

# ACTIVIDADES EVALUABLES:

## ACTIVIDAD 1:

1. Para instalar XAMPP deberemos irnos a la página oficial de descarga:  
<https://www.apachefriends.org/es/download.html>

### Descargar

XAMPP es una distribución de Apache fácil de instalar que contiene MariaDB, PHP y Perl. Simplemente descarga y ejecuta el instalador. ¡Es así de fácil! Instaladores creados usando InstallBuilder.

#### XAMPP para Windows 8.0.30, 8.1.25 & 8.2.12

Versión		Suma de comprobación		Tamaño
8.0.30 / PHP 8.0.30	¿Qué está incluido?.	md5 sha1	<a href="#">Descargar (64 bit)</a>	144 Mb
8.1.25 / PHP 8.1.25	¿Qué está incluido?.	md5 sha1	<a href="#">Descargar (64 bit)</a>	148 Mb
8.2.12 / PHP 8.2.12	¿Qué está incluido?.	md5 sha1	<a href="#">Descargar (64 bit)</a>	149 Mb

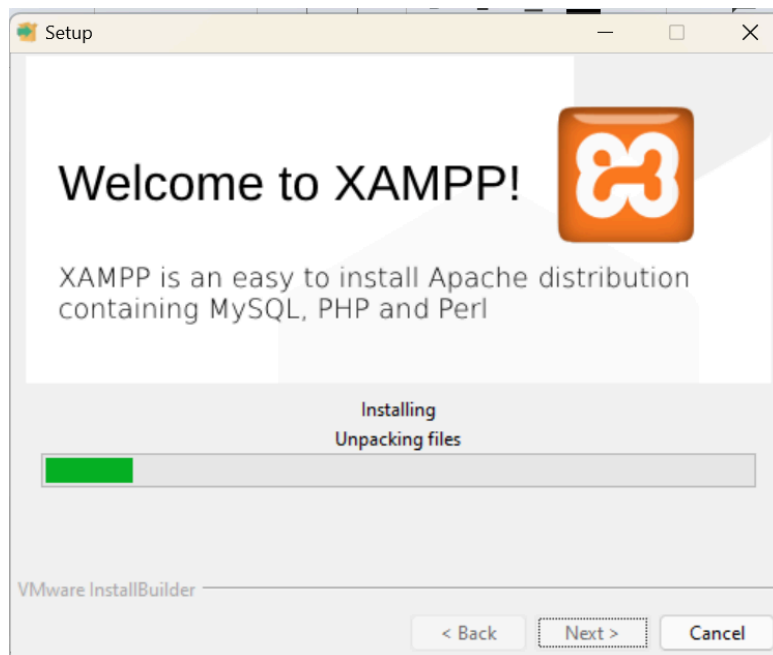
[Requisitos](#) [Más Descargas »](#)

#### Documentación/FAQs

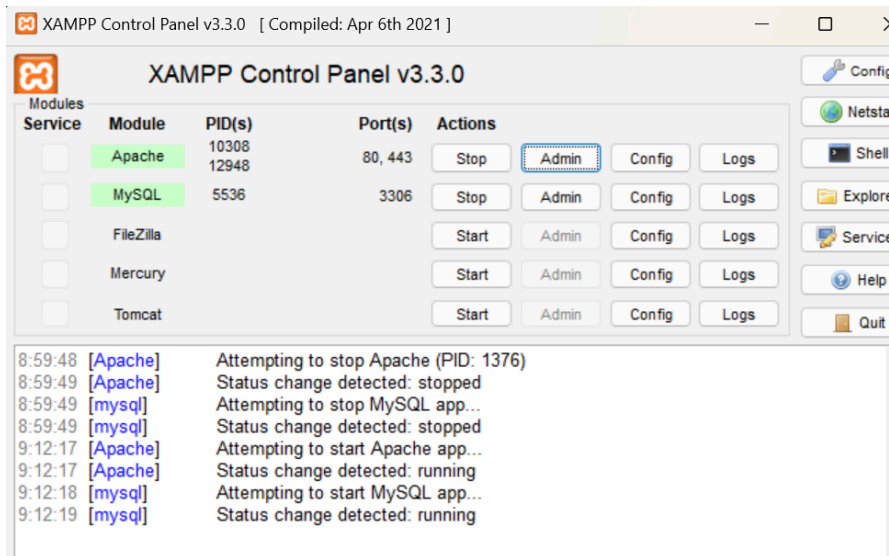
No hay un manual para XAMPP. Escribimos la documentación en forma de preguntas frecuentes (FAQs). ¿Tienes una pregunta que no está respondida? Prueba los Foros o Stack Overflow.

- [Linux Preguntas frecuentes](#)
- [Windows Preguntas frecuentes](#)
- [OS X Preguntas frecuentes](#)

Deberemos seleccionar la descarga compatible para nuestro sistema operativo, en mi caso se trata Windows. Seleccionamos la última versión de XAMPP. Una vez descargada la ejecutamos para su configuración.

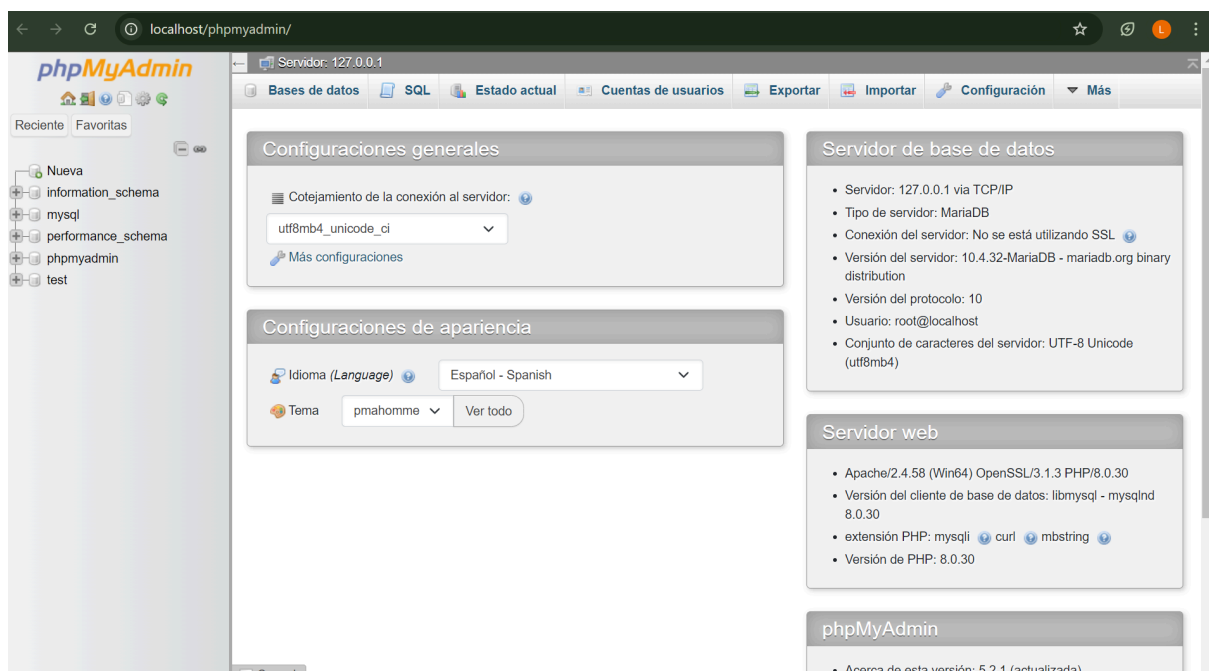


Una vez instalada se desplegará el panel de control deberemos activar los campos Apache para tener un servidor web y MySQL para poder acceder a las bases de datos.

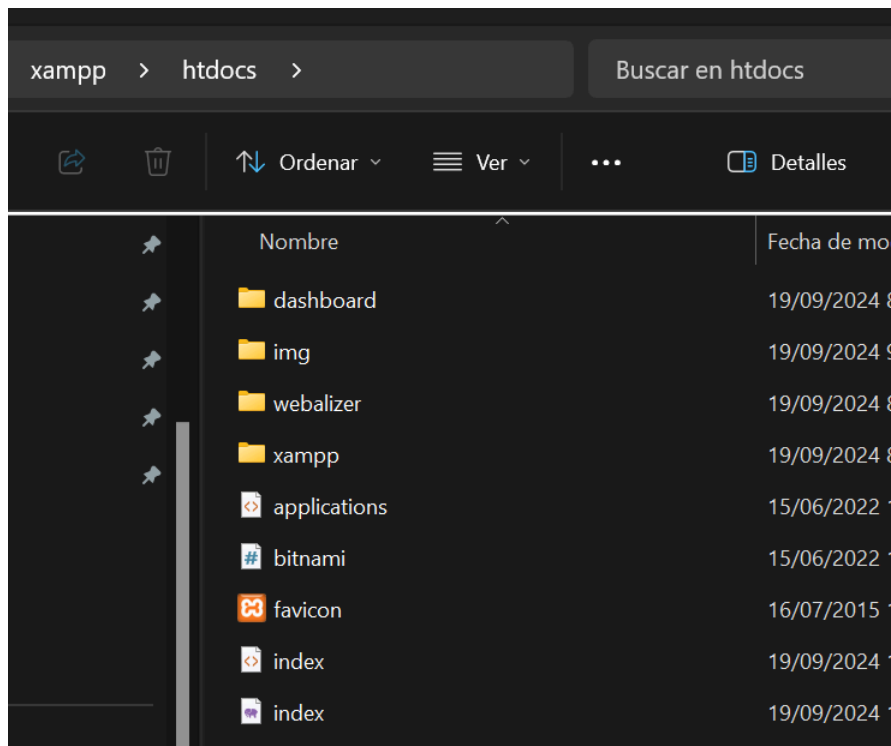


Accederemos a la ruta de nuestro servidor apache en este caso lo haremos de manera local por lo tanto la ruta es localhost:80.

2. Para acceder a phpMyAdmin simplemente deberemos tener activado el servicio de MySQL y dentro del panel de control de XAMPP pinchar sobre el botón( Admin), así nos redireccionará a phpMyAdmin dentro de nuestro servidor web.



3. Para crear nuestro portafolio deberemos acceder a la carpeta del programa XAMPP a la siguiente ruta C:\xampp\htdocs. Dentro de esta ruta crearemos un archivo de tipo html donde crearemos nuestro portafolio.



Accederemos de manera local con la ruta localhost/index.html



## ACTIVIDAD 2:

1. Descargamos el VirtualBox de la página oficial:  
<https://www.virtualbox.org/wiki/Downloads>



Crearemos una máquina virtual con Ubuntu Server e instalaremos dentro Apache, MariaDB, PHP, phpMyAdmin.

Aquí podemos observar como ya tenemos nuestro servidor ubuntu instalado:

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

lucia@lucia:~$
```

Para instalar apache deberemos escribir el siguiente comando en la terminal: “sudo apt install apache2”

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
lucia@lucia:~$ sudo apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1-sqlite3
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1-sqlite3
0 actualizados, 10 nuevos se instalarán, 0 para eliminar y 6 no actualizados.
Se necesita descargar 2.083 kB de archivos.
Se utilizarán 8.094 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://archive.ubuntu.com/ubuntu noble/main amd64 libapr1t64 amd64 1.7.2-3.1build2 [107 kB]
Des:2 http://archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ubuntu7 [91,9 kB]
Des:3 http://archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-1.1ubuntu7 [91,9 kB]
Des:4 http://archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.1ubuntu7 [91,9 kB]
Des:5 http://archive.ubuntu.com/ubuntu noble/main amd64 liblua5.4-0 amd64 5.4.6-3build2 [166 kB]
Des:6 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-1ubuntu8.4 [2,4 MB]
25% [6 apache2-bin 72 B/1.329 kB 0%]
```

Para instalar MariaDB deberemos poner el siguiente comando: “sudo apt install mariadb-server”.

```
Running kernel seems to be up-to-date.
Restarting services...
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
No containers need to be restarted.
User sessions running outdated binaries:
lucia @ user manager service: systemd[1193]
No VM guests are running outdated hypervisor (qemu) binaries on this host.
lucia@lucia:~$ _
```

Para ver que está correctamente instalado haremos el siguiente comando: “sudo systemctl status mariadb”

```
Running kernel seems to be up-to-date.
Restarting services...
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
No containers need to be restarted.
User sessions running outdated binaries:
lucia @ user manager service: systemd[1193]
No VM guests are running outdated hypervisor (qemu) binaries on this host.
lucia@lucia:~$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.11.8 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-09-20 07:24:06 UTC; 3min 3s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
  Main PID: 10242 (mariabdd)
    Status: "Taking your SQL requests now..."
     Tasks: 11 (limit: 30454)
  Memory: 78.8M (peak: 81.8M)
       CPU: 520ms
    CGroup: /system.slice/mariadb.service
            └─10242 /usr/sbin/mariabdd

sep 20 07:24:05 lucia mariabdd[10242]: 2024-09-20 7:24:05 0 [Note] InnoDB: log sequence number 46
sep 20 07:24:05 lucia mariabdd[10242]: 2024-09-20 7:24:05 0 [Note] Plugin 'FEEDBACK' is disabled.
sep 20 07:24:05 lucia mariabdd[10242]: 2024-09-20 7:24:05 0 [Note] InnoDB: Loading buffer pool(s)
sep 20 07:24:05 lucia mariabdd[10242]: 2024-09-20 7:24:05 0 [Warning] You need to use --log-bin t
sep 20 07:24:05 lucia mariabdd[10242]: 2024-09-20 7:24:05 0 [Note] Server socket created on IP:
sep 20 07:24:05 lucia mariabdd[10242]: 2024-09-20 7:24:05 0 [Note] InnoDB: Buffer pool(s) load co
sep 20 07:24:06 lucia mariabdd[10242]: 2024-09-20 7:24:06 0 [Note] /usr/sbin/mariabdd: ready for
```

Para instalar PHP debemos poner este comando: “sudo apt install php”

```
lucia@lucia:~$ sudo apt install php
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libapache2-mod-php8.3 php-common php8.3 php8.3-cli php8.3-common php8.3-opcache php8.3-readline
Paquetes sugeridos:
  php-pear
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-php8.3 php php-common php8.3 php8.3-cli php8.3-common php8.3-opcache php8.3-readlin
0 actualizados, 8 nuevos se instalarán, 0 para eliminar y 6 no actualizados.
Se necesita descargar 4.914 kB de archivos.
Se utilizarán 22,4 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Comprobamos que esté instalado correctamente con este comando: “php -v”

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
lucia@lucia:~$ php -v
PHP 8.3.6 (cli) (built: Jun 13 2024 15:23:20) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
lucia@lucia:~$
```

Para instalar phpMyAdmin usaremos el siguiente comando: “sudo apt install phpmyadmin”

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
lucia@lucia:~$ php -v
PHP 8.3.6 (cli) (built: Jun 13 2024 15:23:20) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
lucia@lucia:~$ sudo apt install phpmyadmin
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  dbconfig-common dbconfig-mysql icc-profiles-free javascript-common libjs-bootstrap5 libjs-codemirror
  libjs-jquery-mousewheel libjs-jquery-tablesorter libjs-jquery-timepicker libjs-jquery-ui libjs-popover
  libmbedtls libzip4t64 node-jquery php-bz2 php-composer-ca-bundle php-curl php-fig-http-message-util
  php-mariadb-mysql-kbs php-mbstring php-mcrypt php-mysql php-nikic-fast-route php-phpmyadmin-motranslator
  php-psr-cache php-psr-container php-psr-http-factory php-psr-http-message php-psr-log php-slim-psr
  php-symfony-config php-symfony-dependency-injection php-symfony-deprecation-contracts php-symfony-serializer
  php-symfony-polyfill-php80 php-symfony-service-contracts php-symfony-var-exporter php-tcpdf php-twig
  php-zip php8.3-bz2 php8.3-curl php8.3-gd php8.3-mbstring php8.3-mcrypt php8.3-mysql php8.3-xml php8.3-zip
Paquetes sugeridos:
  bootstrap-icons libjs-bootstrap5-doc libjs-requirejs libjs-jquery-ui-docs libmbedtls-dev mcrypt php-composer
  www-browser php-gd2 php-pragmarx-google2fa-qrcode php-bacon-qrcode php-code-lts-u2f-php-server
Se instalarán los siguientes paquetes NUEVOS:
  dbconfig-common dbconfig-mysql icc-profiles-free javascript-common libjs-bootstrap5 libjs-codemirror
  libjs-jquery-mousewheel libjs-jquery-tablesorter libjs-jquery-timepicker libjs-jquery-ui libjs-popover
  libmbedtls libzip4t64 node-jquery php-bz2 php-composer-ca-bundle php-curl php-fig-http-message-util
  php-mariadb-mysql-kbs php-mbstring php-mcrypt php-mysql php-nikic-fast-route php-phpmyadmin-motranslator
  php-psr-cache php-psr-container php-psr-http-factory php-psr-http-message php-psr-log php-slim-psr
  php-symfony-config php-symfony-dependency-injection php-symfony-deprecation-contracts php-symfony-serializer
  php-symfony-polyfill-php80 php-symfony-service-contracts php-symfony-var-exporter php-tcpdf php-twig
  php-zip php8.3-bz2 php8.3-curl php8.3-gd php8.3-mbstring php8.3-mcrypt php8.3-mysql php8.3-xml php8.3-zip
0 actualizados, 65 nuevos se instalarán, 0 para eliminar y 6 no actualizados.
Se necesita descargar 17,9 MB de archivos.
Se utilizarán 85,9 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] _
```

3. Para conectar nuestro ordenador a nuestro servidor ubuntu usaremos el protocolo SSH para ello. Debemos instalar SSH con el siguiente comando: “sudo apt openssh-server”.

En mi caso ya está instalado porque al instalar el servidor lo he instalado también:

```
lucia@lucia:~$ sudo apt install openssh-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
openssh-server ya está en su versión más reciente (1:9.6p1-3ubuntu13.5).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 6 no actualizados.
```

Comprobamos su estado con el siguiente comando, en mi caso está inactivo: “sudo systemctl status ssh”

```
lucia@lucia:~$ sudo systemctl status ssh
* ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
   Active: inactive (dead)
 TriggeredBy: ● ssh.socket
             Docs: man:sshd(8)
                  man:sshd_config(5)
lucia@lucia:~$
```

Para activarlo deberemos usar este comando: “sudo systemctl start ssh”

```
lucia@lucia:~$ sudo systemctl start ssh
lucia@lucia:~$ _
```

Comprobamos su estado de nuevo:

```
lucia@lucia:~$ sudo systemctl status ssh
* ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
   Active: inactive (dead)
TriggeredBy: ● ssh.socket
   Docs: man:sshd(8)
        man:sshd_config(5)
lucia@lucia:~$ sudo systemctl start ssh
sudo: systemctl: command not found
lucia@lucia:~$ sudo systemctl start ssh
lucia@lucia:~$ sudo systemctl status ssh
* ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
   Active: active (running) since Fri 2024-09-20 07:47:40 UTC; 27s ago
TriggeredBy: ● ssh.socket
   Docs: man:sshd(8)
        man:sshd_config(5)
   Process: 22014 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 22015 (sshd)
    Tasks: 1 (limit: 4614)
   Memory: 2.0M (peak: 2.3M)
      CPU: 29ms
   CGroup: /system.slice/ssh.service
           └─22015 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

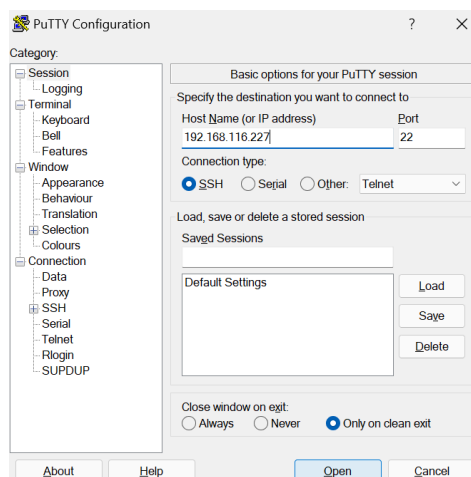
sep 20 07:47:40 lucia systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
sep 20 07:47:40 lucia sshd[22015]: Server listening on :: port 22.
sep 20 07:47:40 lucia systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
lucia@lucia:~$ _
```

Deberemos saber la direccion IP de nuestro servidor:

```
lucia@lucia:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:61:ee:6a brd ff:ff:ff:ff:ff:ff
    inet 192.168.116.227/24 metric 100 brd 192.168.116.255 scope global dynamic enp0s3
        valid_lft 4930sec preferred_lft 4930sec
    inet6 fe80::a00:27ff:fe61:ee6a/64 scope link
        valid_lft forever preferred_lft forever
```

Como podemos ver nuestra IP es 192.168.116.227

Instalamos putty para conectar nuestro equipo al servidor, una vez instalada ejecutamos e introducimos la dirección IP del servidor junto con el puerto que por norma general es 22.



Conectamos y deberemos introducir las credenciales de usuario. En este caso tenemos como usuario y contraseña lucia.

```
192.168.116.227 - PuTTY
login as: lucia
lucia@192.168.116.227's password: [ ]
```

Ya estamos conectados al servidor ubuntu.

```
lucia@lucia: ~
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of vie 20 sep 2024 07:58:41 UTC

System load: 0.0 Processes: 124
Usage of /: 44.4% of 11.21GB Users logged in: 1
Memory usage: 8% IPv4 address for enp0s3: 192.168.116.227
Swap usage: 0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

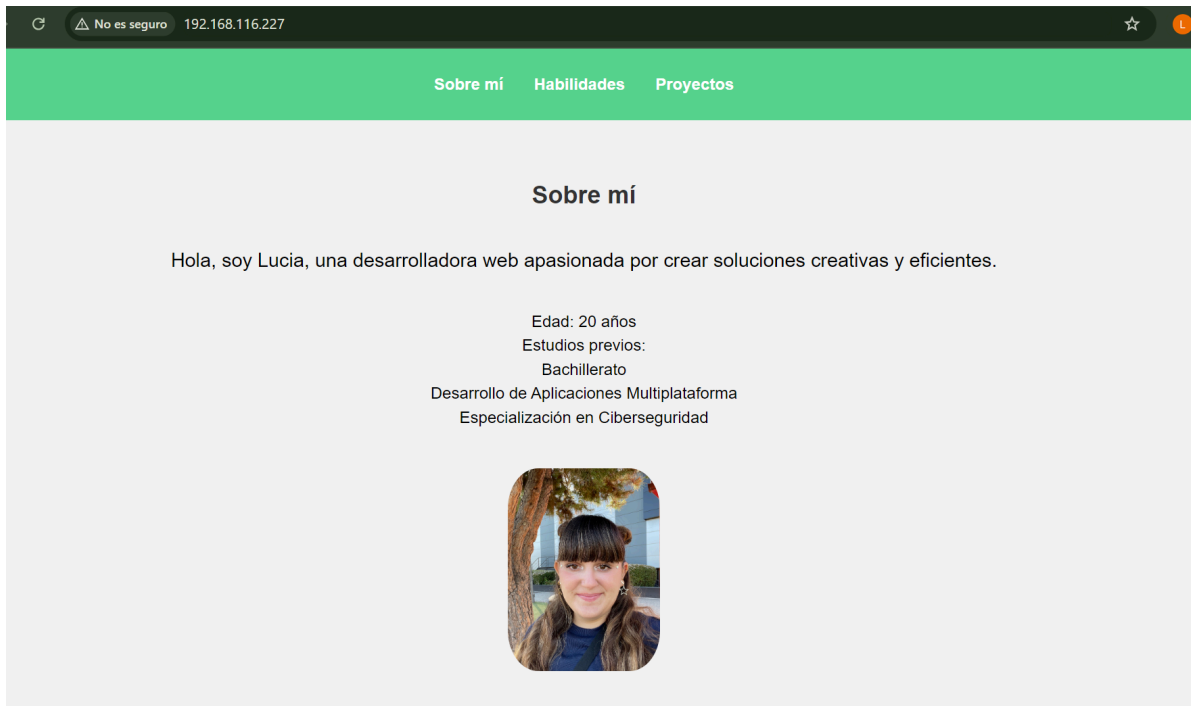
https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado
Se pueden aplicar 0 actualizaciones de forma inmediata.

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

lucia@lucia:~$
```

Una vez conectados, si ponemos en el navegador la dirección IP del servidor obtendremos nuestro portfolio.





### ACTIVIDAD 3:

Lo primero que tenemos que hacer es descargar Vagrant de la página oficial y comprobar la versión que tenemos instalada:

Página oficial: [https://developer.hashicorp.com/vagrant/install?product\\_intent=vagrant](https://developer.hashicorp.com/vagrant/install?product_intent=vagrant)

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

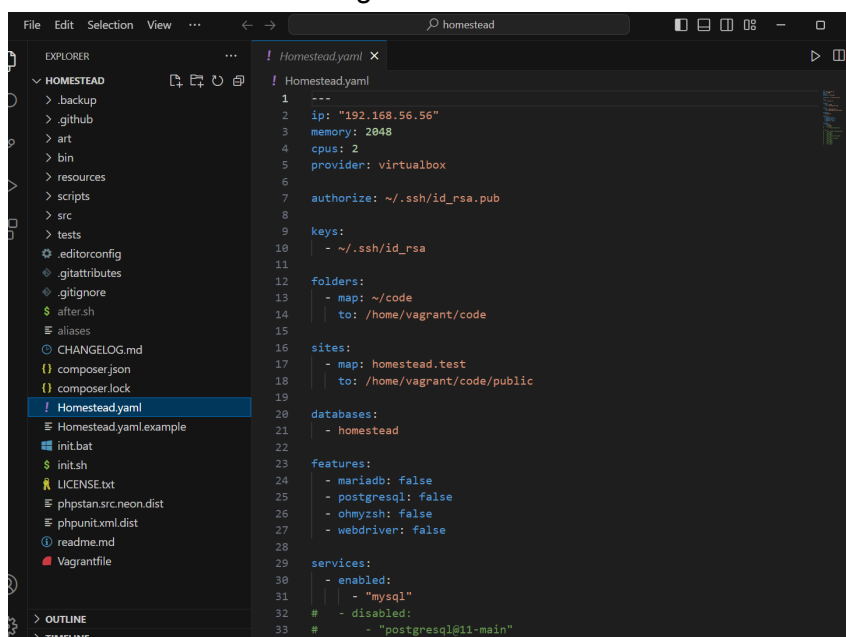
Después deberemos clonar el repositorio: <https://github.com/laravel/homestead.git>

```
C:\Users\lucia>git clone https://github.com/laravel/homestead.git
Cloning into 'homestead'...
remote: Enumerating objects: 6652, done.
remote: Counting objects: 100% (429/429), done.
remote: Compressing objects: 100% (198/198), done.
remote: Total 6652 (delta 278), reused 348 (delta 227), pack-reused 6223 (from 1)
Receiving objects: 100% (6652/6652), 1.65 MiB | 4.05 MiB/s, done.
Resolving deltas: 100% (4197/4197), done.
```

Accedemos a la carpeta homestead y deberemos iniciar el repositorio con el archivo init.bat. Esto generará un archivo de configuración llamado Homestead.yaml.

```
C:\Users\lucia\homestead>init.bat
1 archivo(s) copiado(s).
1 archivo(s) copiado(s).
1 archivo(s) copiado(s).
Homestead initialized!
```

Abrimos el archivo de configuración en un editor de texto:

The screenshot shows a code editor with a dark theme. On the left, there is an 'EXPLORER' sidebar showing a file tree for a project named 'HOMESTEAD'. The tree includes folders like '.backup', '.github', 'art', 'bin', 'resources', 'scripts', 'src', 'tests', and files like '.editorconfig', '.gitattributes', '.gitignore', 'after.sh', 'aliases', 'CHANGELOG.md', 'composer.json', 'composer.lock', 'Homestead.yaml', 'Homestead.yaml.example', 'init.bat', 'init.sh', 'LICENSE.txt', 'phpstan.neon.dist', 'phpunit.xml.dist', 'readme.md', and 'Vagrantfile'. The 'Homestead.yaml' file is selected and highlighted. The main editor area displays the content of 'Homestead.yaml', which is a YAML configuration file. The file starts with '---' and contains settings for IP, memory, CPUs, provider (virtualbox), SSH keys, folders, sites, databases, features, and services. The 'services' section shows 'mysql' as enabled and 'pgsql' as disabled. The file ends with '---'.

Deberemos cambiar la ruta de authorize por : private\_key: C:\Users\lucia\.ssh\id\_rsa

```
authorize:
| private_key: C:\Users\lucia\.ssh\id_rsa
```

Deberemos añadir a nuestro equipo la funcionalidad de openSSH desde configuración > aplicaciones > funciones especiales.

Una vez activado ya podremos crear la clave ssh que nos pide para poder conectarnos.

```
C:\Users\lucia>ssh-keygen -t rsa -C "lucia44ee@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\lucia\.ssh\id_rsa):
Created directory 'C:\Users\lucia\.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\lucia\.ssh\id_rsa
Your public key has been saved in C:\Users\lucia\.ssh\id_rsa.pub
The key fingerprint is:
SHA256:vQgLRxgGjVzikReaUUqStFbTM2p0CF7/3QPAphHYb58 lucia44ee@gmail.com
The key's randomart image is:
+---[RSA 3072]-----+
|o=B@@oo.          |
|o+B%=X o.         |
|+*+.0 .          |
|. o o.o..o        |
|. . +.S.oo        |
|  o o E ..        |
|  . . .           |
+-----[SHA256]-----+
```

En la ubicación C:\users\lucia\.ssh\id\_rsa se han creado dos archivos: id\_rsa (clave privada) / id\_rsa.pub (clave pública).

Al intentar levantar vagrant nos sale un error.

```
C:\Users\lucia>vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'base' could not be found. Attempting to find and install...
default: Box Provider: virtualbox
default: Box Version: >= 0
==> default: Box file was not detected as metadata. Adding it directly...
==> default: Adding box 'base' (v0) for provider: virtualbox
default: Downloading: base
default:
An error occurred while downloading the remote file. The error
message, if any, is reproduced below. Please fix this error and try
again.

Couldn't open file C:/Users/lucia/base
```

Donde no detecta un la caja de Homestead, para arreglarlo debemos acceder a la carpeta de homestead al fichero vagrantfile donde deberemos añadir la siguiente linea: config.vm.box="laravel/homestead".

```
Vagrant.require_version '>= 2.4'

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.box = "laravel/homestead"

  if File.exist? aliasesPath then
    config.vm.provision "file", source: aliasesPath, destination: "/tmp/bash_aliases"
    config.vm.provision "handle_aliases", type: "shell" do |s|
      s.inline = "awk '{ sub(\"\\r$\", \"\\n\"); print }' /tmp/bash_aliases > /home/vagrant/.bash_aliases"
    end
  end

  if File.exist? homesteadYamlPath then
    settings = YAML::load(File.read(homesteadYamlPath))
  elsif File.exist? homesteadJsonPath then
    settings = JSON::parse(File.read(homesteadJsonPath))
  else
    abort "Homestead settings file not found in #{confDir}"
  end
end
```

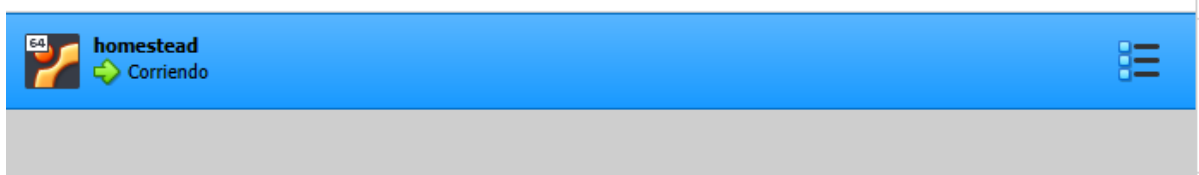
Una vez guardado y actualizado iniciamos vagrant.

Seguía dando error y es porque estaba cogiendo tanto la clave pública como la privada, y en el archivo homestead.rb y este procesa una cadena y no un array para solucionarlo debemos cambiar la ruta en el archivo homestead.yaml

```
authorize: C:\\Users\\lucia\\.ssh\\id_rsa.pub
keys:
  - C:\\Users\\lucia\\.ssh\\id_rsa
```

Tenemos que coger la pública y quitarle el guión. Una vez cambiado arrancamos vagrant.

```
C:\Users\lucia\Homestead>vagrant up
Bringing machine 'homestead' up with 'virtualbox' provider...
==> homestead: Importing base box 'laravel/homestead'...
==> homestead: Matching MAC address for NAT networking...
==> homestead: Checking if box 'laravel/homestead' version '14.0.2' is up to date...
==> homestead: Setting the name of the VM: homestead
==> homestead: Clearing any previously set network interfaces...
==> homestead: Preparing network interfaces based on configuration...
homestead: Adapter 1: nat
homestead: Adapter 2: hostonly
==> homestead: Forwarding ports...
homestead: 80 (guest) => 8000 (host) (adapter 1)
homestead: 443 (guest) => 44300 (host) (adapter 1)
homestead: 22 (guest) => 2222 (host) (adapter 1)
==> homestead: Running 'pre-boot' VM customizations...
==> homestead: Booting VM...
==> homestead: Waiting for machine to boot. This may take a few minutes...
homestead: SSH address: 127.0.0.1:2222
```



Una vez montada la máquina virtual nos conectaremos a ella por ssh. Solo tendremos que poner en el navegador de nuestro equipo la dirección ip de la máquina (192.168.56.56) y nos mostrará que no hay nada. Pero ya la tenemos instalada.



No input file specified.