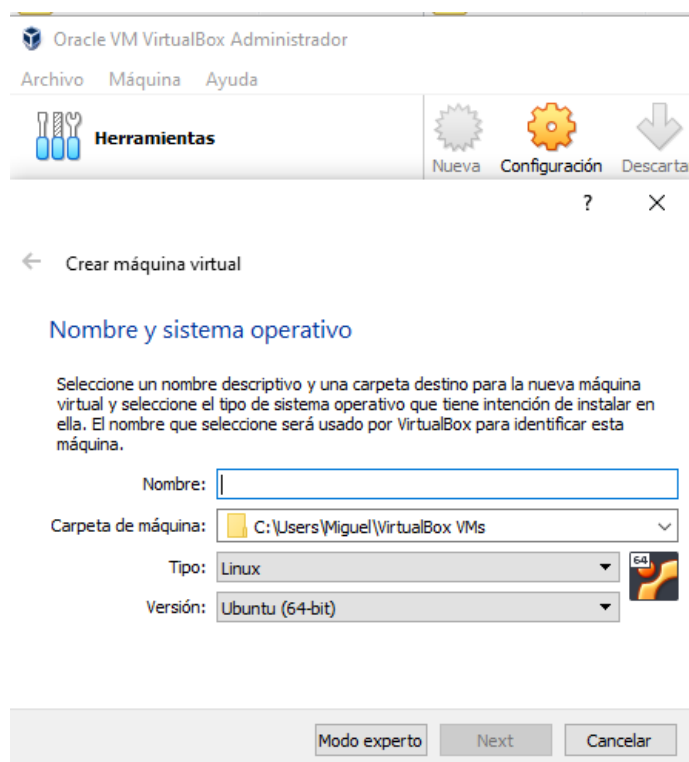


MÁQUINA VIRTUAL

SI AÚN NO TIENES INSTALADO UN SOFTWARE DE VIRTUALIZACIÓN, DESCARGA VIRTUAL BOX EN EL SISTEMA OPERATIVO QUE UTILICES EN TU ORDENADOR O PORTATIL E INSTÁLALO.



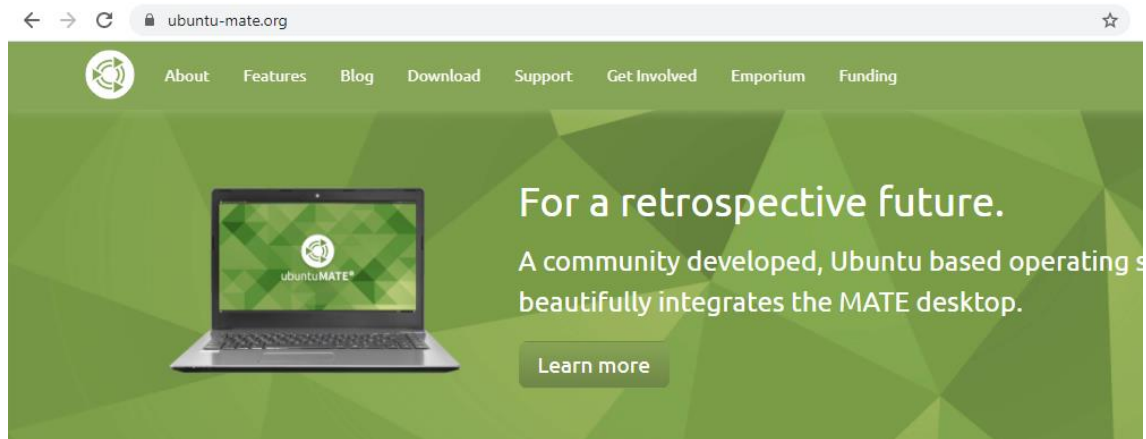
DEBERÁS CREAR UNA NUEVA MÁQUINA VIRTUAL PARA EL UBUNTU QUE DESCARGAREMOS.



DALE UN NOMBRE Y ACEPTA TODOS LOS PASOS SIGUIENTES SIN CAMBIAR NINGUNA OPCIÓN.

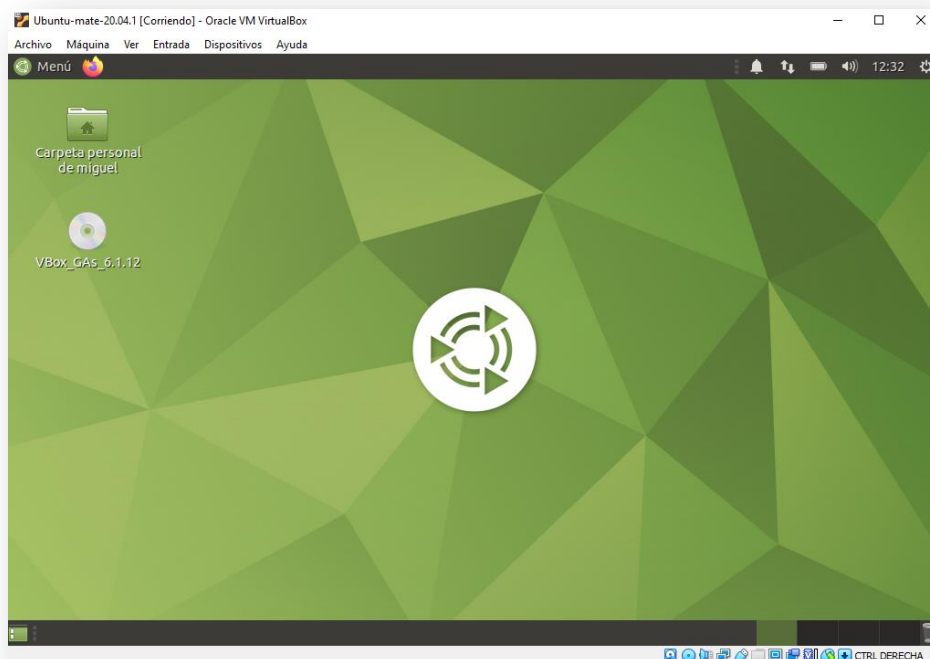
SISTEMA OPERATIVO

DESCARGA UBUNTU MATE 20.04

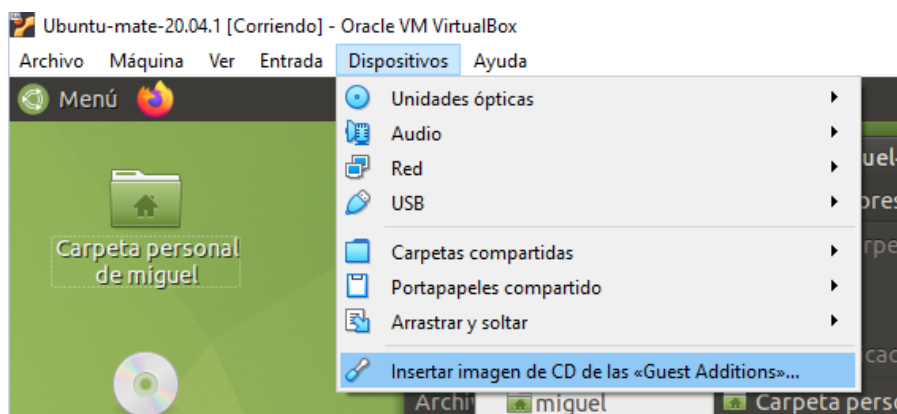


UNA VEZ DESCARGADO, INICIA LA MÁQUINA VIRTUAL CREADA ANTERIORMENTE. NOS PEDIRÁ EL DISCO DE INSTALACIÓN DEL SISTEMA OPERATIVO. LE INDICAREMOS EL ARCHIVO DE INSTALACIÓN DEL UBUNTU-MATE QUE NOS ACABAMOS DE DESCARGAR Y ACEPTAREMOS LAS OPCIONES DE INSTALACIÓN.

EL PROCESO DE INSTALACIÓN DURARÁ VARIOS MINUTOS, TÓMATELO CON CALMA. FINALMENTE NOS APARECERÁ ALGO PARECIDO A ESTO.



PARA PODER REDIMENSIONAR LA VENTANA DE VIRTUAL BOX COMO EN LA IMAGEN DEBES INSTALAR **GUEST ADDITIONS**.



SI TIENES DUDAS, PUEDES SERVIRTE DEL **APARTADO 3** EN EL SIGUIENTE MINI-TUTORIAL:

<https://www.profesionalreview.com/2019/02/06/pantalla-completa-en-virtualbox/>

ES NECESARIO REINICIAR UBUNTU.

AHORA ABRIMOS UN TERMINAL Y ACTUALIZAMOS EL SISTEMA. PRIMERO LANZAMOS UPDATE Y CUANDO TERMINE ESTA INSTRUCCIÓN, LANZAREMOS UPGRADE.

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

¿Qué diferencia hay entre apt update y apt upgrade?

- **apt update:** actualiza la lista de paquetes disponibles y sus versiones, pero no instala o actualiza ningún paquete. Esta lista la coge de los servidores con repositorios que tenemos definidos en el *sources.list*.
- **apt upgrade:** una vez el comando anterior ha descargado la lista de software disponible y la versión en la que se encuentra, podemos actualizar dichos paquetes usando este comando: *apt upgrade*. Instalará las nuevas versiones respetando la configuración del software cuando sea posible.

ENTORNO SERVIDOR

EN NUESTRA MÁQUINA UBUNTU NECESITAMOS CREAR NUESTRO ENTORNO DE SERVIDOR. PARA ELLO USAREMOS, AL MENOS, EL SIGUIENTE SOFTWARE DE CÓDIGO ABIERTO:

- Un servidor web HTTP como **APACHE**.
- Un sistema gestor de bases de datos como **MARIADB/MYSQL** y una herramienta para manejar la administración a través de páginas web como **phpMyAdmin**.
- Un lenguaje de servidor como **PHP**.

PARA FACILITAR EL PROCESO DE INSTALAR TODAS ESTAS HERRAMIENTAS, DESCARGAREMOS E INSTALAREMOS EL PAQUETE LAMP EN NUESTRO UBUNTU.

- **Descargar** Bitnami LAMP desde <https://bitnami.com/stack/lamp/installer>

COMPONENTES

Bitnami LAMP (LINUX, APACHE, MARIADB, PHP) proporciona un entorno de desarrollo completo, totalmente integrado y listo para ejecutar. Además de PHP, MySQL y Apache, incluye phpMyAdmin, openssl, ModSecurity, PageSpeed, Varnish, SQLite, ImageMagick, xDebug, Xcache, OpenLDAP, Memcache, OAuth, PEAR, PECL, APC, GD, cURL y Composer.

INSTALACIÓN

Se descargará un archivo como este:



En Linux, deberás otorgarle permisos de ejecución:

```
chmod 755 bitnami-lampstack-7.4.9-0-linux-x64-installer.run
```

- **chmod 755:** *El propietario del fichero puede leer, escribir y ejecutar el archivo. Todos los otros pueden leer y ejecutar el archivo. Este ajuste es común para los programas que son utilizados por todos los usuarios.*

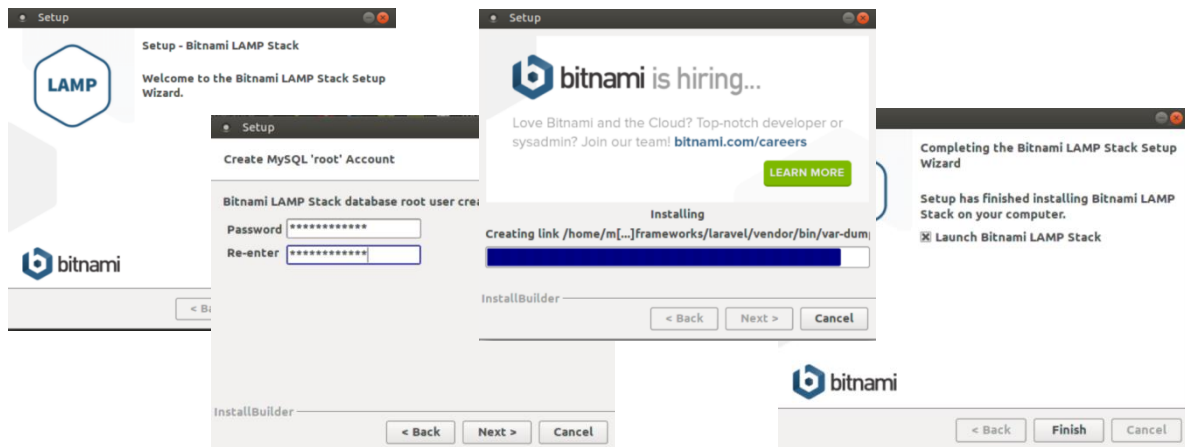
y seguidamente...

```
./bitnami-lampstack-7.4.9-0-linux-x64-installer.run
```

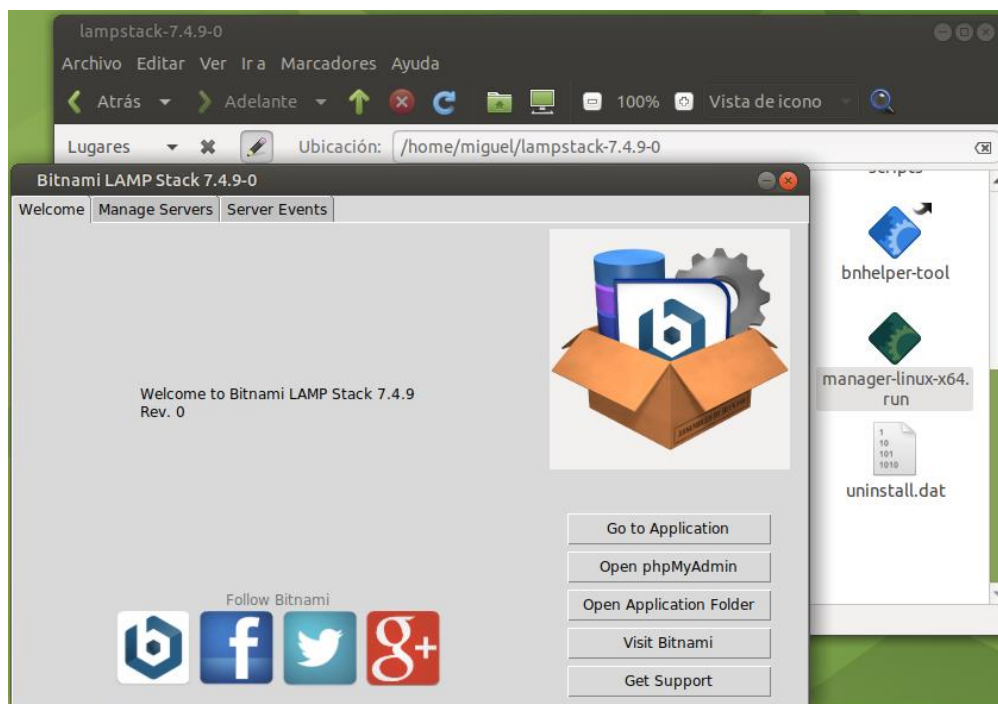
```
miguel@miguel-VirtualBox: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
miguel@miguel-VirtualBox:/$ cd /home/miguel/Descargas
miguel@miguel-VirtualBox:~/Descargas$ chmod 755 bitnami-lampstack-7.3.21-0-linux-x64-installer.run
miguel@miguel-VirtualBox:~/Descargas$ ./bitnami-lampstack-7.3.21-0-linux-x64-installer.run
```

A CONTINUACIÓN, ACEPTA TODOS LOS PASOS DE INSTALACIÓN.

Puede darte algún error de cierta librería de MySQL al principio y al final de la instalación. De ser así, no te preocupes por ello. No nos va a influir.



UNA VEZ INSTALADO PODRÁS ABRIR EL GESTOR (MANAGER) DESDE EL EJECUTABLE CREADO.

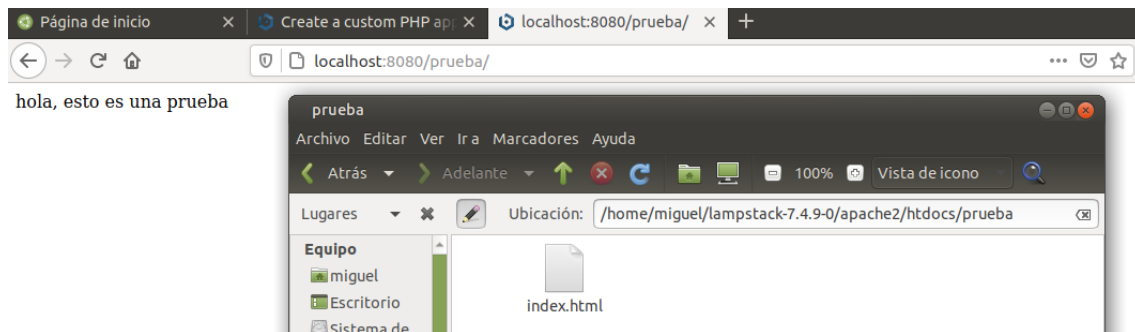


PROBANDO SU FUNCIONAMIENTO

DESDE EL SERVIDOR (UBUNTU)

El directorio principal para desplegar cada aplicación PHP es `.../apache2/htdocs/` (VER IMAGEN)

Puedes copiar los archivos de tu aplicación PHP en carpetas (una por aplicación) dentro de ese directorio y acceder a la aplicación a través del navegador.



DESDE EL CLIENTE (WINDOWS)

Como primer objetivo, vamos a compartir la carpeta `...apache2/htdocs` desde la máquina Linux para poder editar el código fuente desde Windows. Podemos conseguirlo mediante **SAMBA**.

Lo vamos a poder hacer en entorno gráfico, pero debemos tener instalado *Guest Additions*. Abriremos la aplicación “Bienvenido” desde el menú de inicio.



Elegiremos la opción “Primeros Pasos” y después “Tareas Opcionales”

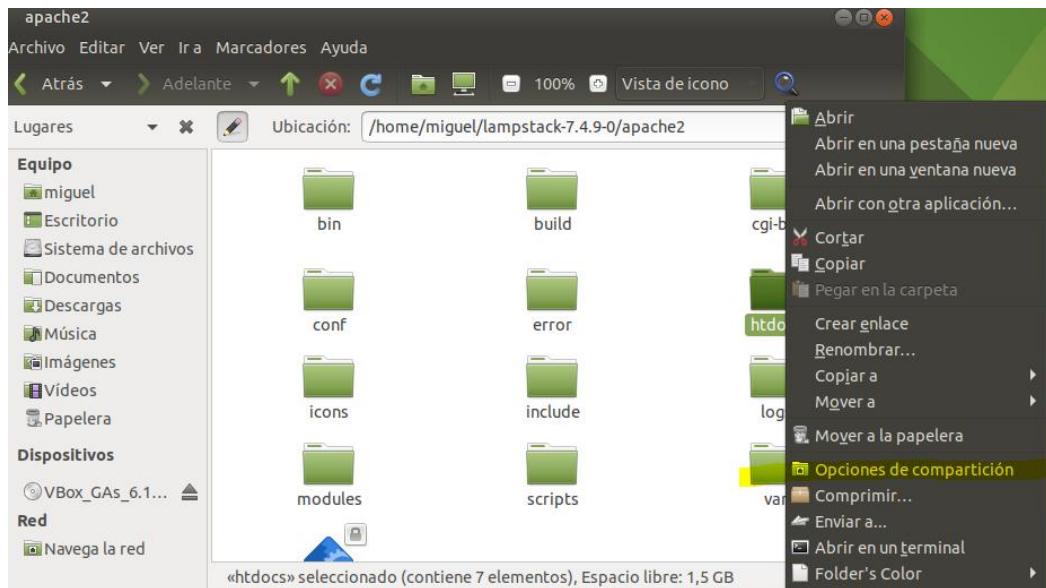


Por último, seleccionaremos la instalación de “Carpetas compartidas de red”.



ES NECESARIO REINICIAR UBUNTU OTRA VEZ.

Ahora ya nos aparece la opción de compartición mediante clic derecho de ratón.



A continuación, marca las opciones que muestra la imagen siguiente y acepta.

Si no te funciona en Windows, o no te pide usuario y contraseña, prueba a seleccionar **Acceso de invitado** para omitir la solicitud de credenciales.

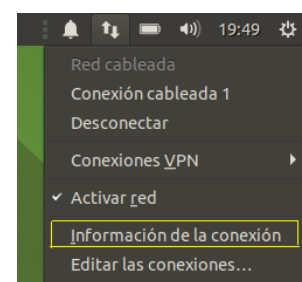


En este momento, el icono cambiará y aparecerán dos flechas junto a él, lo que significa que la carpeta está compartida y debería ser visible desde fuera del terminal.



Ahora tenemos que averiguar cuál es la IP del adaptador de red de la máquina Ubuntu.

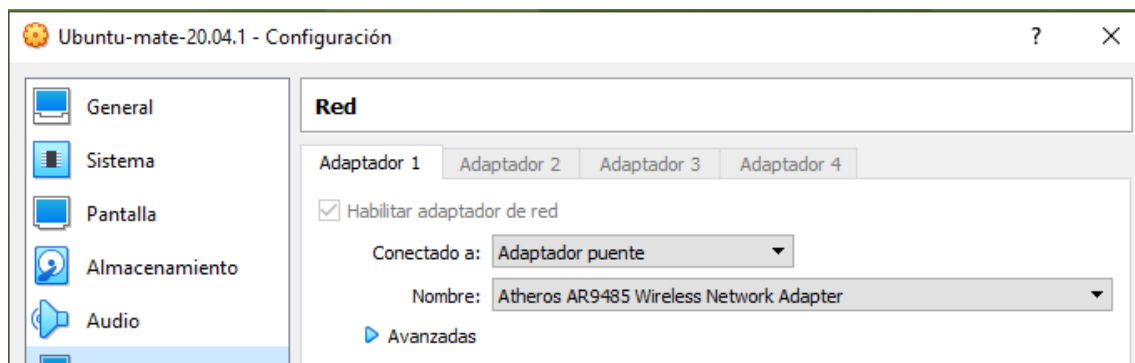
Para ello, accede al icono de conexión de red en la barra superior de Ubuntu. Selecciona "Información de la conexión" para ver la IP que usa el terminal.



Asegúrate de que la red esté activa (Activar red marcado) y no esté inactiva por error.

Si no te aparece una ip de la misma red que la ip de tu sistema Windows, tendremos que cambiar el adaptador de red. Para ello entra en el menú de VirtualBox:

Dispositivos → Red → Preferencias de Red



Vuelve ahora a comprobar tu ip en Ubuntu (tarda unos segundos...).

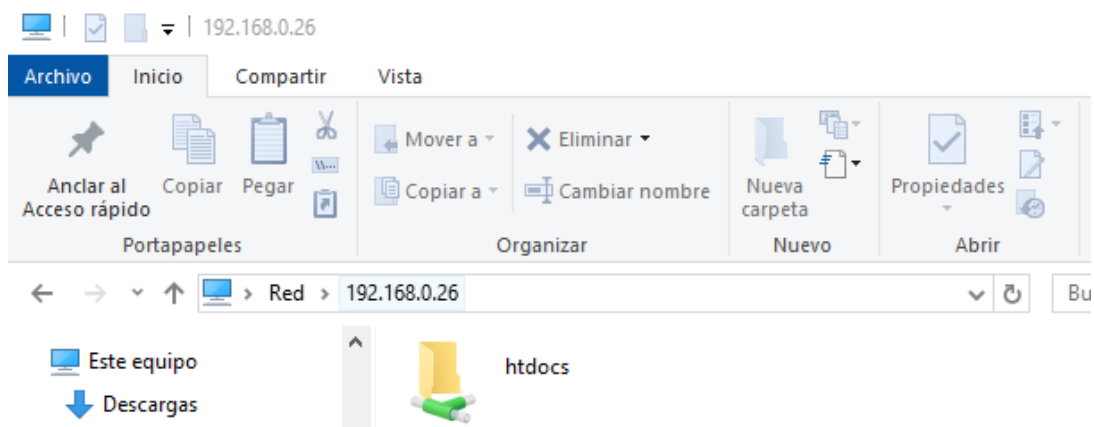
Antes de comprobar que la carpeta se vea desde Windows, vamos a permitir SAMBA (la aplicación que nos permite compartir archivos) en el Firewall de Linux:

```
sudo ufw allow samba
```

Luego, reinicia los servicios de SAMBA para que los cambios tengan efecto.

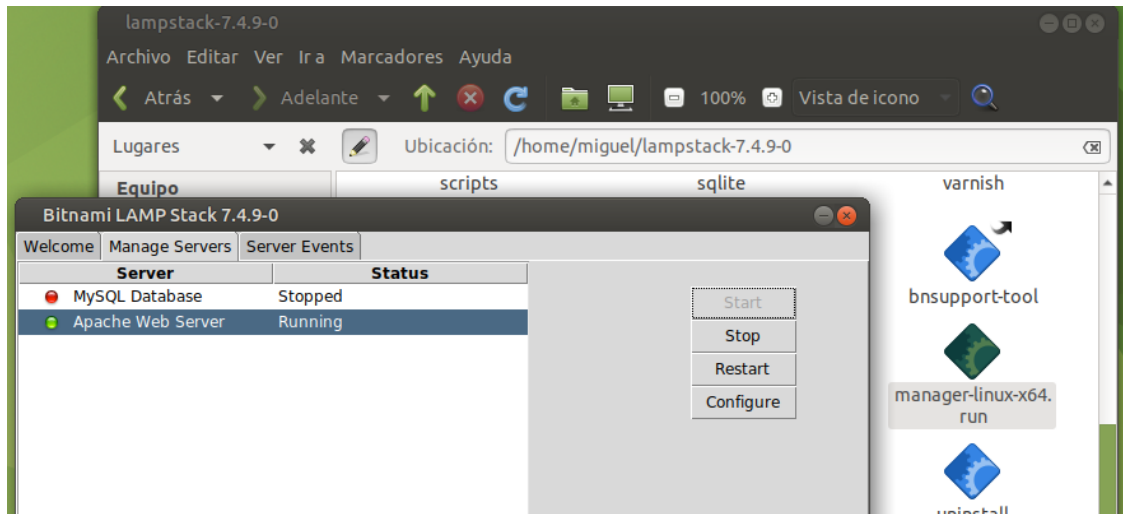
```
sudo /etc/init.d/smbd restart
```

Ahora, desde Windows ya podemos acceder a la carpeta compartida.

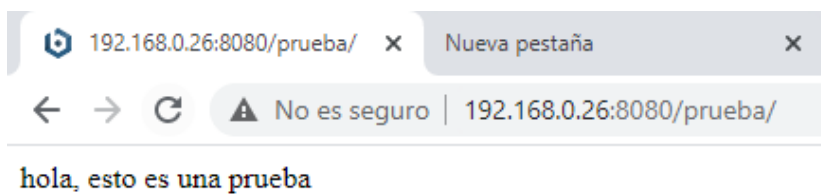


Comprobaremos ahora si tenemos acceso web al servidor.

Abre el manager de LAMP Stack y arranca el servidor APACHE.



Por último, comprobaremos que las páginas se sirven correctamente también desde Windows.



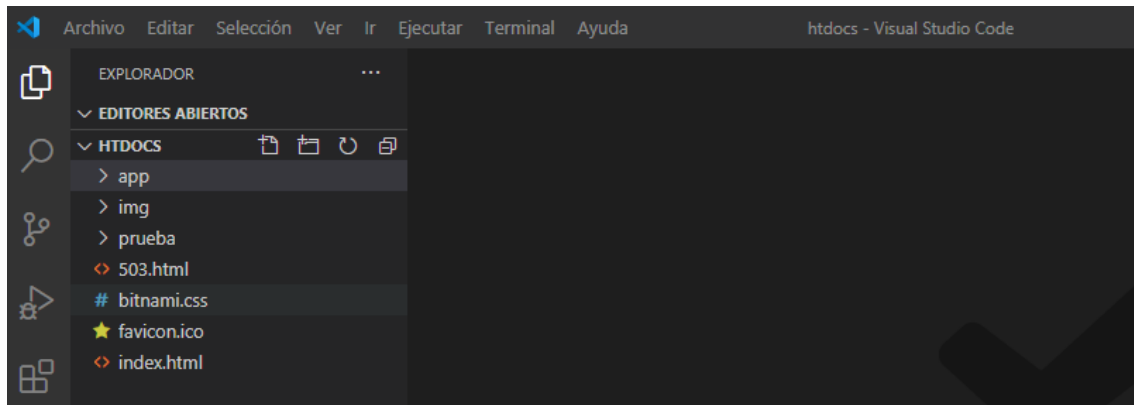
Entorno de desarrollo integrado

Usaremos Visual Studio Code (VSCode) desde Windows.

Descárgalo e instálalo: <https://code.visualstudio.com/>

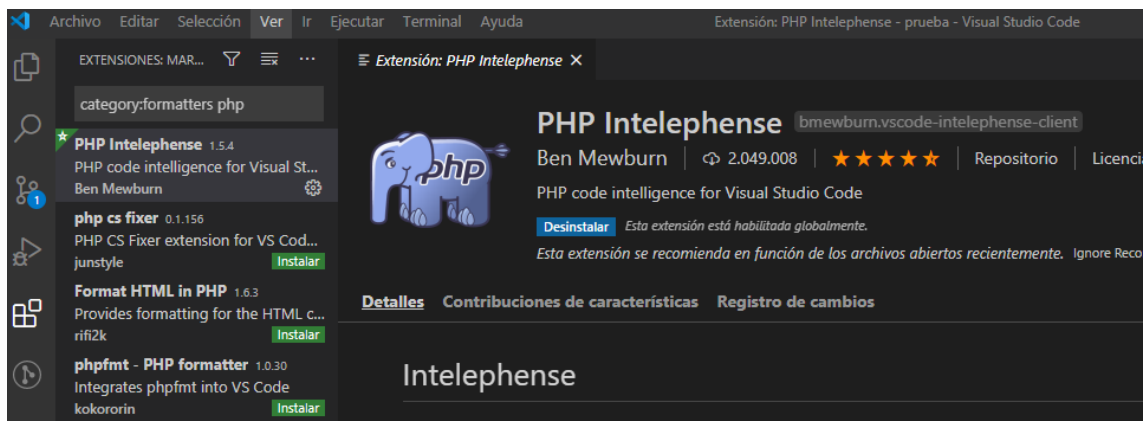


En VS Code puedes ahora abrir la carpeta remota **htdocs**.



INSTALA LAS EXTENSIONES:

- PHPIntelliPhense
- HTML CSS Suport
- PHP Debug (*opcional*)



Gestión de las prácticas con GIT

Vamos a usar el sistema de control de versiones GIT para nuestras prácticas, pero de forma integrada en Visual Studio Code. No usaremos línea de comandos.

A continuación se incluyen algunas órdenes de git a modo de repaso:

Instalar GIT en linux:

```
sudo apt install git
```

Configurar GIT:

```
git config --global user.name "Nombre de Usuario"
git config --global user.email correo@usuario.abc
```

Inicializar un repositorio local en GIT

- Acceder a la carpeta que queremos que sea un repositorio GIT, por ejemplo:

```
cd /var/www/html/ejemplos
```

- Inicializar: `git init`

Esto habrá creado una carpeta oculta con el nombre `.git`, donde el sistema de gestión de versiones almacena los cambios.

Comprobar estado de un repositorio GIT: `git status`

Hacer que ciertos archivos y/o directorios no sean seguidos por GIT

- Creamos un archivo de texto llamado `.gitignore`
- En cada línea de este archivo indicamos un archivo, directorio o patrón

Poner en seguimiento un archivo o directorio: `git add nombre_archivo`

Poner en seguimiento toda una carpeta y sus subcarpetas: `git add .`

- Git muestra, en este caso, en color verde los archivos y directorios en seguimiento.

Dejar de seguir ciertos archivos: `git rm nombre_archivo --cached`

- Con **`git status`** podremos comprobar que el archivo ha dejado de estar en seguimiento.

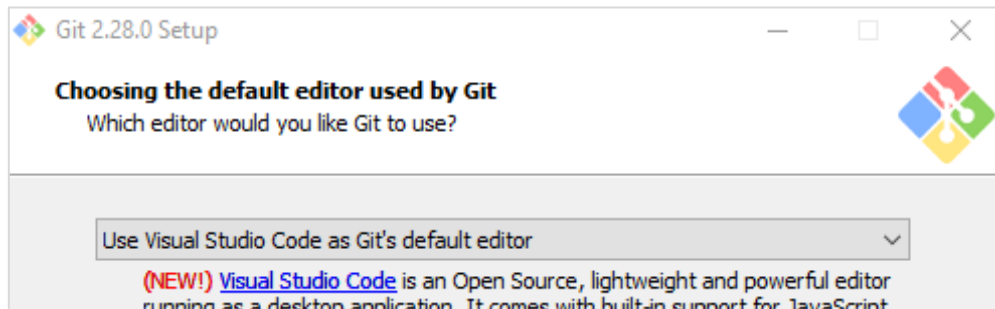
Confirmar cambios y crear instantánea: `git commit -m "Mensaje de confirmación"`

Visualizar el historial de cambios de un repositorio GIT: `git log`

Una vez recordados algunos comandos de git, estamos ya en disposición de inicializar un repositorio de git para **htdocs** desde el entorno gráfico integrado en Visual Studio Code.

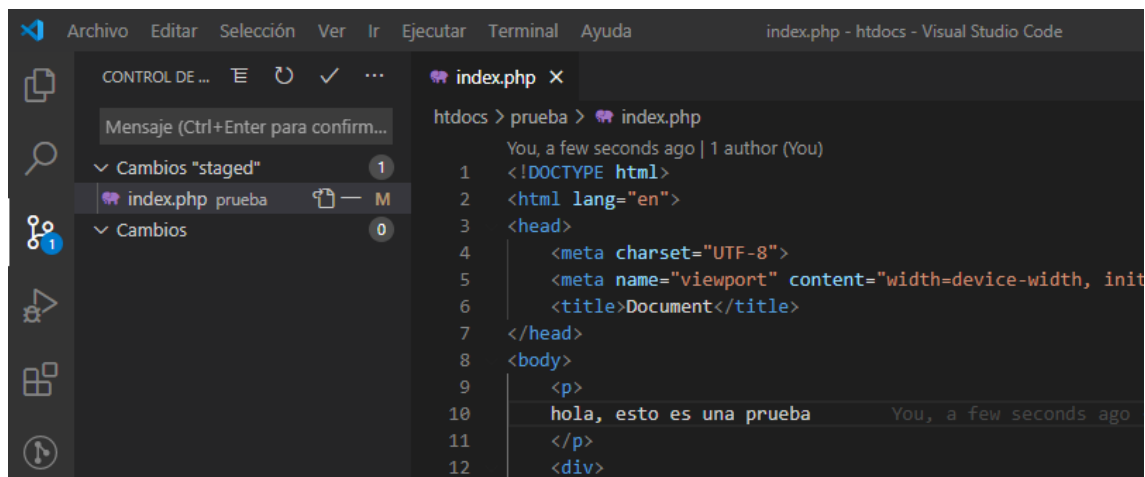
Pero primero descarga e instala git para Windows: <https://git-scm.com/>

En la instalación de git, es muy importante que cambies una opción por defecto, la que se muestra en la imagen. El resto las puedes aceptar tal como están.



Reinicia VS Code.

Ahora ya podemos usar git en Visual Studio Code desde la tercera pestaña.



VIDEOS MUY RECOMENDADOS PARA APRENDER A USAR GIT EN VS CODE (SON CORTITOS):

- [Como Trabajar con Git, Github en Visual Studio Code](#)
- [¿Cómo trabajar con Git desde Visual Studio Code?](#)