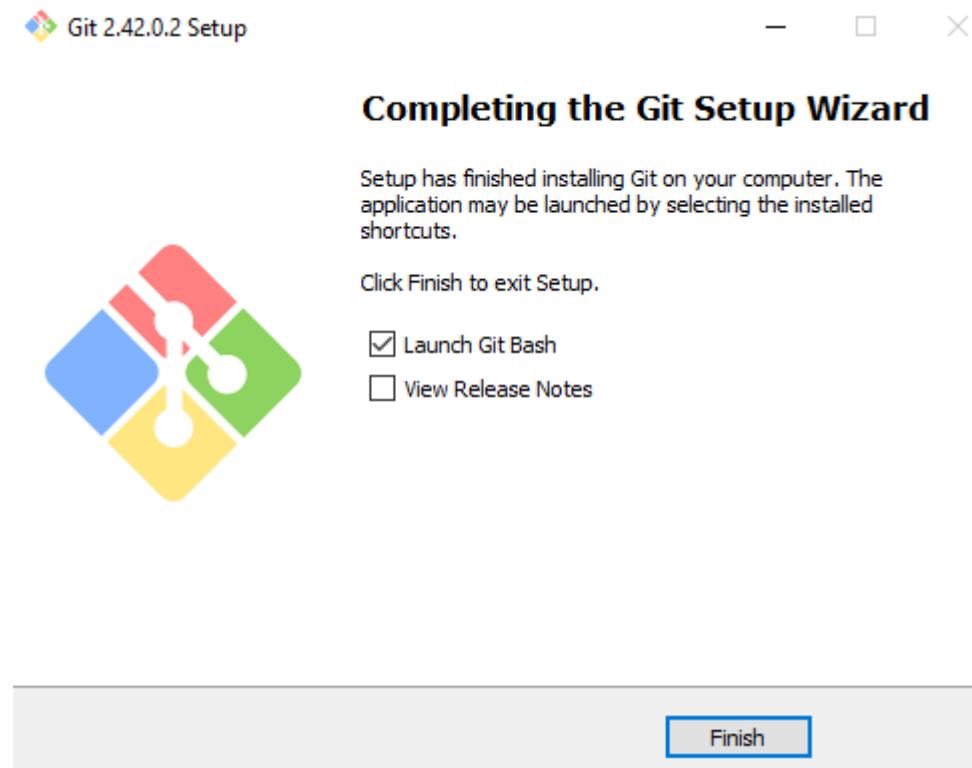
En la última ventana que tenemos que seleccionar la configuración de opciones extra, seleccionaremos la que dice 'Enable file system caching' que se refiere a la posibilidad de habilitar el almacenamiento en caché del sistema de archivos. Esta característica está relacionada con la velocidad y rendimiento de Git al acceder y manipular archivos en el sistema de archivos.



Y por último instalamos Git



Antes de finalizar seleccionamos el siguiente checkbox para pasar a la configuración de credenciales



Configuración de Git

Si hemos seguido el paso anterior de manera correcta, se nos abrirá un ‘Git Bash’ y con los siguientes comandos le especificaremos a Git cuales es nuestro nombre de usuario y correo en Git Hub, que para ello necesitaremos estar registrados.

```
git config --global user.name 'nombreUsuario'
```

```
git config --global user.email 'correo'
```

```
daw2@IS32WX14 MINGW64 ~/Desktop  
$ git config --global user.name Fighter-kun
```

```
daw2@IS32WX14 MINGW64 ~/Desktop  
$ git config --global user.email carlosgarciacachon@gmail.com
```

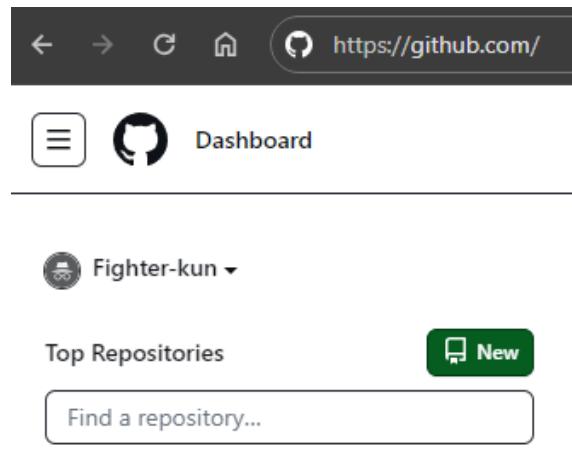
Y para visualizar los cambios usaremos el siguiente comando

```
git config --global --list
```

```
daw2@IS32WX14 MINGW64 ~/Desktop  
$ git config --global --list  
user.email=carlosgarciacachon@gmail.com  
user.name=Fighter-kun
```

Crear repositorio

Usaremos GitHub para realizar los versionados de nuestros proyectos, (suponiendo que tenemos una cuenta, si no debemos registrarnos), luego accederemos con nuestra cuenta. (Podemos usar este link: <https://github.com/>) Y en la parte donde dice ‘Top Repositories’ hacemos click en el botón azul que pone ‘New’.



En la siguiente ventana seleccionaremos el nombre de nuestro nuevo repositorio e indicaremos distintos parámetros

Si es 'Public' o 'Private', en nuestro caso indicaremos que público.

Indicaremos que no queremos inicializar el repositorio sin nada, si no nos crearía por defecto un 'README.md' o un '.gitignore' o indicar el tipo de licencia.

En nuestro caso le daremos a 'Create repository' sin cambiar ningún parámetro más.

Lo siguiente que nos aparecerá será el propio repositorio, aunque vacío, GitHub nos muestra comandos para acceder y subir cosas a nuestro nuevo repositorio

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * Repository name *

 Fighter-kun / 214DAWProyectoDAW

214DAWProyectoDAW is available.

Great repository names are short and memorable. Need inspiration? How about effective-robot ?

Description (optional)

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. Learn more about READMEs.

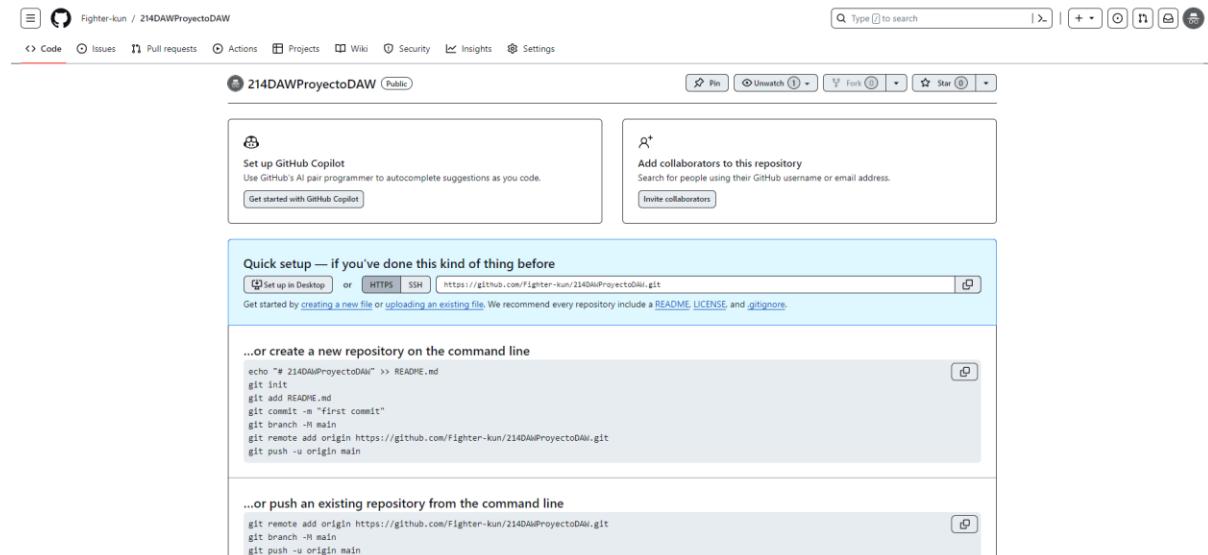
Add .gitignore

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

A license tells others what they can and can't do with your code. Learn more about licenses.

 You are creating a public repository in your personal account.



Iniciar repositorio

Podemos hacerlo de dos maneras, una por comando o otra de manera gráfica usando un IDE.

COMANDO:

Nos posicionaremos en la carpeta donde vayamos a iniciar el repositorio, haremos click derecho y hacemos click en 'Open Git Bash Here' y nos aparecerá esta ventana



En ella utilizaremos el comando 'git init' para iniciar el repositorio

```
git init
```

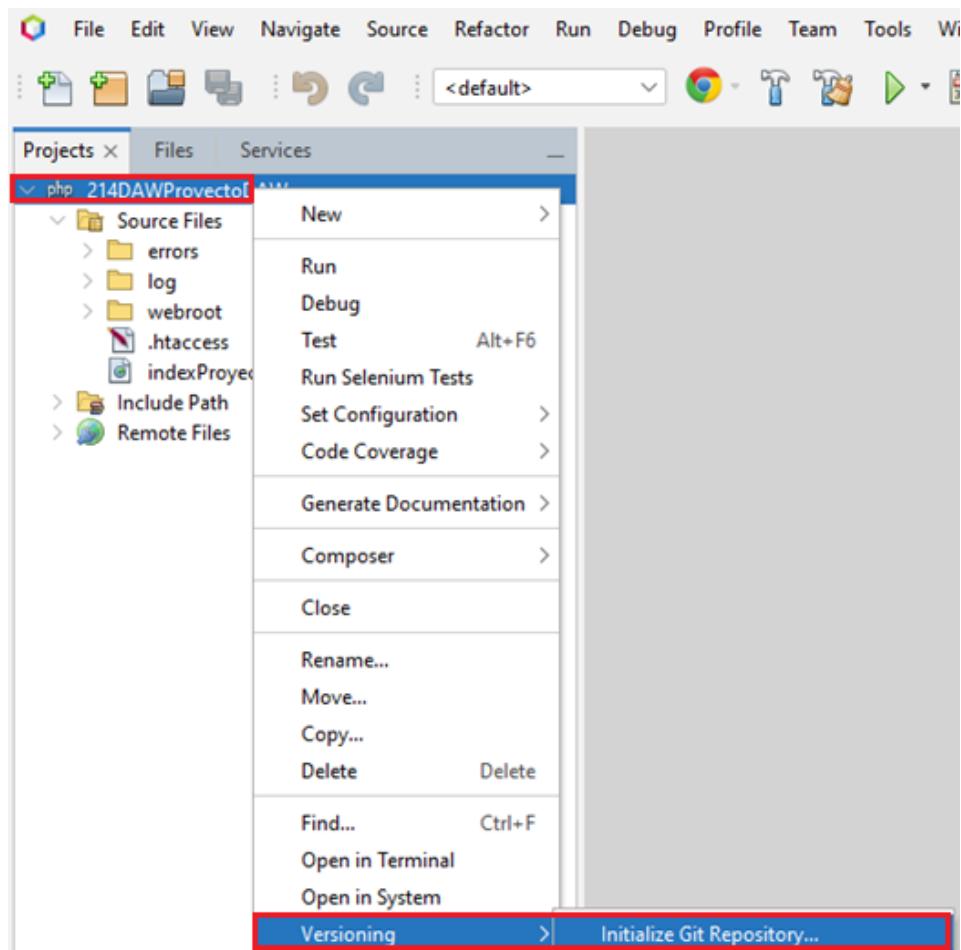
```
MINGW64:/d/ProyectosNB/214DAWProyectoDAW  
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214DAWProyectoDAW  
$ git init
```

Y nos aparecerá esta carpeta oculta

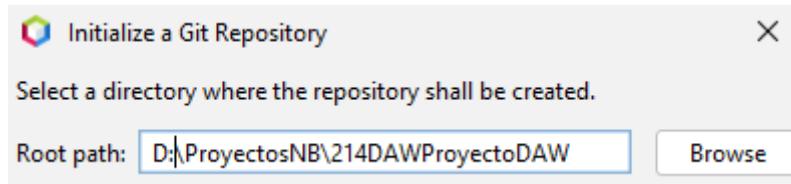


IDE:

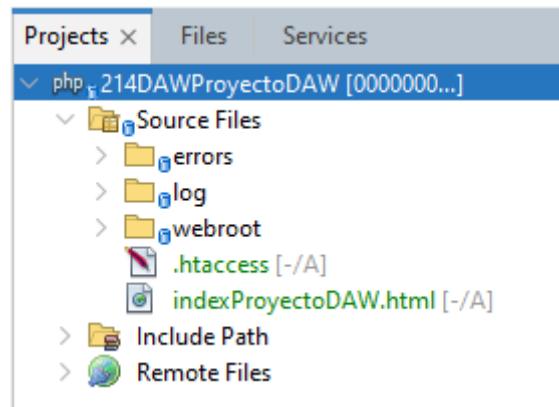
Ahora tendremos que abrir nuestro IDE (en nuestro caso NetBeans 19) e iniciar conexión con nuestro repositorio



Nos aparecerá una ventana donde nos preguntará el Path del directorio donde vamos a iniciar el repositorio

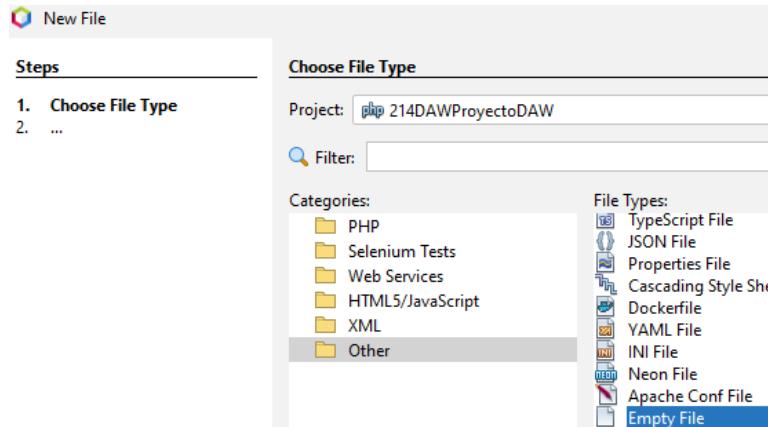


Y nos aparecerá así una vez inicializado

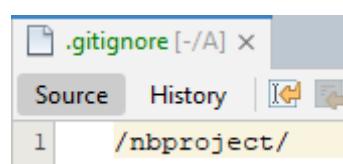


Una vez inicializado el repositorio crearemos los 2 archivos que no creamos antes al inicializar el repositorio, primero el '.gitignore' que sirve para especificar archivos y directorios que deben ser ignorados, es decir, que no deben ser rastreados o versionados por Git.

Crearemos un archivo vacío y le pondremos el nombre siguiente nombre '.gitignore':

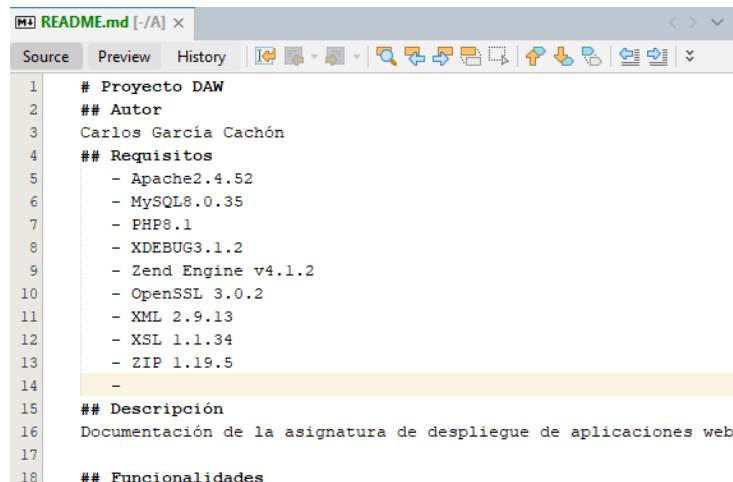


E introducimos la ruta de la carpeta que queremos que ignore, en este caso es la carpeta que se crea por defecto al crear un proyecto en NetBeans



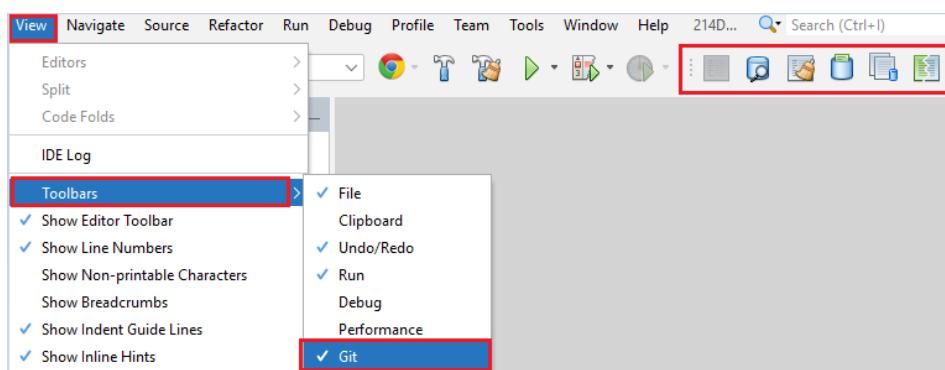
Y con el mismo procedimiento creamos otro archivo vacío llamado ‘README.md’ , el cual es un archivo de texto con formato Markdown que proporciona información sobre nuestro proyecto.

En nuestro caso vamos a añadir los ‘Requisitos’ que serían los módulos que hemos instalado en nuestro servidor (Tiene una estructura determinada, como en HTML, para determinar un H1 aquí usaremos '#' y para H2 ‘##’ , por último para hacer una lista con viñetas usaremos ‘-’.)



```
# Proyecto DAW
## Autor
Carlos Garcia Cachón
## Requisitos
- Apache2.4.52
- MySQL8.0.35
- PHP8.1
- XDEBUG3.1.2
- Zend Engine v4.1.2
- OpenSSL 3.0.2
- XML 2.9.13
- XSL 1.1.34
- ZIP 1.19.5
-
## Descripción
Documentación de la asignatura de despliegue de aplicaciones web
## Funcionalidades
```

Ahora para hacer nuestro IDE más dinámico de cara al uso de GitHub, añadiremos unas opciones que ya vienen por defecto instaladas pero que permanecen ocultas, para mostrarlas accederemos a la barra principal, haremos click en ‘View’ / ‘Toolbars’ / ‘Git’ y las haremos visibles

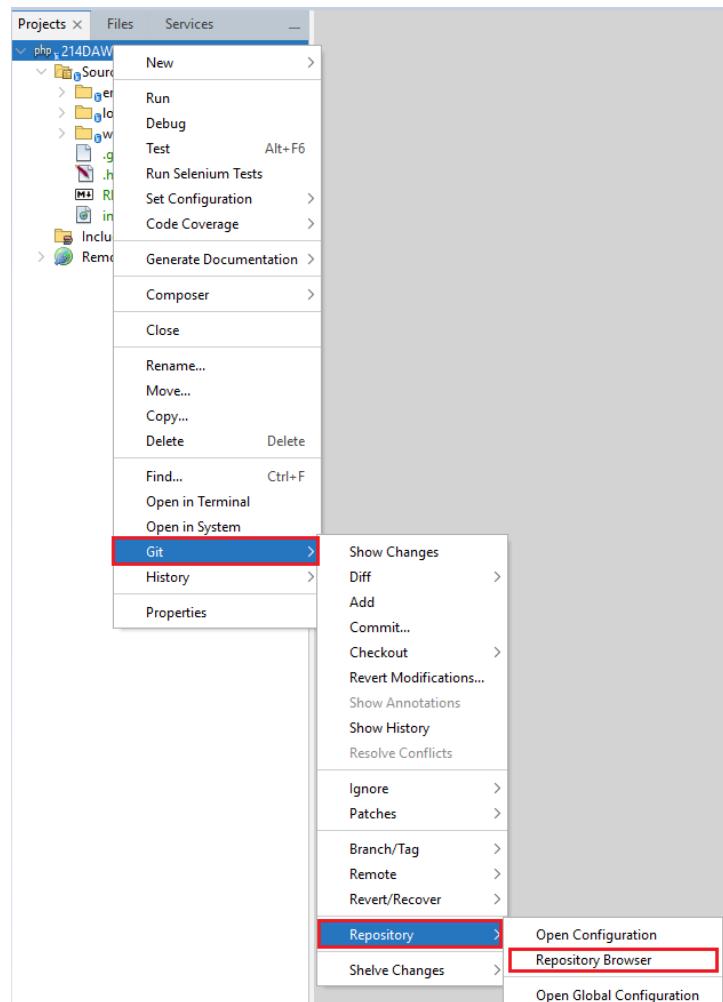


(En orden de aparición) Las cuales sirven para:

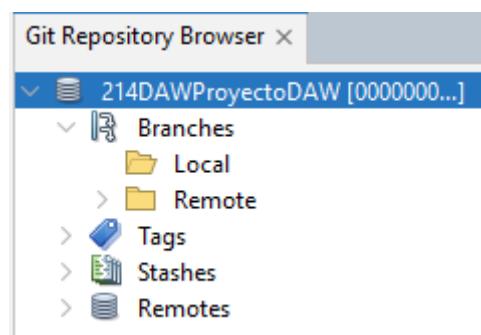
1. Show Annotations: Esta opción te permite ver quién o qué commit modificó cada línea de código en tu archivo. Es útil para rastrear la historia de cada línea y entender quién hizo qué cambios y cuándo.
2. Show History: Muestra el historial de cambios del archivo. Puedes ver todos los commits relacionados con el archivo seleccionado, sus mensajes de commit y las fechas en que se realizaron.
3. Revert Modifications: Permite revertir los cambios locales que aún no han sido confirmados (committed). Útil si realizaste modificaciones y decides que quieres deshacerlas.
4. Commit: Te permite confirmar (commit) los cambios locales en tu repositorio Git. Al hacer clic en esta opción, se abrirá una ventana donde puedes escribir un mensaje de commit y confirmar los cambios.

5. Show Changes: Muestra las diferencias (diff) entre el estado actual de tu archivo y la versión más reciente confirmada en tu repositorio Git. Esto te da una vista detallada de las modificaciones que has realizado.
6. Diff to Head: Similar a "Show Changes", esta opción muestra las diferencias entre tu archivo local y la versión más reciente confirmada en tu rama actual.

Otro paso más a tener en cuenta sería añadir el navegador del repositorio que nos brindara mucha información sobre las ramas, etc. Haremos click derecho sobre el proyecto y seleccionaremos '/Git' / 'Repository' / 'Repository Browser'



Y nos aparecerá en la barra de abajo el navegador



Commit

Podemos hacerlo de dos maneras, una por comando y otra de manera gráfica usando un IDE.

COMANDO:

Abrimos el Git Bash como hemos hecho antes y con el siguiente comando añadimos todos los cambios realizados en el directorio de trabajo al área de preparación. (Podemos sustituir el '.' por el nombre de un archivo en concreto, pero si quisieramos hacerlo de todos, sería de la siguiente manera)

```
git add .
```

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214DAWProyectoDAW
$ git add .
```

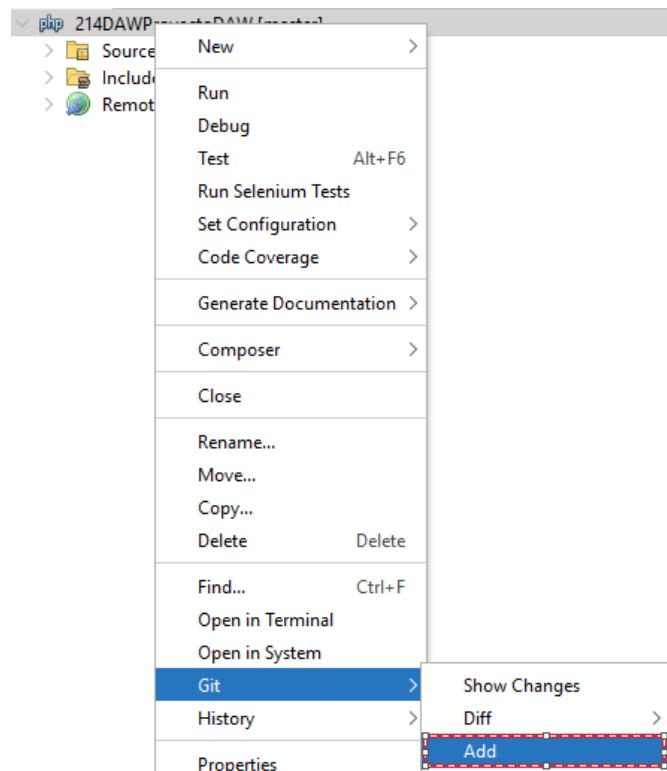
Y luego procederíamos a realizar un commit con el siguiente comando. (El parámetro '-m' es para indicar que el commit se va a realizar con un mensaje)

```
git commit -m "NOMBRE_DEL_COMMIT"
```

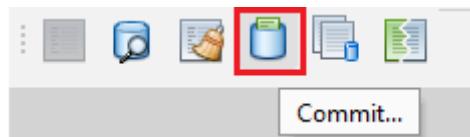
```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214DAWProyectoDAW
$ git commit -m "Commit de Prueba"
```

IDE:

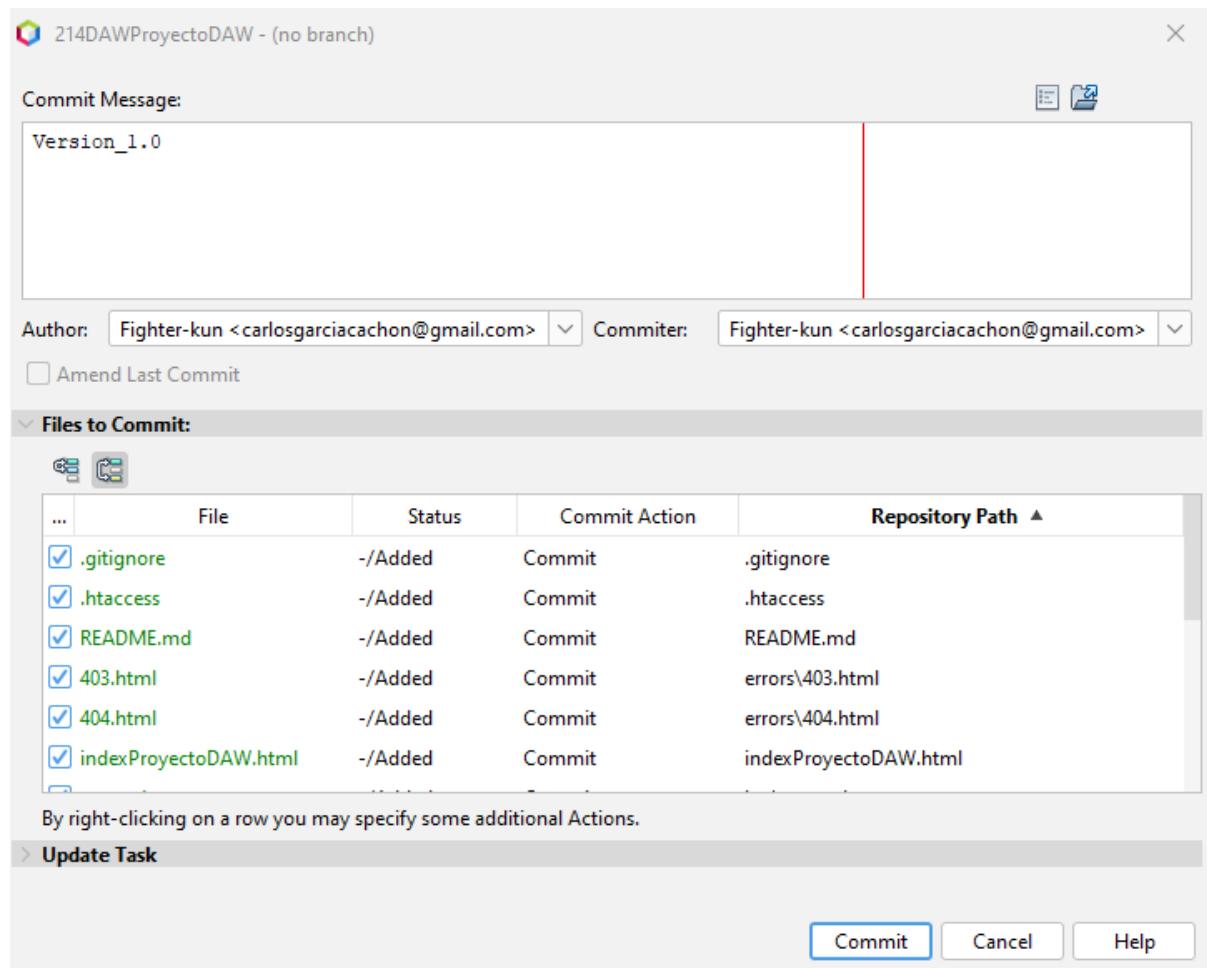
Ahora procederemos a realizar nuestro primer 'Commit' que sería como tomar una instantánea de los cambios que has realizado en tu proyecto y guardarlos en tu repositorio. Para ello primero deberemos agregar los cambios del proyecto al área de preparación (Staging Area) para que incluya los cambios específicos en los archivos en el próximo commit. Entonces hacemos click derecho sobre nuestro proyecto 'Git' / 'Add'



Luego ayudándonos de los iconos que hemos añadido anteriormente y presionamos el que dice 'Commit'



Nos aparecerá la siguiente ventana en la que indicaremos el mensaje de 'Commit' y añadiremos el nombre del autor y el correo asociada a la cuenta



En la parte baja de la ventana veremos todos los archivos menos lo que hemos incluido en el '.gitignore'

Push y generar token

Podemos hacerlo de dos maneras, una por comando y otra de manera gráfica usando un IDE.

COMANDO:

Desde la ventana de Git Bash anterior, usamos el siguiente comando

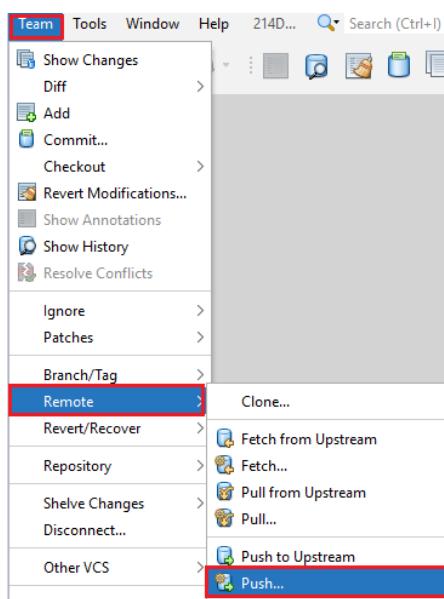
```
git push origin
```

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214DAWProyectoDAW
$ git push origin
```

IDE:

Una vez realizamos correctamente el ‘Commit’ , deberemos hacer un ‘Push’ que es básicamente enviar tus commits locales al repositorio remoto en GitHub.

Para ello accederemos desde la parte superior a ‘Team’ / ‘Remote’ / ‘Push...’



Y nos aparecerá una ventana en la que tendremos que añadir los siguientes parámetros

‘Repository URL’ : Aquí debemos introducir la URL que nos generó al crear el repositorio en la página de GitHub.

‘User’ : El usuario con el que accedes a GitHub

‘Password’ : Aquí tenemos que añadir un token que se genera desde la configuración de la cuenta de GitHub desde el navegador, lo normal es que si es una cuenta nueva no tengamos o quizás la tengamos caducada por no usarse.

Antes de seguir configuraremos nuestro token, nos iremos al siguiente link

<https://github.com/settings/tokens> y nos aparecerá la opción de generar un nuevo token y eliminar los que tengamos. (En mi caso ya tengo creado uno que no caduca, pero vamos a crear otro.)

Hacemos click en ‘Generate new token’ y en el desplegable la opción ‘classica’

The screenshot shows the GitHub 'Personal access tokens (classic)' page. On the left, there's a sidebar with 'GitHub Apps', 'OAuth Apps', and 'Personal access tokens' (which is expanded, showing 'Fine-grained tokens' and 'Tokens (classic)'). The main area is titled 'Personal access tokens (classic)' and contains a table of tokens. At the top right are 'Generate new token' (highlighted with a red box), 'Revoke all', and 'Delete'. A tooltip for 'Generate new token' says 'Generate new token (Beta) Fine-grained, repo-scoped'. Another tooltip for 'Generate new token (classic)' says 'For general use'. A note at the bottom states: 'Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.' A warning below it says: '⚠️ This token has no expiration date.'

En la pestaña que nos abrirá tendremos que seleccionar el nuevo nombre de nuestro token y cuando queremos que caduque, por motivos de control y seguridad, de 7 días a 90 días , personalizado o incluso que no caduque nunca.

The screenshot shows the 'New personal access token (classic)' creation form. It includes fields for 'Note' (with placeholder 'NombreDelNuevoToken'), 'What's this token for?' (with placeholder 'What's this token for?'), and 'Expiration *'. The 'Expiration' dropdown is open, showing options: '30 days' (selected), '7 days', '30 days', '60 days', '90 days', 'Custom...', and 'No expiration'. A tooltip for '30 days' says 'The token will expire on Sat, Dec 9 2023'. Below the dropdown, there are sections for 'Access for personal tokens' (with links to 'Read more about OAuth scopes.', 'Full control of private repositories', and 'Access commit status').

Las opciones de abajo la configuramos de la siguiente manera (para nuestro caso particular)

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	Lea la membresía de la organización y del equipo, lea 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	Ler 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	Ler y escribir discusiones en equipo.
<input checked="" type="checkbox"/> leer: discusión	Ler 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	Ler y escribir datos de facturación empresarial
<input type="checkbox"/> leer:empresa	Ler datos del perfil empresarial
<input type="checkbox"/> registro de auditoría	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 checked="" 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	Ler 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 checked="" type="checkbox"/> escribir:ssh_signing_key	Escribir claves de firma SSH de usuario público
<input checked="" type="checkbox"/> leer:ssh_signing_key	Ler claves de firma SSH de usuarios públicos

Los 'scopes' son permisos que se pueden asignar a un token personal en GitHub. Aquí hay un resumen rápido de algunos de los 'scopes' que puedes seleccionar al crear un token:

- 'repo': Control completo de repositorios privados.
- 'repo:status': Acceso al estado de los commits.
- 'public_repo': Acceso a repositorios públicos.
- 'security_events': Lectura y escritura de eventos de seguridad.
- 'workflow': Actualización de flujos de trabajo de GitHub Actions.

- 'write:packages': Subir paquetes al Registro de paquetes de GitHub.
- 'read:packages': Descargar paquetes del Registro de paquetes de GitHub.
- 'delete:packages': Eliminar paquetes del Registro de paquetes de GitHub.
- 'admin:org': Control total de organizaciones y equipos.
- 'write:org': Lectura y escritura de membresía de organización y equipo.
- 'read:org': Lectura de membresía de organización y equipo.
- 'admin:repo_hook': Control total de ganchos (hooks) de repositorios.
- 'write:repo_hook': Escritura de ganchos (hooks) de repositorios.
- 'read:repo_hook': Lectura de ganchos (hooks) de repositorios.
- 'gist': Crear gists.
- 'notifications': Acceso a notificaciones.
- 'user': Actualizar todos los datos de usuario.
- 'read:user': Lectura de todos los datos de perfil de usuario.
- 'user:email': Acceso a las direcciones de correo electrónico del usuario (solo lectura).
- 'delete_repo': Eliminar repositorios.
- 'write:discussion': Leer y escribir discusiones de equipos.
- 'read:discussion': Leer discusiones de equipos.
- 'admin:enterprise': Control total de empresas.
- 'codespace': Control total de Codespaces.
- 'codespace:secrets': Crear, leer, actualizar y eliminar secretos de Codespaces.
- 'copilot': Control total de la configuración y asignaciones de GitHub Copilot.
- 'manage_billing:copilot': Ver y editar asignaciones de asientos de GitHub Copilot para Empresas.
- 'project': Control total de proyectos.
- 'read:project': Lectura de proyectos.
- 'admin:gpg_key': Control total de claves GPG públicas del usuario.
- 'write:gpg_key': Escritura de claves GPG públicas del usuario.
- 'read:gpg_key': Lectura de claves GPG públicas del usuario.
- 'admin:ssh_signing_key': Control total de claves de firma SSH públicas del usuario.
- 'write:ssh_signing_key': Escritura de claves de firma SSH públicas del usuario.
- 'read:ssh_signing_key': Lectura de claves de firma SSH públicas del usuario.

Y al darle a crear token nos mostrará el código que debemos de copiar y almacenar en un lugar seguro, es muy importante no perderlo:

Personal access tokens (classic)

[Generate new token ▾](#) [Revoke all](#)

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

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

✓ 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 Copy	Delete
---	------------------------

Una vez creado el token y con el código copiado podemos seguir con el 'Push', y lo rellenamos

Push to Remote Repository

Steps	Remote Repository
1. Remote Repository 2. Select Local Branches 3. Update Local References	<input type="radio"/> Select Configured Git Repository Location: <input checked="" type="radio"/> Specify Git Repository Location: Remote Name: origin <input type="button" value="▼"/> <input checked="" type="checkbox"/> Persist Remote Repository URL: https://github.com/Fighter-kun/214DAWProyectoDAW.git http[s]://host.xz[:port]/path/to/repo.git/ User: Fighter-kun (leave blank for anonymous access) Password: <input checked="" type="checkbox"/> Save Password

La siguiente ventana nos aparecerán las ramas, en nuestro caso al solo tener una la seleccionamos y siguiente

Push to Remote Repository

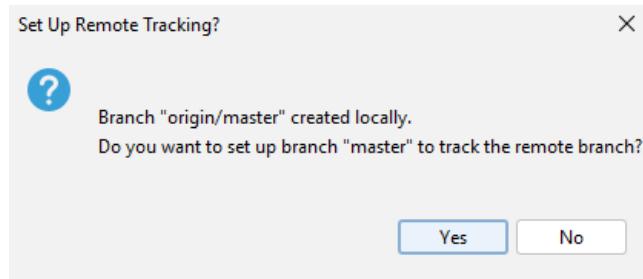
Steps	Select Local Branches
1. Remote Repository 2. Select Local Branches 3. Update Local References	Local Branches <input checked="" type="checkbox"/> master -> master [A]

Al igual que la siguiente

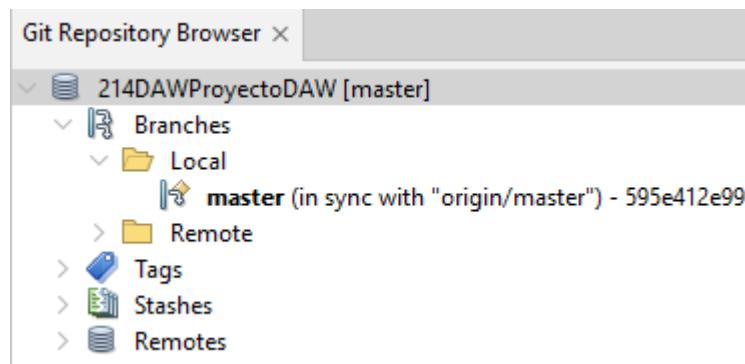
Push to Remote Repository

Steps	Update Local References
1. Remote Repository 2. Select Local Branches 3. Update Local References	Select remote branches you want: Remote Branches <input checked="" type="checkbox"/> master -> origin/master [A]

Nos aparecerá una ventana emergente en la que nos dice si queremos configurar la rama 'master' para hacer seguimiento de la rama remota y le damos 'Yes'



Ahora en el Navegador de Git nos aparece el movimiento



Y en nuestro repositorio dentro de la página de GitHub nos aparecerá así, con los archivos de cuando realizamos el 'commit' anteriormente y el archivo 'README.md' ejecutándose de manera automática

Code

Fighter-kun / 214DAWProyectoDAW

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

214DAWProyectoDAW (master)

master 1 branch 0 tags

Go to file Add file Code

File	Version	Commit
errors	Version_1.0	44 minutes ago
log	Version_1.0	44 minutes ago
webroot	Version_1.0	44 minutes ago
.gitignore	Version_1.0	44 minutes ago
.htaccess	Version_1.0	44 minutes ago
README.md	Version_1.0	44 minutes ago
indexProyectoDAW.html	Version_1.0	44 minutes ago

README.md

Proyecto DAW

Autor: Carlos García Cachón

Requisitos:

- Apache2.4.52
- MySQL8.0.35
- PHP8.1
- XDEBUG1.1.2
- Zend Engine v4.1.2
- OpenSSL 3.0.2
- XML 2.9.13
- XSL 1.1.34
- ZIP 1.19.5

Descripción: Documentación de la asignatura de despliegue de aplicaciones web

Funcionalidades

Create Branch

Podemos hacerlo de dos maneras, una por comando y otra de manera gráfica usando un IDE.

COMANDO:

Desde la ventana de Git Bash anterior, usamos el siguiente comando, el cual con ‘checkout’ indicamos que queremos cambiarnos a esta nueva rama y con ‘-b’ indicamos que la cree, también, podríamos crear primero la rama y luego movernos a ella (Crearla: `git branch 'NOMBRE_DE_LA_RAMA'` Movernos: `git checkout 'NOMBRE_DE_LA_RAMA'`)

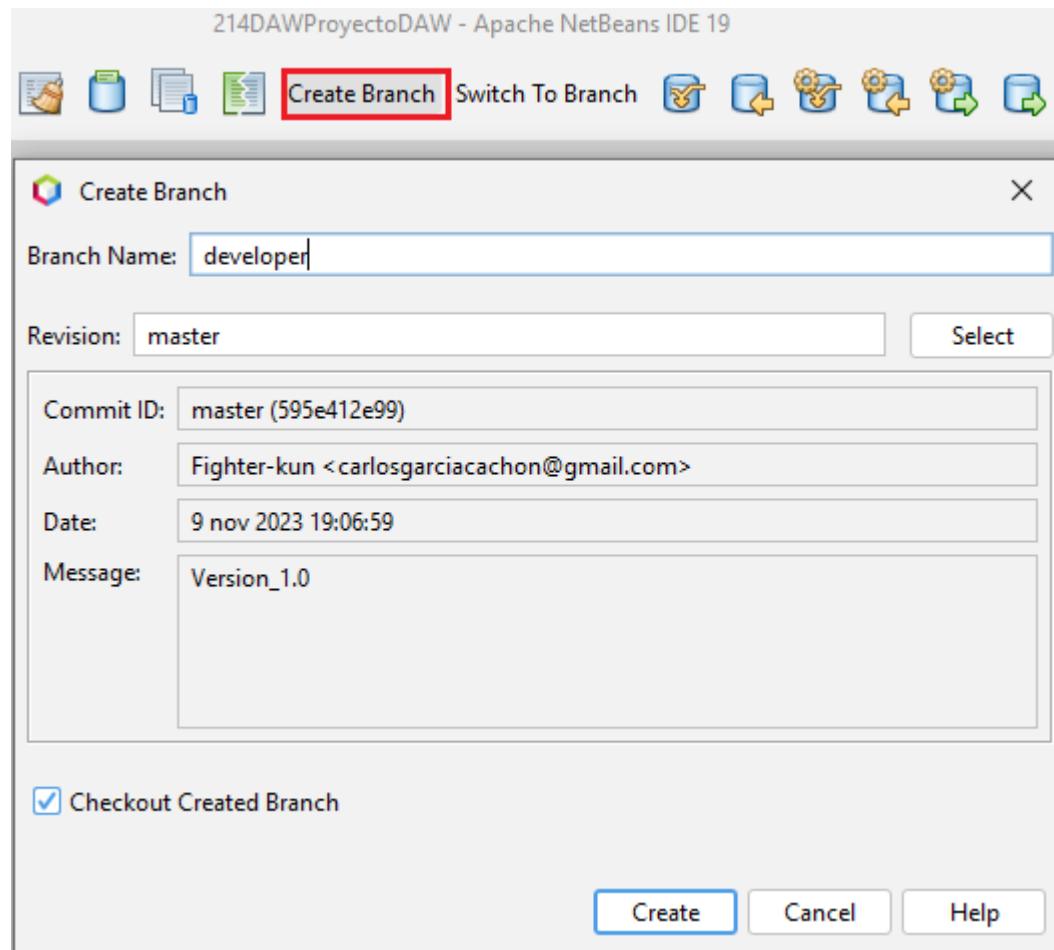
```
git checkout -b 'NOMBRE_DE_LA_RAMA'
```

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214DAWProyectoDAW
$ git checkout -b developer
```

IDE:

La creación de otra rama de trabajo en Git es una práctica común que ofrece varios beneficios en el desarrollo de software y en nuestro caso crearemos una rama llamada “developer” para trabajar directamente en ella si usar la “master”.

Con nuestro proyecto seleccionado, ayudándonos de los iconos que hemos añadido anteriormente, presionamos “Create Branch” y nos aparecerá una ventana con distinta información:



Branch Name: Nombre de la rama que creamos.

Revision: La opción de ‘Revision’ permite seleccionar el punto de partida nuestra nueva rama. Con nuestra selección estamos indicando que queremos basar nuestra nueva rama en el estado actual de la rama principal.

Commit ID : Es una cadena única de caracteres que identifica de manera exclusiva un commit específico.

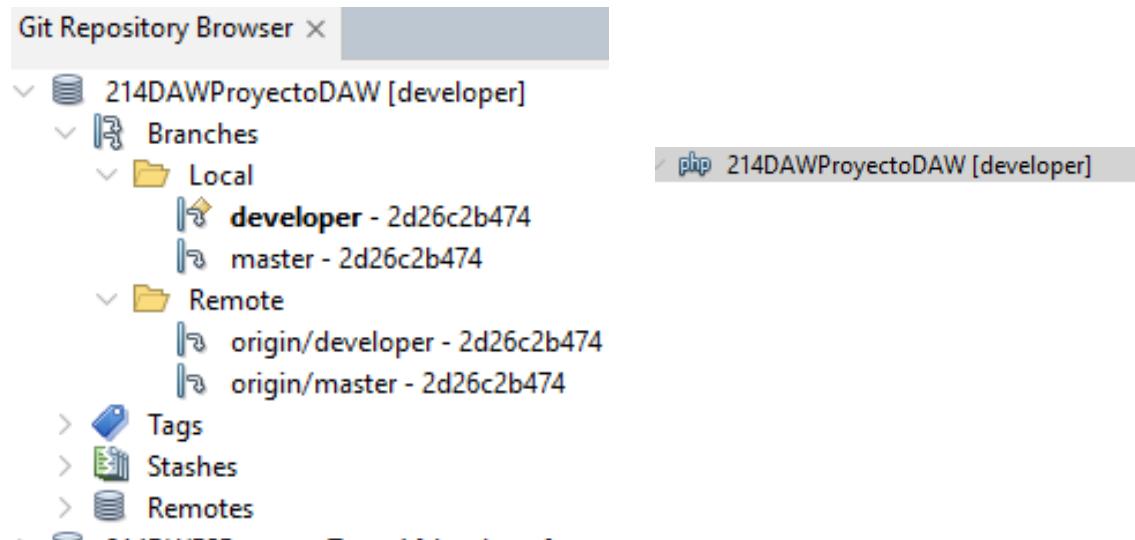
Author: Hace referencia al usuario y correo de nuestra cuenta de Git Hub.

Date: Nos muestra la fecha y hora en la que creamos la nueva rama.

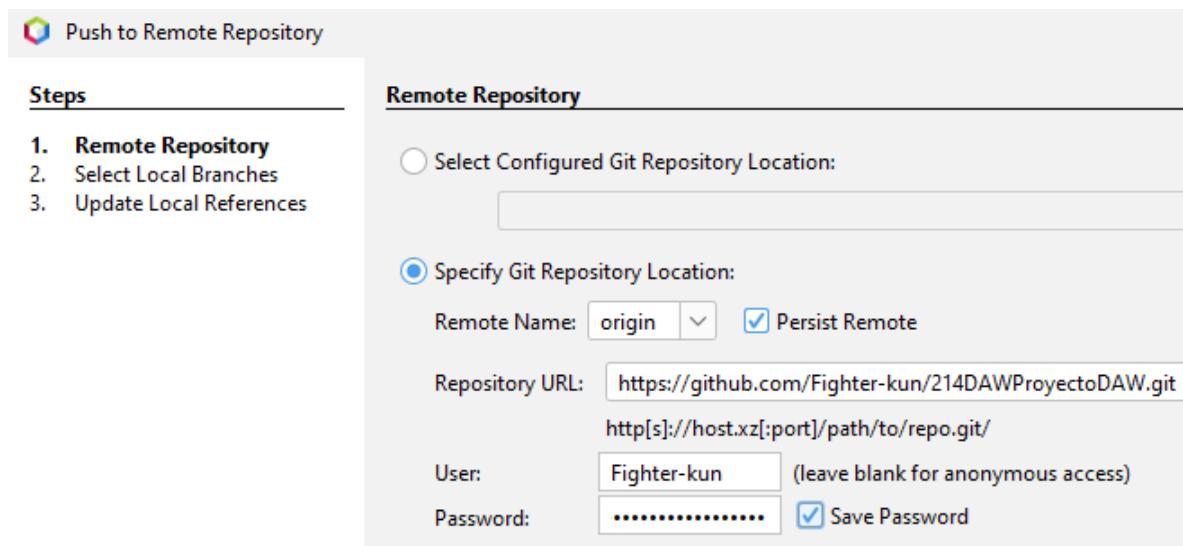
Message: Muestra el mensaje del último ‘commit’ realizado.

Checkout Created Branch: Si el checkbox está marcado, cuando hagamos click en ‘Create’ , no moveremos automáticamente a la nueva rama.

Una vez creado, podremos visualizar la rama desde el ‘Git Repository Browser’ y vemos que se está en local y remoto. También podemos ver que hemos cambiado de rama y ahora trabajamos desde la rama “developer”



Ahora debemos hacer de nuevo un “Push” , para subir el mismo contenido que hemos subido a la rama “master” , con la diferencia que desde ahora todos los cambios que realizaremos los haremos desde la rama “developer” . Seguiremos los pasos anteriores, con la diferencia que elegiremos la nueva rama para realizar el “Push”



Seleccionamos nuestra nueva rama en ambas ventanas y realizamos el “Push” correctamente, es posible que aparezca algún error y es recomendable hacer un “Pull”

Pull

Podemos hacerlo de dos maneras, una por comando y otra de manera gráfica usando un IDE.

COMANDO:

Desde la ventana de Git Bash anterior, usamos el siguiente comando, donde tengo ‘developer’ sería la rama que queremos actualizar en nuestro repositorio local

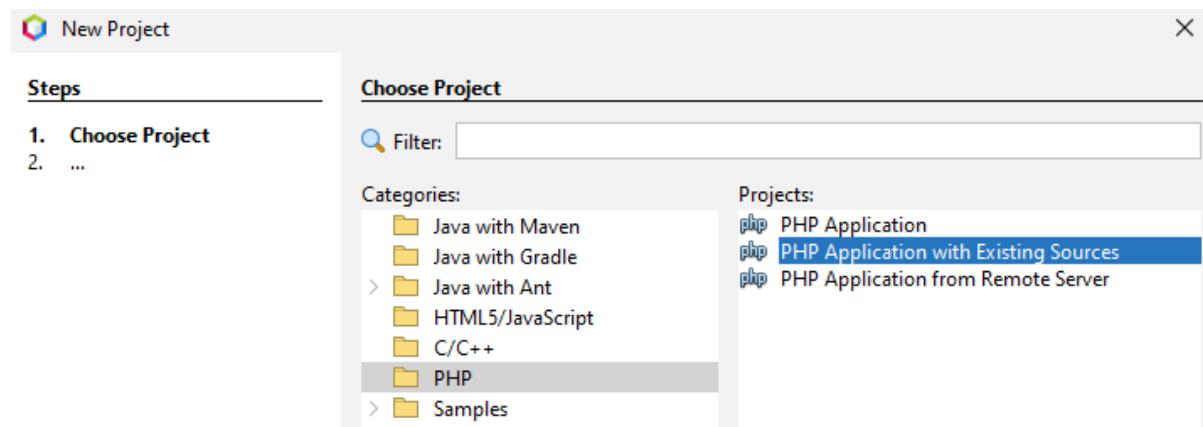
```
git pull origin developer
```

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214DAWProyectoDAW (developer)
$ git pull origin developer|
```

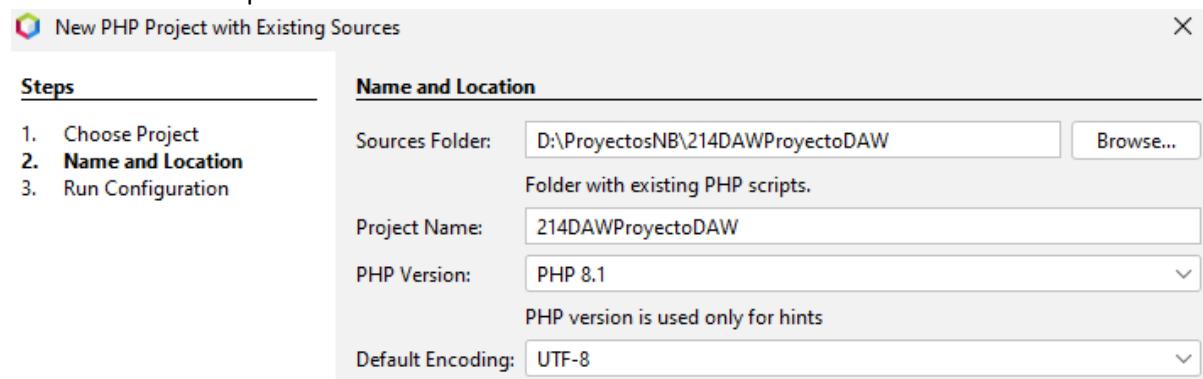
IDE:

Ahora realizaremos un ‘Pull’ , que sirve para obtener los cambios más recientes desde un repositorio remoto y fusionarlos con tu rama local. Lo lógico sería hacerlo cuando estemos trabajando en otro ordenador y no tengamos actualizado nuestro almacenamiento local con el almacenamiento en la nube de nuestro repositorio. (En nuestro caso haremos eliminaremos el proyecto e iniciaremos uno nuevo para hacer un ‘Pull’)

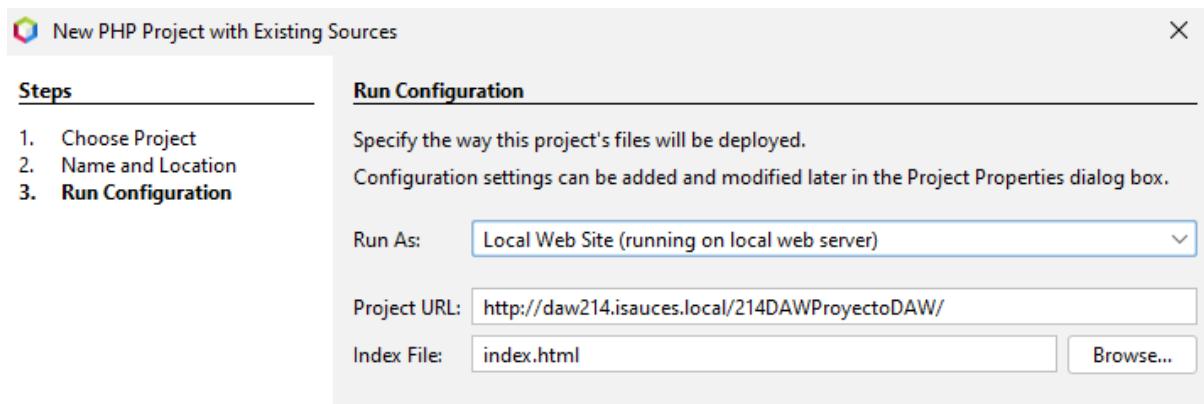
Creamos un proyecto de PHP con existencia de fuentes



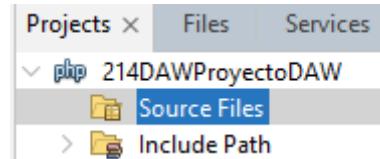
Rellenamos los campos



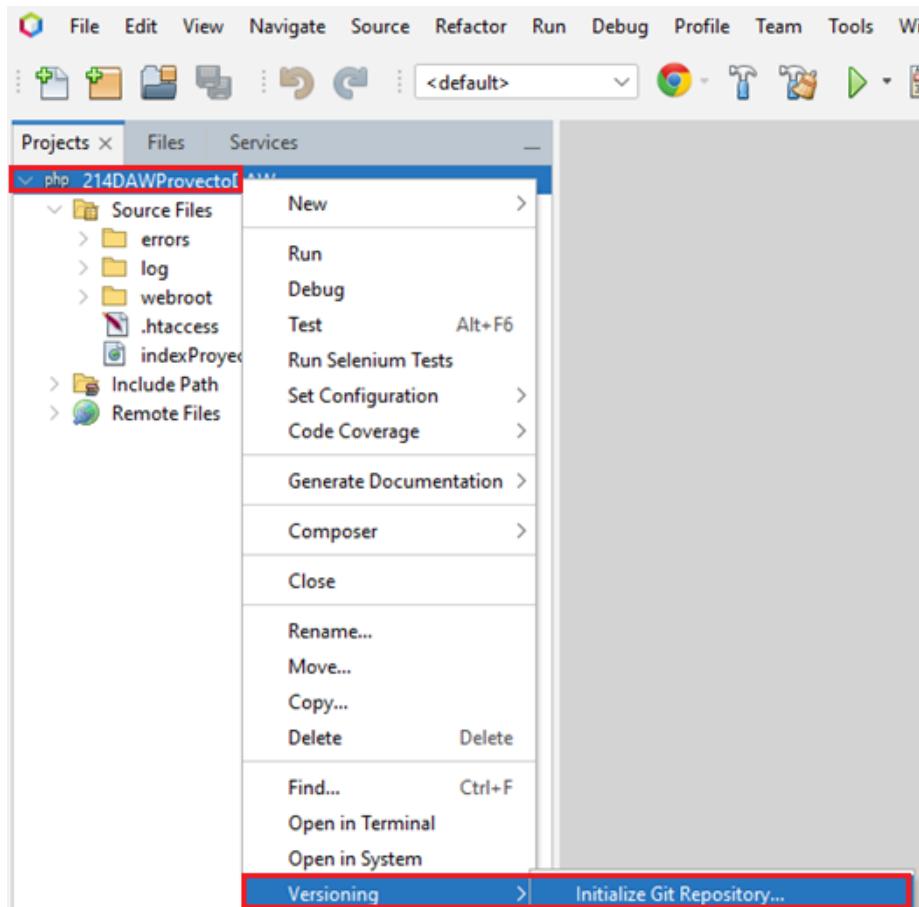
En la carpeta añadimos un archivo con extensión ‘.index’ para que nos deje crear el proyecto, en mi caso lo he hecho de manera manual añadiéndolo a la carpeta desde el explorador de archivos



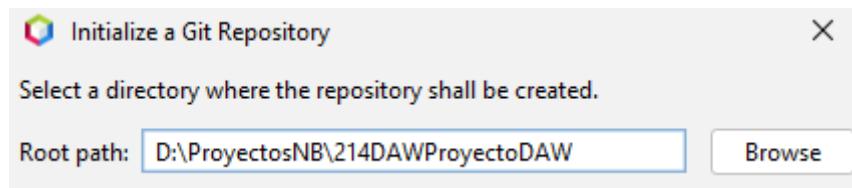
Y nos creara el proyecto, y eliminamos el index vacío que hemos creado.



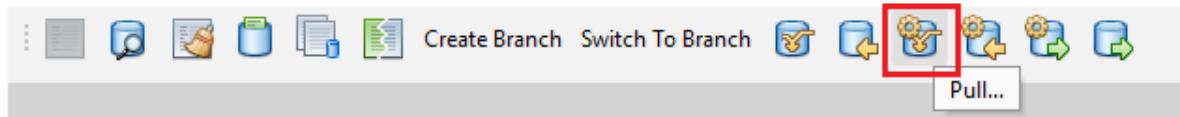
Ahora el siguiente tendremos que inicializar el repositorio en nuestro proyecto, haremos click derecho sobre el proyecto 'Versioning' / 'Initialize Git Repository...'



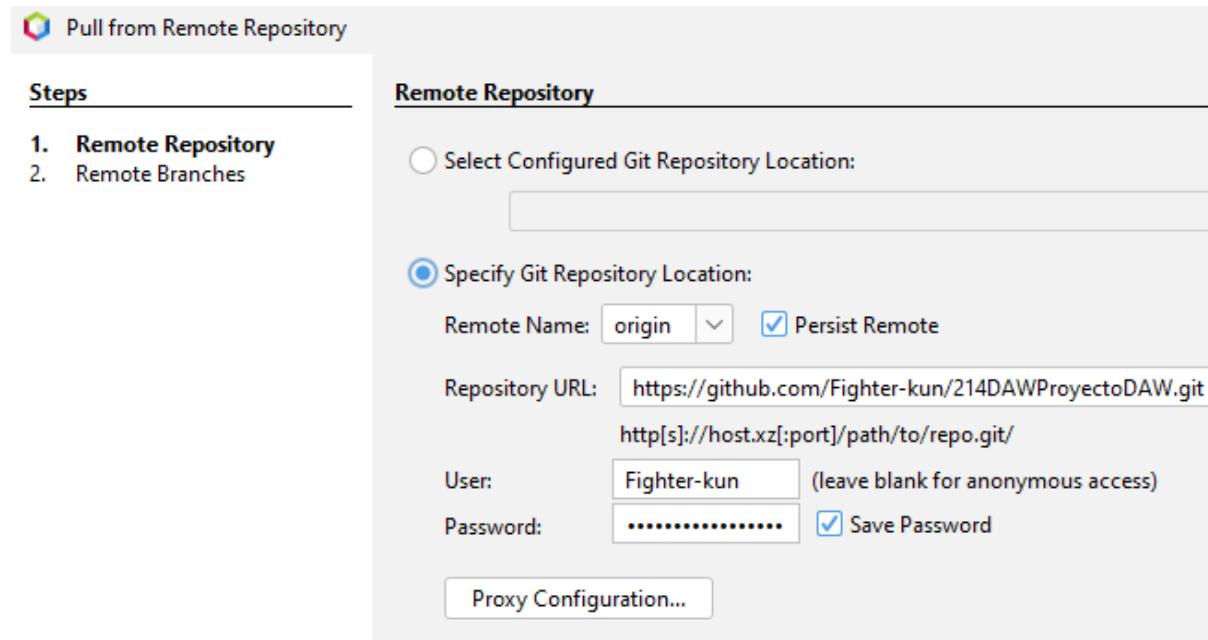
Y nos pedirá un Path como antes para indicarle un directorio donde hacer el Git Init



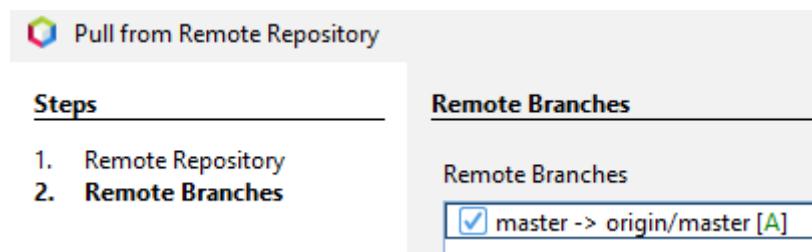
Ahora con las herramientas de la parte superior hacemos un 'Pull'



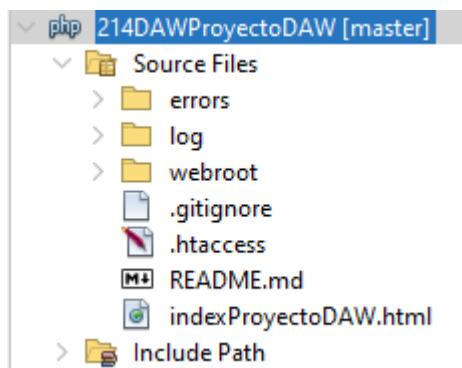
Y nos aparecerá la siguiente ventana, muy parecida a cuando hicimos 'Push', solo tendremos que usar las mismas credenciales que en el punto anterior, URL del repositorio, Usuario y código del TOKEN



Cuando le demos a siguiente nos aparecerá 'Remote Branches', las ramas remotas, tendremos que elegir la que queremos cargar en local, aunque en nuestro caso solo hay una rama



Y al hacer click en 'Finish' nos aparecerá nuestro repositorio en la nube almacenado en local



Hay que tener en cuenta que solo nos a ‘bajado’ los que indicamos en la ‘subida’ , lo archivos que nombramos en el ‘.gitignore’ no nos aparecen.

Esta es el contenido mi carpeta una vez realizado el ‘Pull’

.git	09/11/2023 20:50	Carpeta de archivos
errors	09/11/2023 20:50	Carpeta de archivos
log	09/11/2023 20:50	Carpeta de archivos
nbproject	09/11/2023 20:30	Carpeta de archivos
webroot	09/11/2023 20:50	Carpeta de archivos
.gitignore	09/11/2023 20:50	Documento de te...
.htaccess	09/11/2023 20:50	Archivo HTACCESS
indexProyectoDAW.html	09/11/2023 20:50	Chrome HTML Do...
README.md	09/11/2023 20:50	Archivo de origen ...

Tag

Podemos hacerlo de dos maneras, una por comando y otra de manera gráfica usando Git Hub. Si usamos los comando, nos hará a la vez tanto la creación del 'Tag' como el 'Release'

COMANDO:

Desde la ventana de Git Bash anterior, usamos el siguiente comando

```
git tag -a 'NOMBRE_DEL_TAG' -m 'MENSAJE_DESCRIPTIVO'
```

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214DAWProyectoDAW (developer)
$ git tag -a "Version1.0" -m "Funcionalidad: Indice DAW"
```

('-a' : Este parámetro indica que estás creando una "anotación" o "tag anotado", lo que significa que puedes asociar información adicional, como un mensaje descriptivo, al "tag". Un "tag anotado" es una referencia más completa que incluye el nombre del "tag", el mensaje descriptivo y la información del autor.

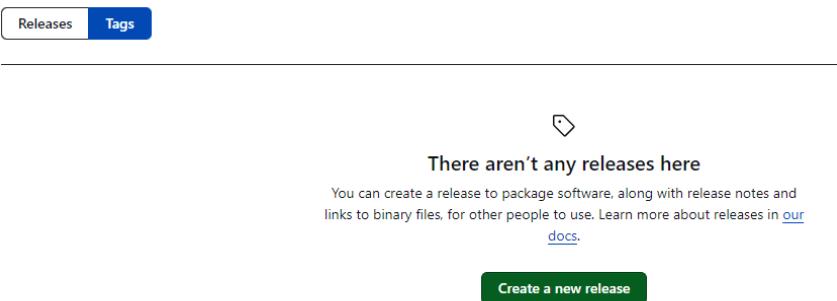
'-m' : Este parámetro te permite proporcionar un mensaje descriptivo que se asociará con el "tag". Es una buena práctica incluir un mensaje que describa brevemente el propósito del "tag" o las características principales de la versión asociada.)

GIT HUB:

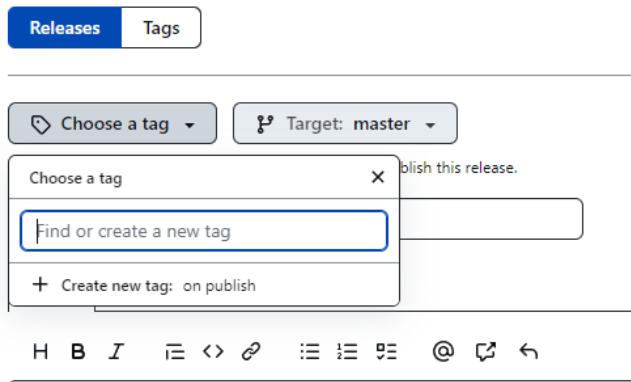
Iremos nuestro repositorio de GitHub y haremos click donde dice 'Tag' , en el siguiente apartado



Una vez dentro nos aparecerá la opción de crear un 'release' , haremos click sobre 'Create a new release'

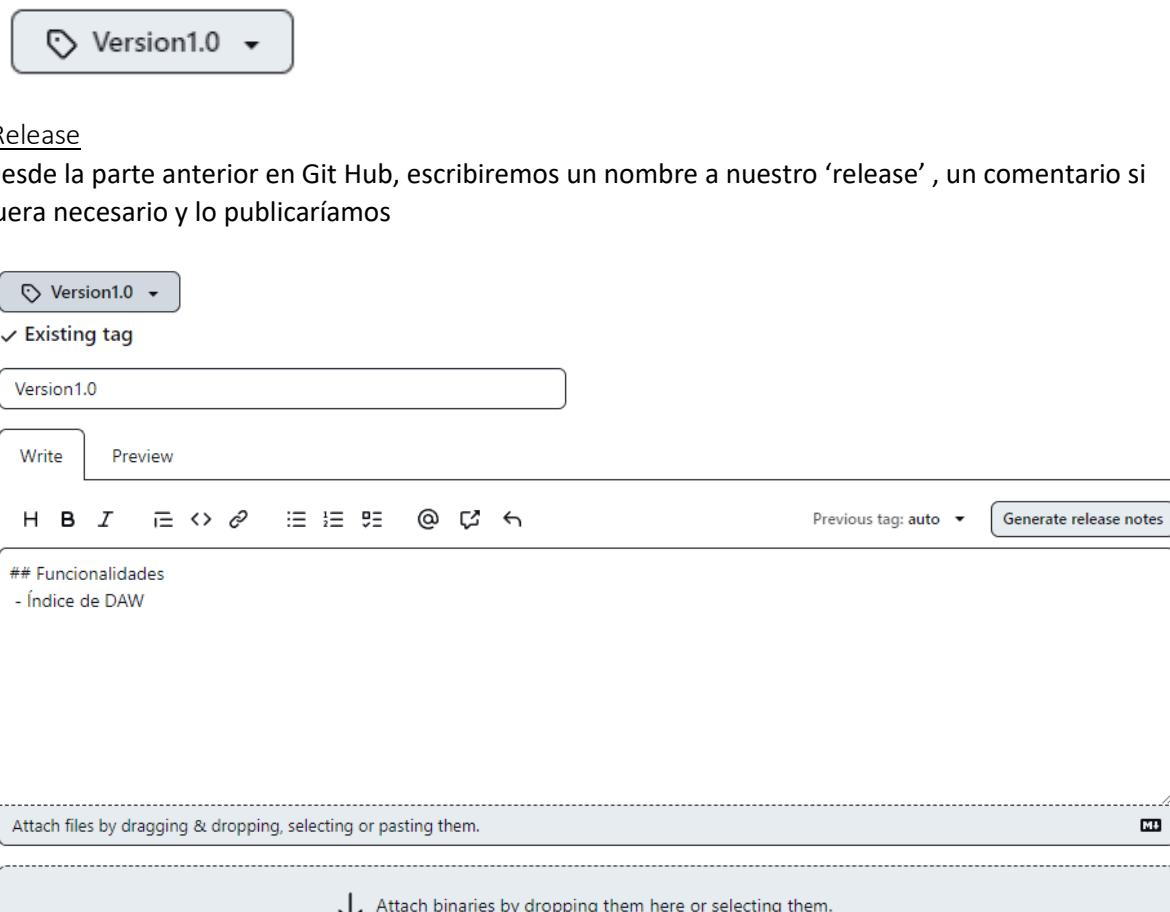


Y en el apartado de 'Choose a tag' nos aparecerá la opción de 'Create new tag: on publish'



Deberemos escribir un nombre y hacer click en 'Create new tag: on publish'

Y nos aparecerá de la siguiente manera



Una vez publicado, nos aparecerá de la siguiente manera

[Releases / Version1.0](#)

Version1.0 Latest

Fighter-kun released this 3 days ago · 4 commits to master since this release · Version1.0 · 32d0899

Funcionalidades

- Índice de DAW

Assets

Source code (zip)	5 days ago
Source code (tar.gz)	5 days ago

[Merge](#)

Merge

Podemos hacerlo de dos maneras, una por comando y otra de manera gráfica usando un IDE.

COMANDO:

Desde la ventana de Git Bash en el directorio local, nos cambiaremos a la rama 'master'

`git checkout master`

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214ProyectoDAW (developer214)
$ git checkout master
Switched to branch 'master'
```

Por seguridad se hace un 'Pull' para evitar conflictos

`git pull origin master`

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214ProyectoDAW (master)
$ git pull origin master
```

Luego tenemos 3 opciones para hacer 'Merge'

'Fast-Forward if possible (--ff)' : Intenta realizar un avance rápido, crea un nuevo commit de fusión si es necesario.

'Fast-Forward only (--ff-only)' : Solo permite un avance rápido, de lo contrario, no realiza la fusión.

'Always create commit (--no-ff)' : Siempre crea un nuevo commit de fusión, incluso si se pudiera haber hecho un avance rápido.

En este caso lo haremos 'fast-forward'

`git merge --ff developer214`

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214ProyectoDAW (master)
$ git merge --ff developer214
```

Si todo sale bien y no existen conflictos, haremos un 'commit' para poner un mensaje al merge

`git commit -m "Index Modificado Para Hacer Merge"`

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214ProyectoDAW (master)
$ git commit -m "Index Modificado Para Hacer Merge"
```

Y un 'Push' para actualizar los cambios en remoto

```
git push origin master
```

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214ProyectoDAW (master)
$ git push origin master
```

Luego verificaremos el estado de repositorio y verificaremos el historial con los dos últimos comandos

```
git status
```

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214ProyectoDAW (master)
$ git status
On branch master
nothing to commit, working tree clean
```

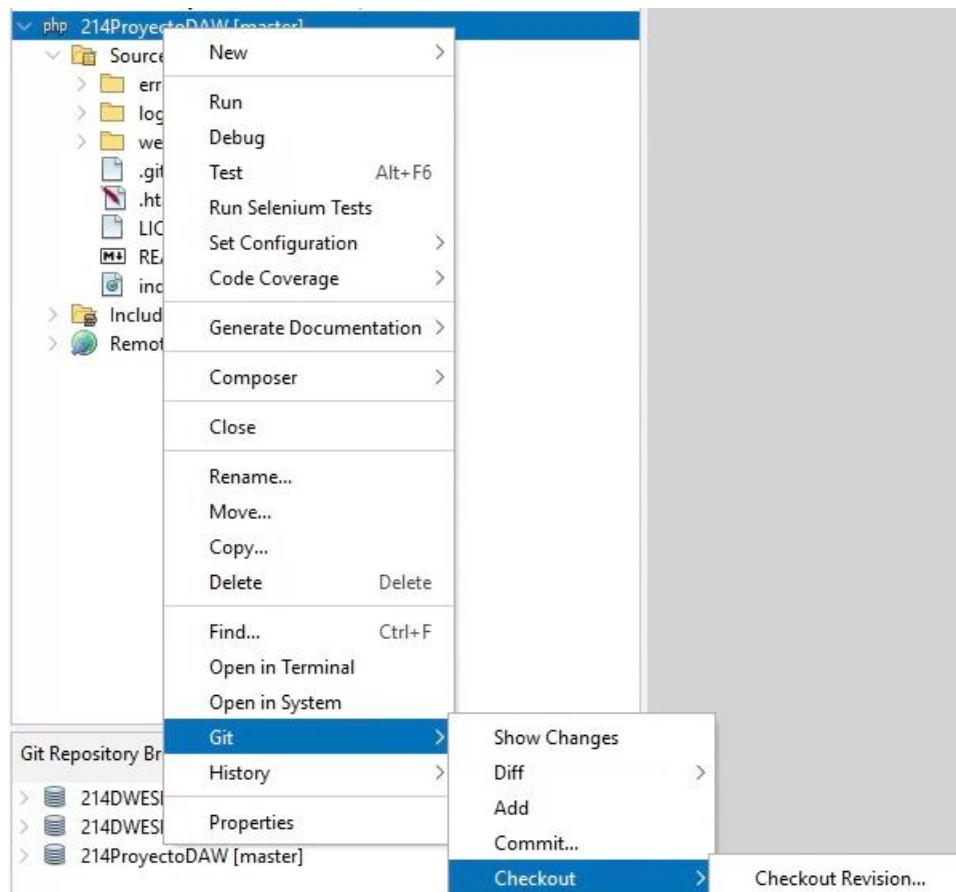
```
git log --oneline
```

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214ProyectoDAW (master)
$ git log --oneline
8f6f5c8 (HEAD -> master, origin/master, origin/developer214, developer214) Index Modificado Para Hacer Merge
c6361a5 V1.0
```

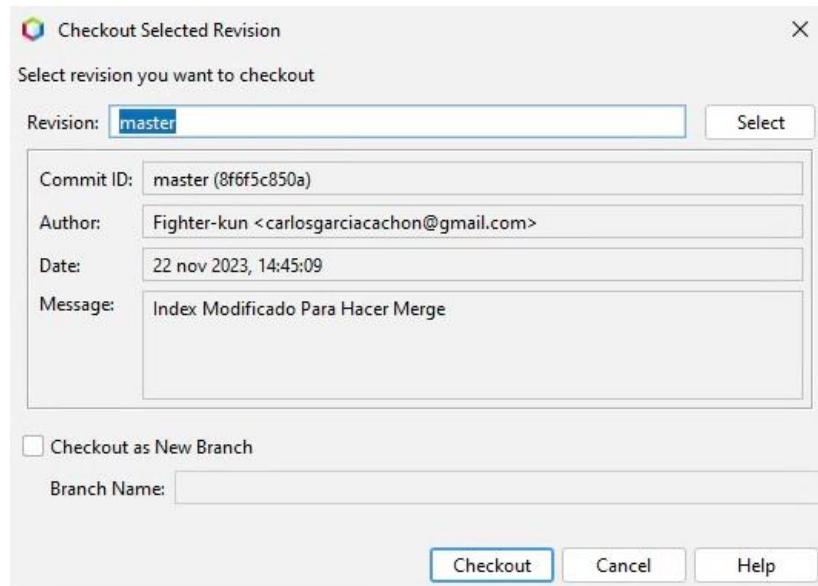
Deberemos tener en cuenta que la próxima vez que trabajemos con otra rama deberemos hacer un 'Pull', para actualizar nuestro repositorio local.

IDE:

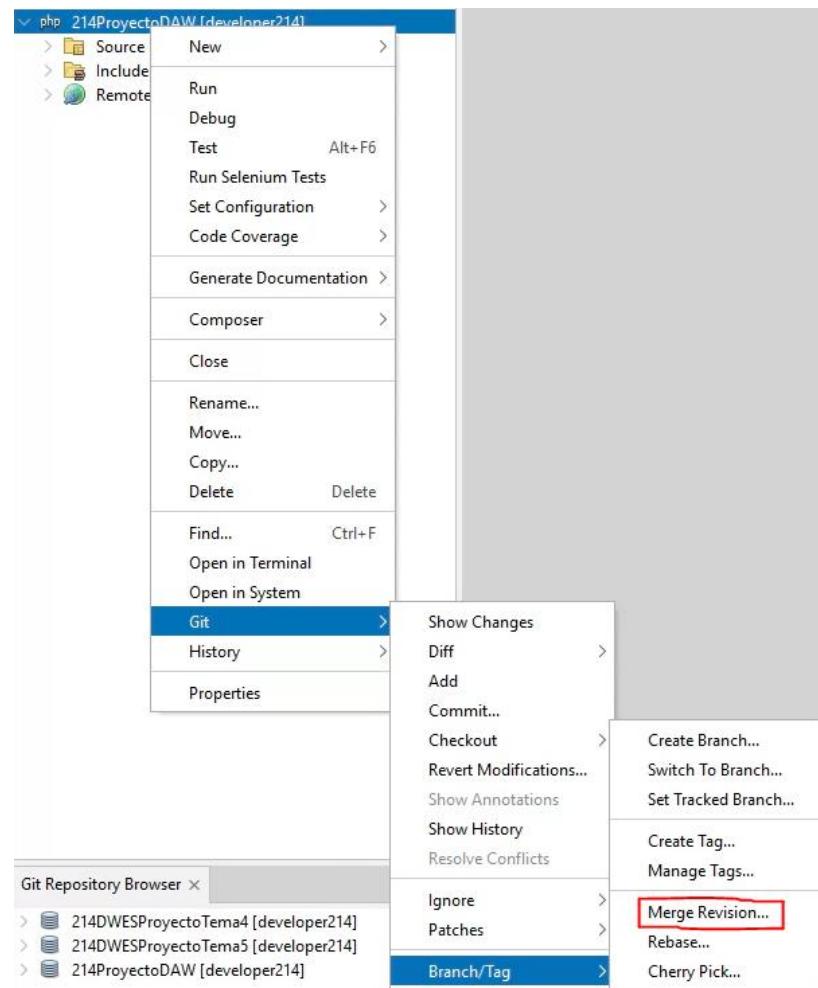
Primero tenemos que cambiar a la rama master:



Nos aparecerá una ventana y seleccionamos la rama a la que queremos cambiar



Hacemos click derecho sobre nuestro proyecto y nos dirigiremos al siguiente desplegable y seleccionamos 'Merge Revision...'



Nos aparecerá la siguiente ventana mostrándonos información sobre lo que vamos a 'mergear'

'Revision' : Revisa la rama que seleccionemos

'Commit ID' : Es el ID del commit

'Author' : Hace referencia al usuario y correo de Git Hub

'Date' : Fecha y hora de cuando se realiza el merge

'Message' : El mensaje del último commit

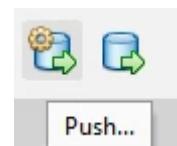
'Fast-Forward if possible (--ff)' : Intenta realizar un avance rápido, crea un nuevo commit de fusión si es necesario.

'Fast-Forward only (--ff-only)' : Solo permite un avance rápido, de lo contrario, no realiza la fusión.

'Always create commit (--no-ff)' : Siempre crea un nuevo commit de fusión, incluso si se pudiera haber hecho un avance rápido.

Dejaremos la opción por defecto y la daremos a 'Merge'

Luego si no nos aparece ningún conflicto haremos 'Push' ayudándonos de los iconos de NetBeans para que se apliquen los cambios en remoto



Y seguimos los pasos como hemos realizado anteriormente.

Rebase

Podemos hacerlo de dos maneras, una por comando y otra de manera gráfica usando un IDE.

COMANDO:

Abrimos el Git Bash y nos posicionamos en el directorio donde tenemos el iniciado git, nos posicionamos en la rama en que vamos a realizar el 'rebase' y ejecutamos el siguiente comando

```
git rebase master
```

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214DAWProyectoDAW (developer214)
$ git rebase master
Successfully rebased and updated refs/heads/developer214.
```

Si no nos da ningún error comprobaremos el estado del repositorio con el siguiente comando

```
git status
```

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214DAWProyectoDAW (developer214)
$ git status
On branch developer214
Your branch is ahead of 'origin/developer214' by 7 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

En este caso no pide hacer un push y luego volvemos a ver el estado

```
git push origin developer214
```

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214DAWProyectoDAW (developer214)
$ git push origin developer214
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 501 bytes | 501.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Fighter-kun/214DAWProyectoDAW.git
  52114fc..e679a0f  developer214 -> developer214
```

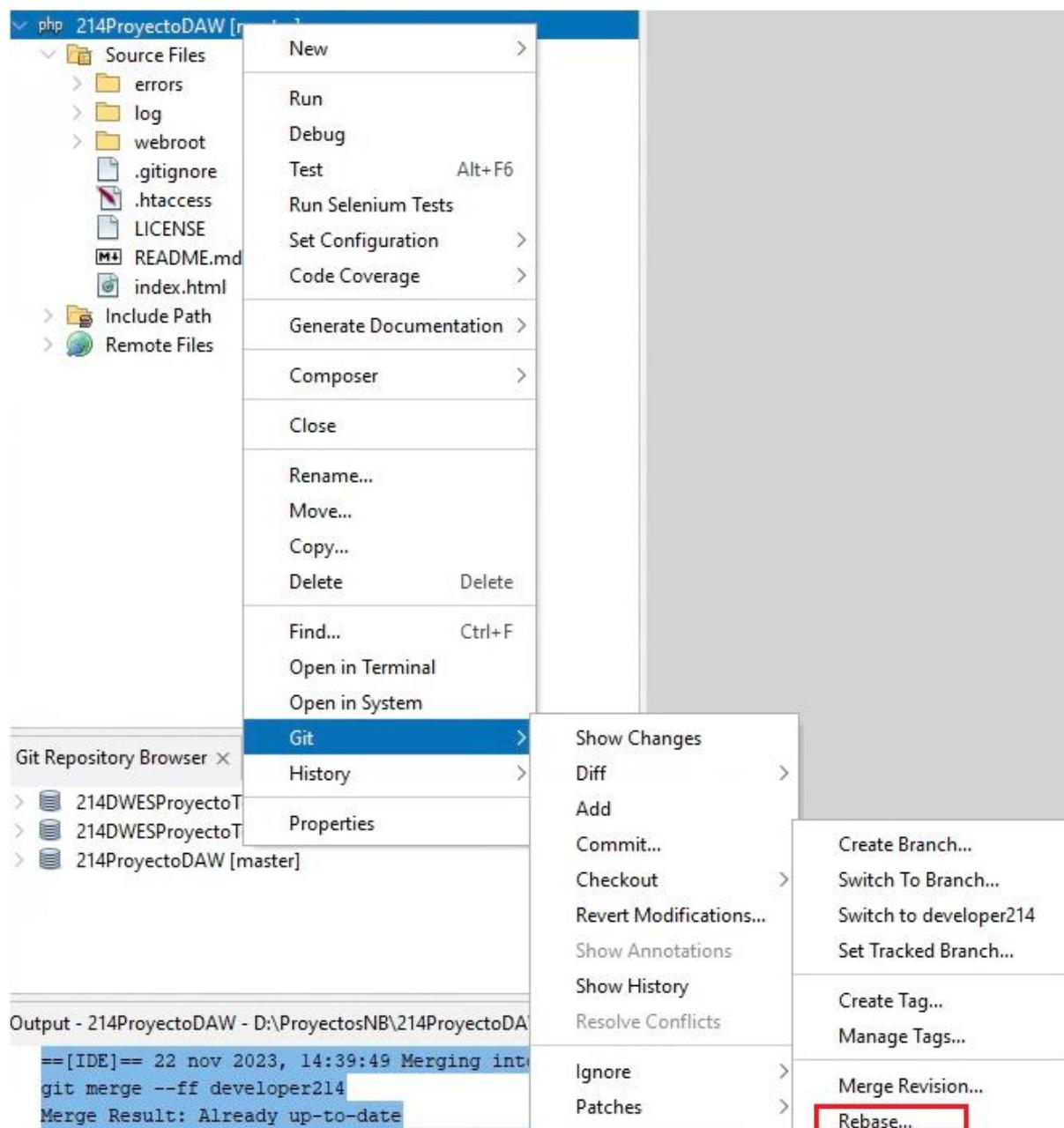
```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214DAWProyectoDAW (developer214)
$ git status
On branch developer214
Your branch is up to date with 'origin/developer214'.

nothing to commit, working tree clean
```

```
daw2@IS32WX14 MINGW64 /d/ProyectosNB/214DAWProyectoDAW (developer214)
```

IDE:

Al igual que en el 'Merge', debemos situarnos en la rama principal 'master' en nuestro caso y desde el desplegable anterior, seleccionamos 'Rebase Revision...'.



SSH Key (Clave SSH)

- **Autenticación Segura:** Permite autenticar de manera segura a un usuario en GitHub sin la necesidad de ingresar constantemente nombre de usuario y contraseña.
- **Conexiones Seguras:** Facilita conexiones seguras entre tu máquina local y los servidores de GitHub.
- **Acceso a Repositorios:** Al agregar tu clave SSH a tu cuenta de GitHub, puedes acceder y realizar operaciones en tus repositorios de manera segura.

Cómo funciona:

- Generas un par de claves (pública y privada) en tu máquina local.
- La clave pública se agrega a tu cuenta de GitHub.
- La clave privada se guarda de forma segura en tu máquina local y se utiliza para autenticar tus conexiones.

Para configurarlo nos iremos a la siguiente link

<https://github.com/settings/keys>

Haremos click en ‘New SSH key’

The screenshot shows the GitHub 'Settings' page under the 'SSH and GPG keys' tab. On the left, there's a sidebar with various account settings like Public profile, Account, Appearance, Accessibility, Notifications, Billing and plans, Emails, Password and authentication, Sessions, and SSH and GPG keys (which is currently selected). The main area shows sections for 'SSH keys' and 'GPG keys'. Both sections indicate that no keys are associated with the account. There are 'New SSH key' and 'New GPG key' buttons at the top of each section respectively. A 'Go to your personal profile' button is also visible.

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"
```

MINGW64:/d/RepositoriosGitHub

```
daw2@IS32WX14 MINGW64 /d/RepositoriosGitHub
$ ssh-keygen -t ed25519 -C "carlosgarciacachon@gmail.com"
```

No aparecerán unas opciones

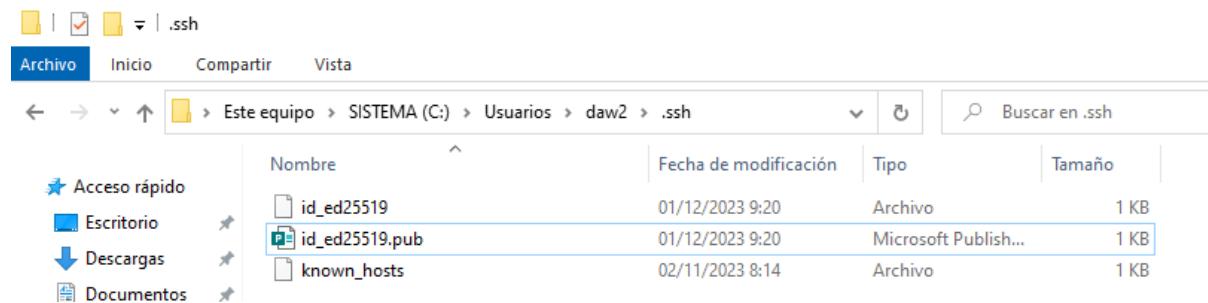
```
daw2@IS32WX14 MINGW64 /d/RepositoriosGitHub
$ ssh-keygen -t ed25519 -C "carlosgarciacachon@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/daw2/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification 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:NTX+j963HrLvqTs6v03IGLxBwRN5+6S16AYkccOH6uY carlosgarciacachon@gmail.com
The key's randomart image is:
++-[ED25519 256] ++
|   . . . o |
| . =.o.+ . |
| + o++ o |
| o . .oo + |
| . o S... * . |
| o .o. o . o |
| o  o+. .... |
| E  o*ooooo.+ |
| .o ==*o*+|
+---[SHA256]---
```

‘Enter file in which to save the key (/c/Users/daw2/.ssh/id_ed25519)’ : 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 (empty for no passphrase)’ : Aquí nos pide una “frase como contraseña” y nos dice que por defecto la deja vacía. En nuestro caso ponemos ‘paso’.

‘Enter same passphrase again’ : Por seguridad no pide de nuevo la misma frase para verificar que no la introducimos mal.

Los siguientes mensajes son de confirmación, nos indica que donde se han guardado los archivos generados e información de las claves.



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’ .

Nos da dos opciones para el tipo de clave:

1. Authentication Key (Clave de Autenticación):

- Esta es la clave SSH que se utiliza para autenticarte en GitHub.
- Te permite acceder y realizar operaciones en tu cuenta de GitHub sin tener que ingresar tu nombre de usuario y contraseña cada vez.
- Se agrega a tu cuenta de GitHub para asociarla con tu identidad.

2. Signing Key (Clave de Firma):

- Esta es una clave GPG (Pretty Good Privacy) que se utiliza para firmar digitalmente tus confirmaciones (commits) en GitHub.
- La firma digital proporciona una forma de verificar que los cambios realizados en un repositorio provienen de la persona legítima y que no han sido alterados.
- La clave de firma se asocia con tu cuenta de GitHub y se utiliza en combinación con la firma GPG cuando confirmas cambios en tus repositorios.

The screenshot shows the GitHub user profile page for 'Carlos García (Fighter-kun)'. On the left, there's a sidebar with account settings like Public profile, Account, Appearance, Accessibility, Notifications, Access (Billing and plans, Emails, Password and authentication, Sessions), SSH and GPG keys (which is selected and highlighted in blue), Organizations, Enterprises, and Moderation. The main content area is titled 'Add new SSH Key' and contains fields for 'Title' (set to 'ClavePersonal') and 'Key type' (set to 'Authentication Key'). Below these is a large text input field labeled 'Key' containing the SSH key value: 'ssh-ed25519 AAAAC3NzaC1IzDI1NTESAAAAILXVKbRMXhcnjqcjTanM2sCcNofnusCjsD1pBNnNPxrn carlosgarciacachon@gmail.com'. At the bottom of this section is a green 'Add SSH key' button. A small 'Go to your personal profile' button is located in the top right corner of the main content area.

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

The screenshot shows the 'SSH keys' page. At the top right is a green 'New SSH key' button. Below it, a message says 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' The main content area is titled 'Authentication Keys' and lists one key: 'ClavePersonal'. This key has a key icon, the SHA256 fingerprint 'SHA256:NTX+j963HrLvqTs6vO3IGLx8wRN5+6S16AYkcc0H6uY', was added on 'Dec 1, 2023', and is described as 'Never used — Read/write'. To the right of the key list is a 'Delete' button. At the bottom of the page is a note: 'Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#)'.

Git Cheat SheetCONFIGURACIONES GLOBALES:

`git config --global core.editor "nano"` : Para cambiar el editor por defecto a 'nano'

CREACION:

`git init` : Iniciar repositorio

`git remote add origin 'URL_REPO.git'` : Añado los ficheros al repositorio "Solo se usa cuando el repositorio está vacío por primera vez"

`git checkout -b 'developer'` : Nos cambiamos de rama 'developer' y con el '-b' la creamos

ESTADO DEL REPOSITORIO:

`git status` : Comprobamos el estado del repositorio local

ACTUALIZACION:

`git add .` : Añadir los ficheros area de trabajo **

`git commit -m "Mensaje"` : Para hacer 'commit': Confirmar cambios area trabajo

`git push origin 'NOMBRE_DE_LA_RAMA'` : Subimos los archivos al repositorio remoto

`git pull origin 'developer'` : Para actualizar el repositorio local de una rama determinada con el remoto

VISUALIZAR HISTORIAL:

`git log --oneline` : Muestra todos los commits con su código hash

GESTION DE RAMAS:

`git merger --no-ff developer` : Nos cambiamos a la rama master y con este comando mergeamos las ramas y un commit del merge que hacemos

`git merge --ff master` : Desde la rama developer para cambiar el puntero a master y luego hacemos un 'git push origin'

`git branch -d 'NOMBRE_DE_LA_RAMA'` : Para eliminar una rama de manera local

`git push origin --delete 'NOMBRE_DE_LA_RAMA'` : Para eliminar una rama de manera remota

ACTUALIZACION Y PUBLICACION:

`git tag -a 'NOMBRE_DEL_TAG' -m 'MENSAJE_DESCRIPTIVO'` : Cuando creamos una versión que pasamos a explotación, creamos un tag indicándole nombre y un mensaje descriptivo del mismo

Paso a explotación

Para el paso a explotación utilizaremos un hosting web (1&1 IONOS), que es una empresa que proporciona servicios de alojamiento web. En nuestro caso se nos ha proporcionado la siguiente información para la conexión con el servidor

Servidor/Host: **home496960781.1and1-data.host**

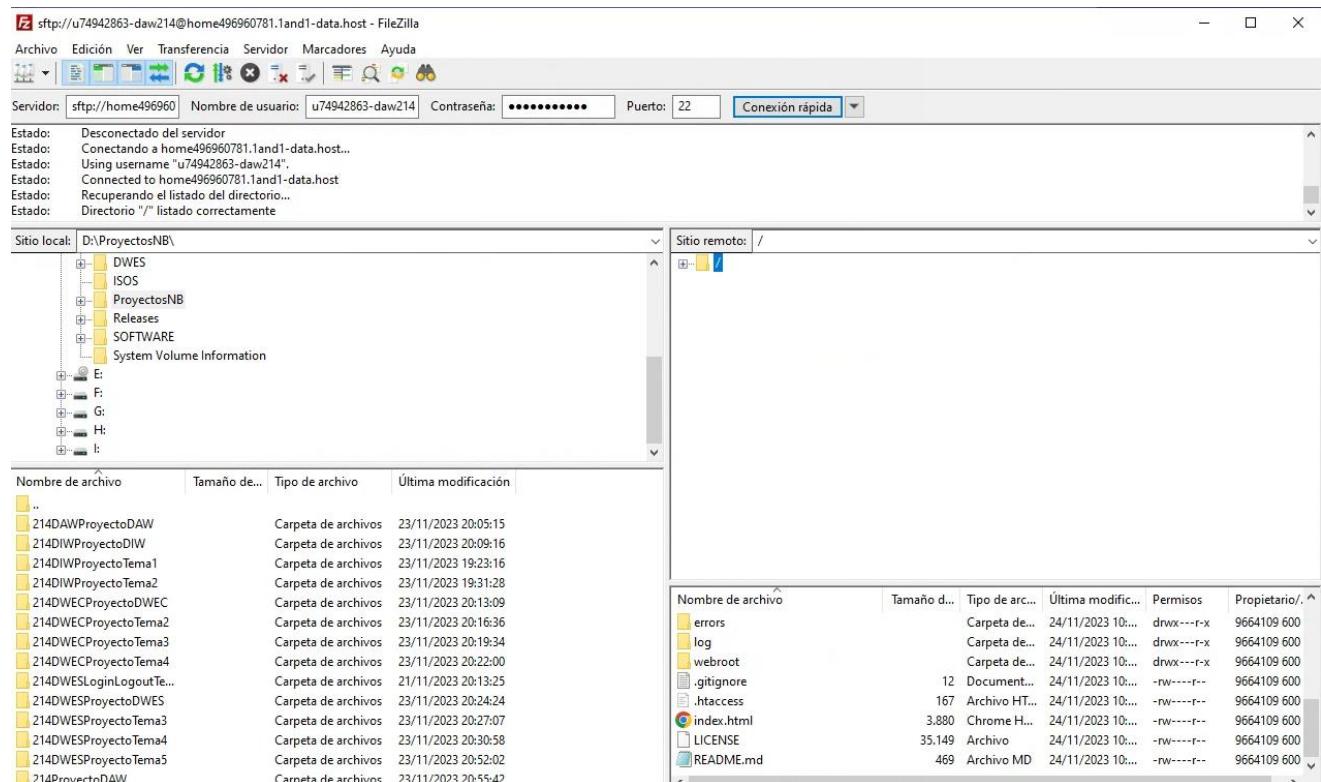
Usuario: **u74942863-daw214 / daw2_Sauces**

Es un usuario enjaulado que no puede utilizar putty, pero si utilizar filezilla (SFTP) para modificar su directorio de publicación. El directorio de publicación es '/2daw/daw214' (Solo se ve /)

Si tuviéramos que hacerlo desde el principio, tendríamos que registrarnos en la página de hosting o usar el que mejor nos convenciera.

<https://www.ionos.es/alojamiento/alojamiento-web>

El siguiente paso sería tener una versión estable de nuestra página web. Usando Git Hub, en el apartado 'Tag' , crearemos un 'Tag and Release' de nuestro repositorio, como hemos hecho en los pasos anteriores, y descargaremos el '.zip' . Luego usando el 'host' , 'usuario' , 'password' y 'puerto' nos conectaremos al servidor por medio de SFTP y subiremos nuestra versión estable.



Una vez subido el contenido, utilizando la siguiente ‘URL’ , accederemos a nuestro sitio web

<https://daw214.ieslossauces.es/>

The screenshot shows a web browser window with the following details:

- Title Bar:** Carlos García Cachón
- Address Bar:** daw214.ieslossauces.es
- Content Area:**
 - Section 1:** Desarrollo web en entorno servidor
 - Section 2:** Desarrollo web en entorno cliente
 - Section 3:** Despliegue de aplicaciones web
 - Section 4:** Desarrollo de interfaces web
- Image:** A photograph of a man with a beard, wearing a white t-shirt, sitting on a dark sofa in a room with warm lighting and decorative elements.
- Welcome Message:**
 - Bienvenido a mi web!**
 - Hola mi nombre es Carlos García Cachón, tengo 23 años y soy técnico en Sistemas Microinformáticos y Redes, actualmente estoy estudiando un ciclo superior en Desarrollo de Aplicaciones Web...
 - Saber más**
- Page Footer:**
 - © Carlos García Cachón IES LOS SAUCES 2023-24
 - Inicio
 -

PHP Doc