

A dark blue vertical bar is positioned on the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the date '27-9-2023'. In the bottom-left corner, there are several thin, curved, light blue lines that sweep upwards and to the right.

27-9-2023

# Manual configuración servidor Linux + GitHub

[Subtítulo del documento]

Fernando Calles Ramos  
SERVIDOR 2º DAW

# Índice

<i>Configuración server Ubuntu .....</i>	3
<i>Conexión SSH del equipo servidor en Visual Studio Code .....</i>	10
<i>Creación de un repositorio en GitHub y su conexión con Visual Studio Code .....</i>	14

### Configuración server Ubuntu

- La primero que haremos será comprobar la dirección IP que tiene nuestra máquina, al hacer *ifconfig* nos indicará que tenemos que instalar la funcionalidad que lo incluye, nos indica el código a introducir para realizar la instalación:

```
fernando@fernando:~$ ifconfig
Command 'ifconfig' not found, but can be installed with:
sudo apt install net-tools
```

- Al introducir el comando se nos instalará el paquete correspondiente

```
fernando@fernando:~$ sudo apt install net-tools
[sudo] password for fernando:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes NUEVOS:
  net-tools
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 27 no actualizados.
Se necesita descargar 204 kB de archivos.
Se utilizarán 819 kB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Descargados 204 kB en 1s (354 kB/s)
Seleccionando el paquete net-tools previamente no seleccionado.
(Leyendo la base de datos ... 74143 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Desempaquetando net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Configurando net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Procesando disparadores para man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

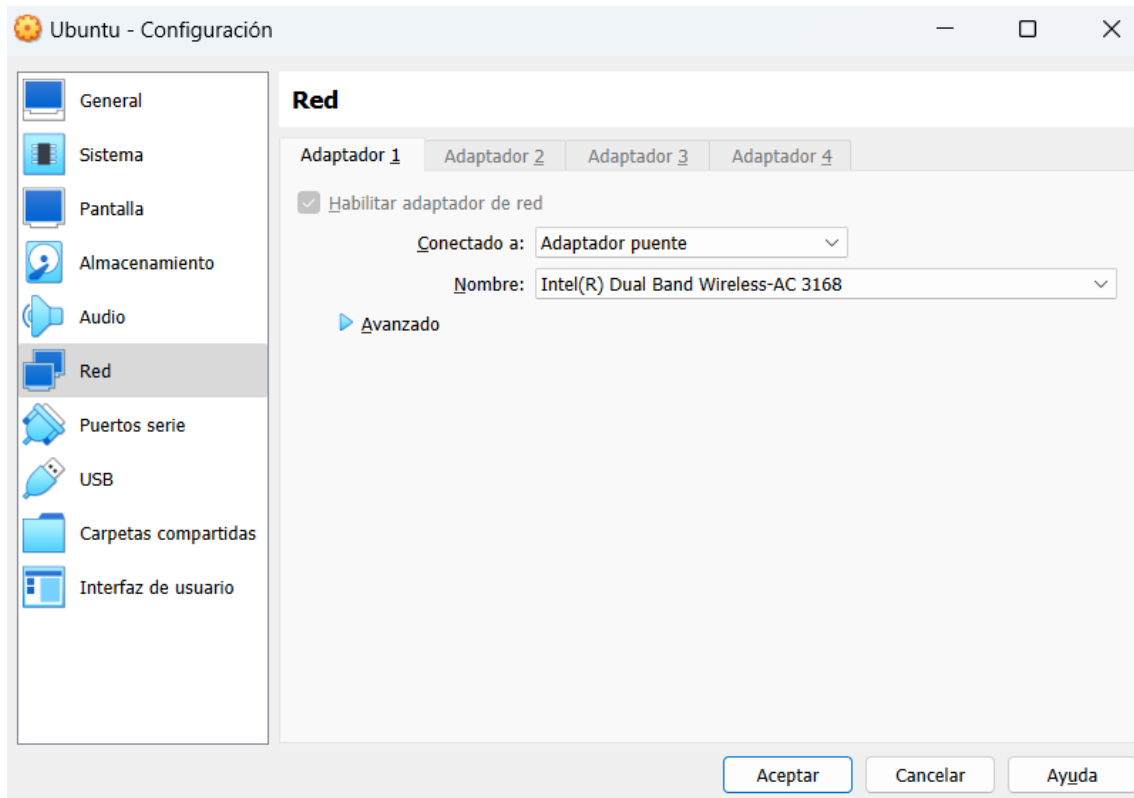
No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

- Ahora ya podremos ver nuestra configuración de red.

```
fernando@fernando:~$ sudo ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:febf:c0c prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:bf:0c:0c txqueuelen 1000 (Ethernet)
    RX packets 424 bytes 521658 (521.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 166 bytes 18992 (18.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 190 bytes 15167 (15.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 190 bytes 15167 (15.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- Esta configuración la vamos a cambiar, pero previamente, vamos a cambiar el tipo de adaptador a “Adaptador puente”.



- Una vez hecho este cambio ahora modificaremos nuestra configuración de red, estos cambios, se deben realizar en el archivo `/etc/netplan/00-installer-config.yaml`, lo abriremos con el comando: `sudo nano`, seguido del nombre del fichero y estableceremos esta configuración:

```
GNU nano 6.2 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.1.140/24
      routes:
        - to: default
          via: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8]
  version: 2
```

- Una vez establecida esa configuración, para que la tome el equipo debemos indicárselo con otro comando:

```
fernando@fernando:~$ sudo netplan apply
```

- Ahora cuando escribamos *ifconfig* nos mostrará la configuración de red que le hemos indicado anteriormente:

```
fernando@fernando:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.140 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:febf:c0c prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:bf:0c:0c txqueuelen 1000 (Ethernet)
    RX packets 6669 bytes 1013788 (1.0 MB)
    RX errors 0 dropped 25 overruns 0 frame 0
    TX packets 230 bytes 24591 (24.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 264 bytes 20625 (20.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 264 bytes 20625 (20.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- Comprobamos que es capaz de establecer conexión tanto con el servidor DNS como con el equipo anfitrión:

```
fernando@fernando:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=41.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=14.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=13.6 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=19.7 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 13.565/22.321/41.703/11.432 ms
fernando@fernando:~$ ping 192.168.1.40
PING 192.168.1.40 (192.168.1.40) 56(84) bytes of data.
64 bytes from 192.168.1.40: icmp_seq=1 ttl=128 time=0.979 ms
64 bytes from 192.168.1.40: icmp_seq=2 ttl=128 time=0.374 ms
64 bytes from 192.168.1.40: icmp_seq=3 ttl=128 time=0.397 ms
64 bytes from 192.168.1.40: icmp_seq=4 ttl=128 time=0.358 ms
^C
--- 192.168.1.40 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3011ms
rtt min/avg/max/mdev = 0.358/0.527/0.979/0.261 ms
```

- Comprobamos que desde el equipo anfitrión también somos capaces de establecer conexión a la máquina servidora.

```
C:\Users\fer19>ping 192.168.1.140

Haciendo ping a 192.168.1.140 con 32 bytes de datos:
Respuesta desde 192.168.1.140: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.1.140: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.1.140: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.1.140: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.1.140:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

- El siguiente paso será instalar el servidor HTTP *apache2*.

```
fernando@fernando:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser bzip2-doc
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
0 actualizados, 13 nuevos se instalarán, 0 para eliminar y 27 no actualizados.
Se necesita descargar 2.137 kB de archivos.
Se utilizarán 8.505 kB de espacio de disco adicional después de esta operación.
```

- Una vez se encuentre instalado, comprobaremos el estado del servicio.

```
fernando@fernando:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-09-26 19:24:32 UTC; 46s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2305 (apache2)
    Tasks: 55 (limit: 2221)
   Memory: 5.0M
      CPU: 53ms
   CGroup: /system.slice/apache2.service
           └─2305 /usr/sbin/apache2 -k start
             └─2307 /usr/sbin/apache2 -k start
               └─2308 /usr/sbin/apache2 -k start

sep 26 19:24:31 fernando systemd[1]: Starting The Apache HTTP Server...
sep 26 19:24:31 fernando apachectl[2304]: AH00558: apache2: Could not reliably determine the server's
sep 26 19:24:32 fernando systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)
```

- Nos muestra que el estado es *active*, en caso de que nos muestre algo diferente, habría que revisar la configuración.
- Para comprobar visualmente el correcto funcionamiento de *apache2* podemos introducir en nuestro navegador la dirección IP de nuestra máquina, esto nos mostrará la página por defecto del servidor.



- El archivo .html de esta página se encuentra en /var/www/html.

```
fernando@fernando:~$ cd /var/www/html/  
fernando@fernando:/var/www/html$ ls -l  
total 12  
-rw-r--r-- 1 root root 10671 sep 26 19:24 index.html
```

- Con nano, podremos modificar el archivo .html del index para comprobar que realmente es el archivo que estamos abriendo en el navegador. Por ejemplo, en este caso vamos a escribir en el encabezado de la página: “Página por defecto del servidor de Fernando”.

```
GNU nano 6.2 index.html  
  
color: #000000;  
}  
  
div.content_section_text a:hover {  
background-color: #000000;  
  
color: #DCDFE6;  
}  
  
div.validator {  
}  
  
</style>  
</head>  
<body>  
  <div class="main_page">  
    <div class="page_header floating_element">  
        
      <div>  
        <span style="margin-top: 1.5em;" class="floating_element">  
          Página por defecto del servidor de Fernando  
        </span>  
      </div>  
    <div class="banner">  
      <div id="about"></div>  
      It works!  
    </div>  
  
  </div>  
  <div class="content_section floating_element">  
    <div class="content_section_text">  
      <p>
```

[ Wrote 363 lines ]

G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location	M-U Undo
X Exit	^R Read File	^N Replace	^U Paste	^J Justify	^_ Go To Line	M-E Redo

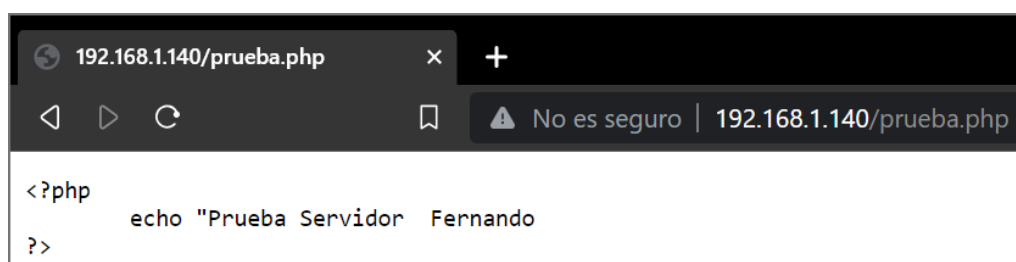
- La página ya aparece modificada



- A continuación, vamos a hacer una prueba, creando una página .php para ver su contenido. Este será el código de la página.

```
GNU nano 6.2 /var/www/html/prueba.php
<?php
    echo "Prueba Servidor Fernando";
?>
```

- Cuando accedemos a esa web a través del navegador, nos encontramos con que nos muestra el código tal cual ha sido escrito. Esto se debe a que tendremos que instalar en nuestro servidor un intérprete para que ejecute este código antes de enviárselo al cliente.

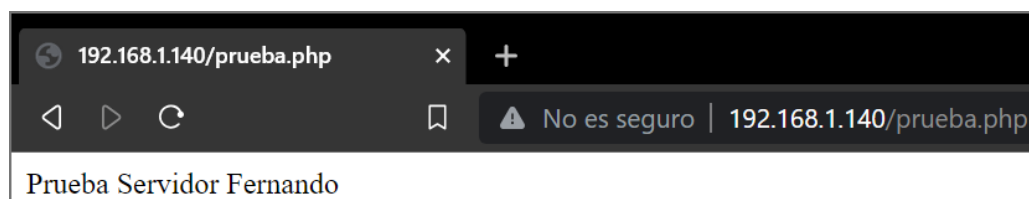




- Con este comando, comenzará la instalación del servicio PHP en el servidor.

```
fernando@fernando:~$ sudo apt-get install php
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
  libapache2-mod-php8.1 php-common php8.1 php8.1-cli php8.1-common php8.1-opcache php8.1-readline
Paquetes sugeridos:
  php-pear
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-php8.1 php php-common php8.1 php8.1-cli php8.1-common php8.1-opcache
  php8.1-readline
0 actualizados, 8 nuevos se instalarán, 0 para eliminar y 27 no actualizados.
Se necesita descargar 5.130 kB de archivos.
Se utilizarán 21,3 MB de espacio de disco adicional después de esta operación.
```

- Cuando la instalación ha finalizado, si actualizamos nuestra página .php del navegador, el código habrá sido interpretado

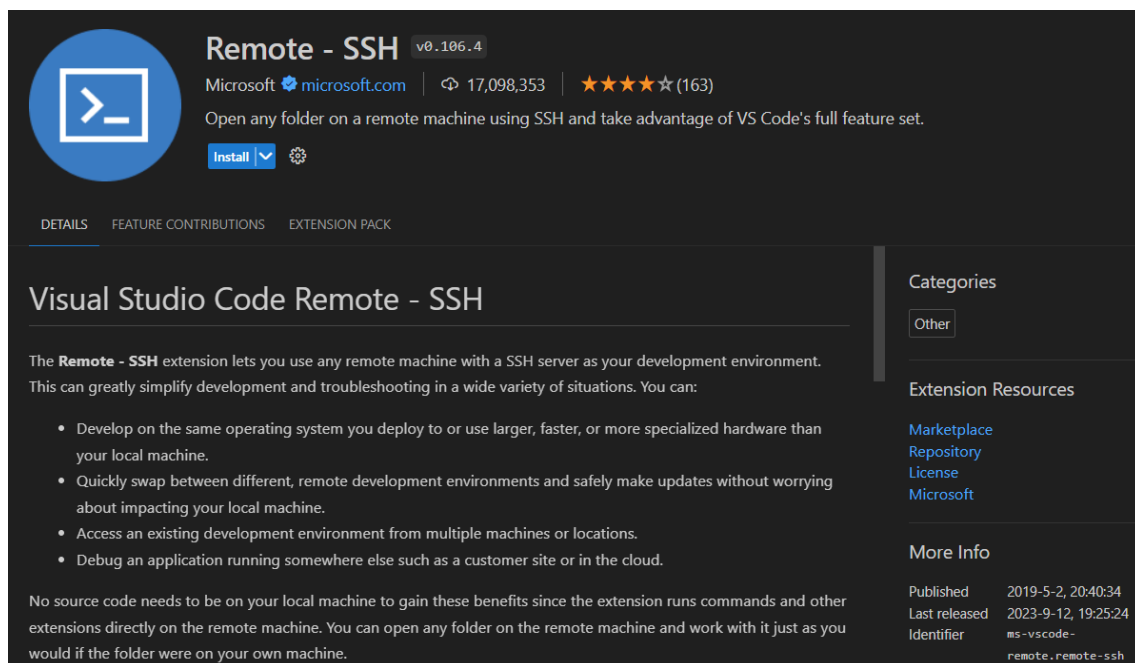


## Conexión SSH del equipo servidor en Visual Studio Code

- Lo primero que haremos será instalar el servicio de SSH en nuestra máquina servidor

```
fernando@fernando:~$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
  openssh-client openssh-sftp-server
Paquetes sugeridos:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard
Se actualizarán los siguientes paquetes:
  openssh-client openssh-server openssh-sftp-server
3 actualizados, 0 nuevos se instalarán, 0 para eliminar y 24 no actualizados.
Se necesita descargar 1.378 kB de archivos.
Se utilizarán 0 B de espacio de disco adicional después de esta operación.
```

- Una vez instalado el servicio necesario en nuestro servidor, instalaremos en Visual Studio Code la extensión “Remote – SSH”.



The screenshot shows the Visual Studio Code extension marketplace page for the "Remote - SSH" extension by Microsoft. The extension is version v0.106.4 and has 17,098,353 installations and 163 five-star ratings. The description states: "Open any folder on a remote machine using SSH and take advantage of VS Code's full feature set." The page includes tabs for DETAILS, FEATURE CONTRIBUTIONS, and EXTENSION PACK. The main content area describes how the extension allows users to develop on remote machines with SSH servers, listing benefits such as using specialized hardware, swapping environments, and accessing development environments from multiple locations. A sidebar on the right contains sections for Categories (Other), Extension Resources (Marketplace, Repository, License, Microsoft), and More Info (Published, Last released, Identifier).

**Remote - SSH** v0.106.4  
Microsoft [microsoft.com](https://microsoft.com) | 17,098,353 | ★★★★★ (163)  
Open any folder on a remote machine using SSH and take advantage of VS Code's full feature set.  
[Install](#) [Settings](#)

DETAILS | FEATURE CONTRIBUTIONS | EXTENSION PACK

### Visual Studio Code Remote - SSH

The **Remote - SSH** extension lets you use any remote machine with a SSH server as your development environment. This can greatly simplify development and troubleshooting in a wide variety of situations. You can:

- Develop on the same operating system you deploy to or use larger, faster, or more specialized hardware than your local machine.
- Quickly swap between different, remote development environments and safely make updates without worrying about impacting your local machine.
- Access an existing development environment from multiple machines or locations.
- Debug an application running somewhere else such as a customer site or in the cloud.

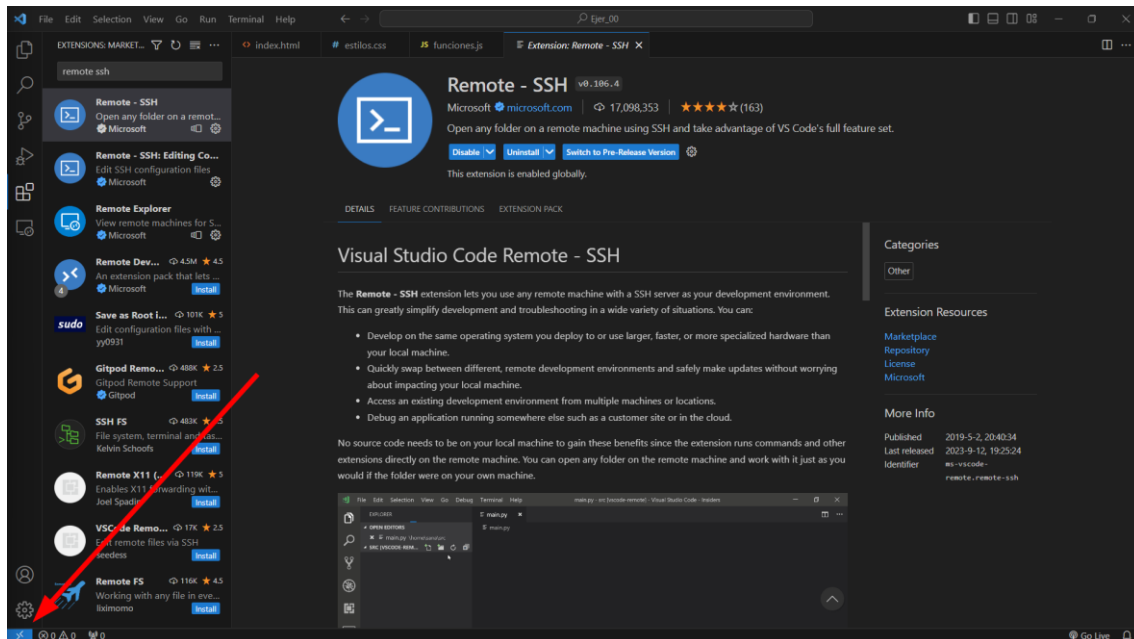
No source code needs to be on your local machine to gain these benefits since the extension runs commands and other extensions directly on the remote machine. You can open any folder on the remote machine and work with it just as you would if the folder were on your own machine.

**Categories**  
Other

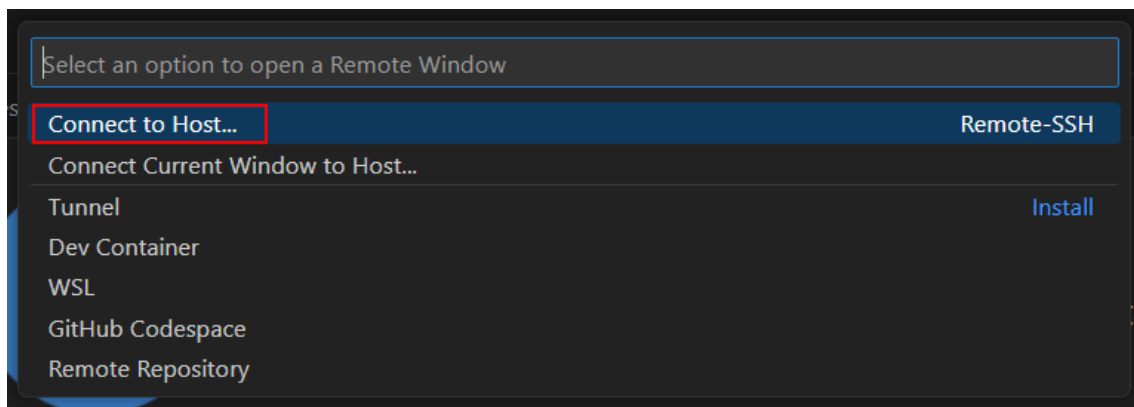
**Extension Resources**  
[Marketplace](#)  
[Repository](#)  
[License](#)  
[Microsoft](#)

**More Info**  
Published 2019-5-2, 20:40:34  
Last released 2023-9-12, 19:25:24  
Identifier ms-vscode-remote.remote-ssh

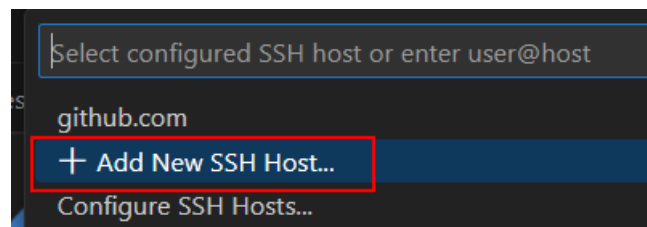
- Una vez finalizada la instalación, hacemos clic en la esquina inferior izquierda, en el rectángulo azul con símbolos de menor y mayor que:



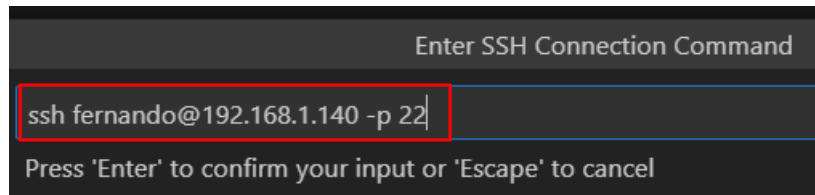
- Al hacer clic, nos pedirá en la parte superior de la pantalla que elijamos una opción, en nuestro caso será “Connect to Host...”



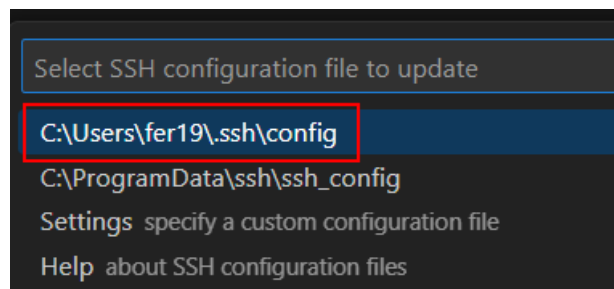
- Hacemos clic en la opción “Add New SSH Host”



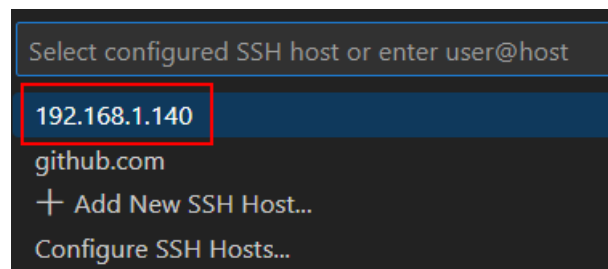
- Escribiremos las siguientes indicaciones, donde establecemos, el nombre del usuario, seguido de la IP de la máquina y por último le indicamos el puerto que servirá para realizar la conexión:



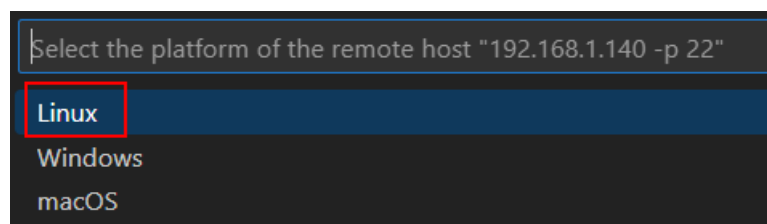
- Luego nos preguntará cual es el archivo de configuración SSH que queremos modificar, hacemos clic sobre el primero



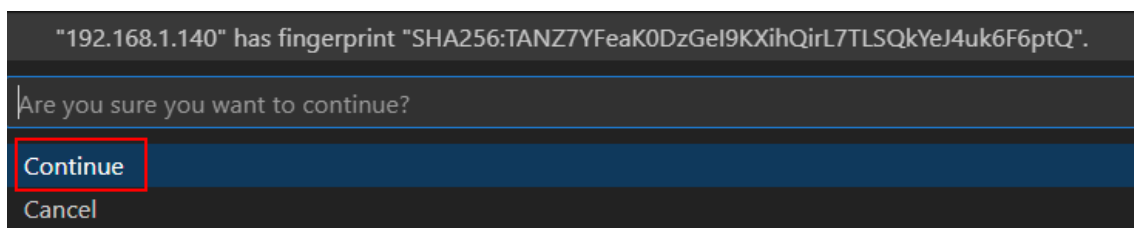
- Ahora, nos saldrá directamente la dirección IP de nuestra máquina servidor, sobre la que nos queremos conectar



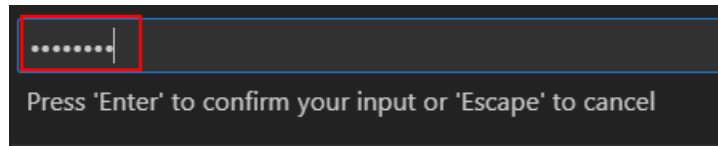
- Nos pedirá que le indiquemos en que plataforma está trabajando el equipo servidor, En nuestro caso, haremos clic en Linux:



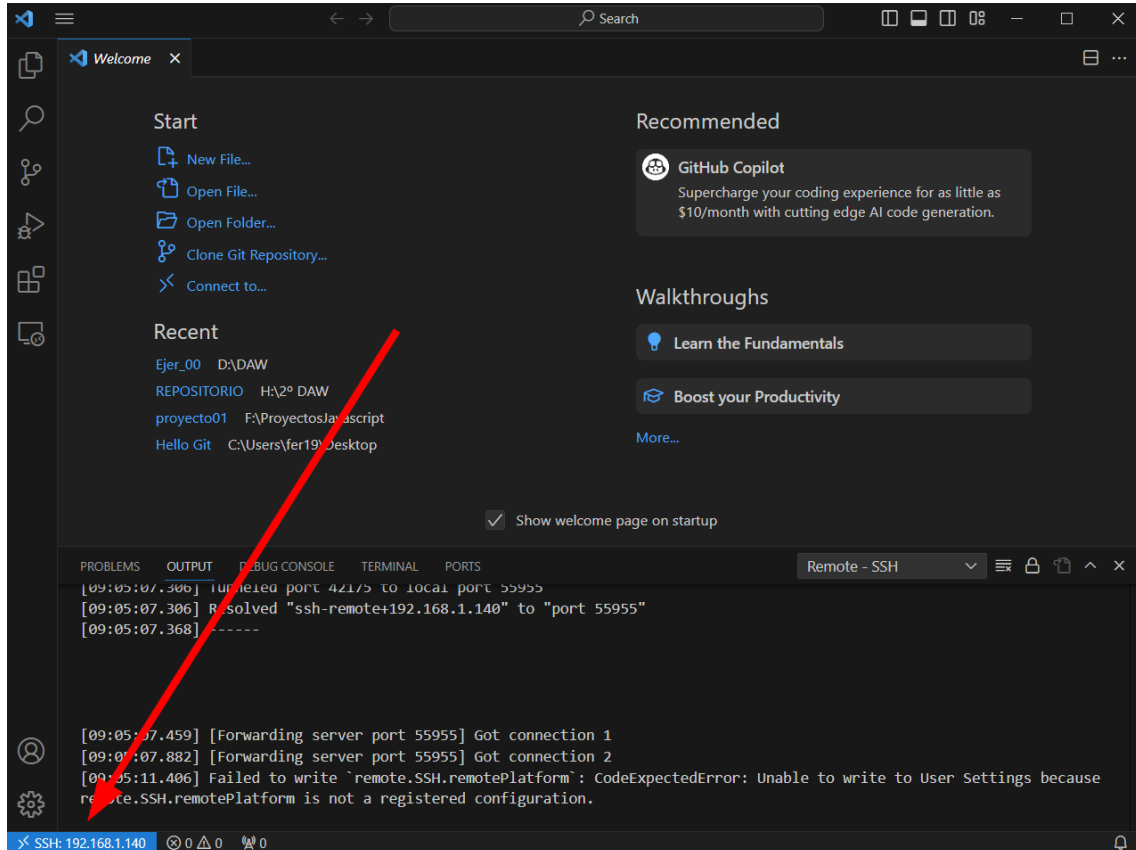
- Nos preguntará si queremos continuar con la acción, hacemos clic en “Continue”



- Por último, nos pedirá que introduzcamos la contraseña del usuario de Linux

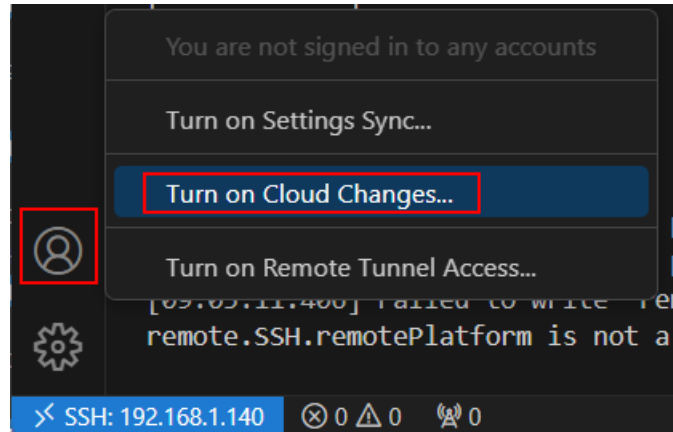


- Ahora en la ventana nueva de Visual Studio Code ya nos aparece en la esquina inferior izquierda, que la conexión ha sido realizada

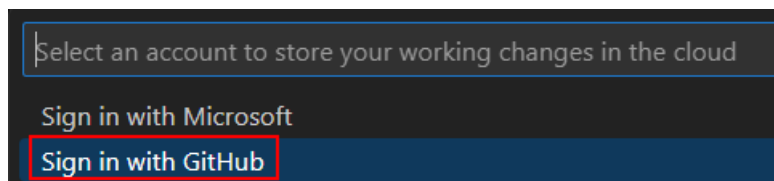


## Creación de un repositorio en GitHub y su conexión con Visual Studio Code

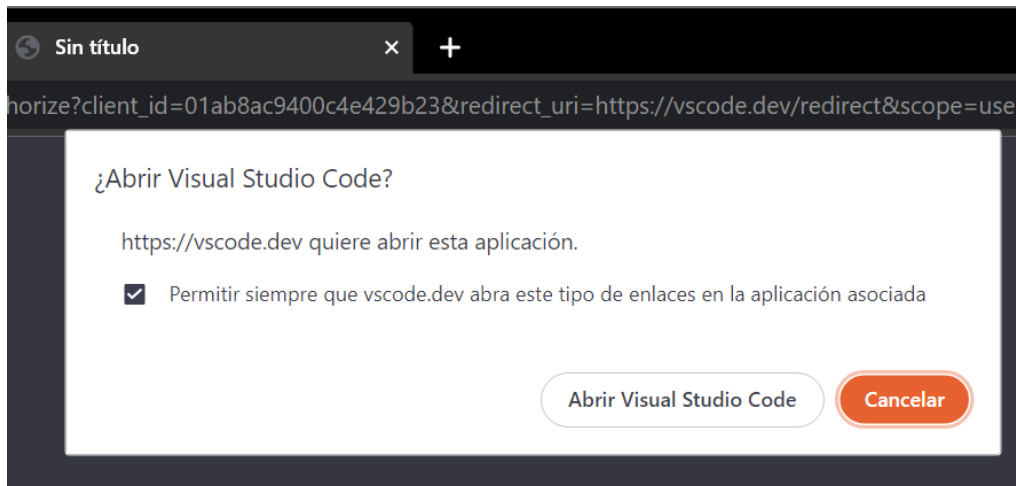
- Dentro de Visual Studio Code, en la esquina inferior izquierda, hacemos clic sobre el dibujo de perfil que aparece, posteriormente hacemos clic en la opción “Turn on Cloud Changes”.



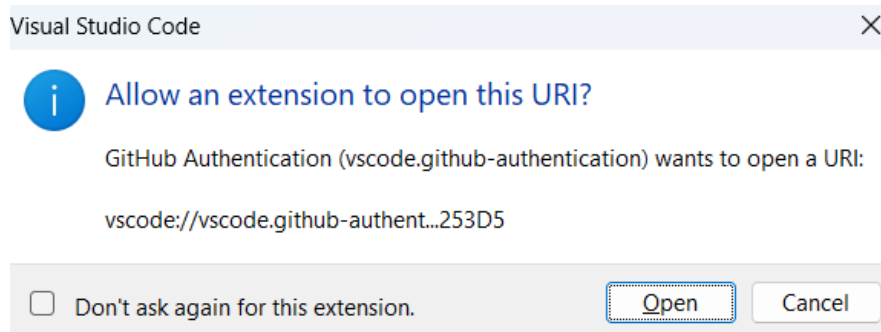
- En la parte superior de la pantalla, nos preguntará que con que aplicación queremos iniciar sesión, en nuestro caso, hacemos clic en “GitHub”.



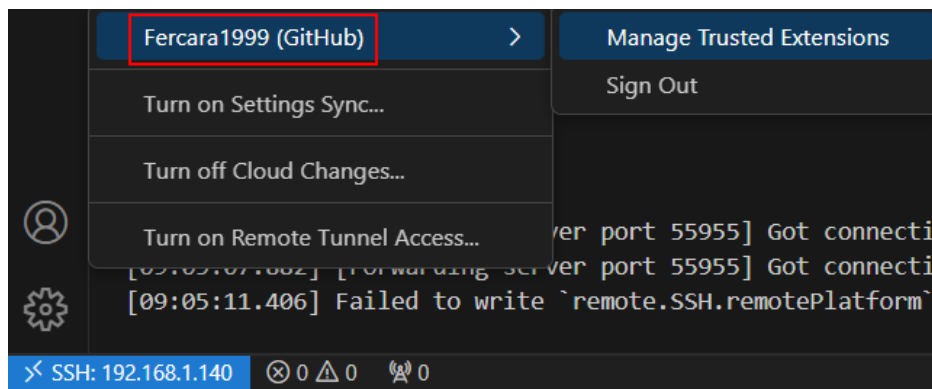
- Nos pedirá nuestros datos de inicio de sesión y verificar mediante un correo electrónico, nosotros como ya tenemos iniciada la sesión, nos redirige de nuevo a Visual Studio Code.



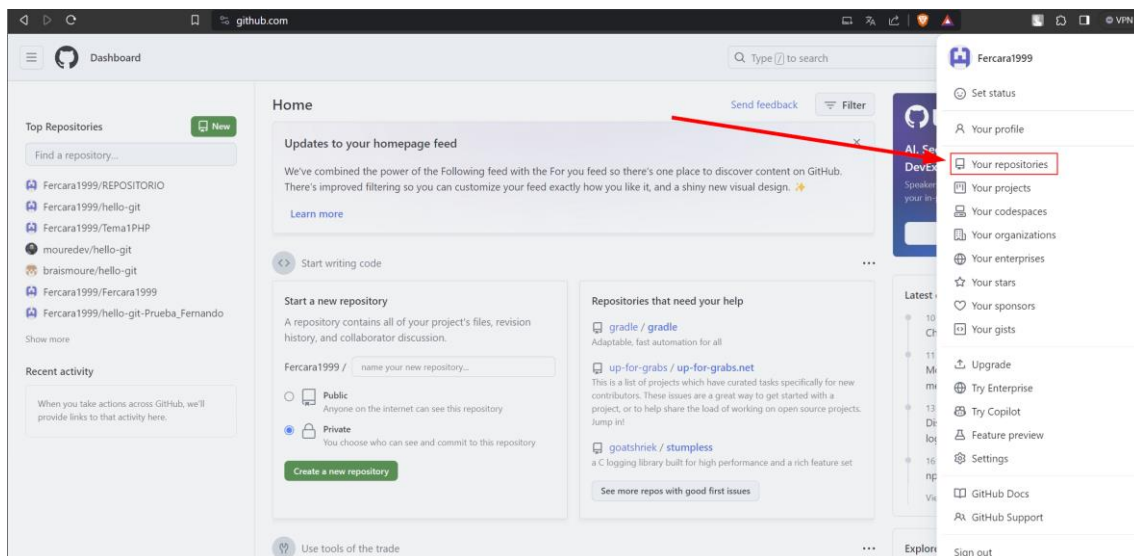
- Nos pregunta si queremos seguir adelante con la autenticación, hacemos clic en “Open”.



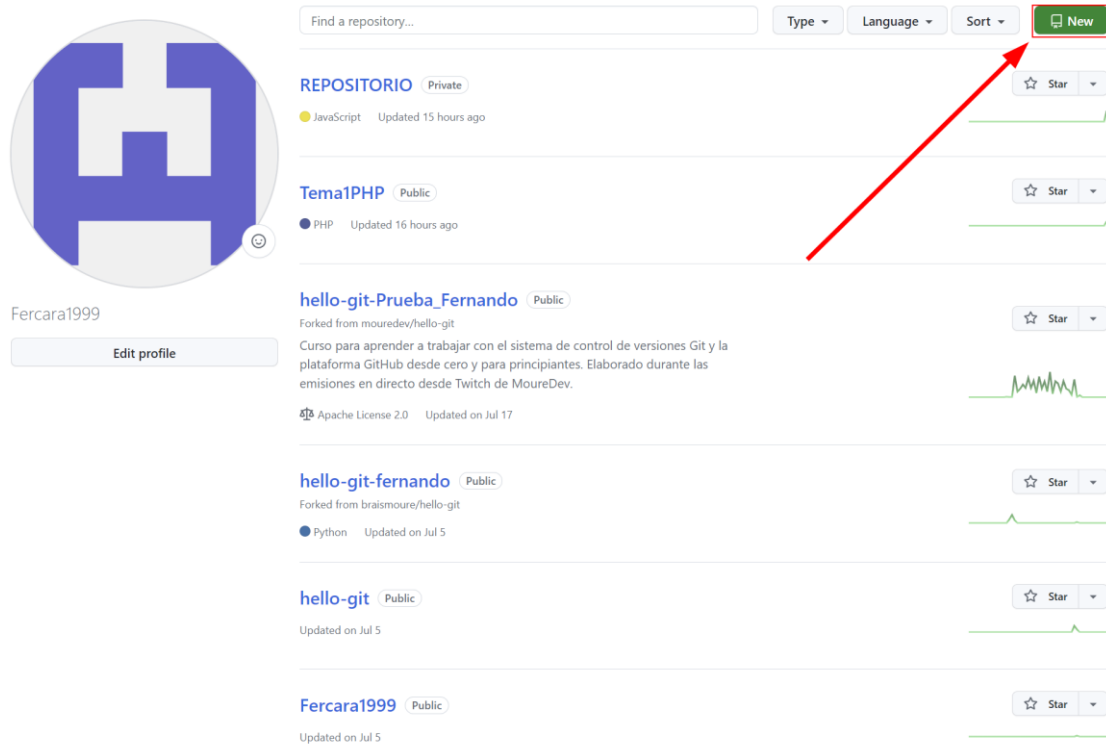
- Ahora cuando accedemos al apartado de perfil, ya tenemos iniciada la sesión en GitHub mediante Visual Studio Code



- El siguiente paso será acceder a github.com, desde aquí nos iremos dentro del icono de nuestro usuario, a la sección “Your repositories”



- En esta pestaña, nos mostrará los repositorios que tenemos creados, y nos dará la opción de crear uno nuevo, que será lo que hagamos en este caso



The screenshot displays the GitHub profile of user Fercara1999. On the left, there is a circular profile picture with a blue and white geometric design, the username 'Fercara1999', and an 'Edit profile' button. The main area shows a list of repositories. At the top of this list is a search bar labeled 'Find a repository...', followed by filters for 'Type', 'Language', and 'Sort'. A red box highlights a 'New' button with a plus icon, and a red arrow points to it from the right. Below the filters, the repository list includes: 'REPOSITORIO' (Private, JavaScript, updated 15 hours ago), 'Tema1PHP' (Public, PHP, updated 16 hours ago), 'hello-git-Prueba\_Fernando' (Public, forked from mouredev/hello-git, updated on Jul 17), 'hello-git-fernando' (Public, forked from braismoure/hello-git, updated on Jul 5), 'hello-git' (Public, updated on Jul 5), and 'Fercara1999' (Public, updated on Jul 5). Each repository entry has a star icon and a dropdown arrow to its right.




- Le daremos a este nuevo repositorio como nombre “Tema1Tarea” y su ámbito será público.

## 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 \*

 Fercara1999

Repository name \*

Tema1Tarea

✔ Tema1Tarea is available.

Great repository names are short and memorable. Need inspiration? How about [urban-pancake](#) ?

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

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

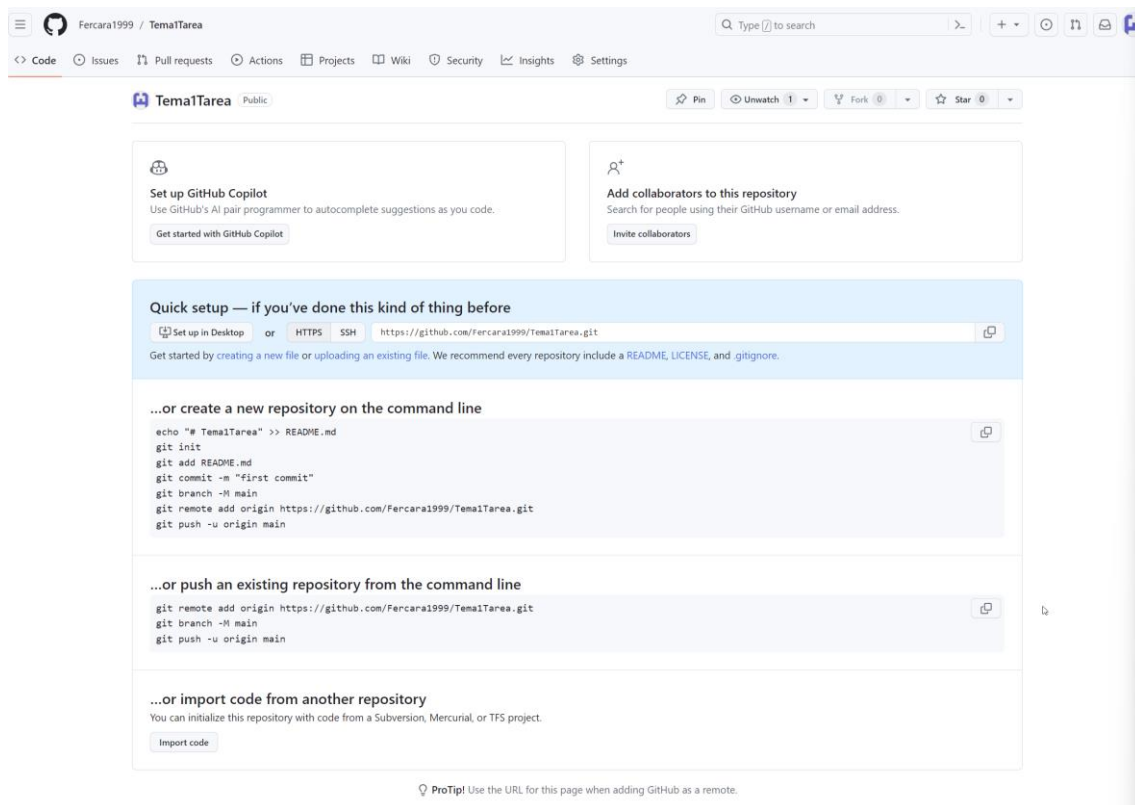
License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

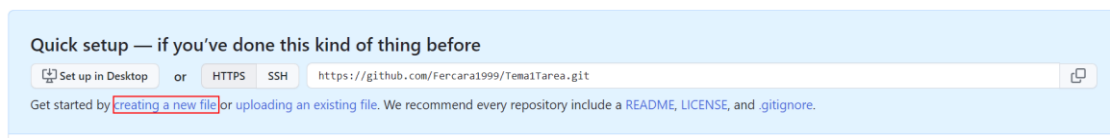
 You are creating a public repository in your personal account.

Create repository

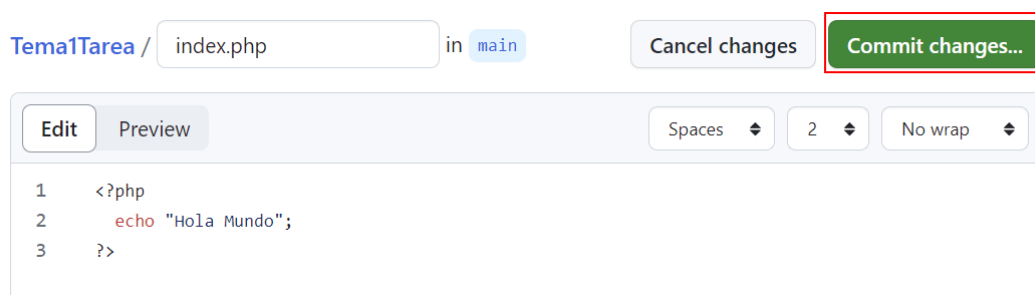
- Ya hemos creado nuestro repositorio.



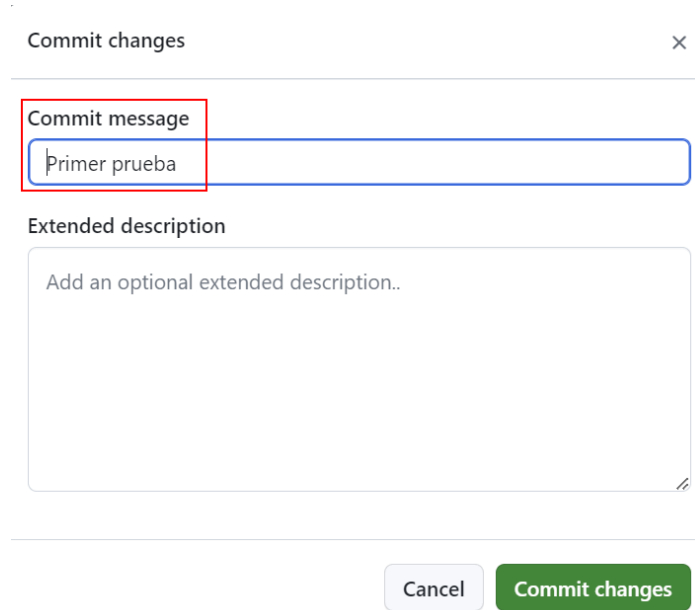
- Ahora vamos a crear un archivo para nuestro repositorio. Lo haremos haciendo clic en la opción “Creating a new file”.



- Este será el contenido de nuestro archivo, le daremos como nombre “index.php” y hacemos clic en “Commic Changes” para que nos confirme los cambios que hemos realizado.



- Le ponemos un nombre al *commit* que vamos a realizar



Commit changes

Commit message

Primer prueba

Extended description

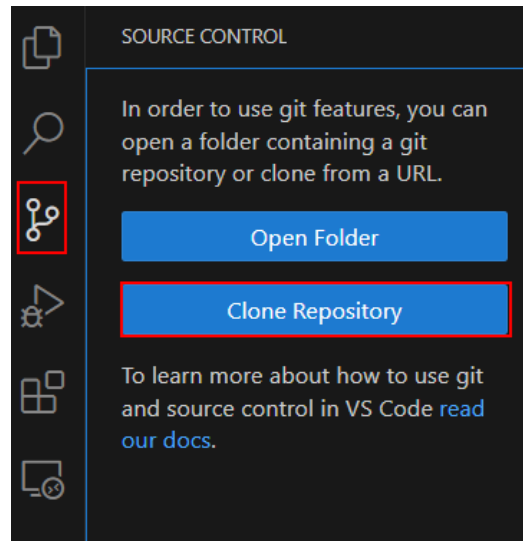
Add an optional extended description..

Cancel Commit changes

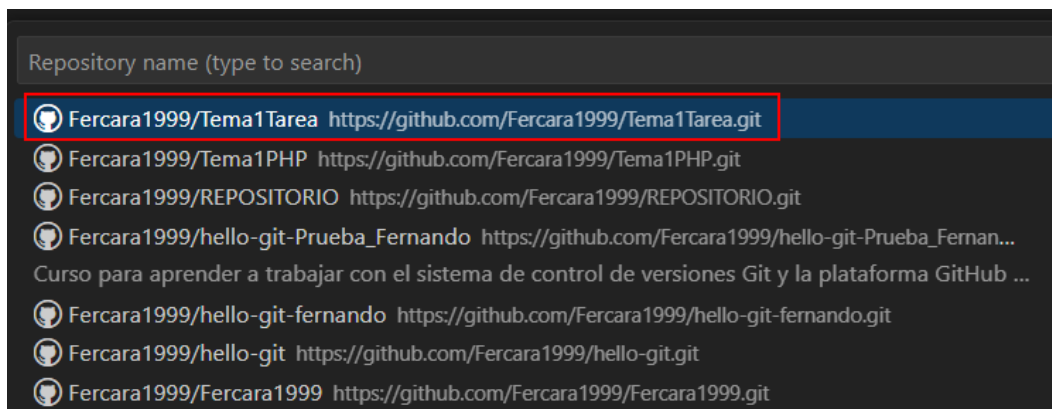
- Antes de continuar, vamos a crear en nuestra máquina Linux, una carpeta que irá asociada al repositorio que hemos creado en GitHub. Esta carpeta se encontrará en la ruta: `/var/www/html` y le daremos el nombre “Tema1Tarea”. Posteriormente vamos a darle control total a todos los usuarios con el comando `chmod`. Como vemos con `ls -la`, ya tenemos la carpeta creada con todos los permisos dados correctamente.

```
fernando@fernando:/var/www/html$ sudo mkdir Tema1Tarea
[sudo] password for fernando:
fernando@fernando:/var/www/html$ sudo chmod 777 Tema1Tarea/
fernando@fernando:/var/www/html$ ls -la
total 28
drwxr-xr-x 3 root root 4096 sep 27 08:07 .
drwxr-xr-x 3 root root 4096 sep 26 19:24 ..
-rw-r--r-- 1 root root 10703 sep 27 06:11 index.html
-rw-r--r-- 1 root root 43 sep 27 06:19 prueba.php
drwxrwxrwx 2 root root 4096 sep 27 08:07 Tema1Tarea
```

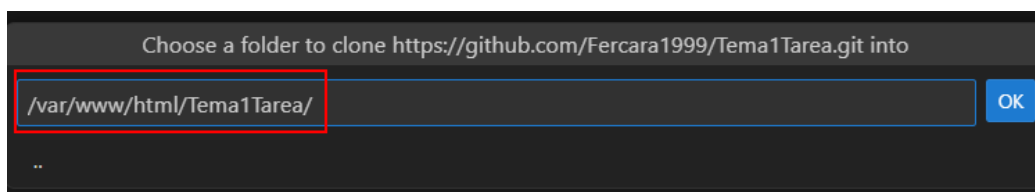
- Ahora, para que nos muestre este repositorio en visual Studio Code, primer debemos hacer clic en el dibujo de la rama, que aparece en la parte izquierda de la pantalla, a continuación, haremos clic sobre la opción “Clone Repository”.



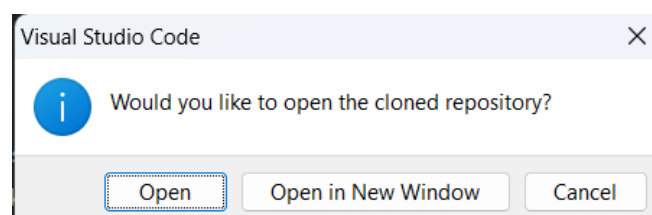
- Nos aparecerá arriba en la barra de búsqueda, los repositorios que se encuentran asociados a nuestra cuenta, hacemos clic en el que hemos llamado “Tema1Tarea”.



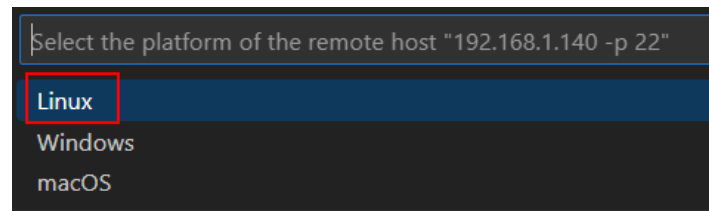
- Ahora le diremos sobre que carpeta de nuestro servidor se debe clonar esa carpeta, en este caso: /var/www/html/Tema1Tarea



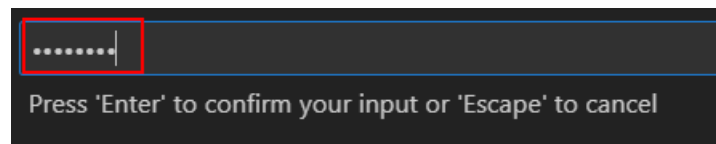
- Nos pregunta si queremos que se nos abra el clon de nuestro repositorio, hacemos clic en *open*.



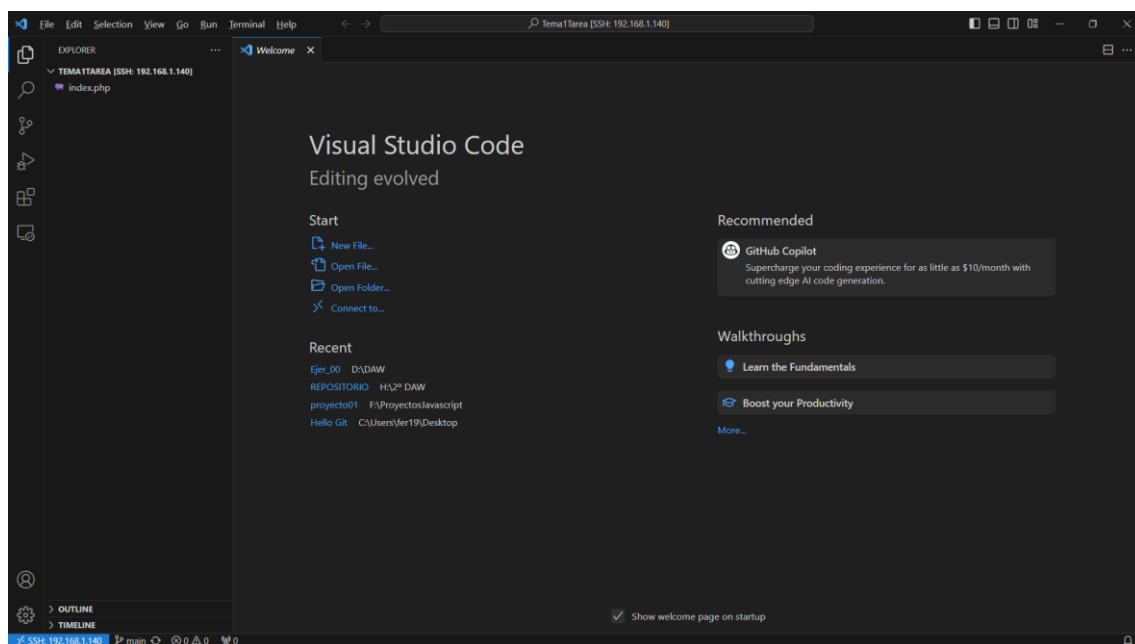
- Nos volverá a preguntar cuál es el sistema operativo de la máquina servidor



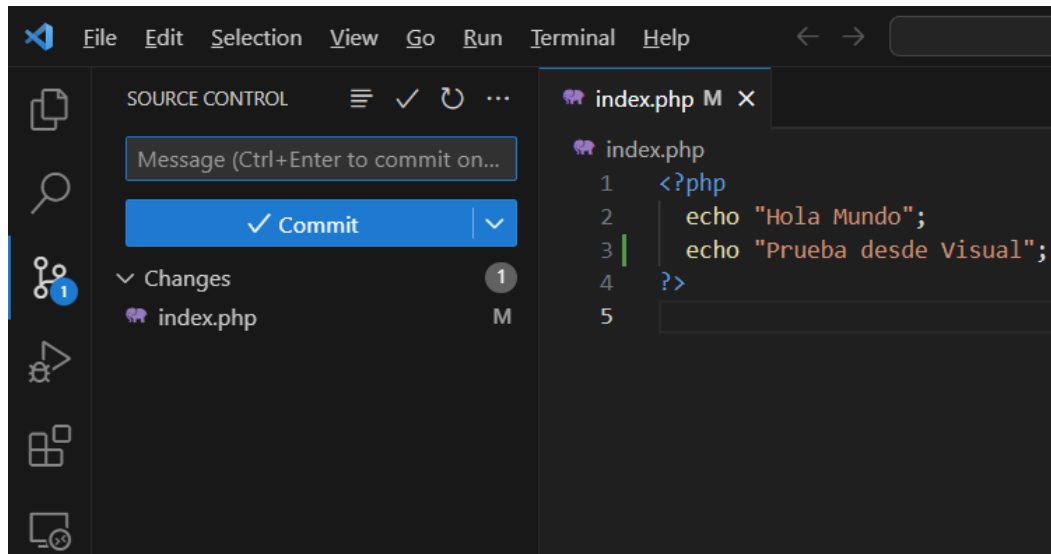
- También nos pedirá que introduzcamos la contraseña de nuestro usuario de la máquina.



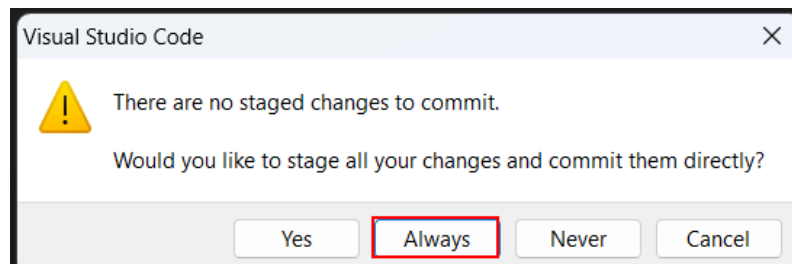
- Tras ello, el repositorio se encuentra clonado en nuestro Visual Studio Code



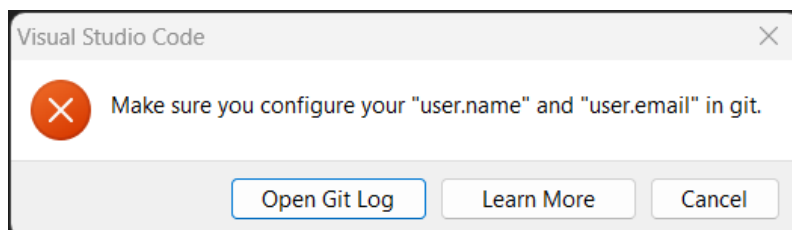
- Ahora haremos una prueba editando el archivo index.php y subiendo los cambios al repositorio de GitHub. Una vez acabada la edición, guardaremos el estado del fichero.



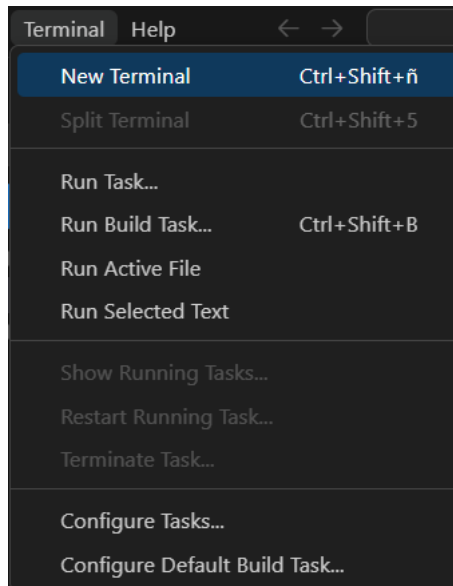
- Hacemos clic en Commit y nos saldrá este mensaje para confirmar los cambios sobre el que haremos clic en "Always" para que no nos vuelva a aparecer



- Nos saldrá este error, que nos indica que no tenemos indicado un Usuario y un correo en el archivo de configuración de Git.



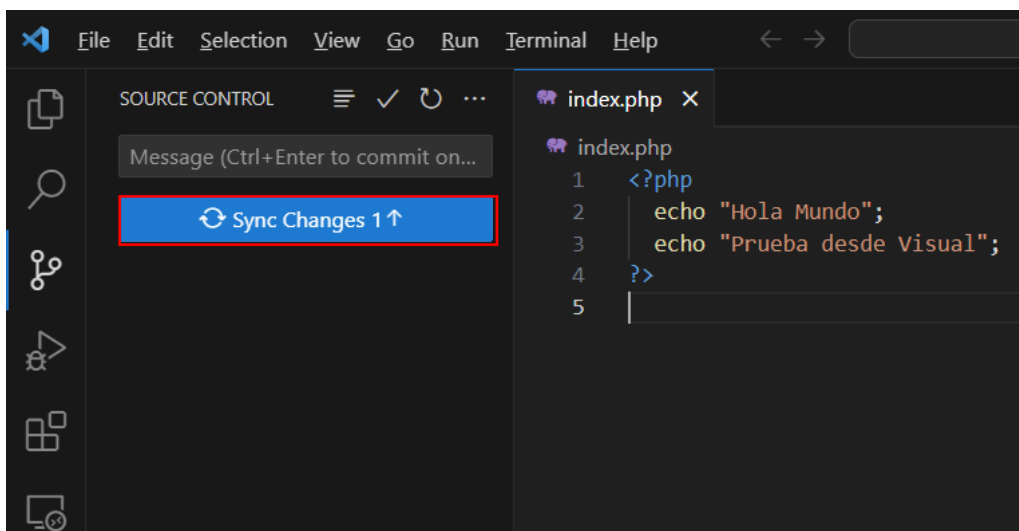
- Para solucionar este error, debemos abrir una nueva terminal, haciendo clic en la parte superior de la pantalla, elegimos la opción “Terminal”, y dentro de ella, hacemos clic en “New terminal”.



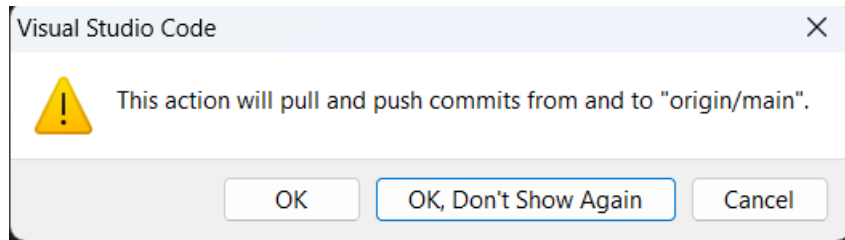
- En la terminal que se nos abre, debemos escribir los siguientes comandos con nuestro nombre y correo para solucionar el problema

```
fernando@fernando:/var/www/html/Tema1Tarea/Tema1Tarea$ git config --global user.name "Fernando Calles"  
fernando@fernando:/var/www/html/Tema1Tarea/Tema1Tarea$ git config --global user.email "fercara1999@gmail.com"
```

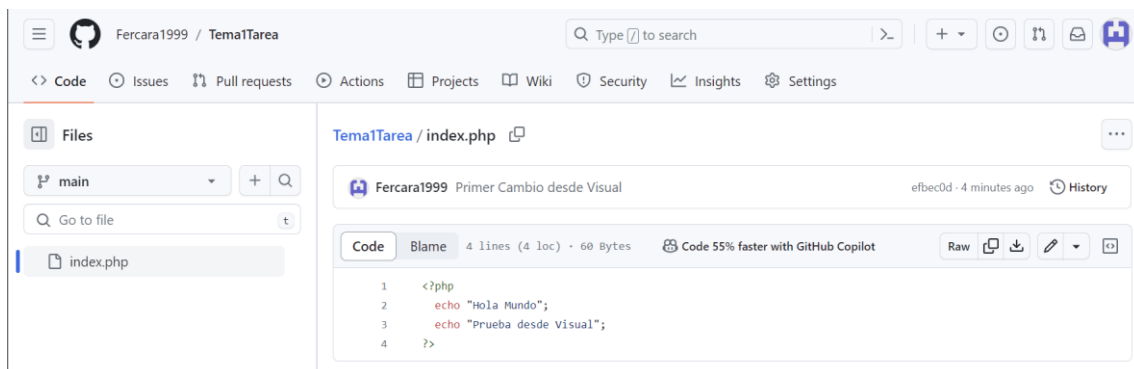
- Ahora ya nos ha dejado hacer el “Commit” sobre el archivo. Ahora los cambios se encuentran guardados, pero no han sido subidos a nuestro repositorio de GitHub, para ello, debemos hacer clic en la opción “Sync Changes”, es lo que se conoce como hacer “Push” de los “Commits” guardados.



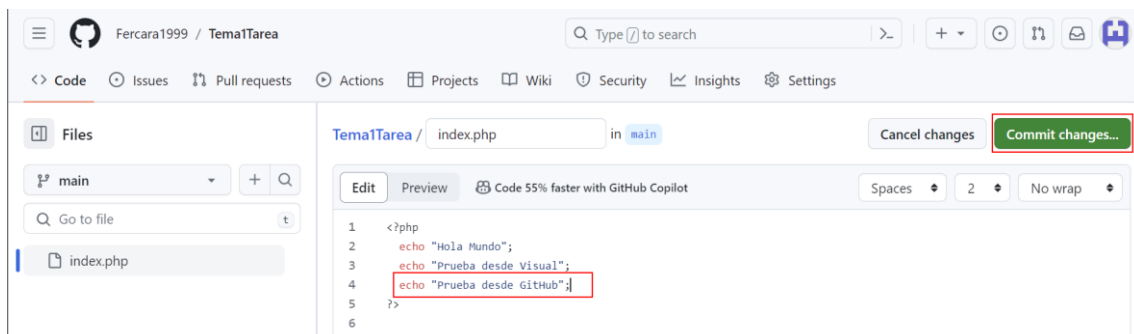
- Nos pregunta si estamos de acuerdo con hacer “*Push*” de los datos y haremos clic en “*Ok, Don’t Show Again*” para que no nos lo vuelva a preguntar de nuevo.



- Ahora si accedemos desde el repositorio al archivo *index.php*, nos encontramos con que ya se muestran los cambios realizados en nuestro archivo.

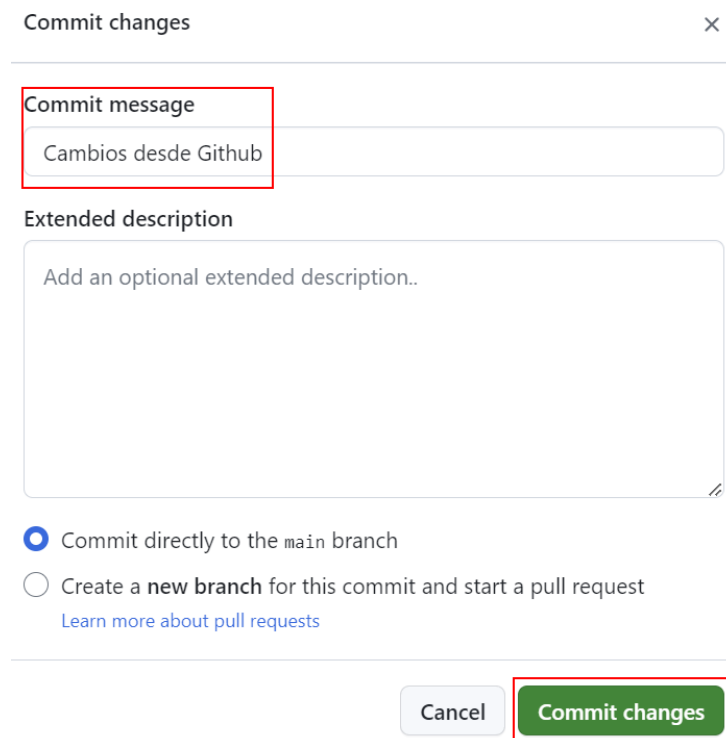


- Lo siguiente será realizar cambio en el código desde GitHub y que estos se muestren en Visual Studio Code. Lo primero será escribir una nueva línea desde GitHub, a continuación, hacemos clic en la opción “*Commit Changes*” para que estos queden guardados.

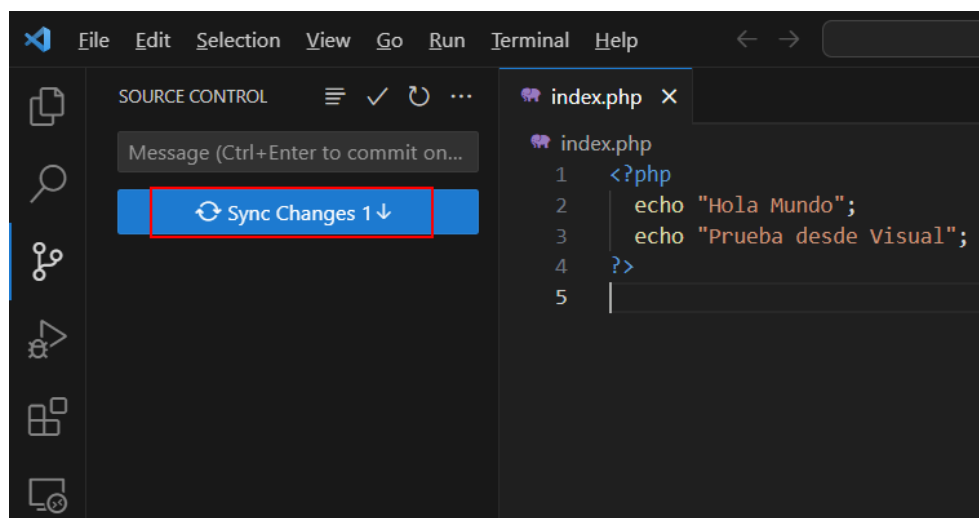




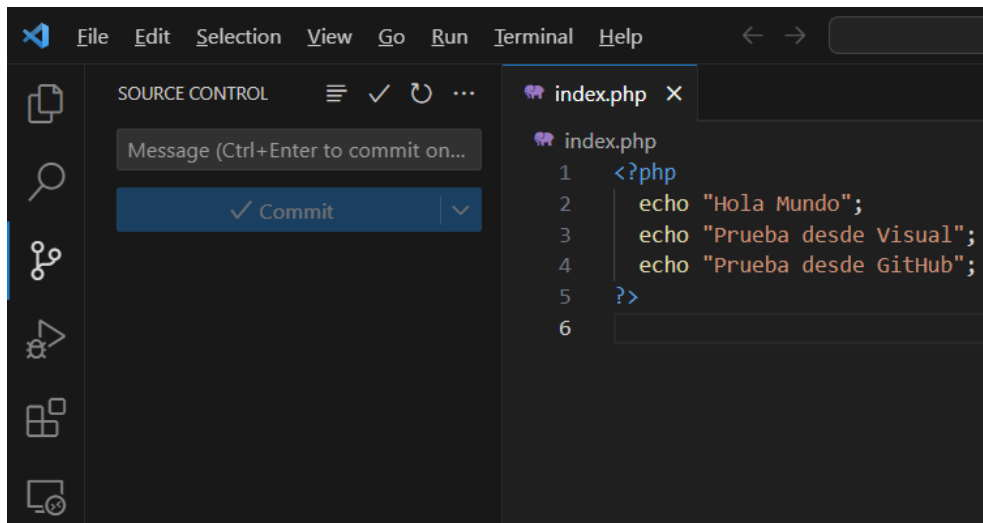
- Le daremos un nombre al Commit que vamos a realizar y hacemos clic sobre “Commit Changes”. Posteriormente, los cambios ya se encuentran guardados.



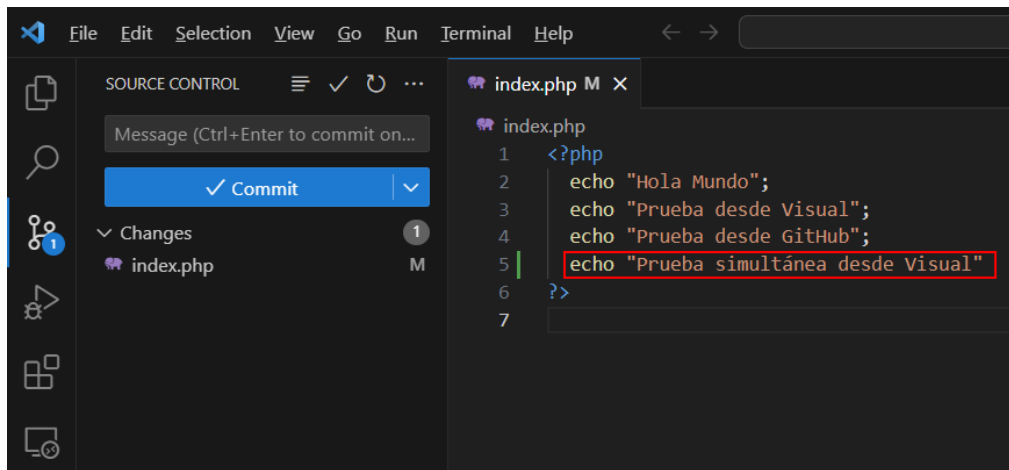
- Ahora, para que estos cambios se actualicen desde nuestro Visual Studio, debemos hacer clic sobre la opción “Sync Changes”, el cual ya nos indica que hay un cambio que descargar



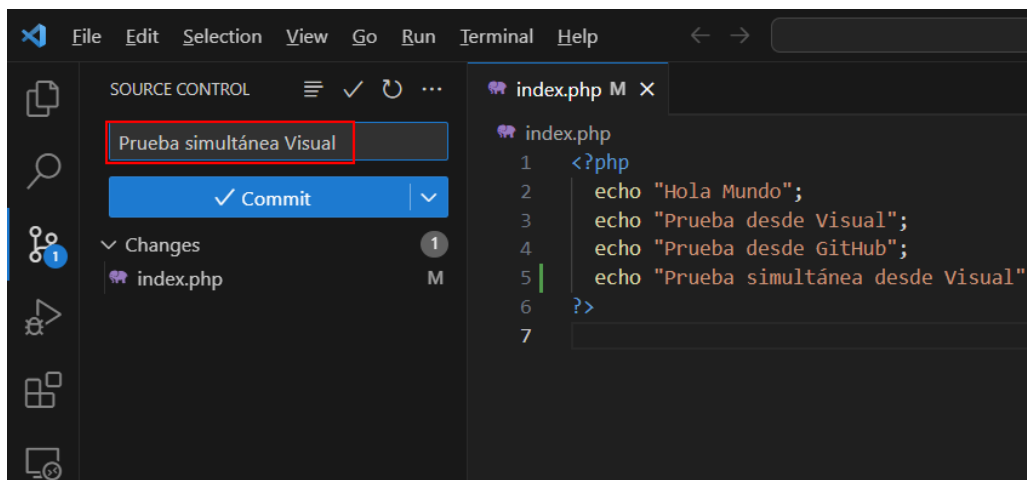
- Ahora ya se encuentran los cambios sincronizados en nuestro Visual Studio Code.



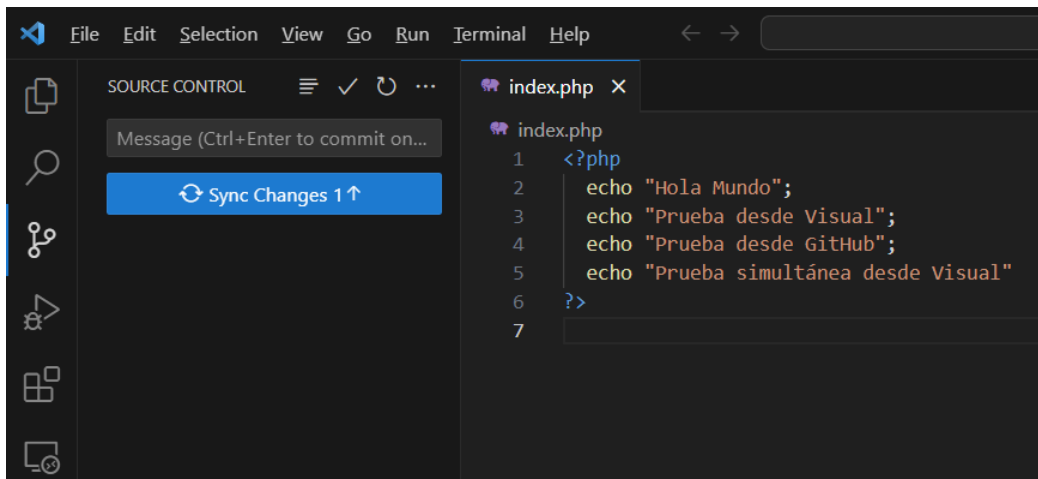
- Ahora vamos a realizar cambios desde ambos sitios, tanto desde GitHub como desde Visual Studio Code. En primer lugar, hacemos este cambio en el Visual Studio Code.



- Hacemos un "Commit" sobre los cambios hechos



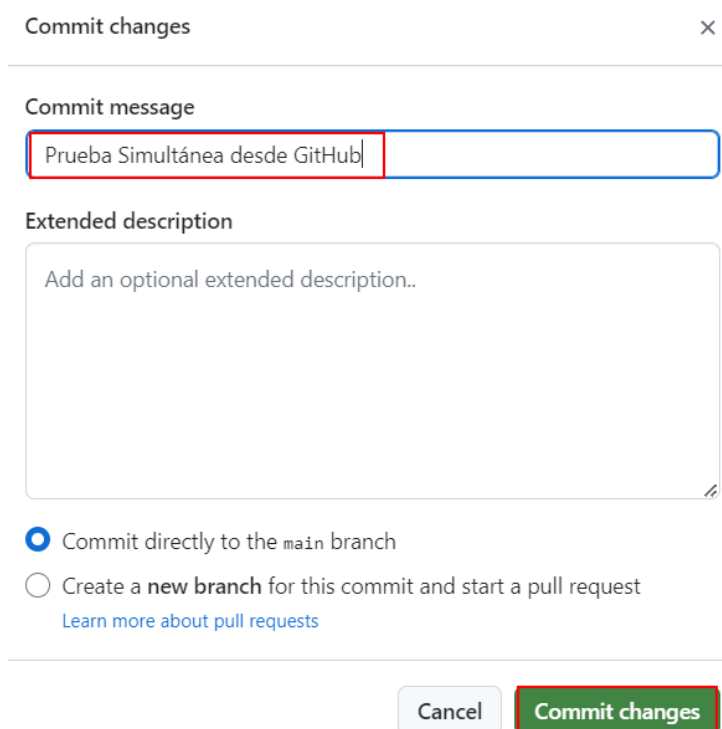
- Una vez realizados estos cambios, es la hora de pasar a realizar cambios desde GitHub



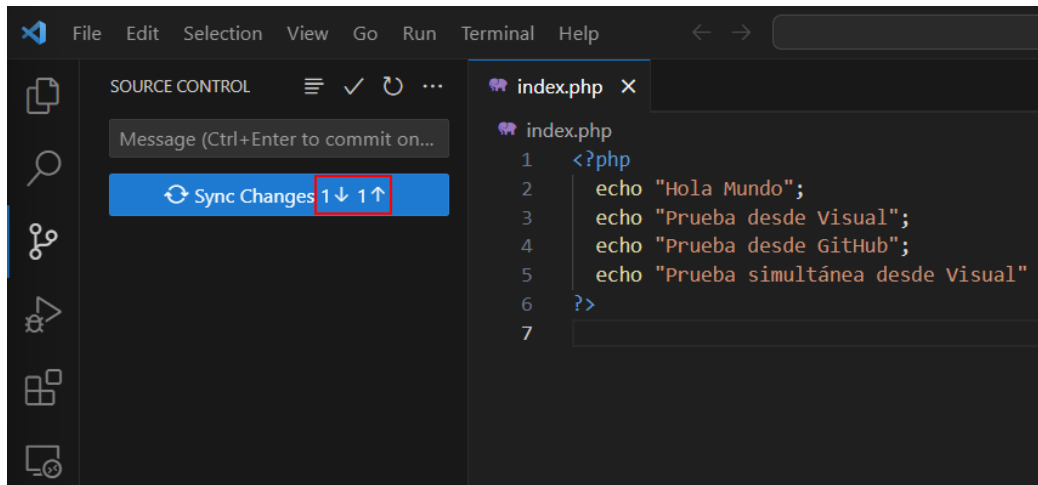
- En GitHub realizamos este cambio de prueba, a continuación, hacemos clic en “Commit Changes”.



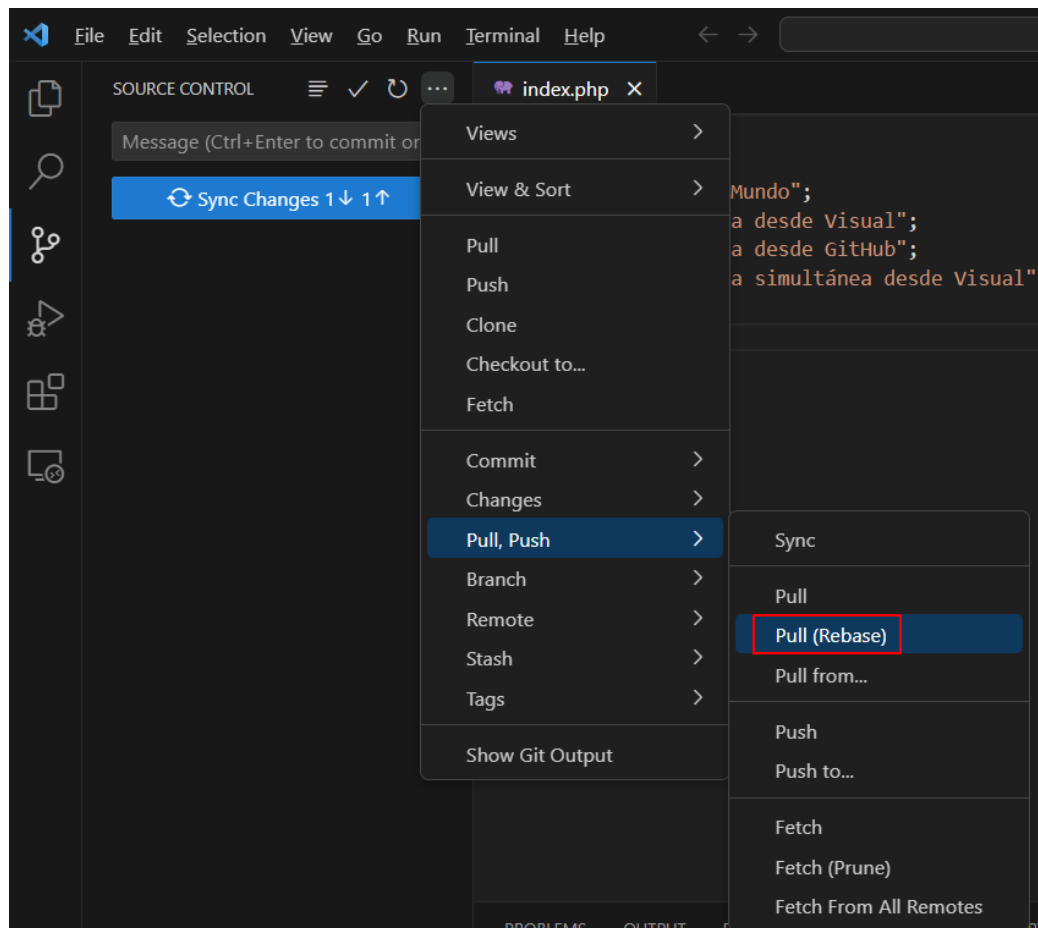
- Le damos un nombre a este “Commit” y hacemos clic en “Commit Changes”.



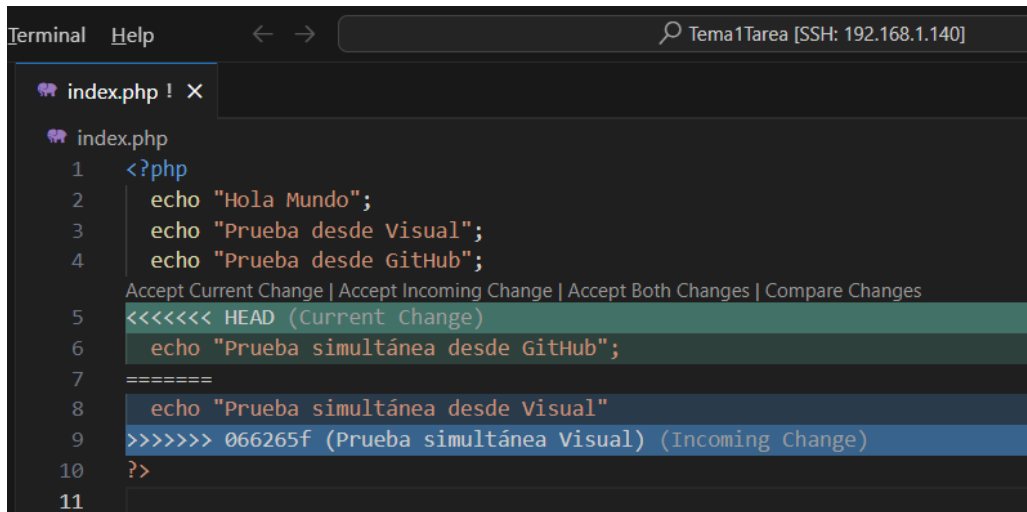
- Ahora que se encuentran los cambios en ambos sitios, iremos a Visual Studio Code, donde ya nos indica que hay un cambio desde cada sitio.



- Lo siguiente será hacer clic en “Pull (Rebase)”, que nos mostrará los cambios que se han hecho en ambos sitios.



- Esta pantalla nos muestra que se han hecho cambios procedentes tanto desde Visual Studio Code (en azul) como desde GitHub (en verde).

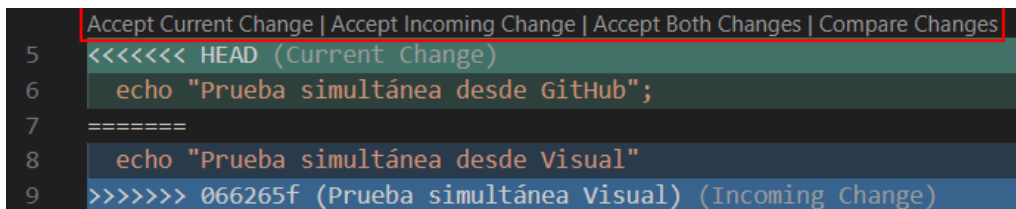


```
Terminal Help  ← →  Tema1Tarea [SSH: 192.168.1.140]

index.php ! ×

index.php
1  <?php
2      echo "Hola Mundo";
3      echo "Prueba desde Visual";
4      echo "Prueba desde GitHub";
5      <<<<<< HEAD (Current Change)
6          echo "Prueba simultánea desde GitHub";
7      =====
8          echo "Prueba simultánea desde Visual"
9      >>>>>> 066265f (Prueba simultánea Visual) (Incoming Change)
10     ?>
11
```

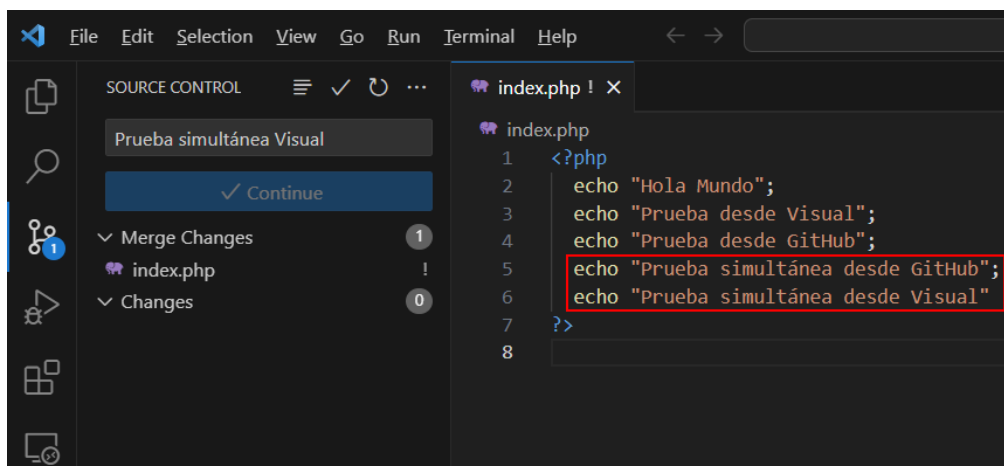
- Para elegir que opción aplicaremos a nuestro código, usaremos el menú que aparece en la parte superior, el cual nos permitirá: mantener los cambios hechos desde Visual Studio Code, mantener los cambios hechos desde GitHub, mantener los cambios de ambos lados o comparar los cambios que se producen en ambos sitios. Nosotros haremos clic en mantener los cambios de ambos sitios.



```
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes

5  <<<<<< HEAD (Current Change)
6      echo "Prueba simultánea desde GitHub";
7      =====
8          echo "Prueba simultánea desde Visual"
9      >>>>>> 066265f (Prueba simultánea Visual) (Incoming Change)
```

- Este será el código que nos queda tras realizar estos cambios



```
File Edit Selection View Go Run Terminal Help  ← →

SOURCE CONTROL  ✓ ↻ ...

Prueba simultánea Visual

Continue

Merge Changes  1
index.php      !
Changes        0

index.php ! ×

index.php
1  <?php
2      echo "Hola Mundo";
3      echo "Prueba desde Visual";
4      echo "Prueba desde GitHub";
5      echo "Prueba simultánea desde GitHub";
6      echo "Prueba simultánea desde Visual";
7  ?>
8
```