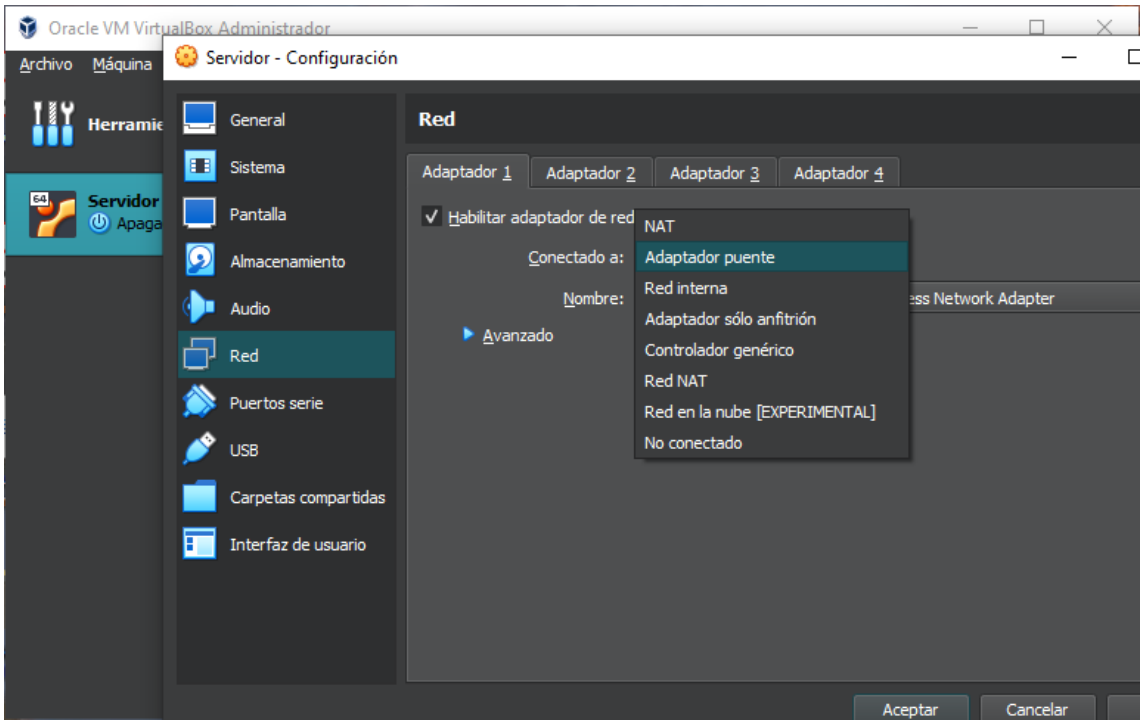
MANUAL DE CONFIGURACIÓN DE UN SERVIDOR APACHE

Para poder programar en web y crear páginas dinámicas, es imprescindible una parte cliente, que es el propio navegador del usuario, y una parte servidor, que habitualmente está ubicado en un superordenador, pero también puede alojarse en una máquina virtual.

En este caso, usaremos una máquina virtual de Ubuntu sin interfaz gráfica en la que instalaremos el servidor de Apache. Una vez configurada la máquina virtual en Virtual Box, deberemos seguir varios pasos a través de este manual para instalar el servidor:

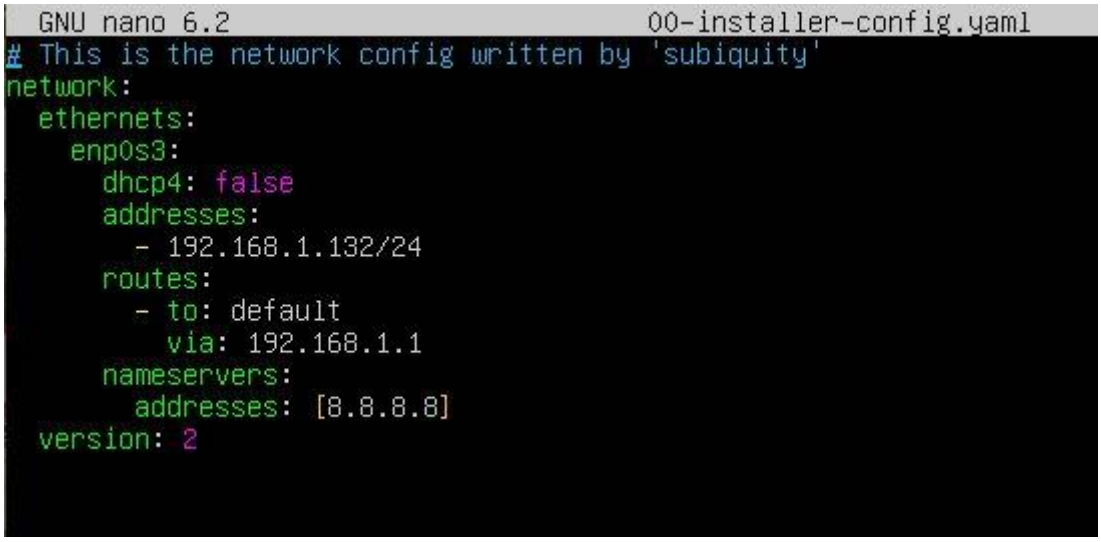
En primer lugar, debemos cambiar la red en Ubuntu. Nos interesa establecer una dirección IP estática, para configurarlo con la IP del equipo anfitrión. Previamente a realizar esta configuración, cambiaremos la red de la máquina virtual a Adaptador Puente, en la configuración de la máquina virtual de Virtual Box:



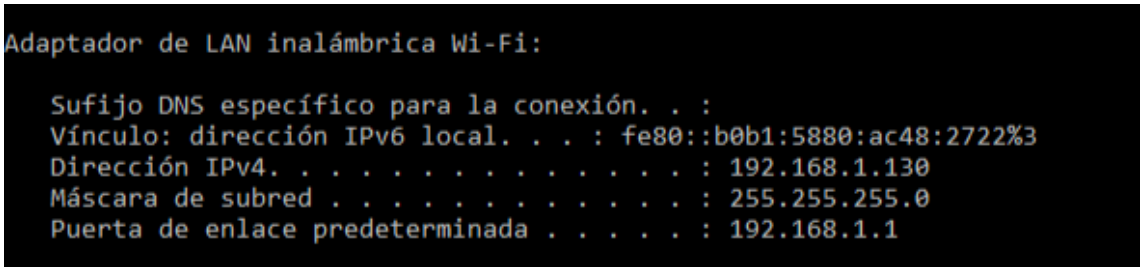
A partir de aquí, ya podremos modificar la IP. Para ello, usaremos la herramienta de configuración de red de Ubuntu llamada NetPlan, la cual se basa en un archivo con la descripción de los adaptadores de red que necesitemos definir. El archivo se llama "`00-installer-config.yaml`" y se localiza en la ruta `/etc/netplan`. Por lo tanto, escribiremos en la consola de nuestra máquina virtual:

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

Accederemos al archivo .yaml y podremos editarlo, escribiremos lo siguiente en el archivo .yaml:



Estos parámetros no son universales, varían para cada usuario dependiendo de los adaptadores de red de su máquina anfitrión. Por lo tanto, hay que revisar estos parámetros con el comando ipconfig (si es Windows) o ifconfig o ip add(en un Linux) en el Símbolo del sistema o terminal del equipo anfitrión. El siguiente ejemplo corresponde a los valores de mi máquina:



La dirección IP no puede coincidir, por eso en lugar de 130 he puesto 132.

El valor que prosigue a la "/", se relaciona con la Máscara de subred: en este caso es "/24", pero si la máscara fuera 255.255.0.0 , sería "/16".

Por último, hay que establecer la misma Puerta de enlace que tiene el equipo anfitrión (192.168.1.1), como se puede observar.

Ya hemos terminado de configurar la red. Ahora escribiremos en la terminal " `sudo netplan apply` " para que se aplique la configuración.

Se puede comprobar que hay conexión a la red haciendo " `ping www.google.es` ".

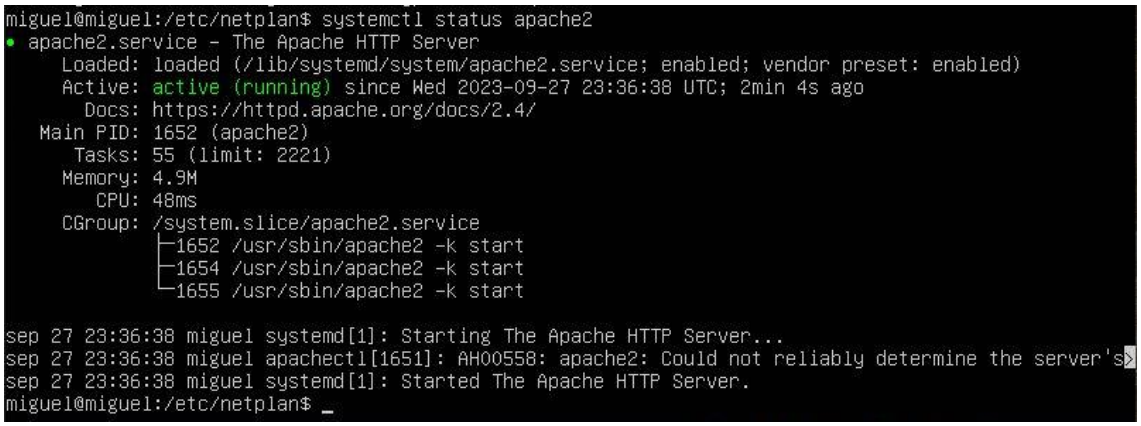
SERVIDOR APACHE

En este momento ya podremos instalar el servidor de Apache, para lo que escribiremos lo siguiente:

```
sudo apt-get install apache2
```

Una vez instalado, para comprobar que está activo:

```
systemctl status apache2
```



En el directorio /var/www/html alojaremos los archivos .html y .php. Aquí podemos crear carpetas para organizar el trabajo que vayamos a realizar. Hay un archivo index.html creado por defecto que podemos modificar. Le cambiaremos el nombre para que al acceder a nuestro servidor no aparezca este html por defecto. Se va a crear otro index.html con información como repositorio de cada alumno, de las tareas que se vayan realizando durante el curso, y aparecerá al acceder al servidor desde el navegador. Para acceder desde el navegador basta con escribir la dirección IP en la barra de búsqueda.

Necesitamos cambiarle los permisos de usuario a la carpeta /var/www/html para poder modificarla posteriormente. Haremos lo siguiente:

```
sudo chmod 777 /var/www/html
```

PHP

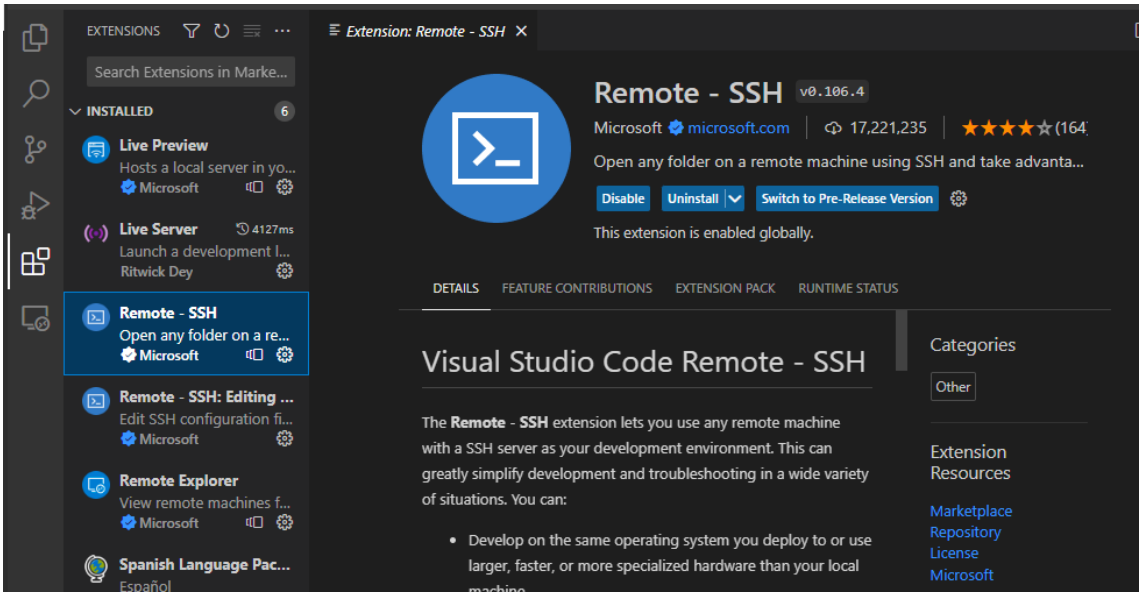
A continuación, ya podemos instalar PHP en la máquina virtual:

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

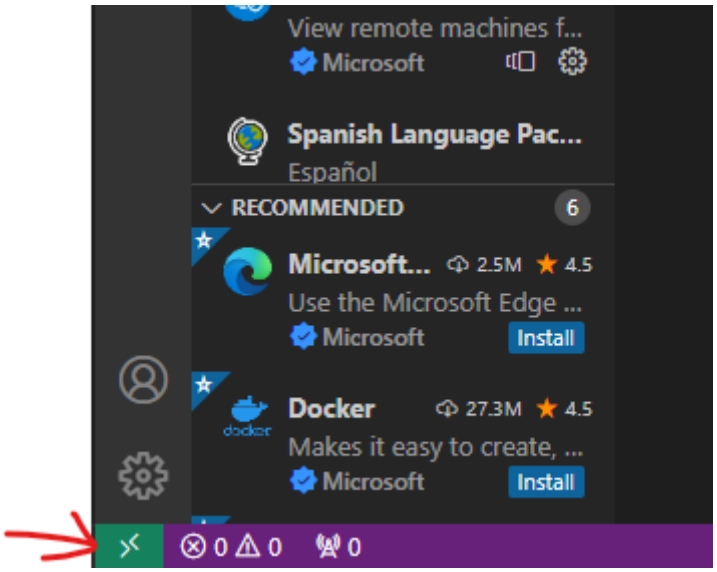
Ahora crearemos un .php en el directorio /var/www/html para comprobar que funciona. Sin embargo, para poder comprobar el funcionamiento debemos realizar la configuración en Visual Studio Code, que nos permita acceder a los ficheros que contiene el servidor y poder trabajar en ellos.

VISUAL STUDIO CODE

En primer lugar, necesitaremos instalar en el Visual Studio Code la extensión “Remote – SSH”.



Abriremos una ventana remota (pulsando la casilla verde de la esquina inferior izquierda), y nos dará la opción de conectar a un host.



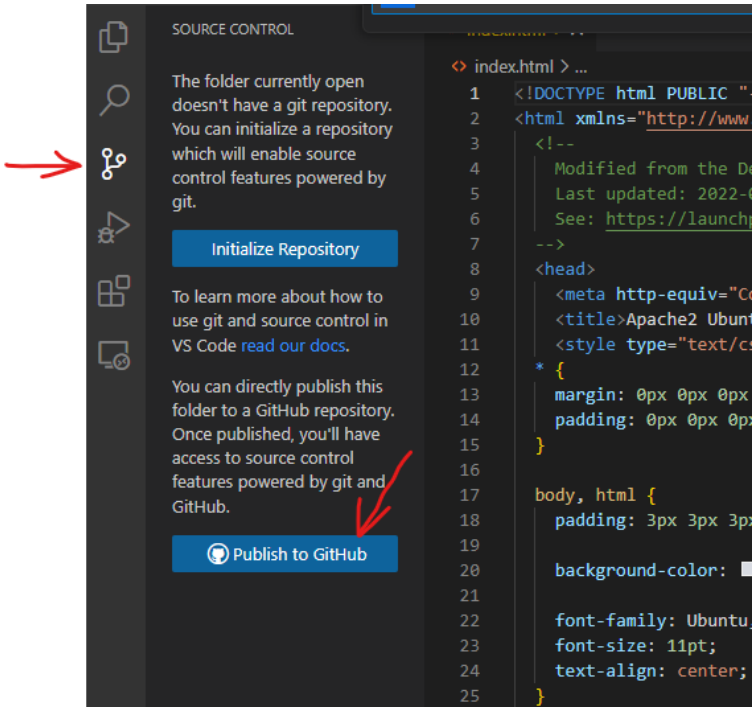
Elegimos esta opción, que nos va a permitir añadir un host SSH. Tenemos que escribir lo siguiente, con el nombre de tu equipo virtual y tu dirección IP en lugar de mis datos:

```
ssh miguel@192.168.7.201 -p 22
```

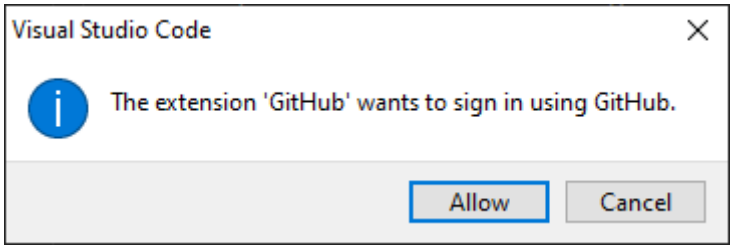
Nos pedirá introducir la contraseña. Una vez introducida, ya habremos conseguido conectar el Visual Studio Code con nuestro servidor Apache. A continuación, abriremos una carpeta, que será la dirección que hemos mencionado anteriormente: /var/www/html. Si se crea alguna carpeta dentro, se puede establecer la que se quiera.

Nos volverá a pedir la contraseña y una vez introducida, ya tendremos acceso a nuestros ficheros.

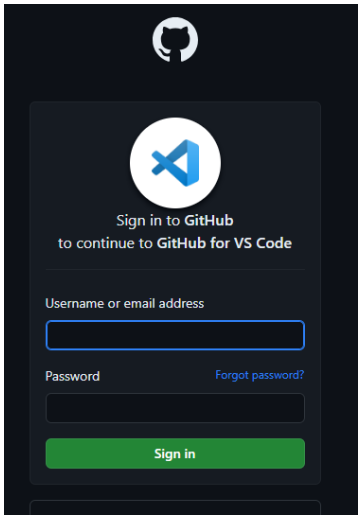
También usaremos un repositorio web, GitHub, para controlar las distintas versiones de nuestros ficheros.



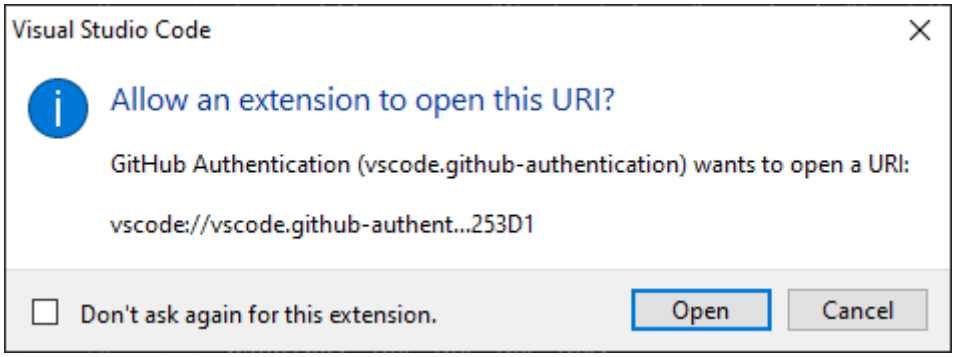
Permitimos acceder a GitHub:



Lo que nos redirigirá a GitHub para que introduzcamos nuestros datos de inicio de sesión.



Después, pulsamos Open:

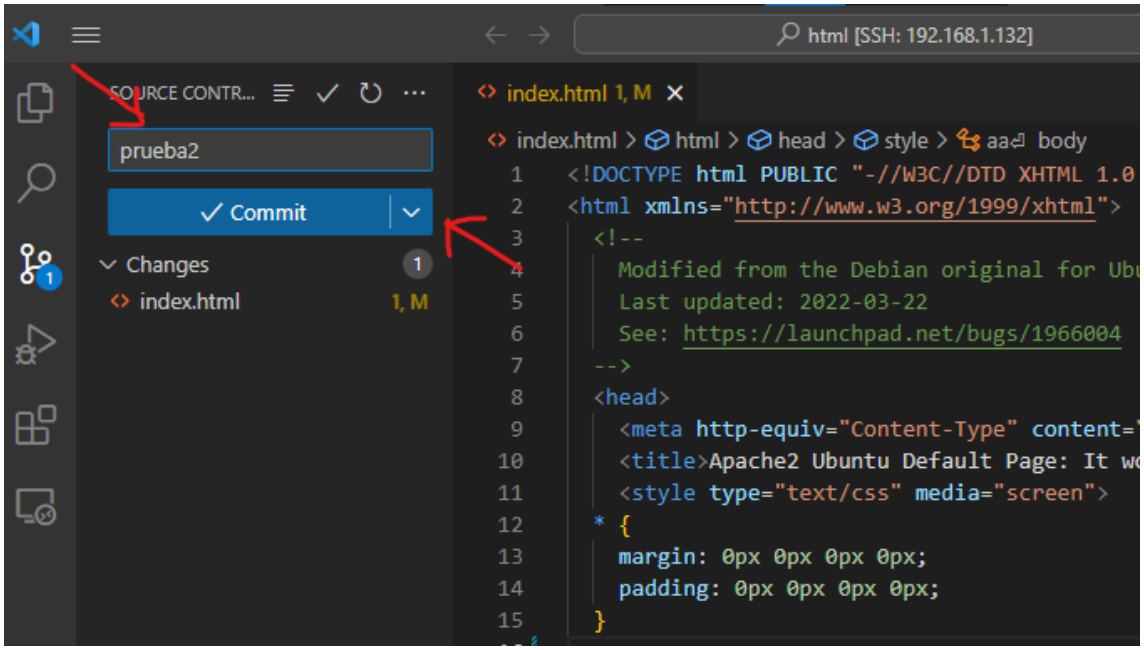


Escribimos un nombre para el repositorio.

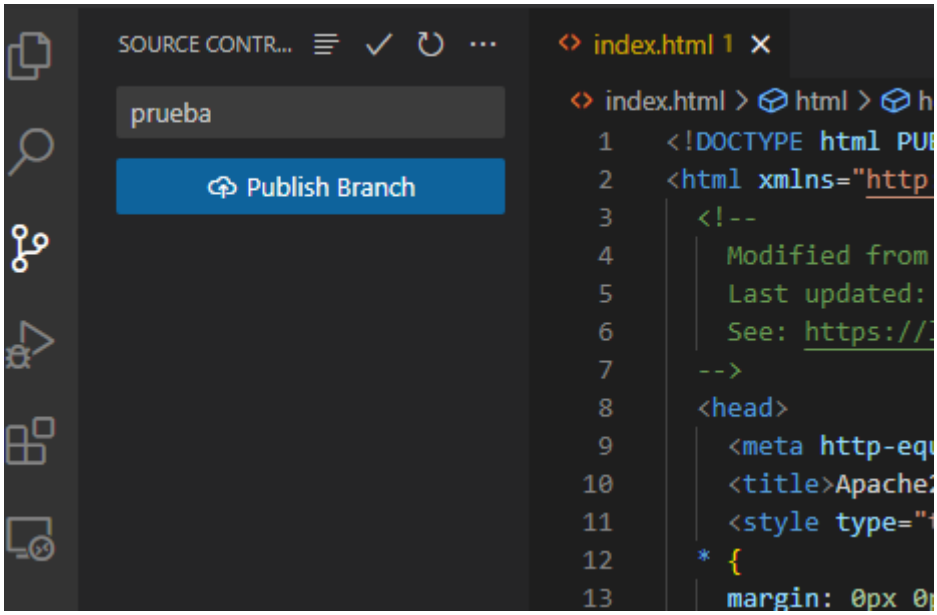
Si da error, hay que introducir nuestros datos de sesión de GitHub en la terminal del Visual Studio Code. Este es un ejemplo:

```
git config --global user.name "Miguel"
git config --global user.email "miguelbarbadguez@gmail.com"
```

Cuando hacemos un cambio en un archivo, tenemos la opción de subir los cambios a GitHub. Para ello, deberemos escribir un mensaje en cada cambio que subamos al repositorio.

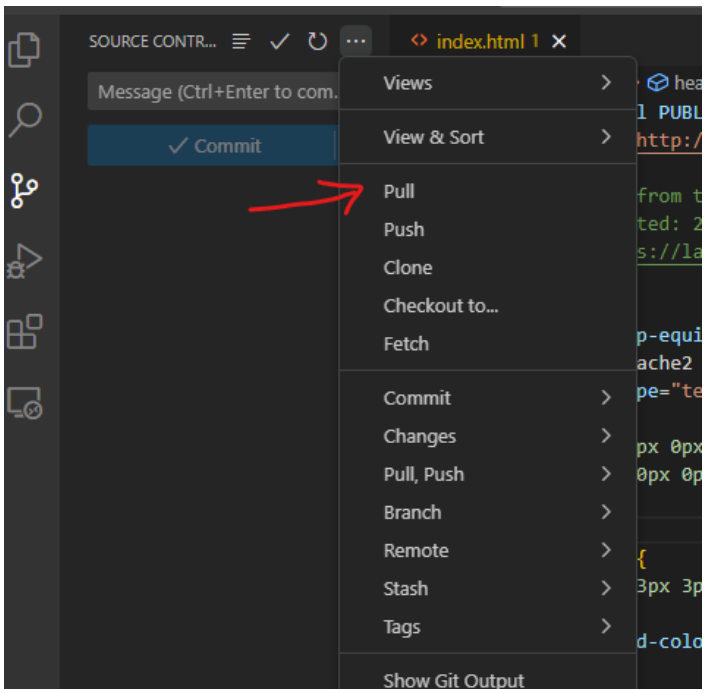


Escribimos un nombre para un conjunto de versiones que vamos a iniciar (Branch), y a continuación le damos un nombre al repositorio.



Ahora ya tenemos nuestro repositorio creado.

Podemos editarlo en GitHub y haremos un Commit para guardar los cambios. Si queremos cargarlo en el Visual, marcaremos Pull y se cargará la versión creada en el GitHub.



Si queremos hacerlo en el sentido contrario, marcaremos la opción de Push.

CONFIGURACIÓN DE PHP

Ahora pasaremos a modificar algunos aspectos de la configuración de PHP. De serie, la configuración de php tiene deshabilitados los errores, lo cual no nos interesa porque queremos ver si tenemos algún fallo al escribir el código. Para cambiar esta configuración, tenemos que acceder al directorio de PHP, que se encuentra en /etc/php/8.1/. En la carpeta apache2 accederemos al fichero php.ini con `sudo nano.`

Buscamos "display_errors" y cambiamos de "Off" a "On". Guardamos el fichero y salimos. Reiniciaremos el servidor con:

```
systemctl restart apache2
```

También podemos eliminar la obligatoriedad de escribir "php" al inicio de cada archivo.php . Para ello volvemos a acceder al fichero, y buscamos "shot_open_tag": lo ponemos en modo "On". Guardamos el fichero y salimos de nuevo, y volvemos a reiniciar el servidor.

Por último, vamos a instalar Xdebug, que es una extensión de PHP que proporciona la capacidad de depuración de código y errores. Para ello, escribiremos en la terminal:

```
sudo apt-get install php-xdebug.
```

Tendremos que acceder al fichero /etc/php/8.1/mods-available/xdebug.ini con `sudo` nano y escribiremos lo siguiente en la segunda línea:

```
xdebug.mode=debug
```

```
xdebug.start_with_request=yes
```

Volvemos a reiniciar el servidor y terminamos. Ya tenemos listo nuestro servidor para poder trabajar con él.