

# Práctica 01

*Crear infraestructura de programación*



# Índice

1- Objetivos.....	3
2- Instalación de VirtualBox.....	3
3- Creación de la máquina virtual en VirtualBox.....	4
4- Instalación de Ubuntu Mate.....	7
4.1- Resolución a pantalla completa con Guest Additions.....	9
5- Creación del entorno servidor.....	11
5.1- Instalación de Bitnami LAMP.....	12
5.1.2- Prueba de Bitnami LAMP (Servidor).....	12
5.1.3- Prueba de Bitnami LAMP (Windows).....	13
6- Entorno de desarrollo integrado (Ide).....	20
7-Instalación de Git en ambos sistemas.....	21
7.1- Instalación en Ubuntu Mate.....	21
7.2- Instalación en Windows.....	22
8-Debug en Visual Studio Code.....	22
8.1-Instalación de Xdebug en el servidor.....	22

## 1- Objetivos

Creación de un entorno de trabajo instalando una máquina virtual con el Sistema operativo (SO) Ubuntu Mate y poder acceder a los archivos del mismo desde nuestro SO local a través de una carpeta compartida y del Servidor Web. Además instalaremos el software Visual Studio Code como editor de código.

## 2- Instalación de VirtualBox

VirtualBox es un software que se utiliza para la creación de máquinas virtuales.

Para instalarlo nos dirigimos a <https://www.virtualbox.org/wiki/Downloads> , seleccionamos nuestro SO y se nos descargara el instalador de la aplicación.

### VirtualBox 6.1.14 platform packages

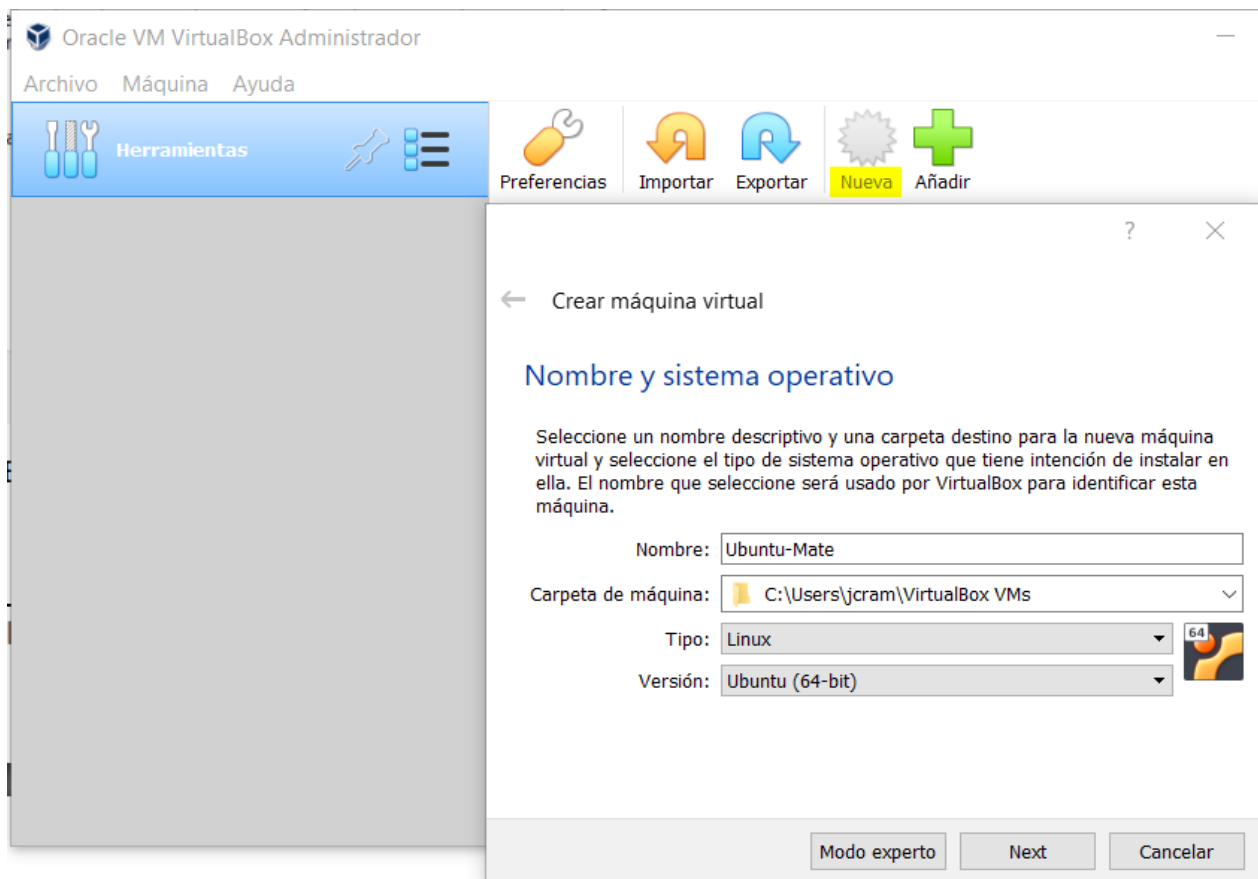
- ➞ [Windows hosts](#)
- ➞ [OS X hosts](#)
- [Linux distributions](#)
- ➞ [Solaris hosts](#)

Una vez descargado, damos doble-click para que inicie la instalación y pulsamos siguiente en todas las opciones. Si todo ha funcionado correctamente, aparecerá una pantalla como esta.



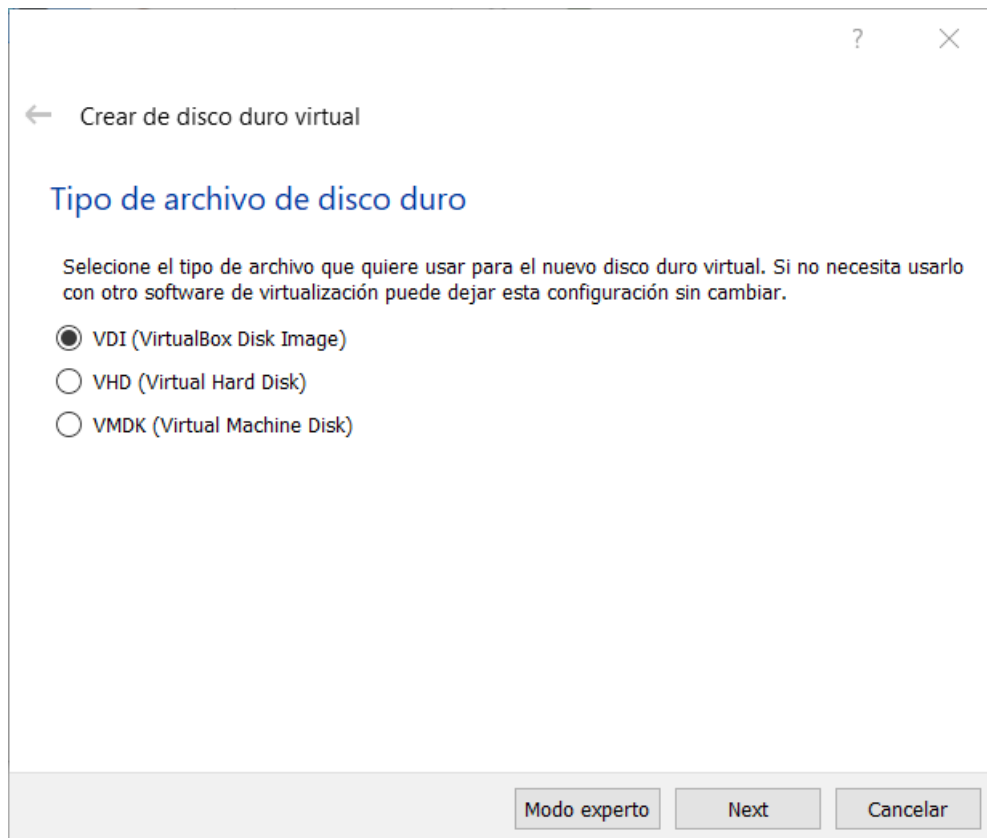
### 3- Creación de la máquina virtual en VirtualBox

Para crear una maquina virtual pulsamos la opción “Nueva”, se abrirá una ventana en la que tenemos que indicar el nombre “Ubuntu-Mate”, seleccionamos el tipo “Linux”, la versión “Ubuntu (64-bit)” y pulsamos “Next”.



Seguidamente se nos pide que indiquemos el tamaño de la memoria y el tamaño del disco duro, en estas opciones pulsamos “Next” para quedarnos con las opciones por defecto que son las que nos recomienda VirtualBox.

A continuación seleccionamos el tipo de archivo de disco duro, elegimos la opción VDI (VirtualBox Disk Image) y pulsamos “Next”.



Elegimos la opción “Reservado dinámicamente” para que el disco duro de la

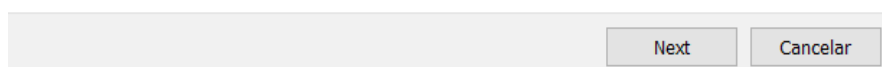
### Almacenamiento en unidad de disco duro física

Seleccione si el nuevo archivo de unidad de disco duro virtual debería crecer según se use (reserva dinámica) o si debería ser creado con su tamaño máximo (tamaño fijo).

Un archivo de disco duro **reservado dinámicamente** solo usará espacio en su disco físico a medida que se llena (hasta un máximo **tamaño fijo**), sin embargo no se reducirá de nuevo automáticamente cuando el espacio en él se libere.

Un archivo de disco duro de **tamaño fijo** puede tomar más tiempo para su creación en algunos sistemas, pero normalmente es más rápido al usarlo.

- ☒ Reservado dinámicamente  
☐ Tamaño fijo



máquina virtual solo ocupe el espacio que estemos utilizando y pulsamos “Next”.

Para finalizar, se nos pide que seleccionemos la ubicación del archivo y el tamaño del disco duro virtual, dejamos las opciones por defecto y pulsamos “Crear”.

### Ubicación del archivo y tamaño

Escriba el nombre del archivo de unidad de disco duro virtual en el campo debajo o haga clic en el icono de carpeta para seleccionar una carpeta diferente donde crear el archivo.

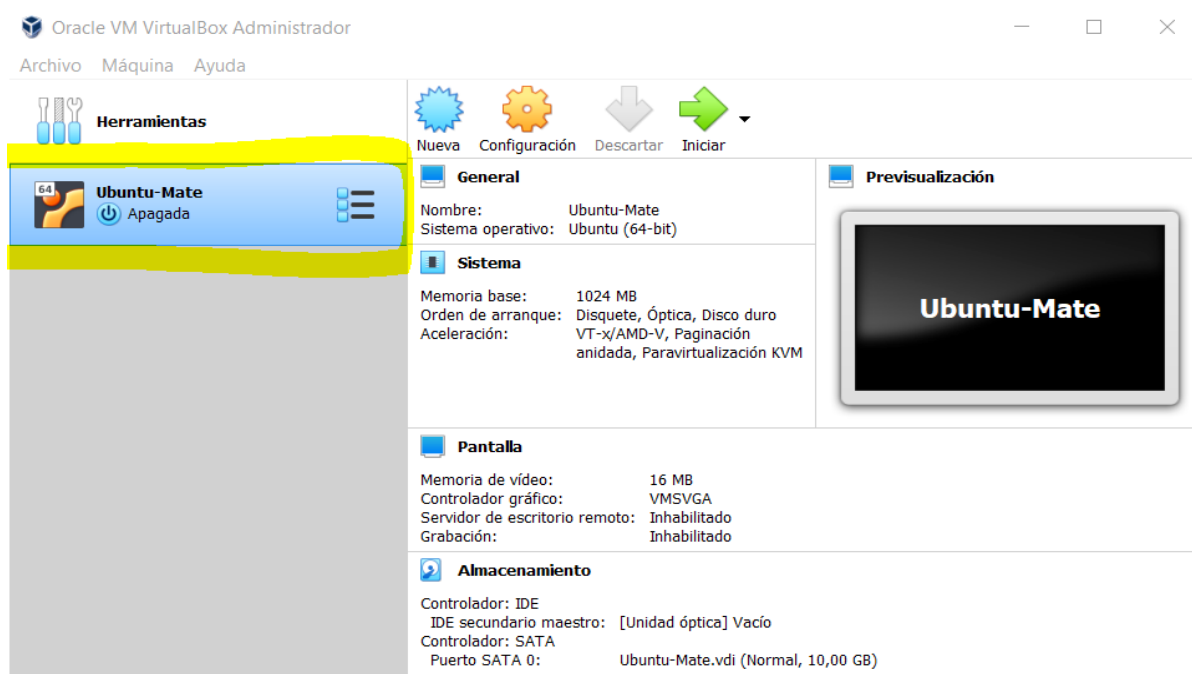
C:\Users\jcram\VirtualBox VMs\Ubuntu-Mate\Ubuntu-Mate.vdi

Seleccione el tamaño de disco duro virtual en megabytes. Este tamaño es el límite para el archivo de datos que una máquina virtual podrá almacenar en el disco duro.

4,00 MB 2,00 TB 10,00 GB

Crear Cancelar

Si todo ha funcionado correctamente podremos ver nuestra nueva máquina virtual en la interfaz de VirtualBox.

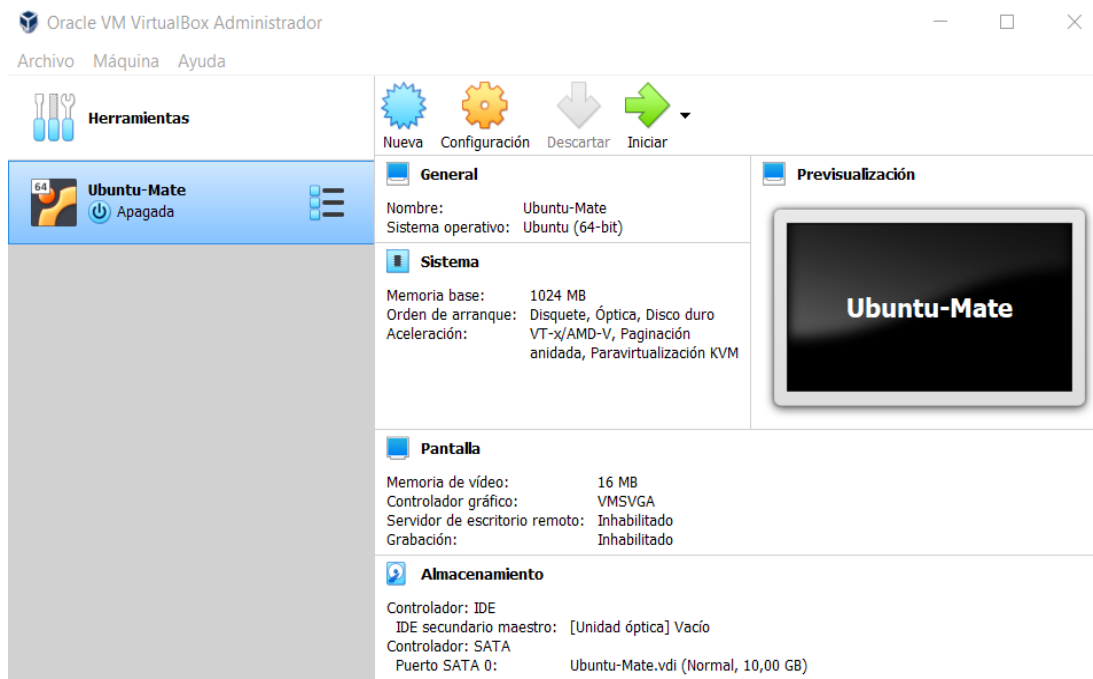


## 4- Instalación de Ubuntu Mate

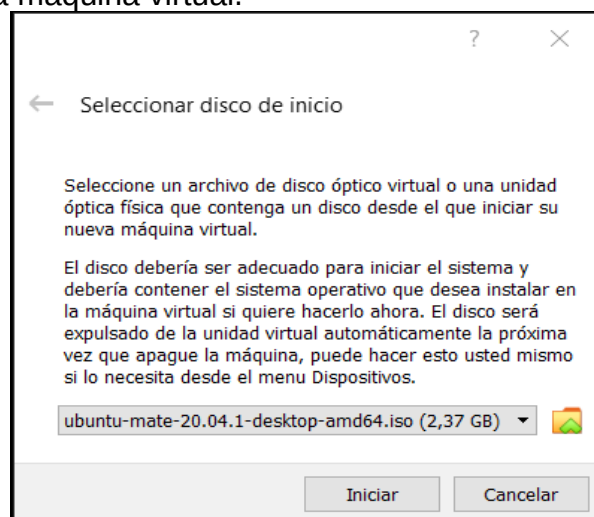
El primer paso es descargar la imagen ISO de Ubuntu Mate, lo podemos hacer desde

<https://ubuntu-mate.org/download/amd64/focal/> y seleccionamos la opción de descarga que más nos guste.

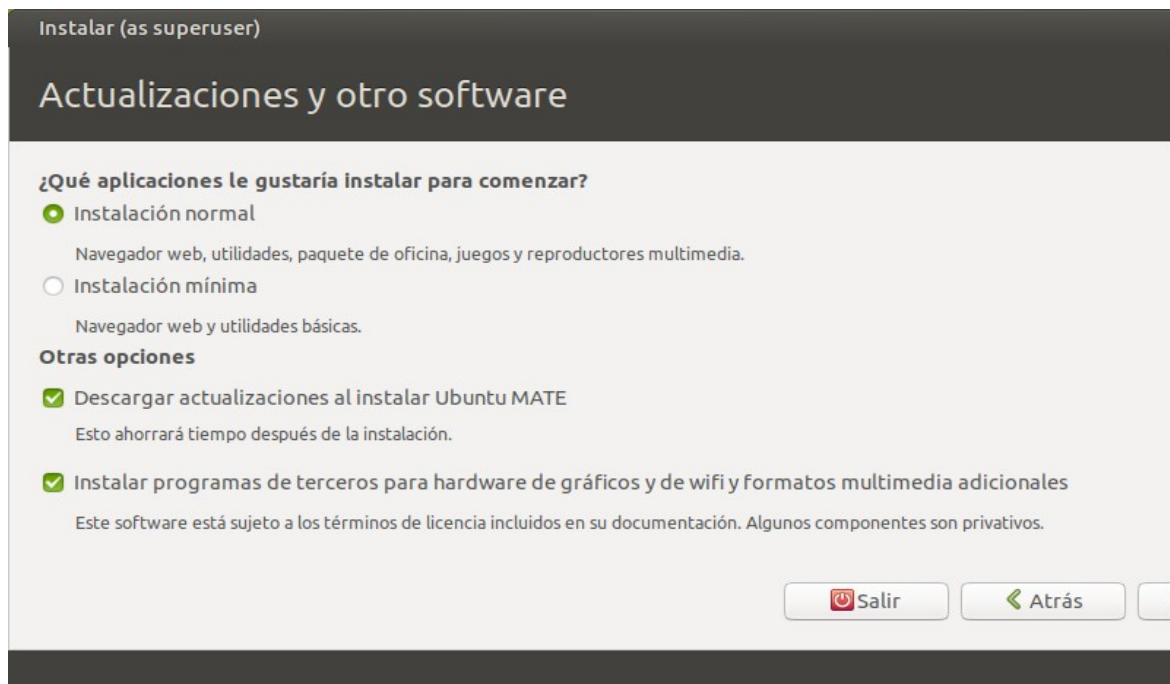
Una vez descargado, iniciamos VirtualBox, seleccionamos la máquina virtual que hemos creado anteriormente y pulsamos en “Iniciar”.



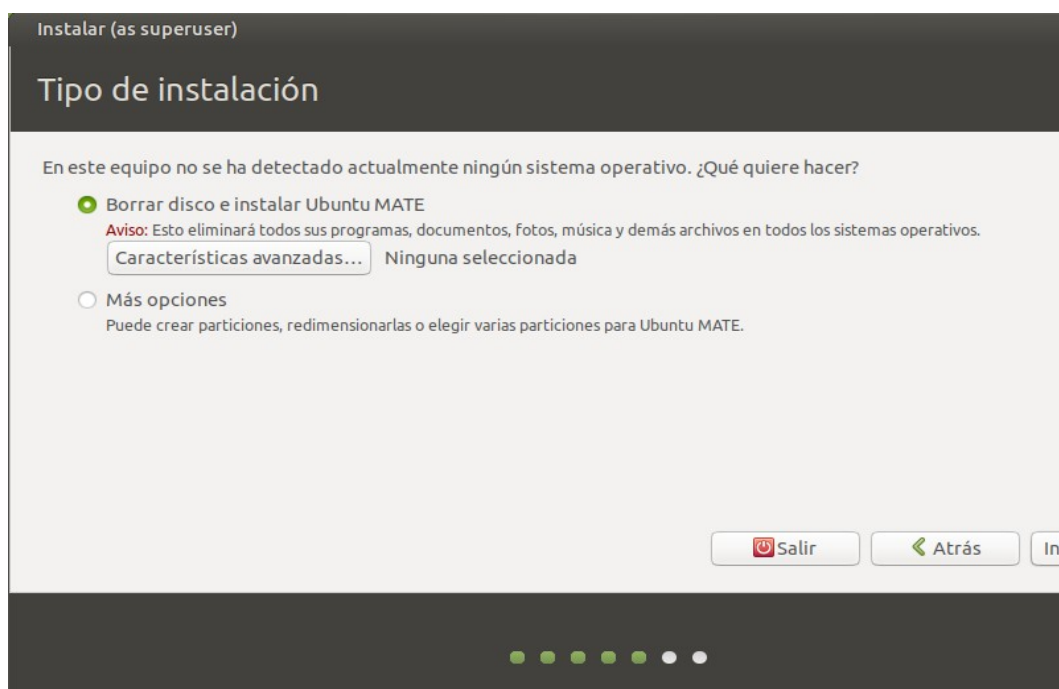
A continuación se nos abrirá un cuadro de dialogo en el que tenemos que localizar la imagen ISO que hemos descargado y pulsamos “Iniciar”, esto iniciara la instalación de Ubuntu Mate en nuestra máquina virtual.



En el proceso de instalación lo primero que se nos pide es que seleccionemos el idioma, lo seleccionamos y pulsamos en “Instalar Ubuntu MATE”, seguidamente seleccionamos la distribución del teclado y pulsamos “Siguiente”, seleccionamos el tipo de instalación y pulsamos “Siguiente”.



En la siguiente pantalla, seleccionamos el tipo de instalación , pulsamos en “Instalar” y damos a “Continuar”.

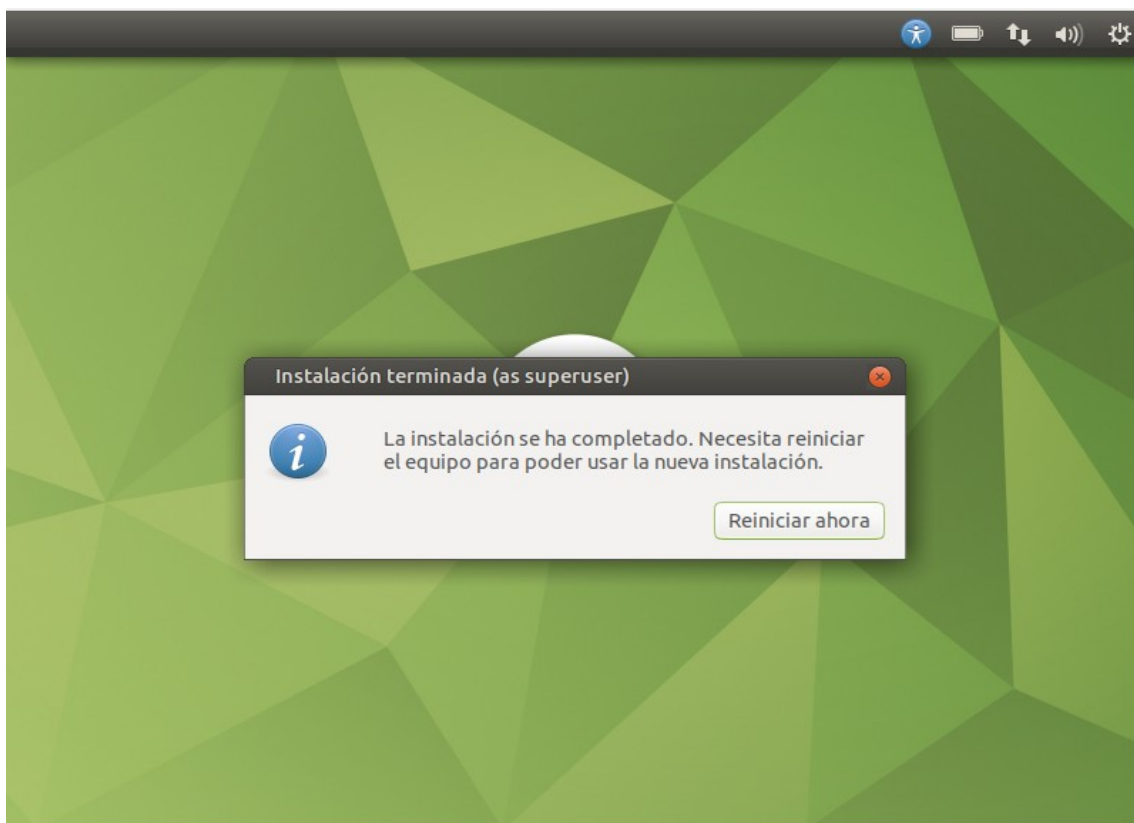




Nos pedirá que seleccionemos la zona horaria, indicamos “Madrid” y pulsamos en siguiente.

Rellenamos los campos del formulario para indicar nuestro nombre y contraseña y pulsamos en “Siguiente”.

Si todo ha funcionado correctamente deberíamos ver una pantalla como esta.

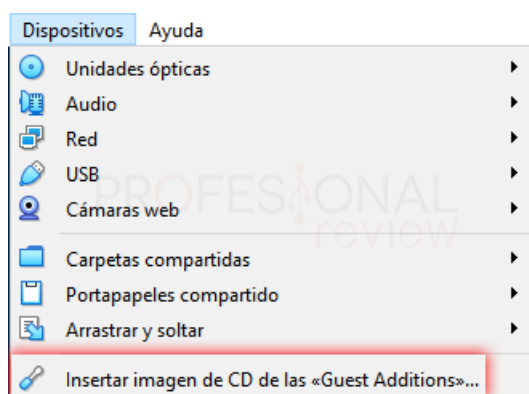


Pulsamos en “Reiniciar ahora” para completar el proceso de instalación.

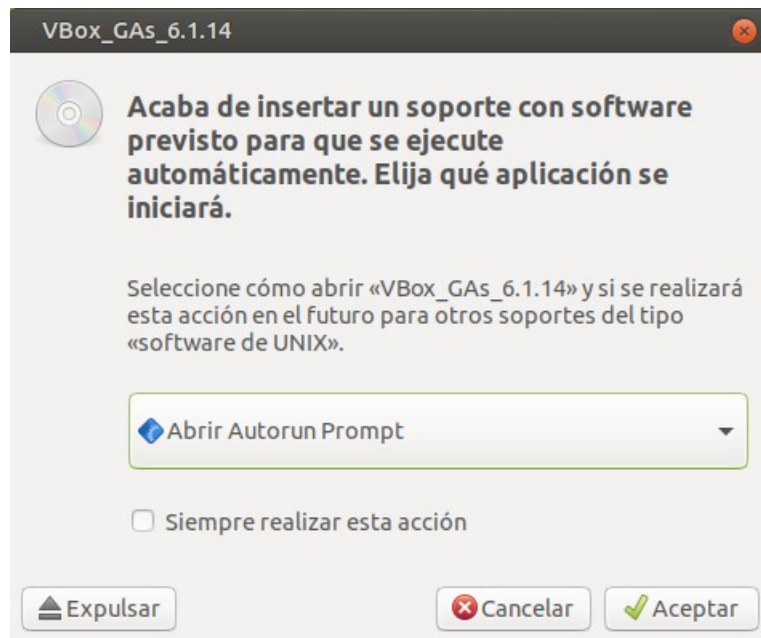
#### **4.1- Resolución a pantalla completa con Guest Additions**

Para poder redimensionar a pantalla completa debemos instalar las VirtualBox Guest Additions.

En primer lugar en el menú de herramientas de la máquina virtual, pulsamos sobre “Dispositivos” y elegimos la opción “Insertar imagen de CD de las <<Guest Additions>>”.



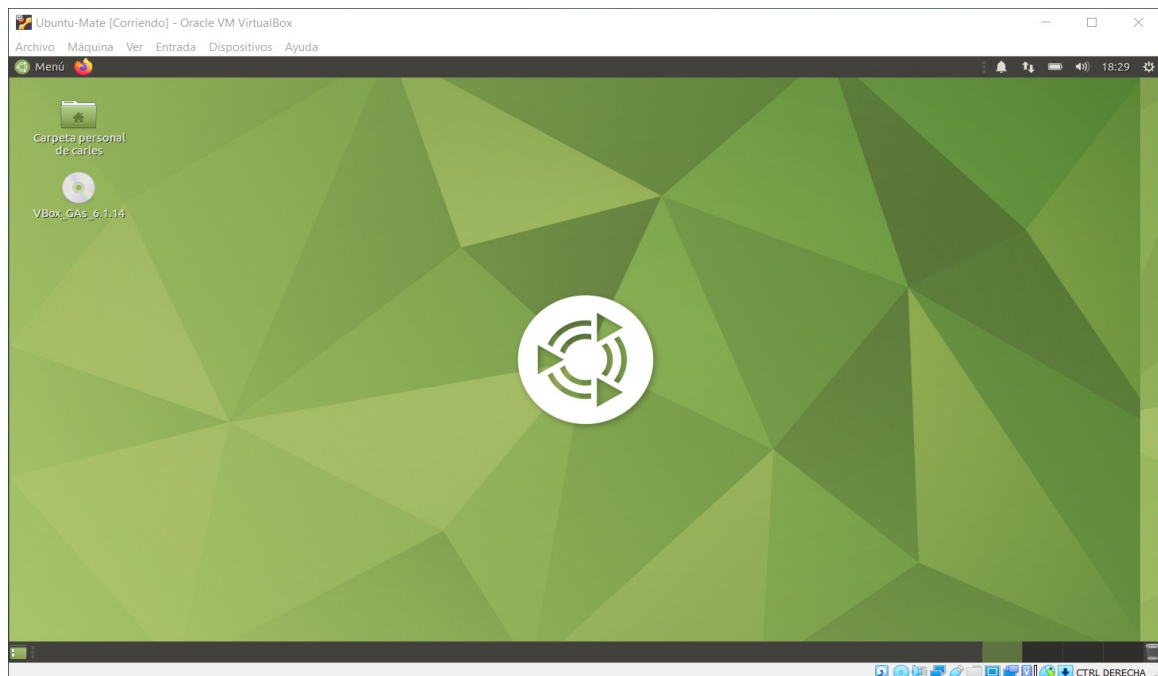
Nos aparece una pantalla para elegir la aplicación con la que iniciaremos la instalación en este caso he seleccionado “Abrir Autorun Prompt”.



Pulsamos “Aceptar” luego “Ejecutar” y se iniciará el proceso de instalación. Cuando termine, nos pedirá que reiniciemos Ubuntu para completar la instalación.

```
VirtualBox Guest Additions installation
Archivo Editar Ver Buscar Terminal Ayuda
Verifying archive integrity... All good.
Uncompressing VirtualBox 6.1.14 Guest Additions for Linux.....
VirtualBox Guest Additions installer
Removing installed version 6.1.14 of VirtualBox Guest Additions...
update-initramfs: Generating /boot/initrd.img-5.4.0-47-generic
Copying additional installer modules ...
Installing additional modules ...
VirtualBox Guest Additions: Starting.
VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel
modules. This may take a while.
VirtualBox Guest Additions: To build modules for other installed kernels, run
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup <version>
VirtualBox Guest Additions: or
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup all
VirtualBox Guest Additions: Building the modules for kernel 5.4.0-47-generic.
update-initramfs: Generating /boot/initrd.img-5.4.0-47-generic
VirtualBox Guest Additions: Running kernel modules will not be replaced until
the system is restarted
Press Return to close this window...
```

Para comprobar si todo ha ido correctamente, ensanchamos la pantalla y deberíamos ver como se adapta.



## 5- Creación del entorno servidor

Previamente actualizaremos el SO, para ello iniciamos una terminal con la combinación de teclas “Ctrl + Alt + T” e introducimos los siguientes comandos:

- “sudo apt update”: Con este comando, actualizaremos la lista de paquetes, utilizamos la palabra “sudo” para dar permisos de administrador.

```
carles@carles-VirtualBox: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
carles@carles-VirtualBox:~$ sudo apt update  
[sudo] contraseña para carles:  
Obj:1 http://es.archive.ubuntu.com/ubuntu focal InRelease  
Obj:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease  
Obj:3 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease  
Obj:4 http://security.ubuntu.com/ubuntu focal-security InRelease  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Todos los paquetes están actualizados.  
carles@carles-VirtualBox:~$
```

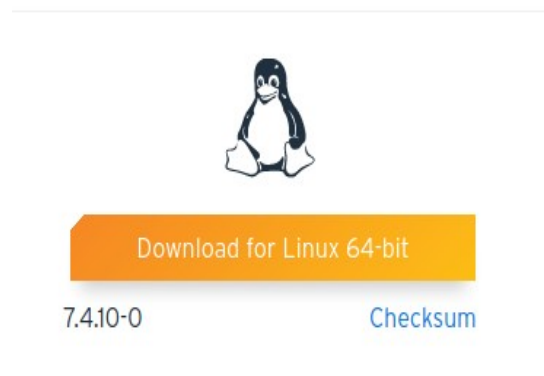
“sudo apt upgrade”: Con este comando instalamos los paquetes nuevos. En mi caso ya los tengo actualizados, por eso no aparece ninguno en la captura anterior.

```
carles@carles-VirtualBox:~$ sudo apt upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
carles@carles-VirtualBox:~$
```

Para poder realizar las pruebas de comunicación necesitamos crear un entorno servidor en nuestro Ubuntu Mate, para facilitar este proceso, descargaremos el paquete “Bitnami LAMP” que lo podemos encontrar en <https://bitnami.com/stack/lamp/installer>.

“Este paquete es bueno para la realización de pruebas pero no se aconseja para aplicaciones en producción.”

En la página seleccionamos la opción “Win / Mac / Linux” y pulsamos en “Download for Linux 64-Bit” pulsamos en siguiente y empezará el proceso de descarga.



## 5.1- Instalación de Bitnami LAMP

En primer lugar tenemos que darle permisos de ejecución al instalador, para ello abrimos una Terminal, nos situamos en la carpeta de descargas e introducimos el comando “**sudo chmod 755 bitnami-lampstack-7.4.10-0-linux-x64-installer.run**”.

Instalamos la librería requerida por Mysql para evitar errores en la instalación, simplemente insertamos el comando “**sudo apt install libncurses5**”

Ejecutamos el instalador introduciendo “**./ bitnami-lampstack-7.4.10-0-linux-x64-installer.run**” damos a siguiente en todas las pantallas, introducimos una contraseña y pulsamos en “Instalar” (*En mi caso he deshabilitado la opción de ejecutar en la nube porque no se instalaba correctamente*).

### 5.1.2- Prueba de Bitnami LAMP (Servidor)

Para comprobar que todo ha salido correctamente, introducimos lo siguiente en el navegador “<file:///home/carles/LAMP/apache2/htdocs/index.html>” esta ruta es el directorio principal donde se desplegará nuestra aplicación PHP. Es posible que la ruta sea diferente, ya que depende de donde se haya instalado Bitnami LAMP.

Una buena práctica es crear carpetas para nuestros proyectos y así tener una mejor organización.

Para crear una carpeta nos situamos dentro de “/htdocs “ pulsamos botón derecho del ratón y elegimos “Abrir en una terminal”. Una vez en la Terminal escribimos “**mkdir prueba**” para crear la carpeta, nos situamos en ella y tecleamos “**touch index.html**” para crear el archivo índice de nuestra app. A continuación introducimos “**nano index.html**” para abrir el editor de texto e introducimos el texto “Hola esto es una prueba” pulsamos “Ctrl + x” para guardar los cambios.

Ahora comprobamos si todo ha ido correctamente introduciendo la ruta que apunta a nuestro archivo “index.html” y veremos la siguiente pantalla.



### 5.1.3- Prueba de Bitnami LAMP (Windows)

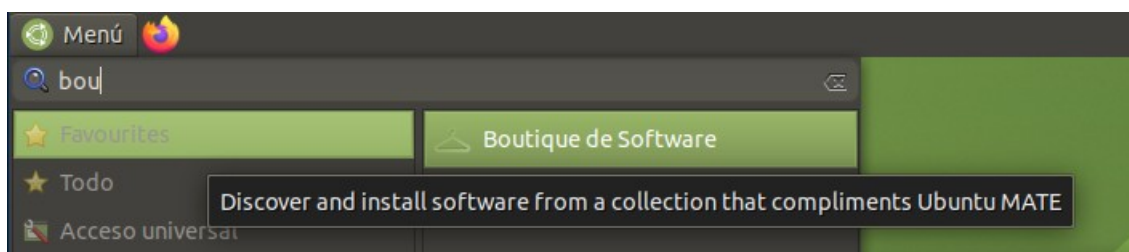
En primer lugar hay que compartir la carpeta “apache2/htdocs” desde la máquina virtual a nuestro Windows local para poder editar los archivos que allí se encuentren. Para conseguirlo utilizaremos SAMBA que es un protocolo de comunicaciones para conectar Windows o Mac Os, con el sistema de archivos de Linux.

“Para utilizar SAMBA, hay que tener instaladas las Guest Additions que hemos instalado anteriormente.”

Tenemos que instalar la aplicación “Caja Share” que utiliza SAMBA para que nuestros archivos puedan ser accedidos desde cualquier SO.

Hay varias formas de instalarla, he optado por la siguiente.

Primero nos situamos en el menú de Ubuntu y buscamos Boutique de Software.



Pulsamos y nos aparecerá una ventana con un buscador para encontrar aplicaciones que podemos instalar en nuestro Ubuntu de forma sencilla.

En el buscador introducimos “samba”, pulsamos Enter y nos aparecera la aplicación Caja Share y pulsamos la opción “Instalar”.

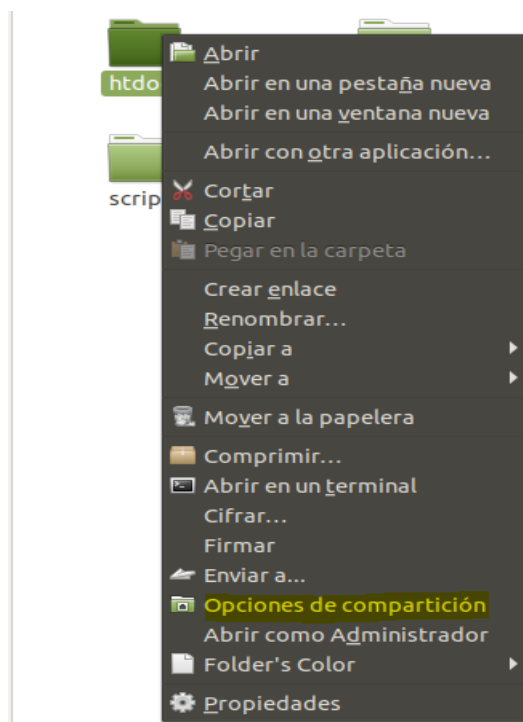


Durante el proceso de instalación se nos dirá que ingresemos la contraseña. Una vez finalizada la instalación deberíamos ver una pantalla como esta.



Para finalizar la instalación reiniciamos Ubuntu nuevamente.

Ahora que tenemos habilitado Caja Share, nos situamos en el directorio donde tenemos la carpeta “htdocs” , pulsamos sobre ella con el botón derecho del ratón y elegimos la opción “Opciones de compartición”.

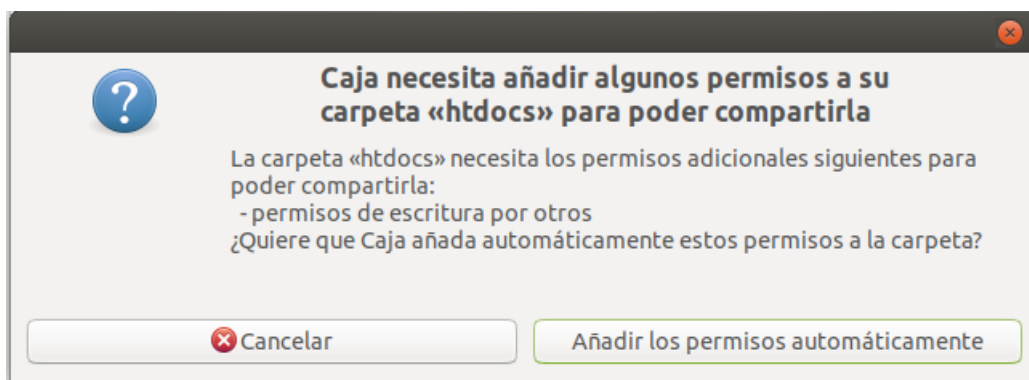




Se nos abre una ventana en la que tenemos que marcar las opciones de “Compartir esta carpeta”, “Permitir a otros crear y eliminar archivos en esta carpeta” y pulsamos en “Crear recurso compartido” (opcionalmente se puede agregar un comentario).



Se nos abrirá una ventana indicándonos que para compartir la carpeta es necesario tener habilitada la opción de permisos de escritura para otros, esto lo podríamos hacer con “chmod”, pero Ubuntu nos da la opción de hacerlo automáticamente.

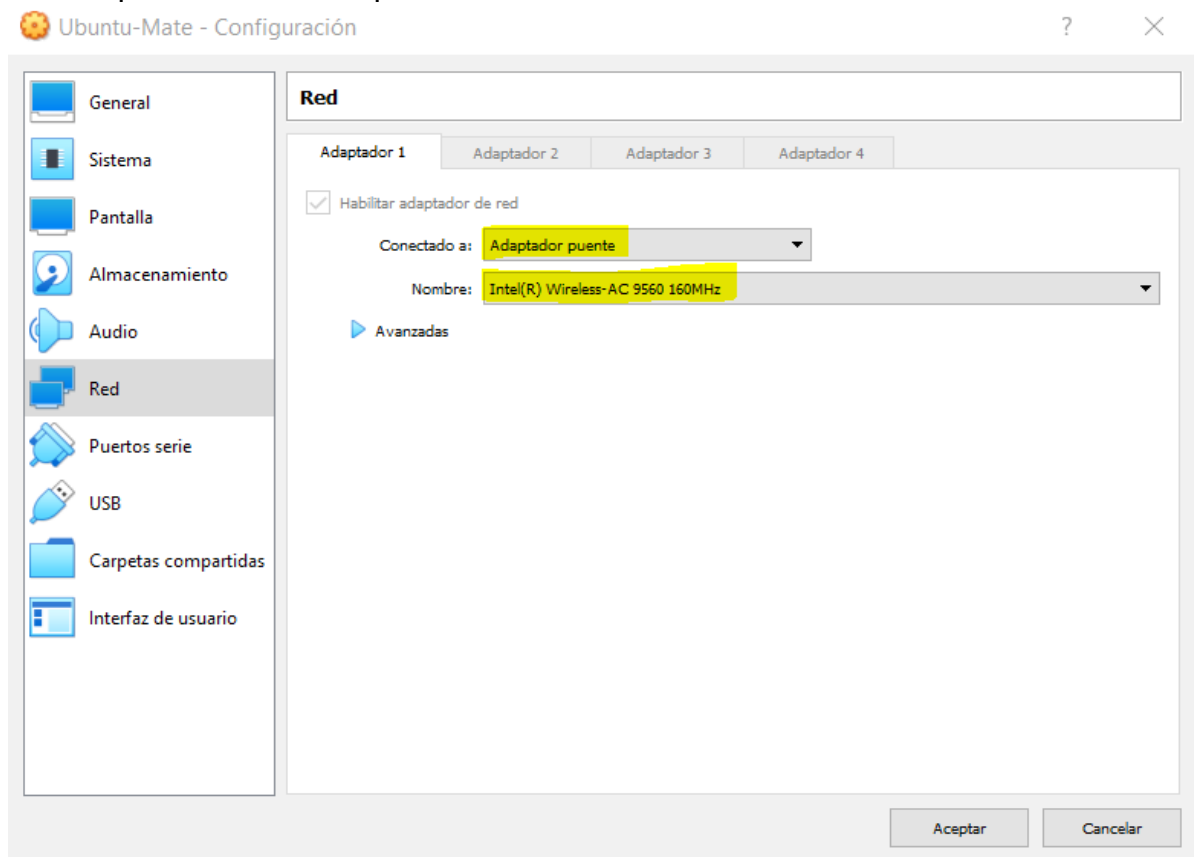


Si todo ha funcionado correctamente deberíamos ver que el icono de la carpeta a cambiado y le aparecen dos flechas en la parte superior.



Ahora necesitamos que ambos sistemas estén en la misma red, para ello pulsamos en la opción “Dispositivos” de nuestro VirtualBox y en “Red” => “Preferencias de red”.

Veremos una ventana donde podremos editar el tipo de conexión, seleccionamos “Adaptador puente”, en el apartado “Nombre”, debería aparecer nuestra tarjeta de Red y finalmente pulsamos en “Aceptar”.



Ahora si comprobamos la dirección IP de nuestro Ubuntu debería aparecer una parecida a la que tenemos en nuestro equipo Windows.

Antes de comprobar que podemos acceder desde nuestro Windows, hay que permitir SAMBA en el Firewall de Ubuntu, para ello Abrimos una Terminal e introducimos los comandos

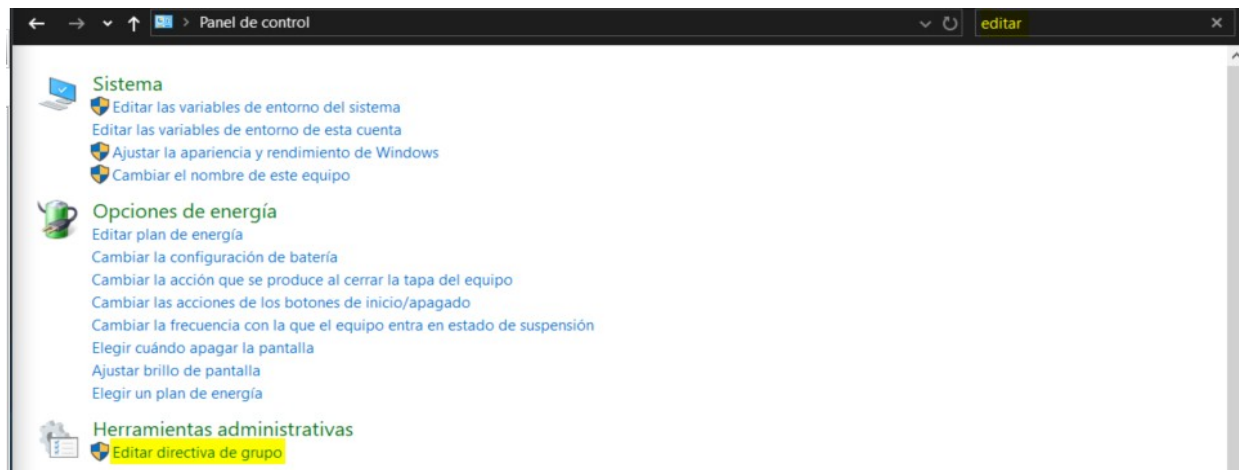
“sudo ufw allow samba “: Da los permisos.

“sudo /etc/init.d/smbd restart “: Reinicia los servicios de SAMBA.

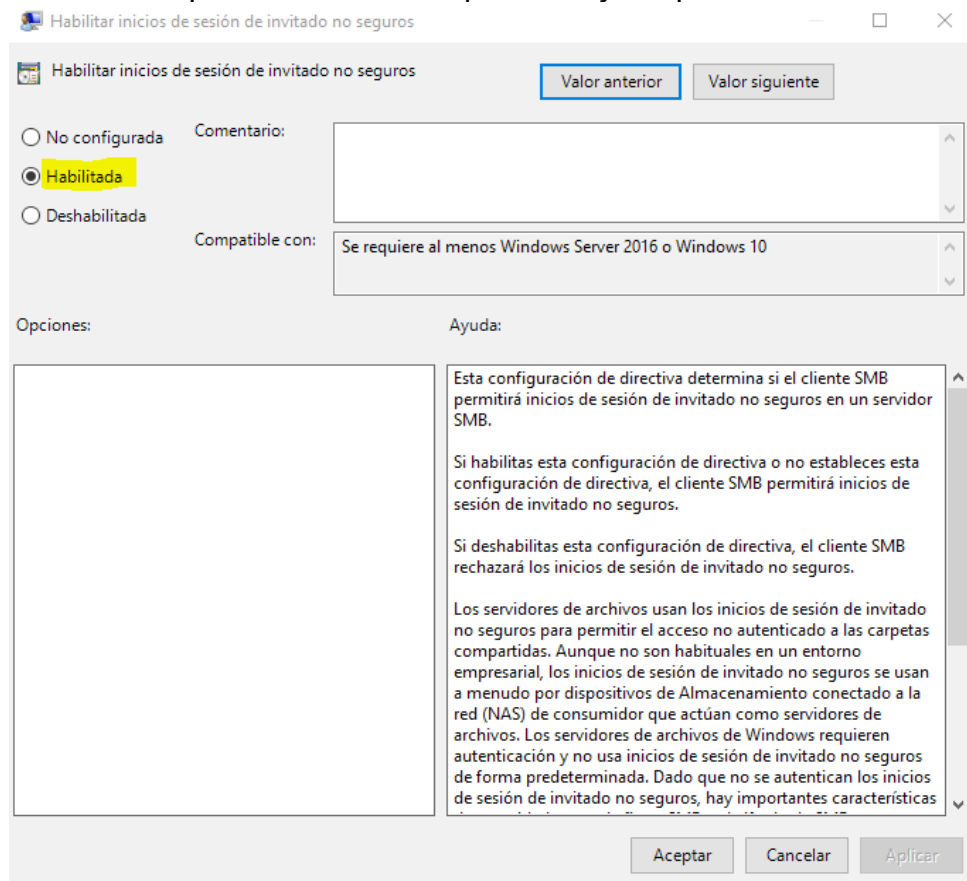
```
Archivo Editar Ver Buscar Terminal Ayuda
carles@carles-VirtualBox:~$ sudo ufw allow samba
[sudo] contraseña para carles:
Reglas actualizadas
Reglas actualizadas (v6)
carles@carles-VirtualBox:~$ sudo /etc/init.d/smbd restart
Restarting smbd (via systemctl): smbd.service.
carles@carles-VirtualBox:~$
```

Una vez en Windows, para evitar problemas, tenemos que ir a “Panel de Control”, escribimos “editar” en el buscador y seleccionamos la opción “Editar directiva de grupo”.

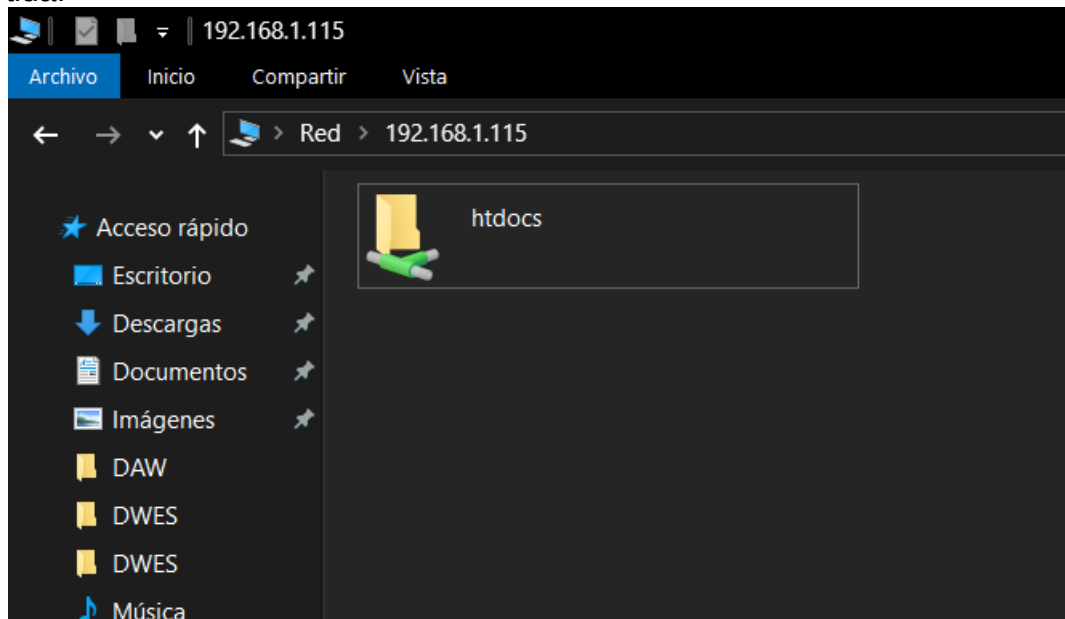




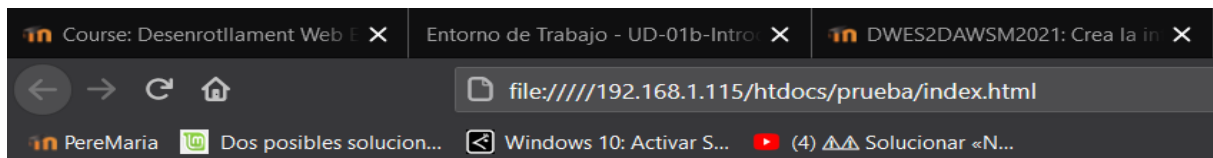
Ahora tenemos que navegar hacia “Plantillas administrativas” => “Red” => “Estación de trabajo Lanman” => “Habilitar inicios de sesión de invitado no seguros”. Una vez dentro, accionamos la opción “Habilitada” aplicamos y aceptamos.



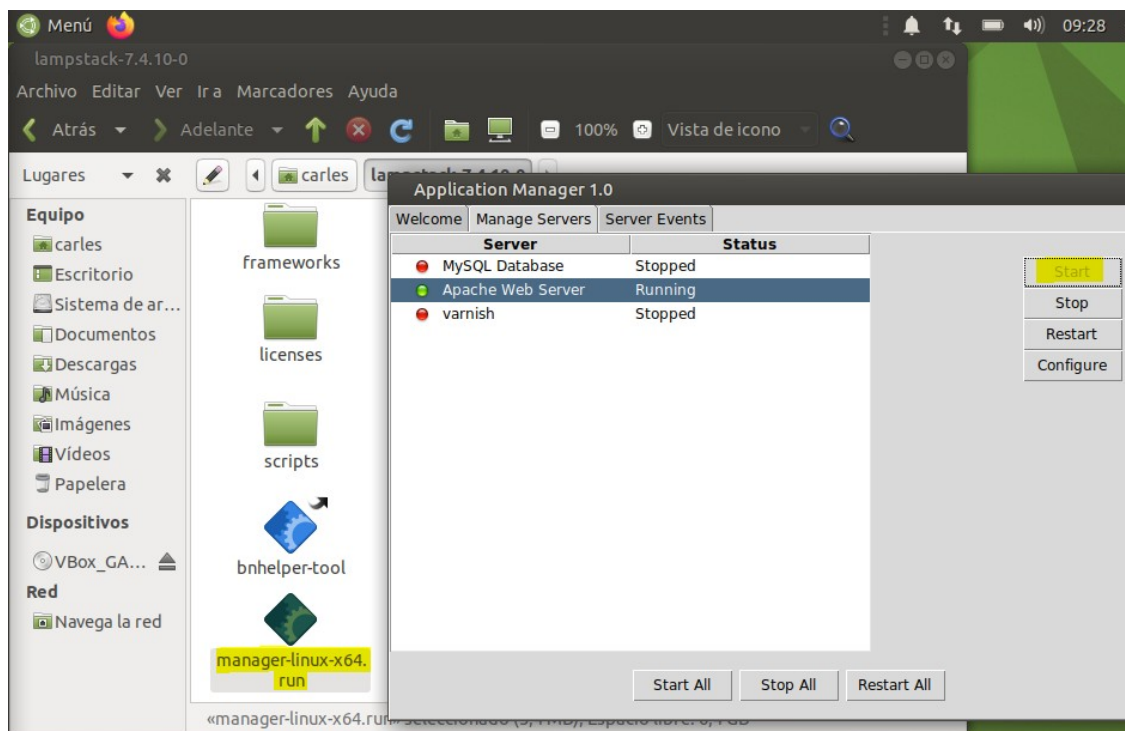
Una vez realizados los cambios anteriores nos situamos en el administrador de archivos e introducimos “\\192.168.1.115” la IP puede variar ya que es la del Ubuntu Virtualizado. Si todo ha funcionado correctamente seremos capaces de ver la carpeta compartida.



Finalmente comprobamos que podemos ejecutar el archivo “index.html” que situamos en la carpeta “prueba” creada anteriormente.

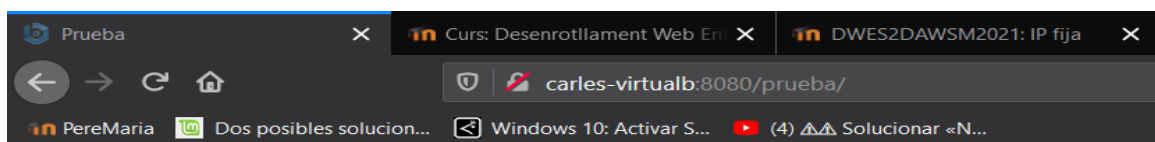


Ahora comprobaremos si tenemos acceso web al servidor, en primer lugar hay que encender el servidor Apache, para ello nos situamos en la carpeta “lampstack-7.4.10-0”, damos doble-click en “manager-linux-x64.run” y se nos abrirá la aplicación para administrar los servidores. Seleccionamos la pestaña “Manage Servers” y seleccionamos “Apache Web Server” y pulsamos en “Start”, si todo ha funcionado deberíamos ver lo siguiente.



Estamos en disposición de probarlo desde nuestro navegador en Windows. Nos situamos en el navegador y escribimos “[Nombre-Nuestra-Máquina]:8080/prueba”.

Utilizamos el nombre de nuestra máquina porque así no nos importa la dirección Ip que se le asigne, ya que esto depende de la red en la que nos conectemos. Si todo funciona correctamente veremos lo siguiente.



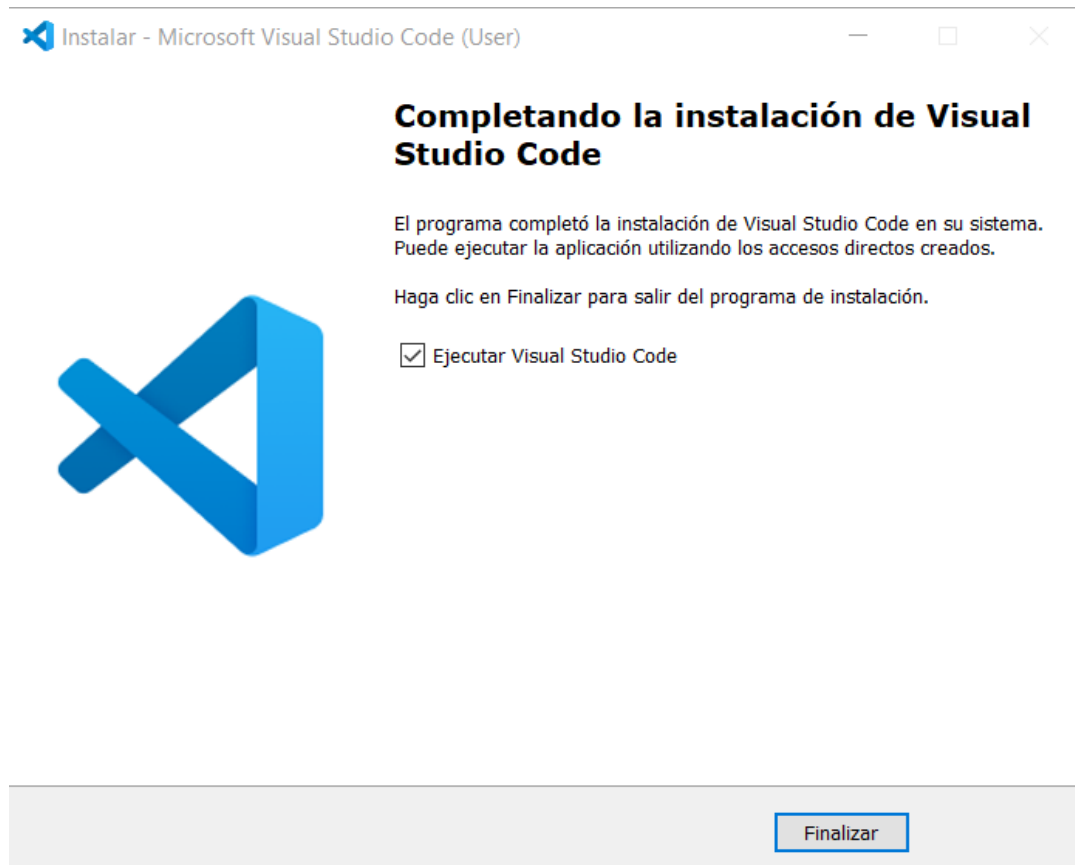
**Hola desde la Web!!**

## 6- Entorno de desarrollo integrado (Ide)

Un Ide es una aplicación informática que proporciona servicios integrales para facilitarle al programador el desarrollo de Software.

Vamos a instalar Visual Studio Code en nuestro Windows que lo podemos encontrar en <https://code.visualstudio.com/download> simplemente seleccionamos nuestro SO y empezará la descarga.

Una vez descargado, lo ejecutamos, aceptamos los términos y vamos pulsando en “Siguiente” y por último en “Instalar”. Si todo fue correctamente veremos lo siguiente.

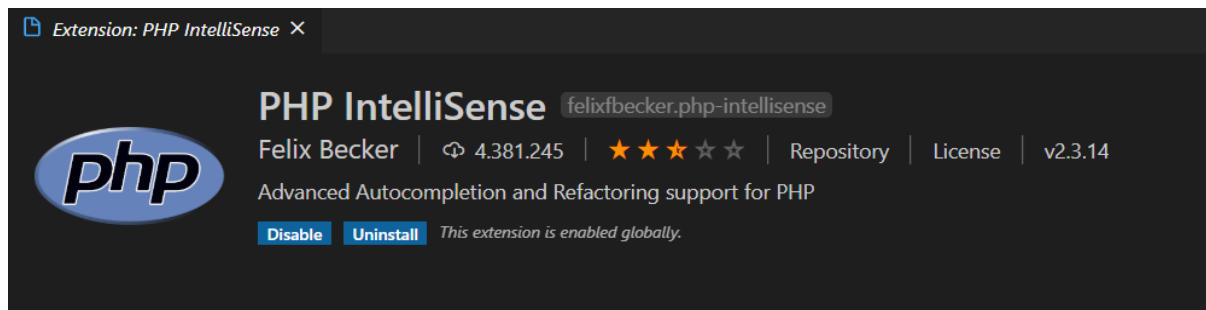


Pulsamos en “Finalizar” y se nos abrirá el editor.

Para facilitar las labores de codificar en PHP instalaremos las extensiones (plugins) “PHP Debug”, “PHPIntellisense” y “HTML CSS Suport” en nuestro VsCode.

Para ello pulsamos en el símbolo  para entrar en la sección “Extensions”.

Buscamos las extensiones y damos a “Instalar”. Para comprobar que se instalan correctamente deberíamos ver lo siguiente.



## 7-Instalación de Git en ambos sistemas

Git es un sistema de control de versiones y de proyectos ampliamente utilizado en el desarrollo de software, permitiendo el trabajo en grupo de forma segura.

### 7.1- Instalación en Ubuntu Mate

Para instalar Git simplemente abrimos una terminal y ejecutamos el comando “sudo apt install git”.

Para configurar Git debemos indicarle un nombre de usuario y un correo, para ello ejecutamos los siguientes comandos:

“git config --global user.name “Nuestro nombre”

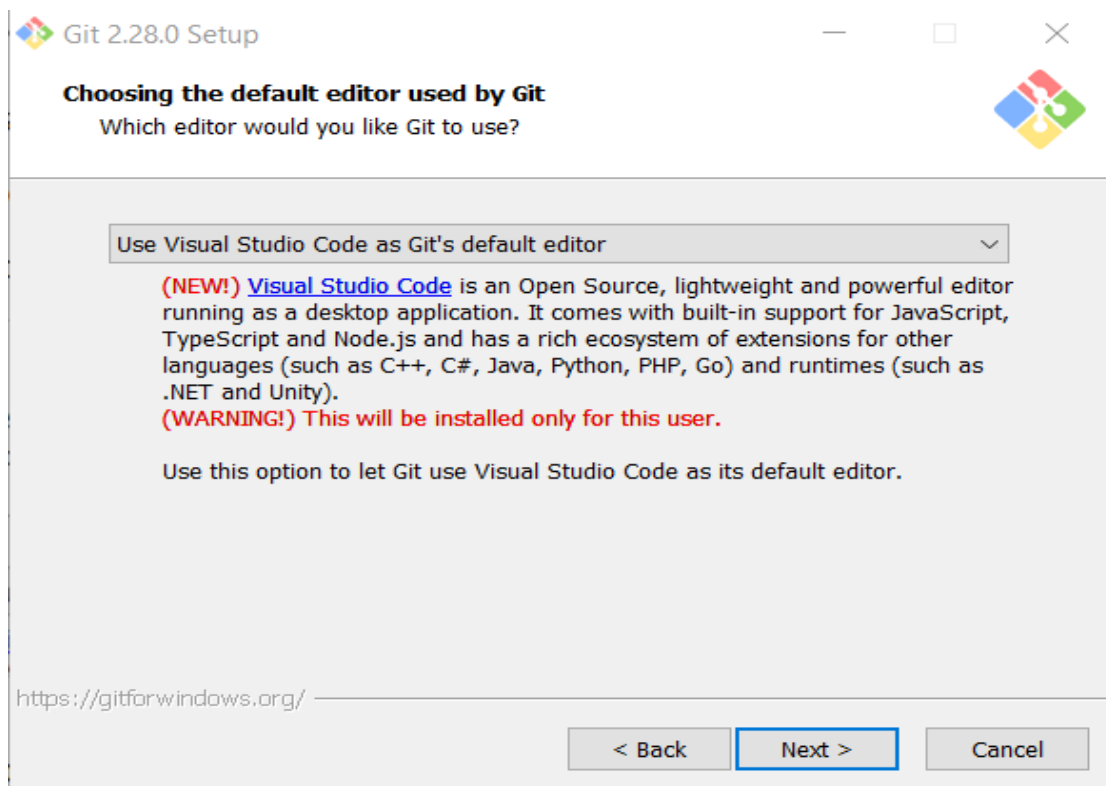
“git config --global user.email “[miemail@miservicio.com](mailto:miemail@miservicio.com)”

```
carles@carles-VirtualBox: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
carles@carles-VirtualBox:~$ git config --global user.name "Carles-Ubuntu"  
carles@carles-VirtualBox:~$ git config --global user.email "jcramos8282@gmail.com"  
carles@carles-VirtualBox:~$
```

Para crear un repositorio, nos situamos en la carpeta en la que lo queremos y ejecutamos el comando “git init”.

## 7.2- Instalación en Windows

Lo primero es descargar Git desde <https://git-scm.com/download/win> ejecutamos y damos a "Next" hasta que nos pida elegir el editor por defecto y seleccionamos Visual Studio Code y pulsamos en "Next" hasta que empiece la instalación.



## 8-Debug en Visual Studio Code


Hacer Debug en nuestro código es imprescindible en cualquier entorno de trabajo, ya que nos permite localizar los fallos deteniendo el flujo del programa mediante breakpoints.

### 8.1-Instalación de Xdebug en el servidor.

Lo primero que tenemos que hacer es cambiar nuestro código PHP y escribir lo siguiente.

```
index.php x Extension: PHP Debug
htdocs > prueba > index.php
1
2 <?php
3 | phpinfo();
4 ?>
5
6 |
```

Con la función `phpinfo()`, al ejecutarlo obtendremos la información sobre nuestra versión de PHP, Apache, equipo y otros componentes.

<div> <div>PHP Version 7.4.10</div>  </div>	
System	Linux carles-VirtualBox 5.4.0-47-generic #51-Ubuntu SMP Fri Sep 4 19:50:52 UTC 2020 x86_64
Build Date	Sep 7 2020 08:41:47
Configure Command	<pre>'./configure' '--prefix=/bitnami/lamp74stack-linux-x64/output/php' '--enable-fpm' '--with-fpm-user=daemon' '--with-fpm-group=daemon' '--with-apxs2=/bitnami/lamp74stack-linux-x64/output/apache2/bin/apxs' '--with-zlib-dir=/bitnami/lamp74stack-linux-x64/output/common' '--enable-mbstring=all' '--enable-soap' '--enable-bcmath' '--enable-ftp' '--with-xmllib' '--enable-cgi' '--with-imap=/bitnami/lamp74stack-linux-x64/src/imap-2007f' '--with-imap-ssl=/bitnami/lamp74stack-linux-x64/output/common' '--with-png-dir=/bitnami/lamp74stack-linux-x64/output/common' '--with-gd' '--with-zlib' '--with-curl=/bitnami/lamp74stack-linux-x64/output/common' '--with-jpeg-dir=/bitnami/lamp74stack-linux-x64/output/common' '--enable-exif' '--with-openssl=/bitnami/lamp74stack-linux-x64/output/common' '--enable-calendar' '--enable-ctype' '--enable-pcntl' '--enable-session' '--enable-zip' '--with-bz2=/bitnami/lamp74stack-linux-x64/output/common' '--enable-sockets' '--with-mcrypt=/bitnami/lamp74stack-linux-x64/output/common' '--with-icu-dir=/bitnami/lamp74stack-linux-x64/output/common' '--with-tidy=/bitnami/lamp74stack-linux-x64/output/common' '--with-mysqli=mysqlnd' '--with-pdo-mysql=mysqlnd' '--with-pdo-sqlite=/bitnami/lamp74stack-linux-x64/output/sqlite' '--with-sqlite3=/bitnami/lamp74stack-linux-x64/output/sqlite' '--with-iconv=/bitnami/lamp74stack-linux-x64/output/common' '--with-libxml-dir=/bitnami/lamp74stack-linux-x64/output/common' '--with-xsl=/bitnami/lamp74stack-linux-x64/output/common' '--enable-mbstring' '--with-freetype-dir=/bitnami/lamp74stack-linux-x64/output/common' '--with-gmp=/bitnami/lamp74stack-linux-x64/output/common' '--with-gettext' '--enable-intl' '--with-readline=/bitnami/lamp74stack-linux-x64/output/common' '--with-libzip=/bitnami/lamp74stack-linux-x64/output/common' '--with-pear' '--enable-gd' '--with-jpeg' '--with-libwebp' '--with-zip' '--disable-huge-code-pages' 'PKG_CONFIG_PATH=/bitnami/lamp74stack-linux-x64/output/sqlite/lib/pkgconfig:/bitnami/lamp74stack-linux-x64/output/common/lib/pkgconfig' 'JPEG_CFLAGS=-I/bitnami/lamp74stack-linux-x64/output/common/include' 'JPEG_LIBS=-L/bitnami/lamp74stack-linux-x64/output/common/lib -ljpeg'</pre>
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/bitnami/lamp74stack-linux-x64/output/php/lib
Loaded Configuration File	(none)
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20190902

Ahora hacemos click derecho sobre la página y seleccionamos “Ver código fuente”, seleccionamos todo el código, lo copiamos y nos dirigimos a <https://xdebug.org/wizard>

Veremos una página con un cuadro de texto en el que pegaremos el código, esta herramienta sirve para que nos indique la versión del debugger que tenemos que instalar según la información que le hemos introducido.

Seguidamente pulsamos en “Analyse my phpinfo() output”.

# Installation Wizard

This page helps you finding which file to download, and how to configure PHP to get Xdebug running. Please paste the **full** output of phpinfo() (either a copy & paste of the HTML version, the HTML source or `php -i` output) and submit the form to receive tailored download and installation instructions.

```
Brown </td></tr>
<tr><td class="e">Network Infrastructure </td><td class="v">Daniel P. Brown
</td></tr>
<tr><td class="e">Windows Infrastructure </td><td class="v">Alex Schoenmaker
</td></tr>
</table>
<h2>PHP License</h2>
<table>
<tr class="v"><td>
<p>
This program is free software; you can redistribute it and/or modify it under
the terms of the PHP License as published by the PHP Group and included in the
distribution in the file:  LICENSE
</p>
<p>This program is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
FOR A PARTICULAR PURPOSE.
</p>
<p>If you did not receive a copy of the PHP license, or have any questions
about PHP licensing, please contact license@php.net.
</p>
</td></tr>
</table>
</div></body></html>
```

The information that you upload will not be stored. The script will only use a few regular expressions to analyse the output and provide you with instructions. You can see the code [here](#).

Analyse my phpinfo() output

Ahora vemos una página con las recomendaciones para la instalación, lo que nos interesa esta en la sección "Instructions"



## Instructions

1. Download [xdebug-2.9.7.tgz](#)
2. Install the pre-requisites for compiling PHP extensions.  
On your Ubuntu system, install them with: `apt-get install php-dev autoconf automake`
3. Unpack the downloaded file with `tar -xvzf xdebug-2.9.7.tgz`
4. Run: `cd xdebug-2.9.7`
5. Run: `phpize` (See the [FAQ](#) if you don't have `phpize`).

As part of its output it should show:

```
Configuring for:
...
Zend Module Api No:      20190902
Zend Extension Api No:   320190902
```

If it does not, you are using the wrong `phpize`. Please follow [this FAQ entry](#) and skip the next step.

6. Run: `./configure`
7. Run: `make`
8. Run: `cp modules/xdebug.so /bitnami/lamp74stack-linux-x64/output/php/lib/php/extensions`
9. Create `/bitnami/lamp74stack-linux-x64/output/php/lib/php.ini` and add the line  
`zend_extension = /bitnami/lamp74stack-linux-x64/output/php/lib/php/extensions/xdebug.so`
10. Restart the webserver

Seguimos todos los pasos que se nos indican.

**Nota:** “*No he podido completar el proceso, pero ahora en Visual Studio Code aparece el icono de debug*”.

