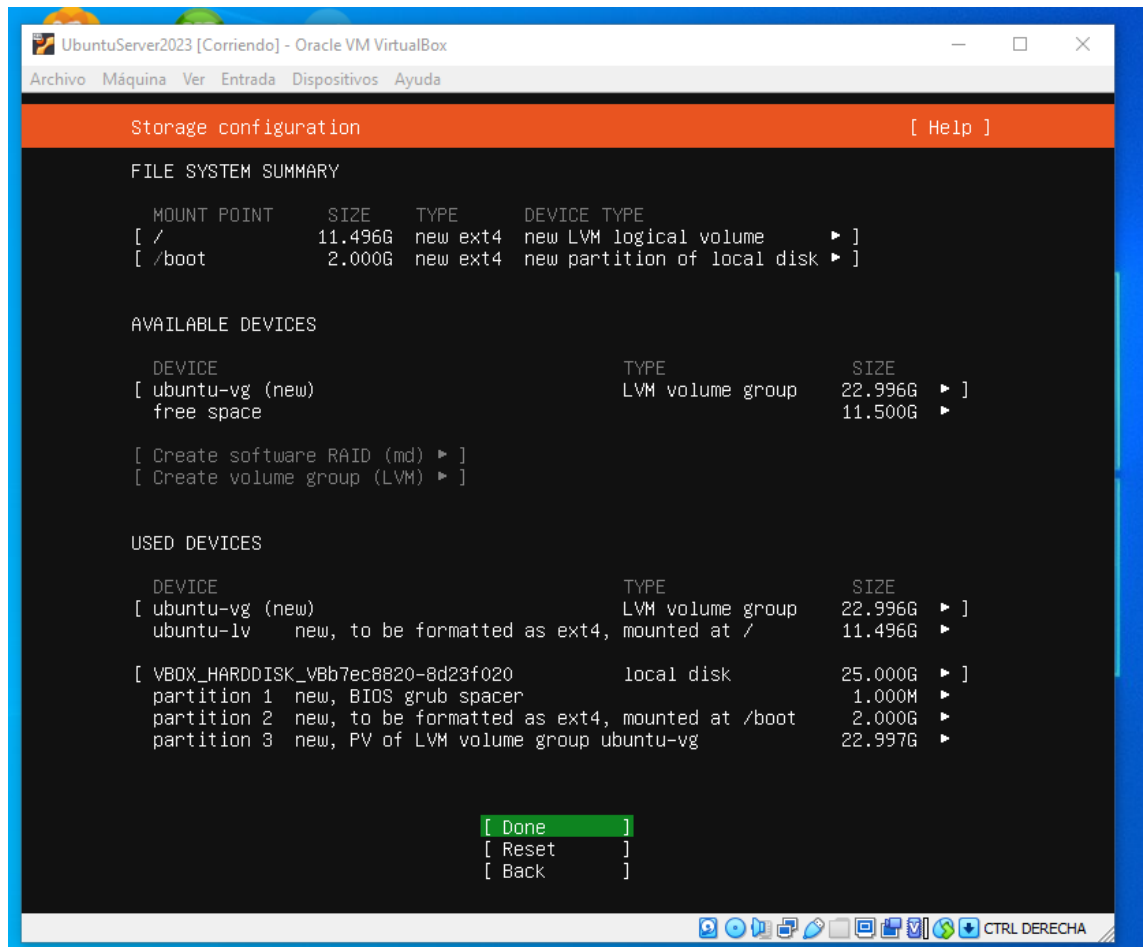
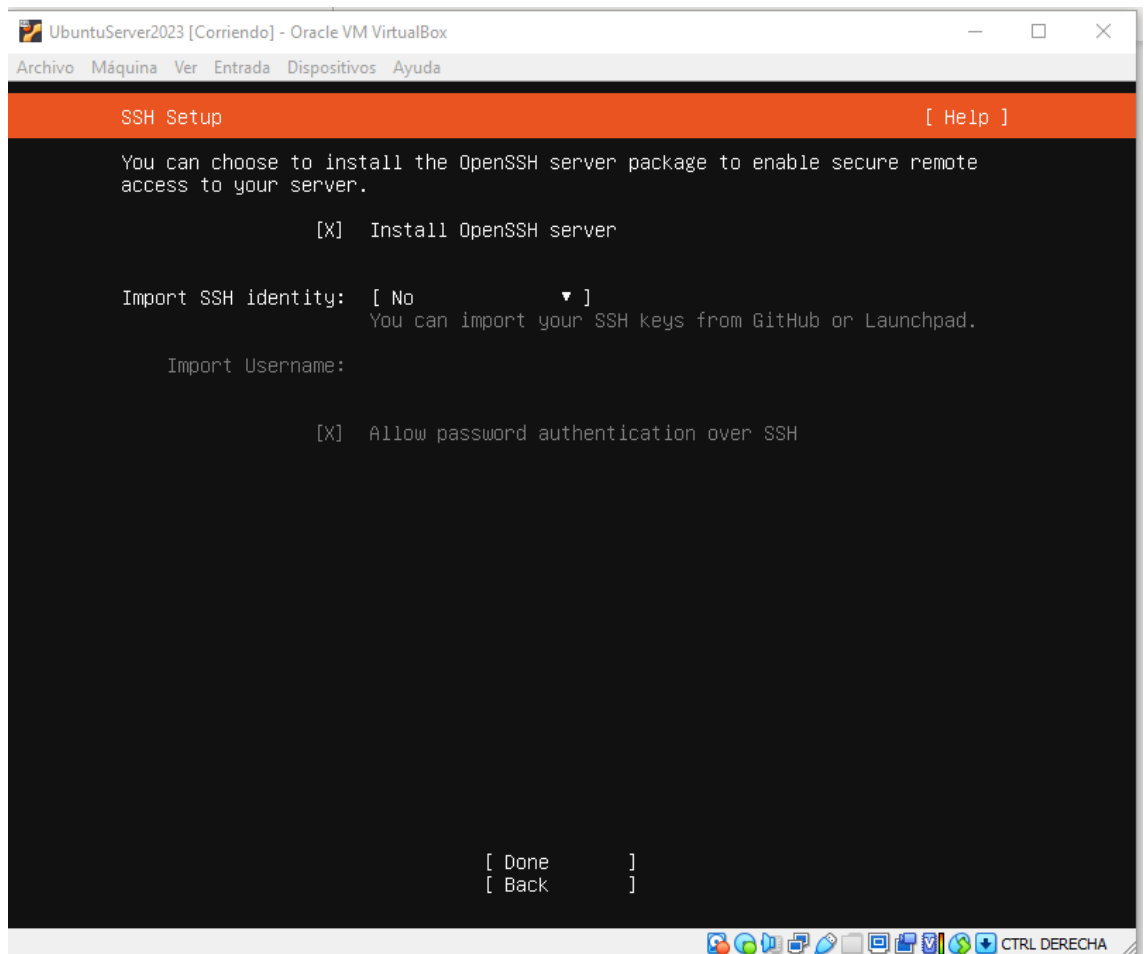


CREAR SERVIDOR CON UBUNTU

Raul Ferrero Vicente

1. Descargamos ISO de Ubuntu Server y la instalamos en Virtualbox, configurando el adaptador de red en modo adaptador puente.





Sudo apt-get install net-tools para poder ejecutar ifconfig.

Configuramos netplan para estar en la misma red y hacemos ping al equipo anfitrión.

- `sudo netplan apply`

LaRed5G

Propiedades

SSID:

Protocolo:

Tipo de seguridad:

Banda de red:

Canal de red:

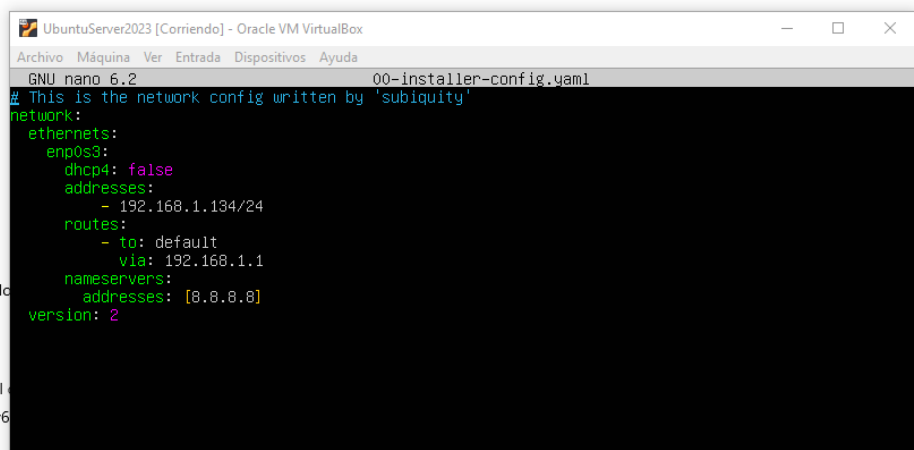
Velocidad de vínculo
(transmisión):

Dirección IPv6:

Dirección IPv6 local:

Servidores DNS IPv6:

Dirección IPv4:



- `sudo netplan apply` (para aplicar cambios de netplan)

```
UbuntuServer2023 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
raul@raulserver:/etc/netplan$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.134 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fd8c:15c7:3565:8800:a00:27ff:fe25:deef prefixlen 64 scopeid 0x0<global>
    inet6 fe80::a00:27ff:fe25:deef prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:25:de:ef txqueuelen 1000 (Ethernet)
    RX packets 854 bytes 557702 (557.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 246 bytes 25814 (25.8 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 114 bytes 9667 (9.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 114 bytes 9667 (9.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

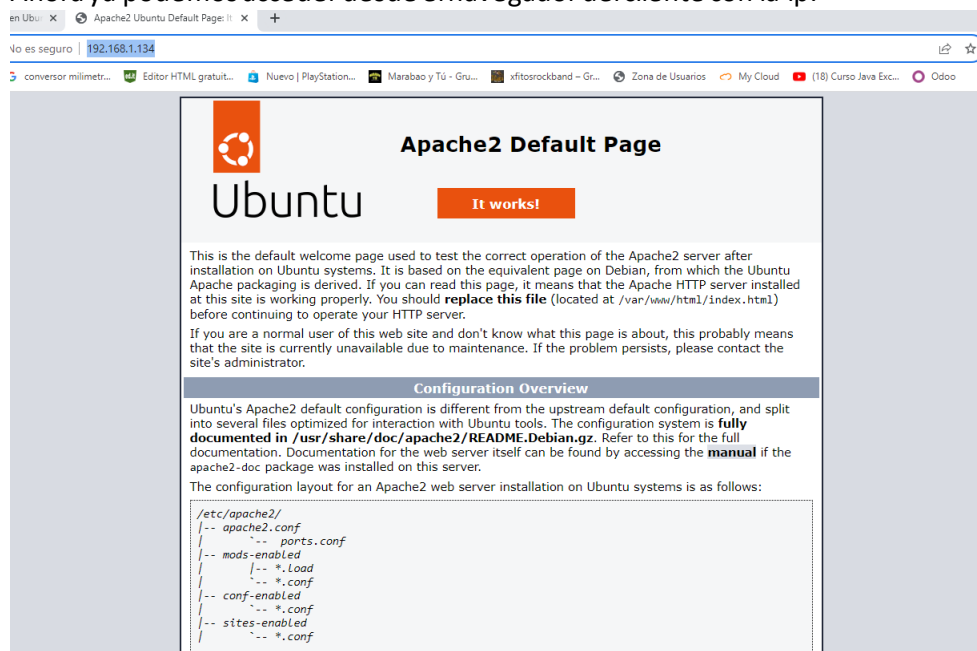
raul@raulserver:/etc/netplan$ ping 168.1.133
PING 168.1.133 (168.1.0.133) 56(84) bytes of data.
^C
--- 168.1.133 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5099ms

raul@raulserver:/etc/netplan$ ping 192.168.1.133
PING 192.168.1.133 (192.168.1.133) 56(84) bytes of data.
64 bytes from 192.168.1.133: icmp_seq=1 ttl=128 time=0.327 ms
64 bytes from 192.168.1.133: icmp_seq=2 ttl=128 time=0.870 ms
64 bytes from 192.168.1.133: icmp_seq=3 ttl=128 time=0.868 ms
64 bytes from 192.168.1.133: icmp_seq=4 ttl=128 time=0.818 ms
^Z
[1]+  Stopped                  ping 192.168.1.133
raul@raulserver:/etc/netplan$
```

Instalamos el servicio apache:

Sudo apt-get install apache2

Ahora ya podemos acceder desde el navegador del cliente con la ip:



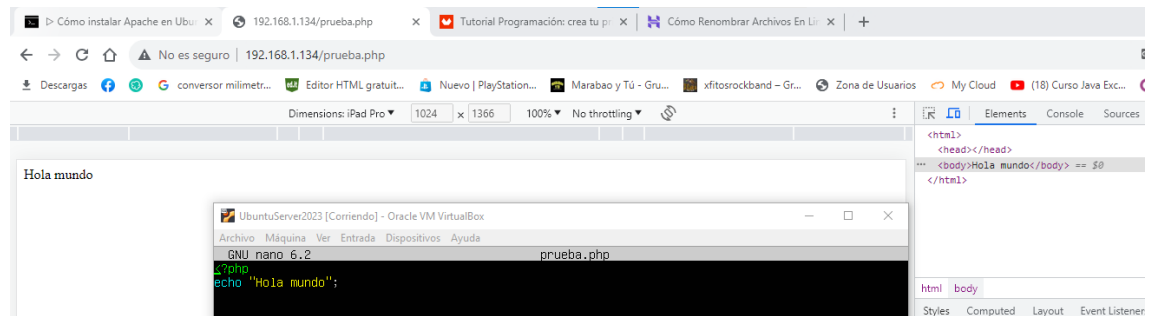
Desde `var/www/html` podemos editar la pagina `index.html`

Creamos otra pagina php con código php para probar.

Como no tenemos interprete php lo instalamos con el siguiente código:

`Sudo apt-get install php`

Ahora si es visible desde el navegador del cliente



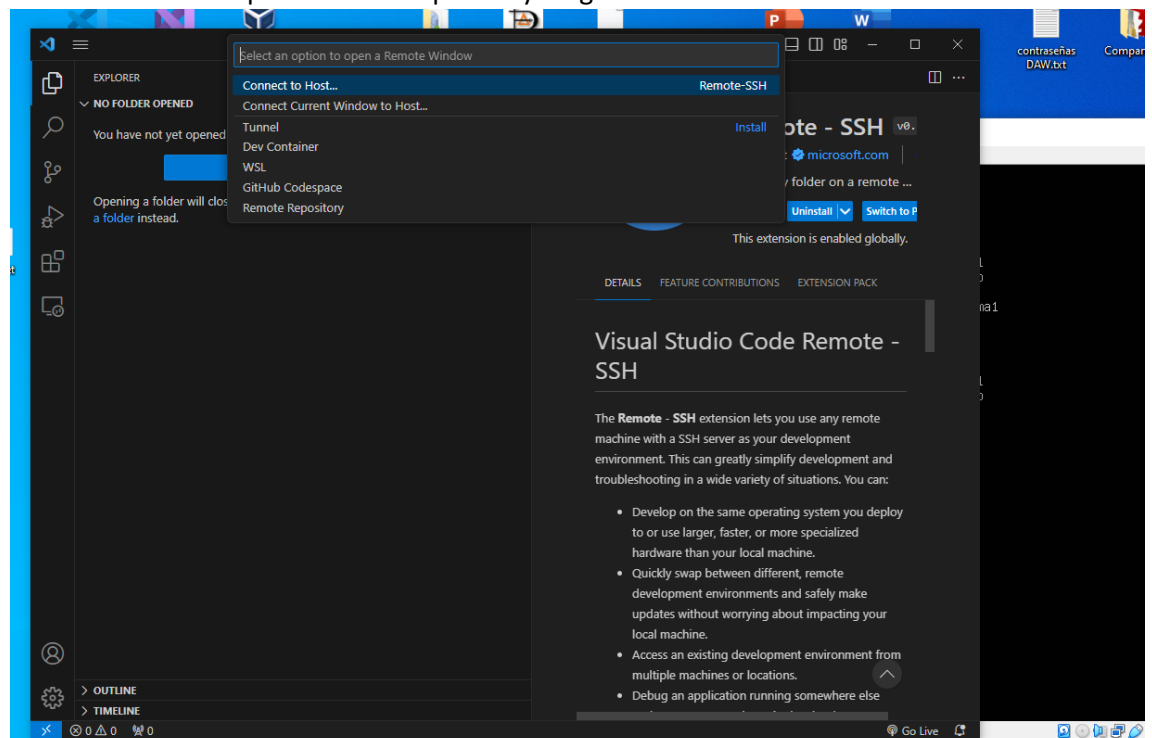
Para editar la carpeta desde visualStudio Code creamos una carpeta llamada Tema1 en nuestro servidor y le otorgamos todos los permisos.

`Sudo mkdir Tema1`

`Sudo chmod -R 777 Tema1`

Instalamos en vs Code del cliente el plugin remote ssh.

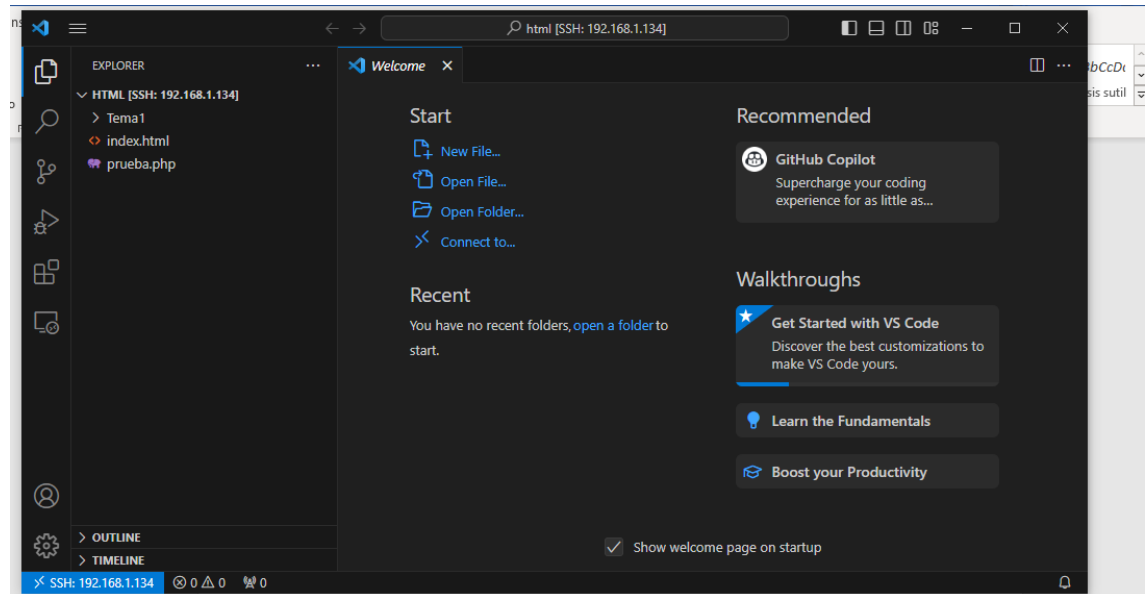
Pinchamos en la esquina inferior izquierda y luego connect to Host..



Add new Host.

ssh raul@192.168.1.134 -p 22 (donde raul es tu usuario de Ubuntu y la ip del servidor.
Pinchamos sobre Linux y tecleamos nuestra contraseña.

Ahora ya tenemos acceso a los archivos del servidor desde VS CODE:



Desde vs Code abrimos la carpeta Tema1 q tiene acceso permitido. Open folder.
Creamos un repositorio en Github y copiamos el enlace.

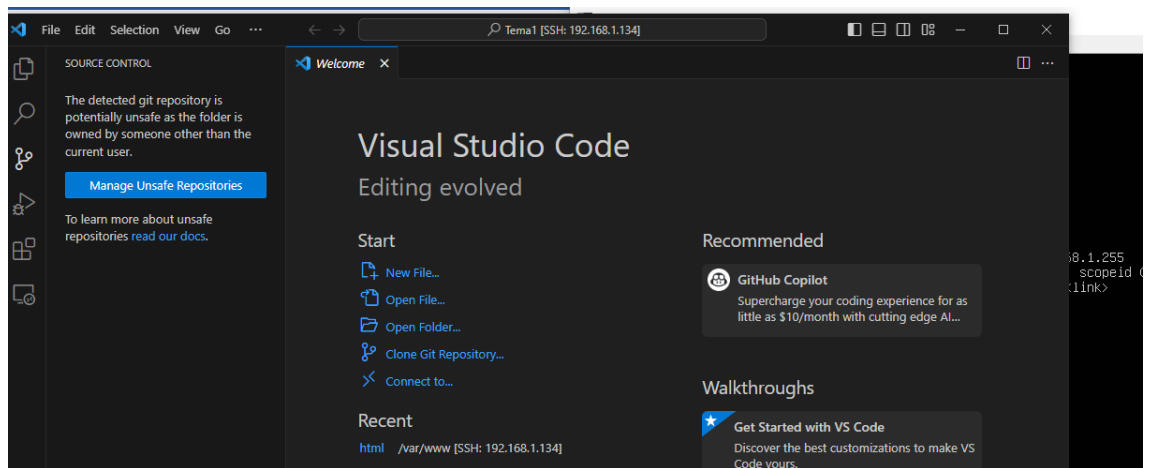
Con la Carpeta creada abierta en VSCode inicializamos el repositorio.

Creamos un archivo html desde github para después clonarlo.
Desde VS Code pinchamos en el icono de usuario Turn on Cloud changes.
Sing with github. Aceptamos.

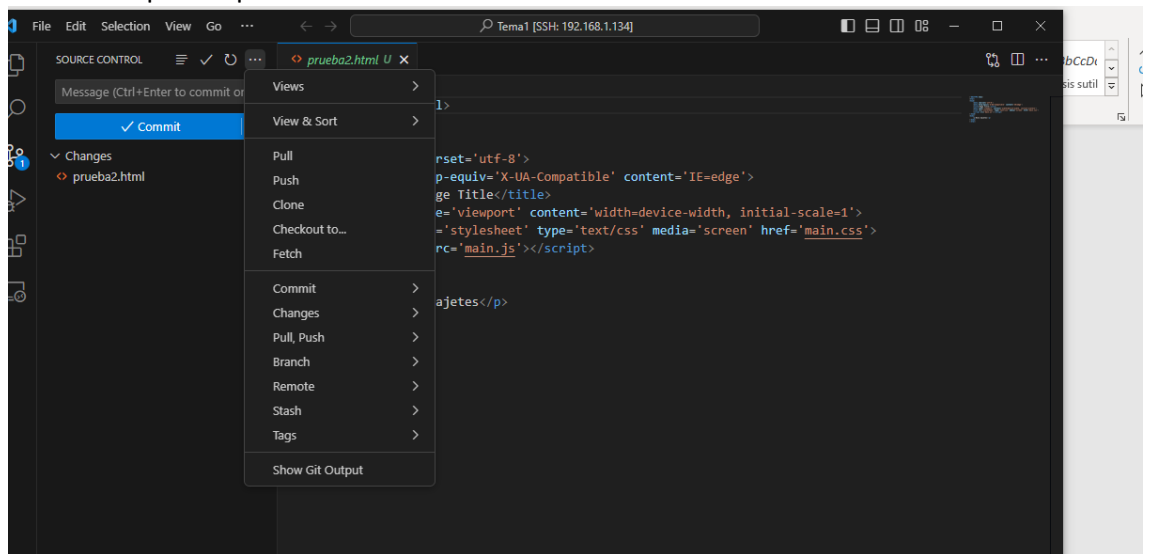
Desde la terminal de vs Code ejecutamos las siguientes instrucciones

```
$ git config --global user.name "usuario"
$ git config --global user.email email
```

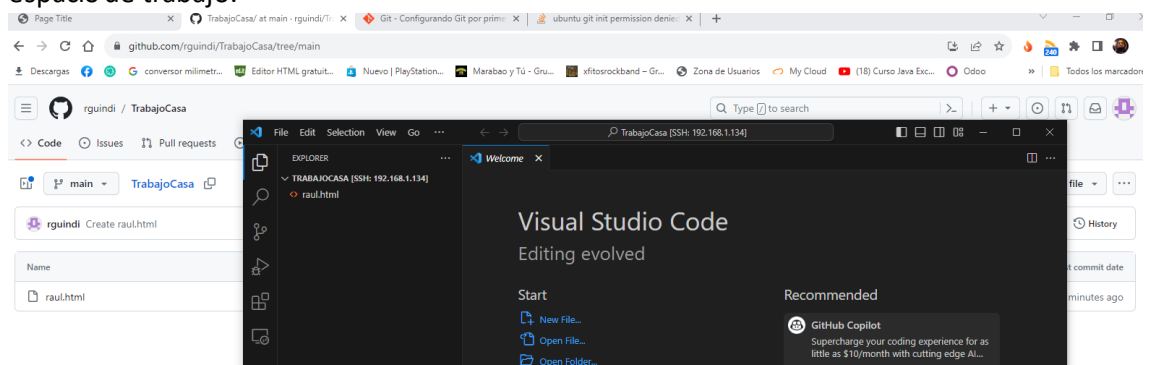
Pinchamos en el icono repositorio y lo inicializamos



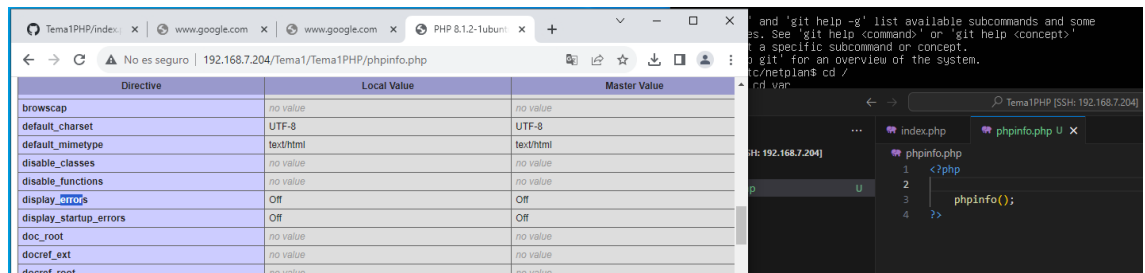
Sobre los 3 puntos pinchamos en clonar:



Seguimos las instrucciones y ya tendríamos nuestro archivo de Github en nuestro espacio de trabajo.



Para ver la configuración de PHP ponemos el código `phpinfo();`



La configuración se cambia en `etc/php/8.1/apache2/php.ini`

Ponemos `Display_errors` poner en “on” para mostrar errores de código mal escrito.

También ponemos en “on” `short_open_tag` para no tener que escribir php al inicio de cada etiqueta.

Luego reiniciamos apache.

`Sudo systemctl restart apache2`

Instalamos el Debugger

`sudo apt-get install php-xdebug`

Luego reiniciamos apache.

`Sudo systemctl restart apache2`

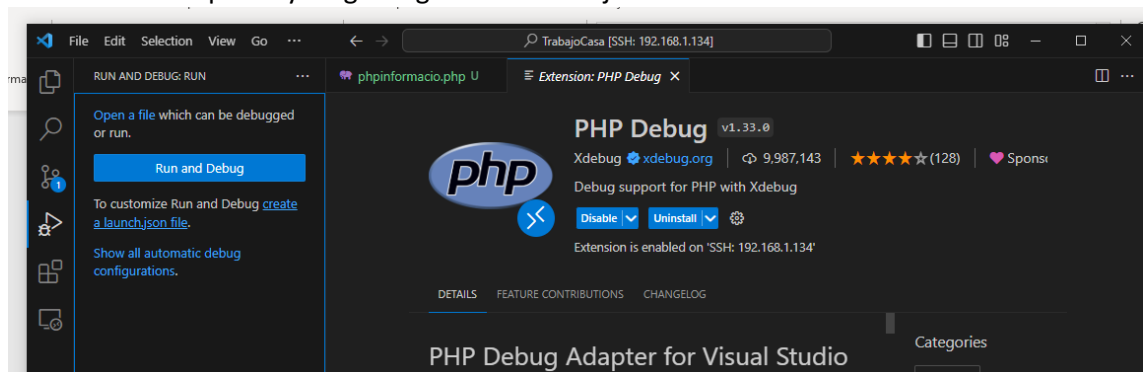
Añadimos las 2 ultimas líneas en `/etc/php/8.1/mods-available/xdebug.ini`



`Sudo systemctl restart apache2`

Instalamos vscode la extensión Php Debug para poder usarlo.

Pinchamos en la polilla y luego en generar `launch.json`.



Ya podemos utilizar el debugger.