

Documentar y publicar las siguientes actividades evaluables.

ACTIVIDADES EVALUABLES

Actividad 1.

Objetivo: Instalar y configurar un paquete software que integre Apache, PHP y MySQL.

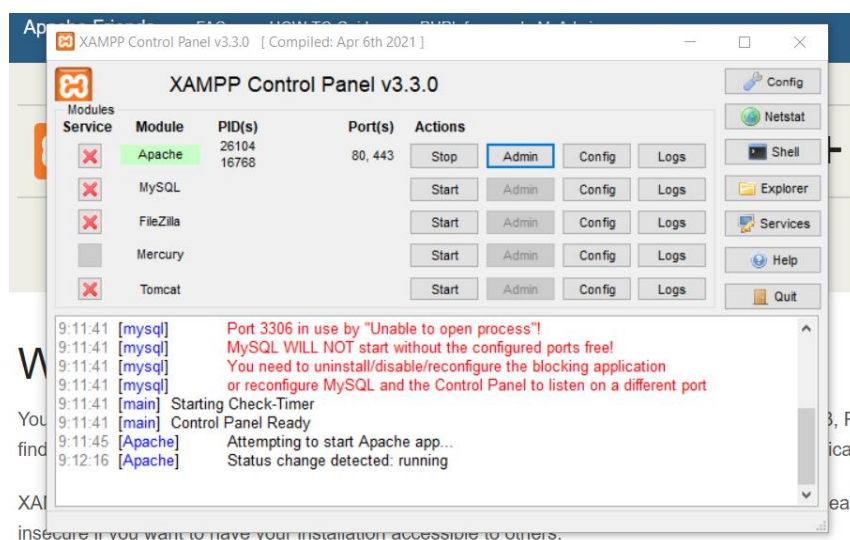
1. Instalar y configurar XAMPP.

Vamos a instalar la aplicación de **XAMPP** para ello vamos a irnos a la [página web oficial](https://www.apachefriends.org/es/index.html) del programa. Y nos vamos a el apartado e descargar y instalamos el instalador del programa para nuestro S.O.



En mi caso pos yo seleccioné **Window** y cuando le das se **descarga** en tu equipo un **programa de instalación**.

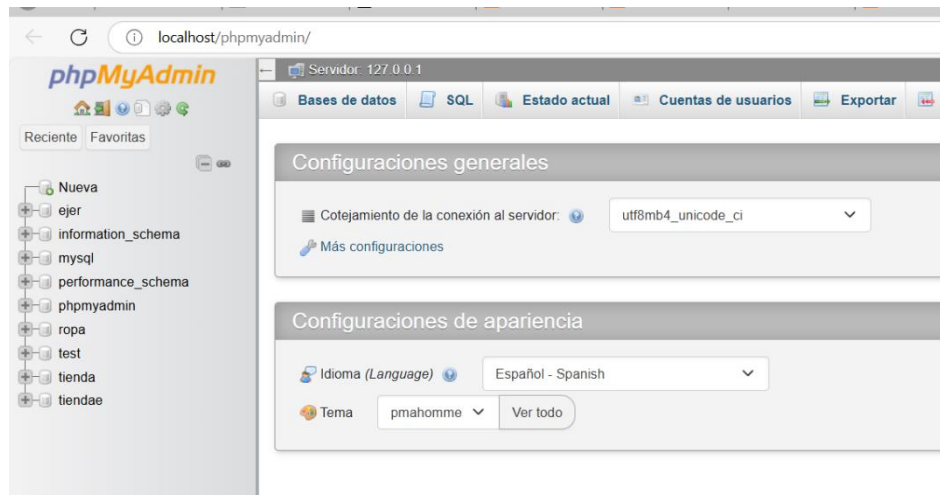
En el **programa de instalación** tenemos que darle a siguiente y a aceptar y se te instalará el **XAMPP** y una vez instalada la abrimos y se nos abrirá el menú.



2. Instalar phpmyadmin.

Para instalar el **phpmyadmin** lo que tenemos que hacer en el menú es iniciar **MySQL**, para ello le damos a **“Start”**.

Un **problema** que he tenido es que tenía otro programa con el **servicio mysql**, así que lo he **detenido** y he vuelto a **iniciar** mi **phpmyadmin** y me ha **dejado**.



3. Probar la aplicación creando un portfolio personal básico.

Para **crear** un **portfolio** que se acceda con nuestra **IP** lo que haremos primero será **activar** el **servidor de apache** en nuestro **XAMPP**.

Una vez que lo hayamos hecho **accederemos** a la siguiente **ruta** y eliminaremos el **index.php** y crearemos uno que sea **index.html**, **index** es el archivo **principal** que reconoce el **servidor web**.












Portapapeles

Organizar

Nuevo

Abrir

> Este equipo > Windows (C:) > xampp > htdocs

	Nombre	Fecha de modificación	Tipo	Tamaño
lo	 Canalplad	16/08/2024 12:44	Carpeta de archivos	
	 dashboard	01/04/2024 11:42	Carpeta de archivos	
	 img	19/09/2024 10:22	Carpeta de archivos	
	 portfolio	30/08/2024 12:33	Carpeta de archivos	
JS	 webalizer	01/04/2024 11:42	Carpeta de archivos	
	 xampp	01/04/2024 11:42	Carpeta de archivos	
NS	 .hintrc	19/09/2024 9:54	Archivo HINTRC	1 KB
	 applications.html	15/06/2022 18:07	Microsoft Edge HT...	4 KB
	 bitnami.css	19/09/2024 10:16	Documento de hoj...	2 KB
	 favicon.ico	16/07/2015 17:32	Archivo ICO	31 KB
	 index.html	19/09/2024 10:22	Microsoft Edge HT...	2 KB

Una vez que hayamos creado el archivo procederemos a crear la **estructura** de nuestra **aplicación web**. Aquí pongo la captura del **body**, en el **head** es el que crea por defecto **HTML5** más el enlace del archivo **css**.

```
<body>
  <header>
    <h3>Javier R.M</h3>
    <nav>
      <a href="">Inicio</a>
      <a href="">Sobre Mi</a>
      <a href="">Portfolio</a>
      <a href="">Contacto</a>
    </nav>
  </header>
  <main>
    <h1>PORTFOLIO</h1>
    <div class="divy">
      
      <div class="divInfo">
        <h4>Datos Personales</h4>
        <p>Javier Ruiz Molero</p>
        <p>19 Años</p>
        <p>Villafranca de Córdoba</p>
      </div>
      <div class="estudios">
        <h4>Estudios</h4>
        <p>ESO</p>
        <p>Grado Medio de Microinformática y redes</p>
        <p>En proceso: Grado superior de Desarrollo de Aplicaciones Web</p>
      </div>
    </div>
  </main>
</body>
```

Y ahora procederemos a crear los estilos de nuestra página creados en el archivo **bitnami.css** . Una parte del código:

```
}
main{
  padding: 4rem 10rem;
  background-color: rgb(33, 33, 33);
}
h1{
  color: white;
  font-family: system-ui, -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, C
  font-size: 3rem;
  margin-bottom: 2rem;
}
main img{
  height: 400px;
}
.divy{
  display: flex;
}
.divInfo{
  padding: 2rem 4rem;
  width: 55%;
}
.divInfo p {
  color: white;
  font-family: system-ui, -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, C
  font-size: 1.5rem;
}
.estudios{
  margin-top: 2rem;
}
```

Este es el **resultado final** del proyecto. Un poco simple pero **rápido**.



Actividad 2.

Objetivo: Instalación y configuración de un entorno web utilizando una máquina virtual.

1. Descargar e instalar VirtualBox.

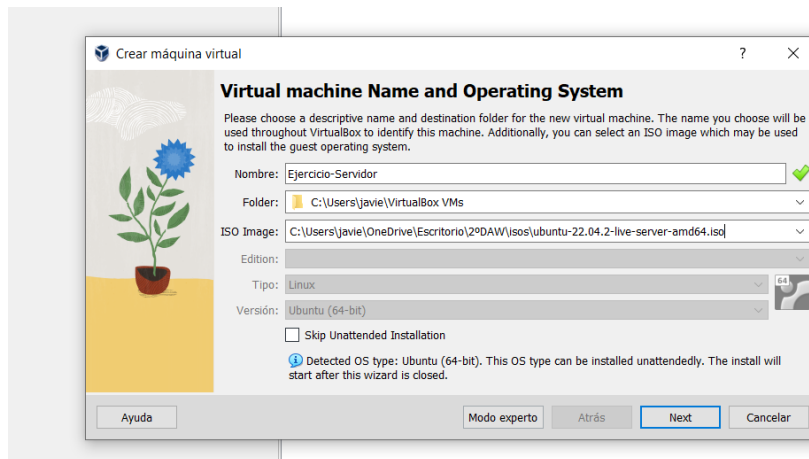
Para la instalación de virtualbox(**software de virtualización**), lo primero que debemos hacer es acceder a la [página web oficial](#) y descargar el instalador de nuestro **S.O.**



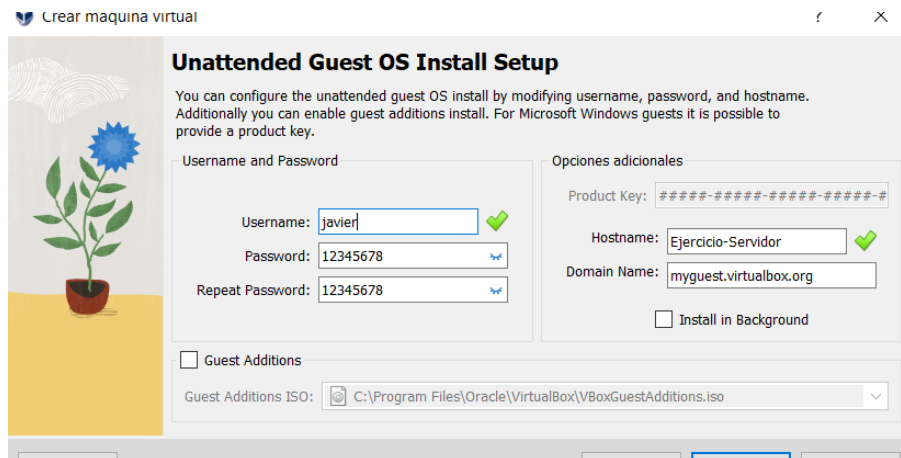
Una vez lo tengamos descargado abrimos el instalador y le damos a siguiente y aceptamos los términos y condiciones.

2. Crear una máquina virtual con Ubuntu Server e instalar de manera independiente Apache, MariaDB, PHP y phpMyAdmin.

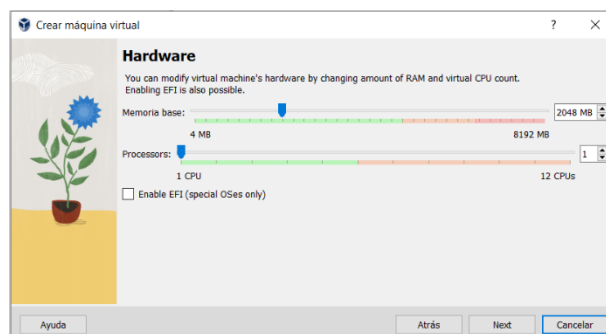
Para crear una máquina virtual dentro de **VirtualBox** lo que haremos será pinchar en **Máquina > Nueva** y en la pantalla inicial pondremos el **nombre** y la **iso** descargada anteriormente, para instalar la iso nos iremos a la **página web oficial de Ubuntu**.



Ahora ponemos el usuario y contraseña que queramos

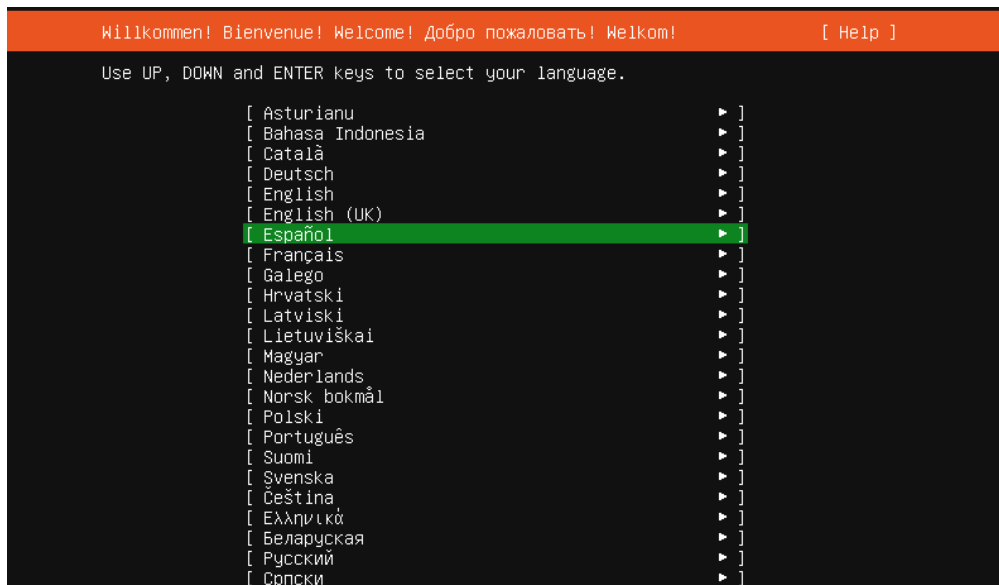


Dejamos por defecto el siguiente hardware

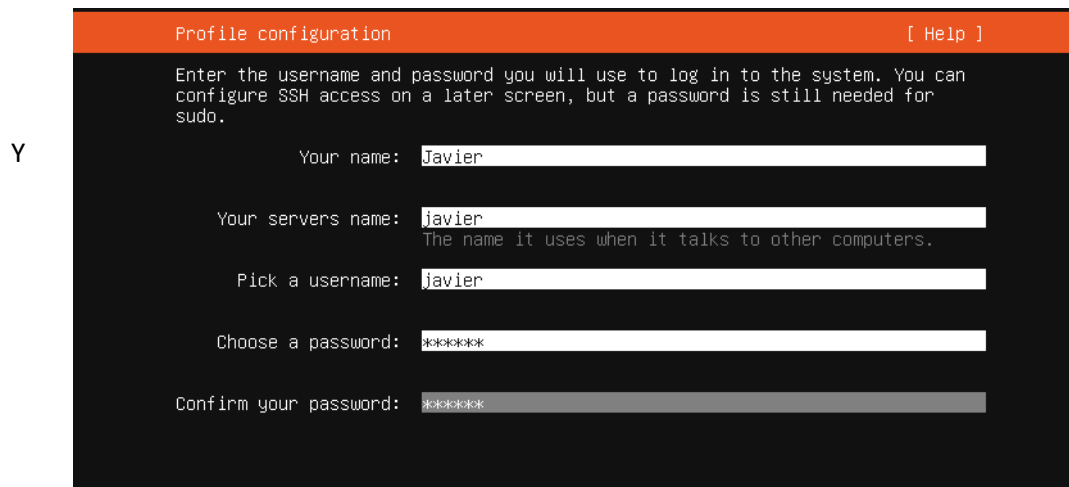


Y por último nos pedirá el tamaño del disco duro, al ser una práctica pequeña ponemos 15GB y ya tendremos la instalación terminada.

Ahora abrimos la máquina virtual y esperamos y nos saldrá la selección del idioma.



Ahora nos aparecerá más cosas de configuración y las dejaremos por defecto hasta la siguiente pantalla.



pincharemos en **siguiente** lo demás y se procederá a **iniciar la instalación** del Ubuntu server.

Ahora vamos a proceder con la instalación de los servicios necesarios, primero **actualizaremos el sistema**.

```
see "man sudo_root" for details.

javier@javier:~$ sudo apt update && sudo apt upgrade
[sudo] password for javier:
Des:1 http://security.ubuntu.com/ubuntu jammy-security InRe
Obj:2 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Des:3 http://es.archive.ubuntu.com/ubuntu jammy-updates InR
Obj:4 http://es.archive.ubuntu.com/ubuntu jammy-backports I
```

Instalación Apache

Primero lo que haremos será **descargarlo**, para ello vamos a utilizar el comando: **sudo apt install apache2**

Una vez lo hayamos hecho lo que haremos será **iniciarlo** y **habilitarlo** con los siguientes comandos:

sudo systemctl start apache2

sudo systemctl enable apache2

Y por ultimo lo que vamos a hacer es comprobar si esta funcionando: **curl http://localhost**

```
_html</a>      <a href="http://httpd.apache.org/docs/2.4/mod/mod_userdir.html" rel="nofollow">
                directories (when enabled) and <tt>/usr/share</tt> (for web
                applications). If your site is using a web document root
                located elsewhere (such as in <tt>/srv</tt>) you may need to whitelist your
                document root directory in <tt>/etc/apache2/apache2.conf</tt>.
            </p>
            <p>
                The default Ubuntu document root is <tt>/var/www/html</tt>. You
                can make your own virtual hosts under /var/www.
            </p>
        </div>
        <div class="section_header">
            <div id="bugs"></div>
            Reporting Problems
        </div>
        <div class="content_section_text">
            <p>
                Please use the <tt>ubuntu-bug</tt> tool to report bugs in the
                Apache2 package with Ubuntu. However, check <a
                href="https://bugs.launchpad.net/ubuntu/+source/apache2"
                rel="nofollow">existing bug reports</a> before reporting a new bug.
            </p>
            <p>
                Please report bugs specific to modules (such as PHP and others)
                to their respective packages, not to the web server itself.
            </p>
        </div>
    </div>
</div>
<div class="footnote">
```

Instalación MariaDB

Primero lo que haremos será **instalarlo** con: **sudo apt install mariadb-server**

Ahora lo que haremos será iniciarlo y habilitarlo como lo hemos hecho antes pero **sustituyendo apache por mariadb**.

Ahora vamos a **ejecutar un script de seguridad** que te pedirá configurar una **contraseña root**, **eliminar usuarios** anónimos y **eliminar** la base de datos de prueba: **sudo mysql_secure_installation**

Y ahora probamos la conexión:

```
javier@javier:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.6.18-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> _
```


Instalación PHP

Primero lo que haremos será **instarlo** con: `sudo apt install libapache2-mod-php php-mysql`

Instalación PhpMyAdmin

Primero lo que haremos será **instarlo** con: `sudo apt install phpmyadmin`

Y verificamos si esta funcionando correctamente al conectarlo con apache2:

```
javier@javier:~$ sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin
javier@javier:~$ curl http://localhost/phpmyadmin
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="http://localhost/phpmyadmin/">here</a>.</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at localhost Port 80</address>
</body></html>
javier@javier:~$ _
```

4. Probar la aplicación creando un portfolio personal básico.

Ahora lo que vamos a hacer es conectarnos por **ssh** desde mi **máquina** al **servidor**, para ello tenemos que tener la máquina en **adaptador puente** para la red.

```
C:\Users\javier>ssh javier@192.168.1.22
The authenticity of host '192.168.1.22 (192.168.1.22)' can't be established.
ECDSA key fingerprint is SHA256:D0zb995WMEZRLNLuevqAmqZsHwDwwQGzkk1OM29uaM8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.22' (ECDSA) to the list of known hosts.
javier@192.168.1.22's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-122-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
```

Ahora lo que vamos a hacer es **trasferir** los **archivos** de mi **window** del portfolio anterior al **servidor**, para ello lo que tenemos que hacer es desde el servidor darle **permisos** para ello con **chown** y después utilizar el **scp** para la **transferencia** de los **archivos** de desarrollo(html y css)

```
Last login: Thu Sep 19 18:56:32 2024 from 192.168.1.15
javier@javier:~$ sudo chown -R javier:javier /var/www/html
[sudo] password for javier:
javier@javier:~$ exit
logout
Connection to 192.168.1.22 closed.

C:\xampp\htdocs>sudo scp index.html javier@192.168.1.22:/var/www/html/
"sudo" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\xampp\htdocs>scp index.html javier@192.168.1.22:/var/www/html/
javier@192.168.1.22's password:
index.html                                100% 1219   243.1KB/s   00:00

C:\xampp\htdocs>

C:\xampp\htdocs>scp bitnami.css javier@192.168.1.22:/var/www/html/
javier@192.168.1.22's password:
bitnami.css                              100% 2010   671.8KB/s   00:00

C:\xampp\htdocs>
```

Actividad 4.

Objetivo: Crear un entorno virtual para desarrollo en el lado del servidor.

1. Descargar e instalar Vagrant.

Primero tenemos que **entrar** en la página web de vagrant y descargamos el instalador de **Window**.

Vagrant es una **herramienta o aplicación de líneas de comando utilizada en el sector IT**, especialmente por desarrolladores. Permite la creación de entornos de desarrollo virtualizados que pueden ser reproducidos y compartidos de una forma muy fácil.

Comprobación de la versión de vagrant

```
C:\Users\javier>vagrant --version
Vagrant 2.4.1
```

3. Añadir el proyecto Laravel Homestead seleccionando VirtualBox

En nuestro equipo, abrimos la terminal y añadimos el proyecto con virtualbox

```
C:\Users\javier>vagrant box add laravel/homestead
==> box: Loading metadata for box 'laravel/homestead'
    box: URL: https://vagrantcloud.com/api/v2/vagrant/laravel/homestead
This box can work with multiple providers! The providers that it
can work with are listed below. Please review the list and choose
the provider you will be working with.

1) parallels
2) virtualbox

Enter your choice: 2
==> box: Adding box 'laravel/homestead' (v14.0.2) for provider: virtualbox (amd64)
    box: Downloading: https://vagrantcloud.com/laravel/boxes/homestead/versions/14.0.2.box
```

4. Clonar proyecto Laravel Homestead en el directorio de trabajo

Lo que hemos hecho para clonar el proyecto de github es:

1. Hemos abierto en el directorio de trabajo: **git bash**
2. Hemos hecho un **git init**

3. Hemos hecho un `git clone <url>`

Portapapeles	Organizar	Nuevo	Abrir	Seleccionar
Este equipo > Escritorio > 2ºDAW > Servidor > homestead				
Nombre	Fecha de modificación	Tipo	Tamaño	
.backup	21/09/2024 11:29	Carpeta de archivos		
.git	21/09/2024 11:29	Carpeta de archivos		
.github	21/09/2024 11:29	Carpeta de archivos		
.vagrant	21/09/2024 11:30	Carpeta de archivos		
art	21/09/2024 11:29	Carpeta de archivos		
bin	21/09/2024 11:29	Carpeta de archivos		
resources	21/09/2024 11:29	Carpeta de archivos		
scripts	21/09/2024 11:29	Carpeta de archivos		
src	21/09/2024 11:29	Carpeta de archivos		

5. Inicia homestead

Para iniciarlo en el repositorio abriremos el archivo: `init.bat`

Homestead.yaml.example	21/09/2024 11:29	Archivo EXAMPLE	1 KB
init.bat	21/09/2024 11:29	Archivo por lotes ...	1 KB
init.sh	21/09/2024 11:29	Archivo SH	1 KB

6. Observar el archivo de configuración `Homestead.yaml`, especialmente en `folders`, y `sites`. Es posible tener múltiples proyectos en la misma máquina.

```
---
ip: "192.168.56.10"
memory: 2048
cpus: 2
provider: virtualbox

authorize: ~/.ssh/id_rsa.pub

keys:
  - ~/.ssh/id_rsa

folders:
  - map: ~/code
    to: /home/vagrant/code

sites:
  - map: homestead.dw
    to: /home/vagrant/code/public

databases:
  - homestead

features:
  - mariadb: false
  - postgresql: false
  - ohmyzsh: false
  - webdriver: false

services:
  - enabled:
      "nginx"
```

Lo que hemos modificado en el archivo de configuración es:

Ip: por una nueva ip que le asignemos al vagrant

Folder/map: tenemos que poner la ruta de nuestro equipo que tengas los archivos

7. Lanza la máquina.

Para lanzar nuestra máquina tendremos que ejecutar el comando que aparece en la captura, lo que hace es crear una máquina virtual que corre en segundo plano con el vagrant.

```
==> box: Successfully added box 'hashicorp/precise32' (v1.0.2) for 'virtualbox' provider.
C:\Users\javier>vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'hashicorp/precise32'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'hashicorp/precise32' version '1.0.0' is up to date...
==> default: Setting the name of the VM: javie_default_1726913818930_21289
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
    default: SSH auth method: private key
```

8. Prueba a conectarte a la máquina por ssh.

Un problema que he tenido en la práctica es la clave ssh que o estaba mal creada o no la tenía agragada y con el siguiente comando añadimos el ssh.

```
C:\Users\javier>ssh-keygen -t rsa -C "javierrumo2@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\javier\.ssh/id_rsa):
C:\Users\javier\.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\javier\.ssh/id_rsa.
Your public key has been saved in C:\Users\javier\.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:TbSnpYspm3h1bz0Xbzv4Tsl86pgHILusqDtRc+QqHYJ javierrumo2@gmail.com
The key's randomart image is:
+---[RSA 3072]-----+
|
|..
|..
|.oo
|E=oo=
|+S+..o
|o+o..o o..
|..+o..o +..+
|.o+..=+o=
|..+..oood=Bo|
```

Para conectarnos por **ssh** utilizaremos el comando de la **imagen**:

```
C:\Users\javier>ssh javier@192.168.1.19
The authenticity of host '192.168.1.19 (192.168.1.19)' can't be established.
ECDSA key fingerprint is SHA256:Gwn7QGeQAgBKPwbk/HipY2aCuBTS1K0QNd7T26BMrA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.19' (ECDSA) to the list of known hosts.
javier@192.168.1.19's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-122-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of sáb 21 sep 2024 10:19:15 UTC

System load:  0.0          Processes:      106
Usage of /:   44.5% of 11.21GB Users logged in:  1
Memory usage: 13%         IPv4 address for enp0s3: 192.168.1.19
Swap usage:   0%
```

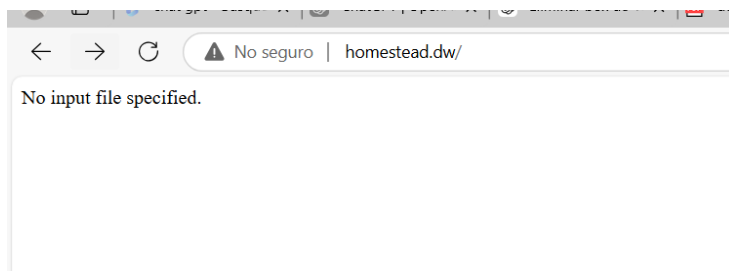
9. Modifica el fichero de hosts para asignar nombre y dirección ip del archivo de configuración Homestead.yaml.

En el archivo de host que esta en **Window/System32/driver/etc** lo que hacemos es agregar dicha **línea** donde ponemos la **ip** del archivo **homestead.yaml** y el **nombre** que le hayamos puesto.

```
# End of section
192.168.56.10 homestead.dw
```

10.Crea una página que muestre el mensaje “Hello Vagrant”

Otro **problema** que me ha surgido al llegar a este punto del **ejercicio** es que al entrar a la **página** me decía que no *podía encontrar el archivo*, esto se debe que yo tenía en la ruta de mi usuario una carpeta que era **code**, que lo asigno al .yaml pero tenía que **añadir** una **subcarpeta public** para que funcione.



Y dentro del subdirectorio añadimos un archivo **php** con un echo **hello word**

