



LARAVEL A TRAVÉS DE VAGRANT Y XAMPP

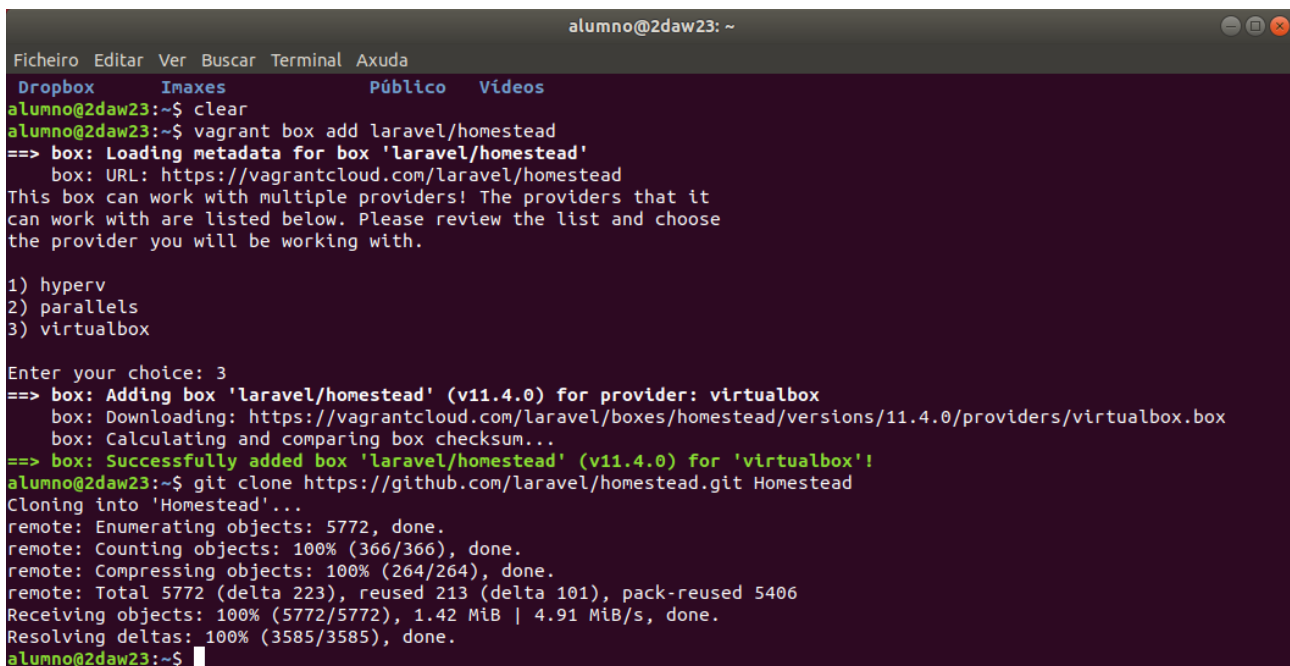
Prueba de servidor en Ubuntu y
Windows 10

Rodríguez Jácome, David
Desenvolvimento Web en Contorno Servidor
2º DAW

1. Instalación de Laravel en Vagrant con Ubuntu

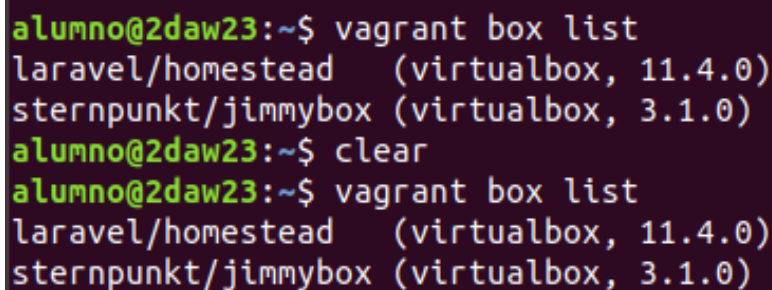
En la siguiente tarea instalaremos y configuraremos el framework de PHP Laravel usando los programas Vagrant (instalado en un sistema Ubuntu) y XAMPP, este último montado en un sistema virtualizado Windows 10.

Partimos de la base de que ya tenemos instalado Vagrant en nuestro PC. Abrimos una terminal, nos situamos en el directorio raíz y escribimos el comando “vagrant box add laravel/homestead” para añadir el box laravel/homestead configurando como proveedor de máquinas virtuales VirtualBox. Seguidamente, usaremos el comando “git clone https://github.com/laravel/homestead.git Homestead” para clonar desde GitHub este repositorio Homestead.



```
alumno@2daw23: ~  
Ficheiro Editar Ver Buscar Terminal Axuda  
Dropbox Imaxes Público Vídeos  
alumno@2daw23:~$ clear  
alumno@2daw23:~$ vagrant box add laravel/homestead  
==> box: Loading metadata for box 'laravel/homestead'  
box: URL: https://vagrantcloud.com/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) hyperv  
2) parallels  
3) virtualbox  
  
Enter your choice: 3  
==> box: Adding box 'laravel/homestead' (v11.4.0) for provider: virtualbox  
box: Downloading: https://vagrantcloud.com/laravel/boxes/homestead/versions/11.4.0/providers/virtualbox.box  
box: Calculating and comparing box checksum...  
==> box: Successfully added box 'laravel/homestead' (v11.4.0) for 'virtualbox'!  
alumno@2daw23:~$ git clone https://github.com/laravel/homestead.git Homestead  
Cloning into 'Homestead'...  
remote: Enumerating objects: 5772, done.  
remote: Counting objects: 100% (366/366), done.  
remote: Compressing objects: 100% (264/264), done.  
remote: Total 5772 (delta 223), reused 213 (delta 101), pack-reused 5406  
Receiving objects: 100% (5772/5772), 1.42 MiB | 4.91 MiB/s, done.  
Resolving deltas: 100% (3585/3585), done.  
alumno@2daw23:~$
```

Cuando terminen ambos procesos, podemos comprobar que el box está correctamente instalado con el comando “vagrant box list” y aparecerá que acabamos de agregar.

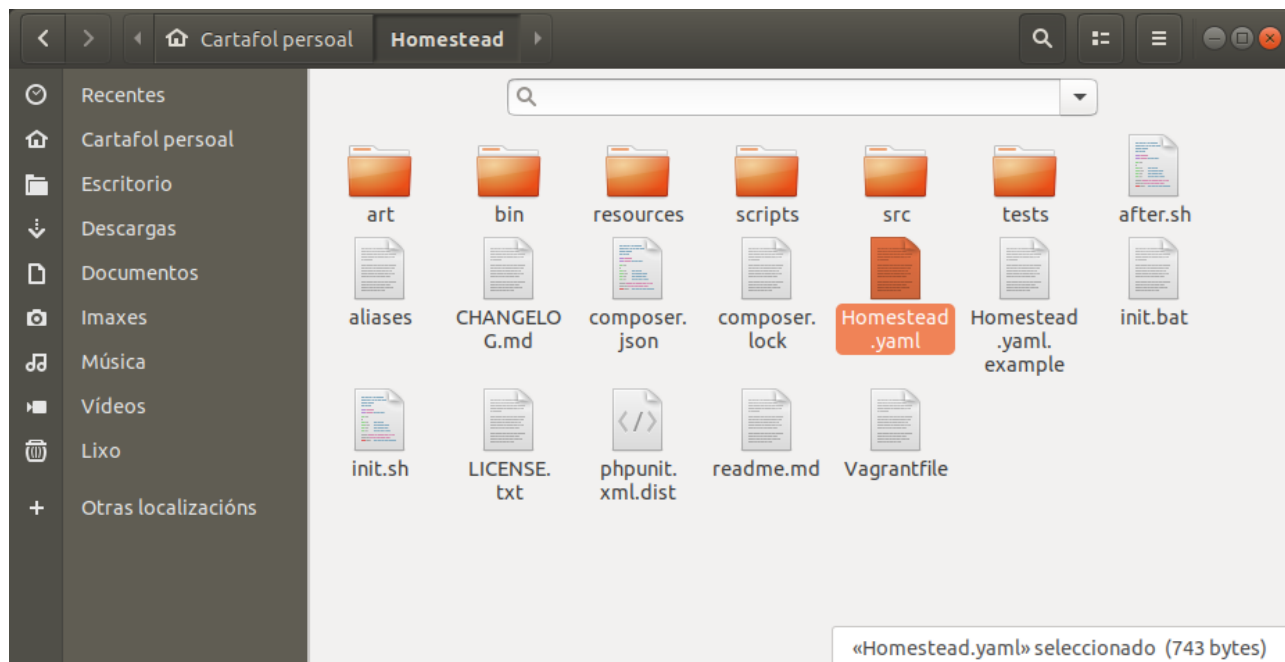


```
alumno@2daw23:~$ vagrant box list  
laravel/homestead (virtualbox, 11.4.0)  
sternpunkt/jimmybox (virtualbox, 3.1.0)  
alumno@2daw23:~$ clear  
alumno@2daw23:~$ vagrant box list  
laravel/homestead (virtualbox, 11.4.0)  
sternpunkt/jimmybox (virtualbox, 3.1.0)
```

El siguiente paso será ejecutar el comando “ssh-keygen -t rsa -C “xxxxx@email.com” (cambiando convenientemente el correo electrónico por uno propio donde corresponde).

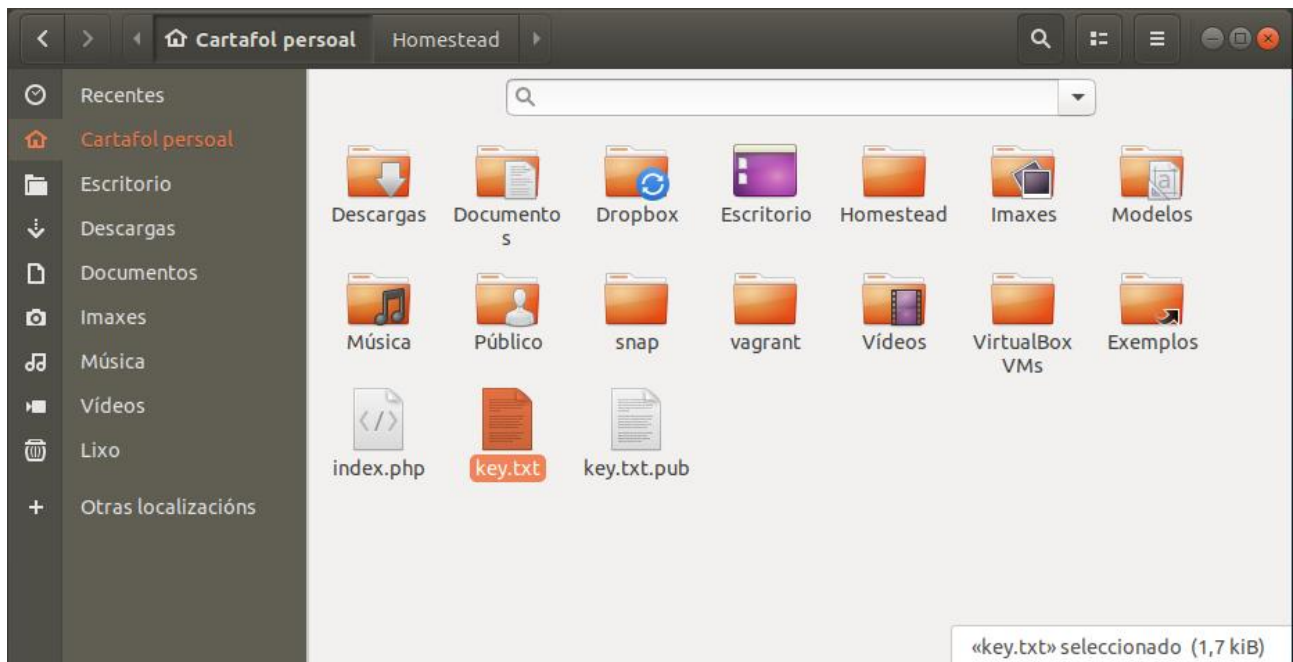
```
alumno@2daw23:~$ ssh-keygen -t rsa -C "jacodaro@hotmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alumno/.ssh/id_rsa): key.txt
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in key.txt.
Your public key has been saved in key.txt.pub.
The key fingerprint is:
SHA256:13x+K62FwUh0BykxMYLh9Byr/qI1AIL1BUyCS3tKeD0 jacodaro@hotmail.com
The key's randomart image is:
+---[RSA 2048]---+
| .00+=+. 0+.. |
|00.0+.0.0+... |
|++.. .+. .0 . |
|000.E . +0+ |
|..0. 0 S .000. |
| . 0 . 00 |
| + .0.. |
| ..0 ..0. |
| .. .. .0. |
+---[SHA256]-----+
```

Ahora nos situamos en el directorio *Homestead* y modificamos el archivo *Homestead.yaml* que se encuentra por defecto en esta carpeta con un editor (en este caso usaremos Visual Studio Code):

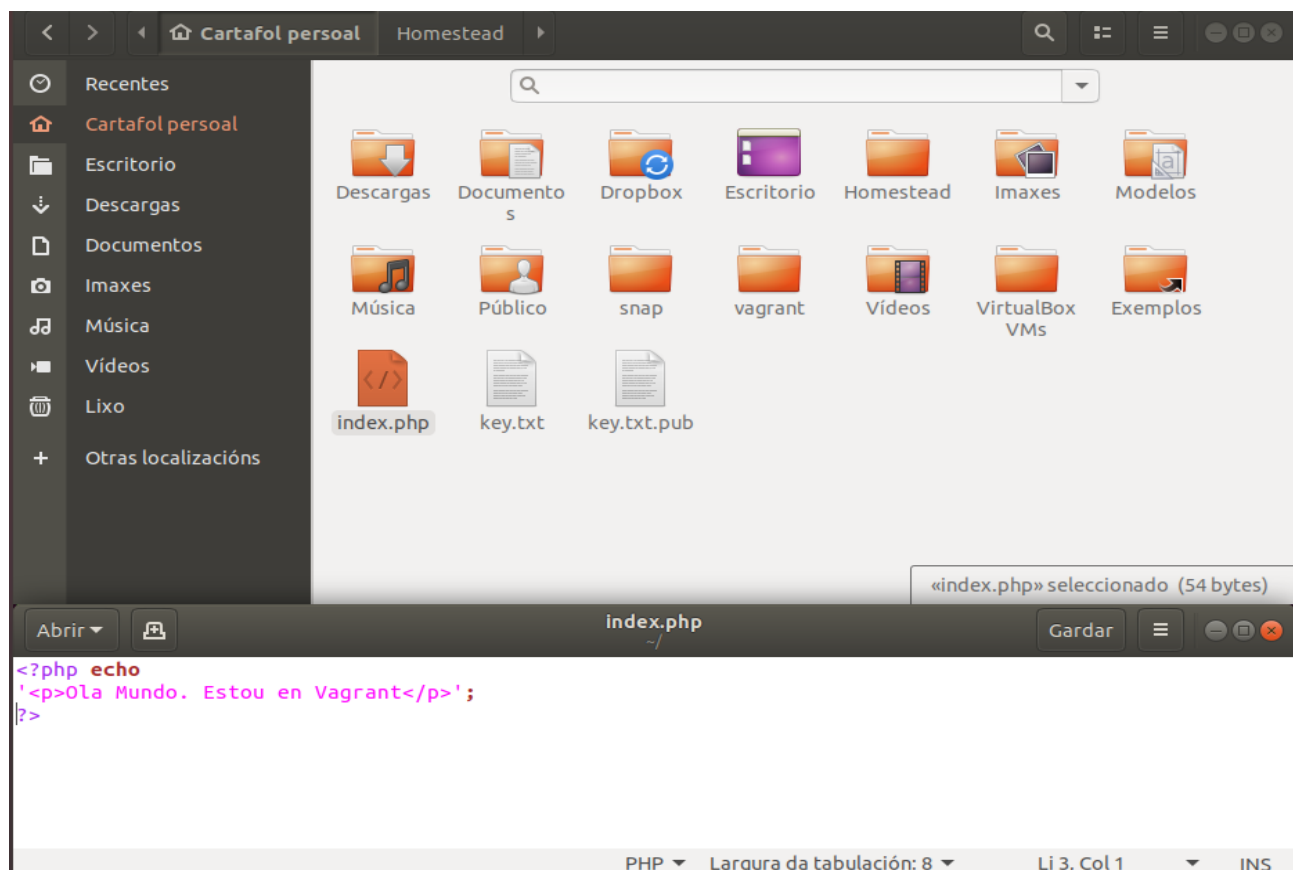


```
alumno@2daw23:~$ cd Homestead/
alumno@2daw23:~/Homestead$ code Homestead.yaml
```

Necesitaremos crear un archivo adicional en el directorio *Homestead* llamado “key.txt”, ya que no se encuentra por defecto en dicha carpeta y que es necesario debido al comando anterior “ssh-keygen -t rsa -C “xxxxx@email.com” y que está referenciado en el archivo *Homestead.yaml*, como veremos a continuación.

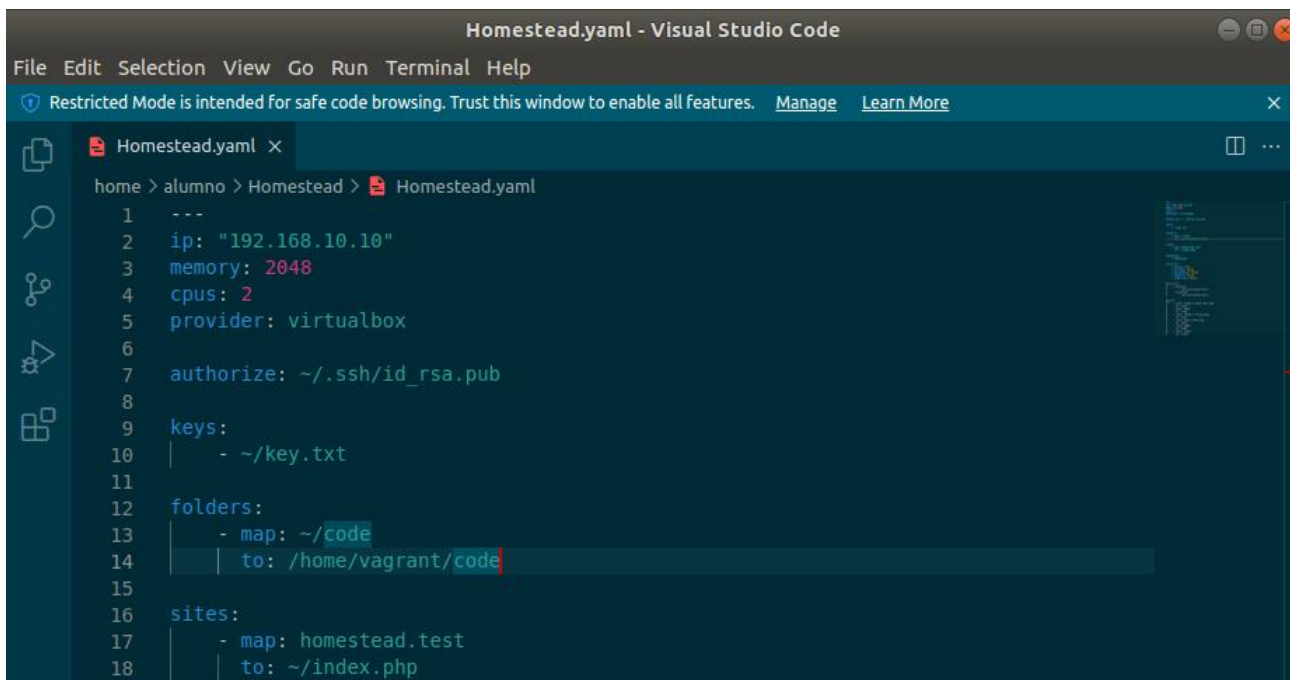
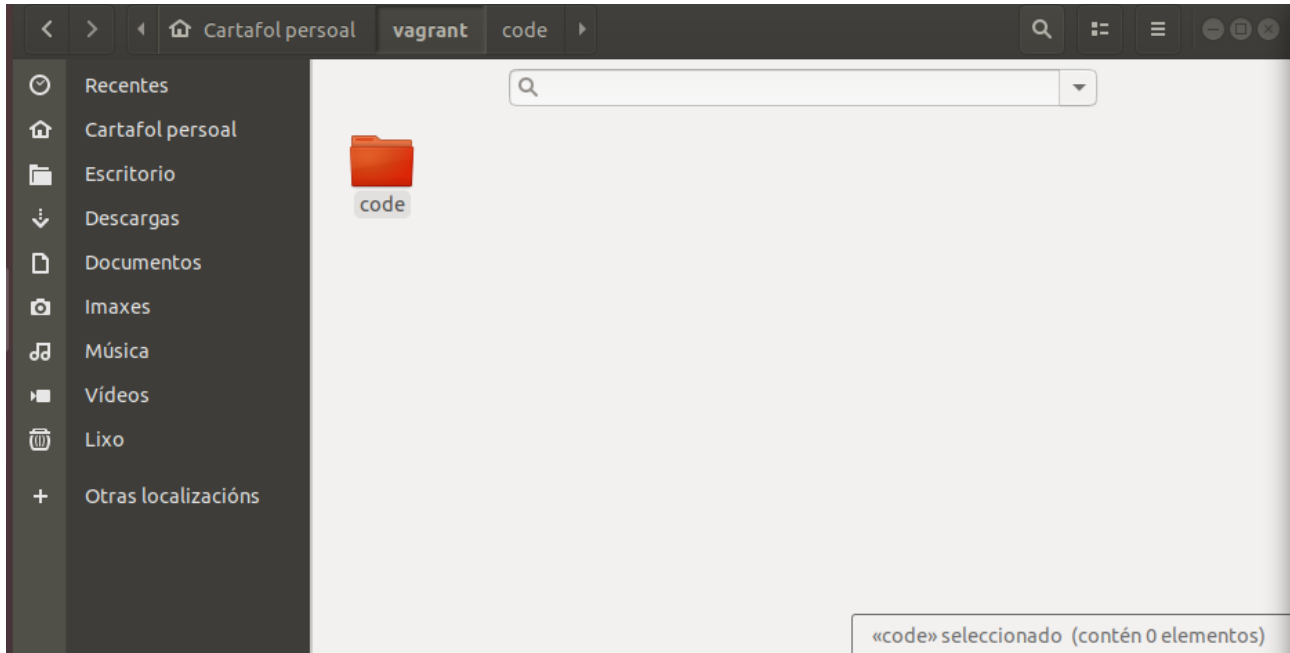


En VS Code vemos la línea 9 “Keys:”, donde está la referencia “~/key.txt” de antes. Además, necesitamos modificar la línea 18 y escribir “~/index.php”, que referenciará a un archivo llamado



“index.php” que crearemos en el directorio *Homestead*, y que modificaremos con un editor de texto donde escribiremos el mensaje “Hola Mundo. Estoy en Vagrant” en PHP.

Adicionalmente, tenemos que crear una carpeta “Code” en el directorio “vagrant”, para que no haya error en la referencia que se encuentra en la línea 14 del archivo *Homestead.yaml*.

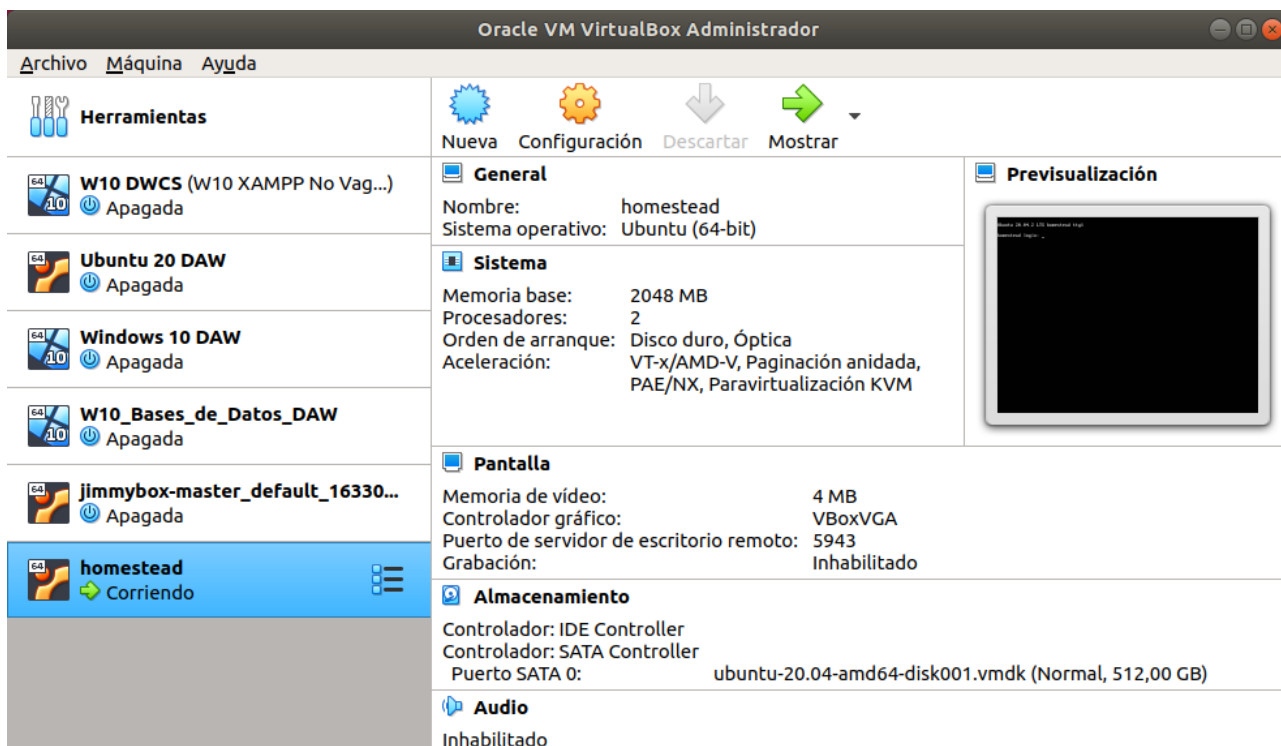


Ahora abrimos una terminal, nos situamos en el directorio *Homestead* e iniciamos el box con “vagrant up”.

```
alumno@2daw23:~/Homestead$ vagrant up
Bringing machine 'homestead' up with 'virtualbox' provider...
==> homestead: Checking if box 'laravel/homestead' version '11.4.0' is up to date...
==> homestead: Clearing any previously set forwarded ports...
==> 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
homestead: SSH username: vagrant
homestead: SSH auth method: private key
==> homestead: Machine booted and ready!
==> homestead: Checking for guest additions in VM...
==> homestead: Setting hostname...
==> homestead: Configuring and enabling network interfaces...
==> homestead: Mounting shared folders...
homestead: /vagrant => /home/alumno/Homestead
==> homestead: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> homestead: flag to force provisioning. Provisioners marked to run always will still run.
```

Una vez terminado el proceso, haremos una prueba de conexión con “vagrant ssh” y podemos comprobar que la máquina Vagrant está funcionando:

```
alumno@2daw23:~/Homestead$ vagrant ssh
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-81-generic x86_64)
```



Ahora, nos situaremos en la carpeta “Code” y creamos un nuevo proyecto de Laravel con el comando “composer create-project laravel/laravel demo”, lo ejecutamos y esperamos a que termine el proceso:

```
vagrant@homestead:~$ mkdir Code
vagrant@homestead:~$ ls
Code
vagrant@homestead:~$ cd Code/
vagrant@homestead:~/Code$ composer create-project laravel/laravel demo
Creating a "laravel/laravel" project at "./demo"
Installing laravel/laravel (v8.6.3)
- Downloading laravel/laravel (v8.6.3)
- Installing laravel/laravel (v8.6.3): Extracting archive
Created project in /home/vagrant/Code/demo
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking asm89/stack-cors (v2.0.3)
- Locking brick/math (0.9.3)
- Locking dflydev/dot-access-data (v3.0.1)
- Locking doctrine/inflector (2.0.3)
- Locking doctrine/instantiator (1.4.0)
- Locking doctrine/lexer (1.2.1)
- Locking dragonmantank/cron-expression (v3.1.0)
- Locking egulias/email-validator (2.1.25)
- Locking facade/flare-client-php (1.9.1)
- Locking facade/ignition (2.14.0)
- Locking facade/ignition-contracts (1.0.2)
- Locking fakerphp/faker (v1.16.0)
- Locking filp/whoops (2.14.4)
- Locking fruitcake/laravel-cors (v2.0.4)
- Locking graham-campbell/result-type (v1.0.2)
- Locking guzzlehttp/guzzle (7.3.0)
- Locking guzzlehttp/promises (1.4.1)
- Locking guzzlehttp/psr7 (2.0.0)
```

```
Package manifest generated successfully.
76 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi
No publishable resources for tag [laravel-assets].
Publishing complete.
> @php artisan key:generate --ansi
Application key set successfully.
vagrant@homestead:~/Code$ exit
logout
Connection to 127.0.0.1 closed.
alumno@2daw23:~/Homestead$
```

El último paso sería modificar el archivo “hosts” en el directorio */etc* en el sistema anfitrión de Ubuntu, pero por desgracia no tenemos permisos de administrador para modificar este archivo de sistema. Es por eso por lo que no podremos sacar por pantalla nuestro mensaje “Ola Mundo. Estou en Vagrant”.

2. Instalación de Laravel con XAMPP y Composer en sistema virtualizado Windows 10.

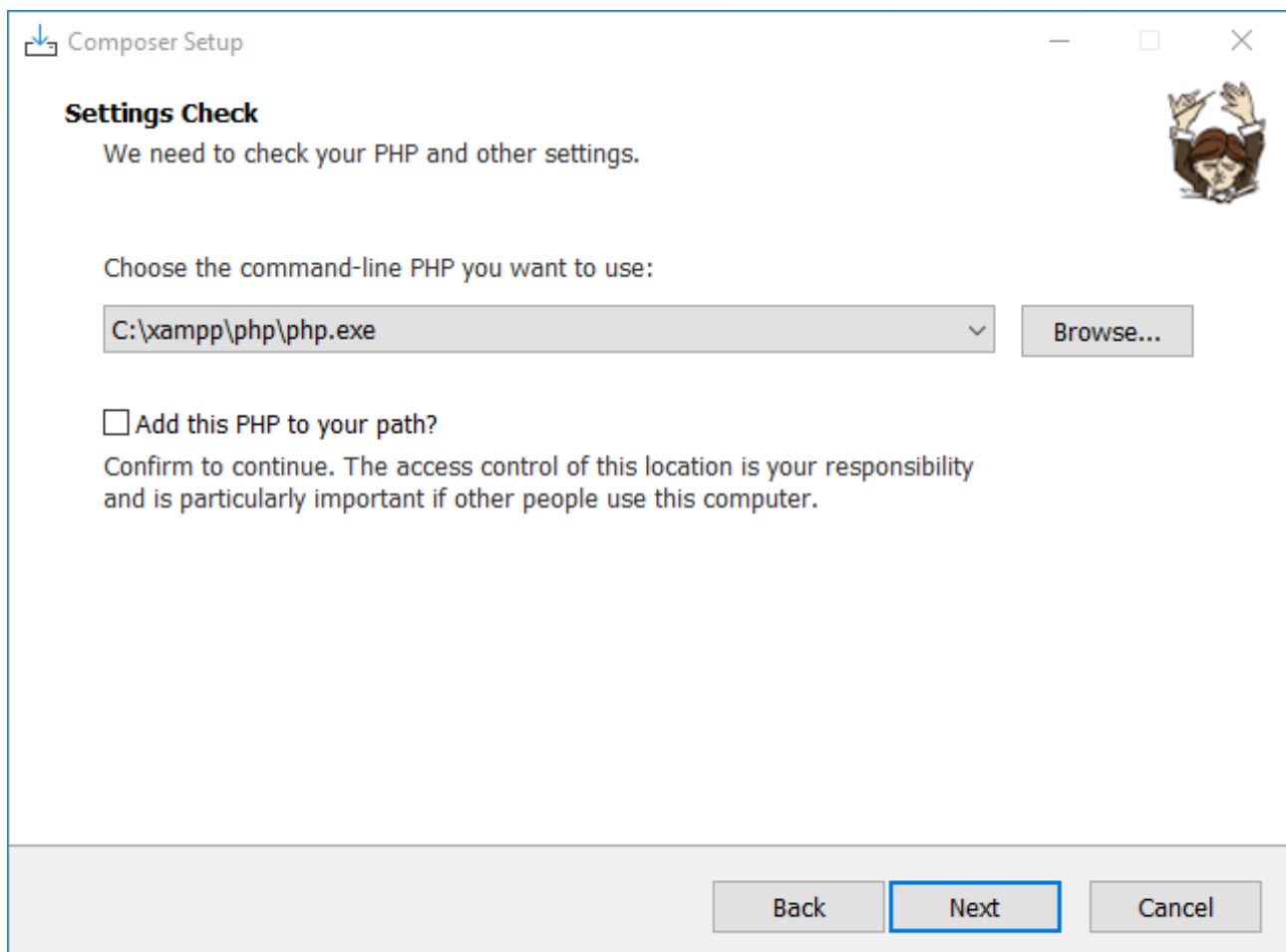
En este apartado veremos cómo replicar el proceso anterior, pero usando esta vez una máquina virtual de Windows 10. Partiremos de la base de que ya tenemos instalado XAMPP en Windows. En primer lugar, descargaremos *Composer* y procederemos a su instalación cumplimentando todos los pasos requeridos:

Download Composer Latest: v2.1.9

Windows Installer

The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.



Select Destination Location

Where should Composer be installed?



Setup will install Composer into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

At least 3,2 MB of free disk space is required.

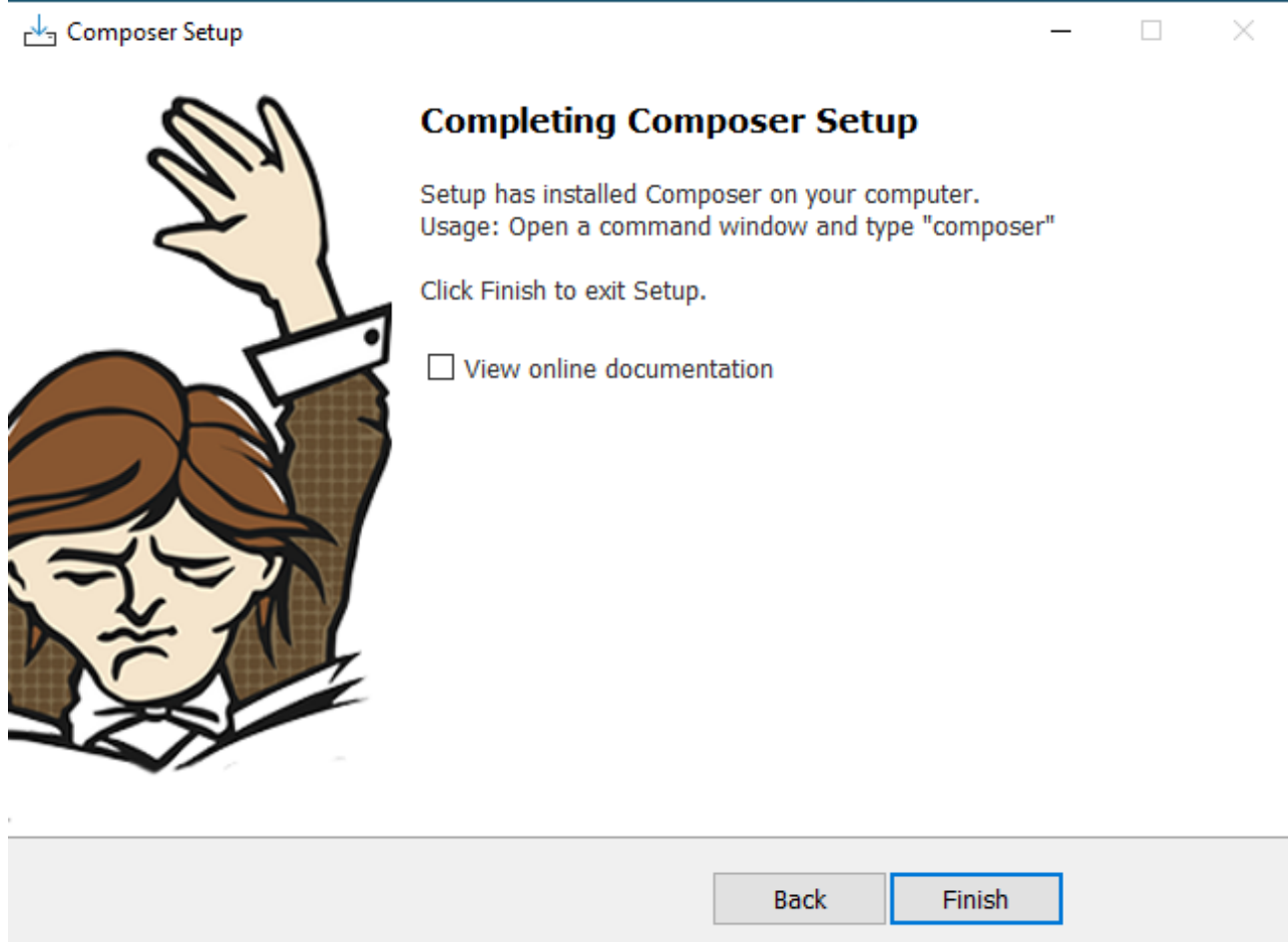
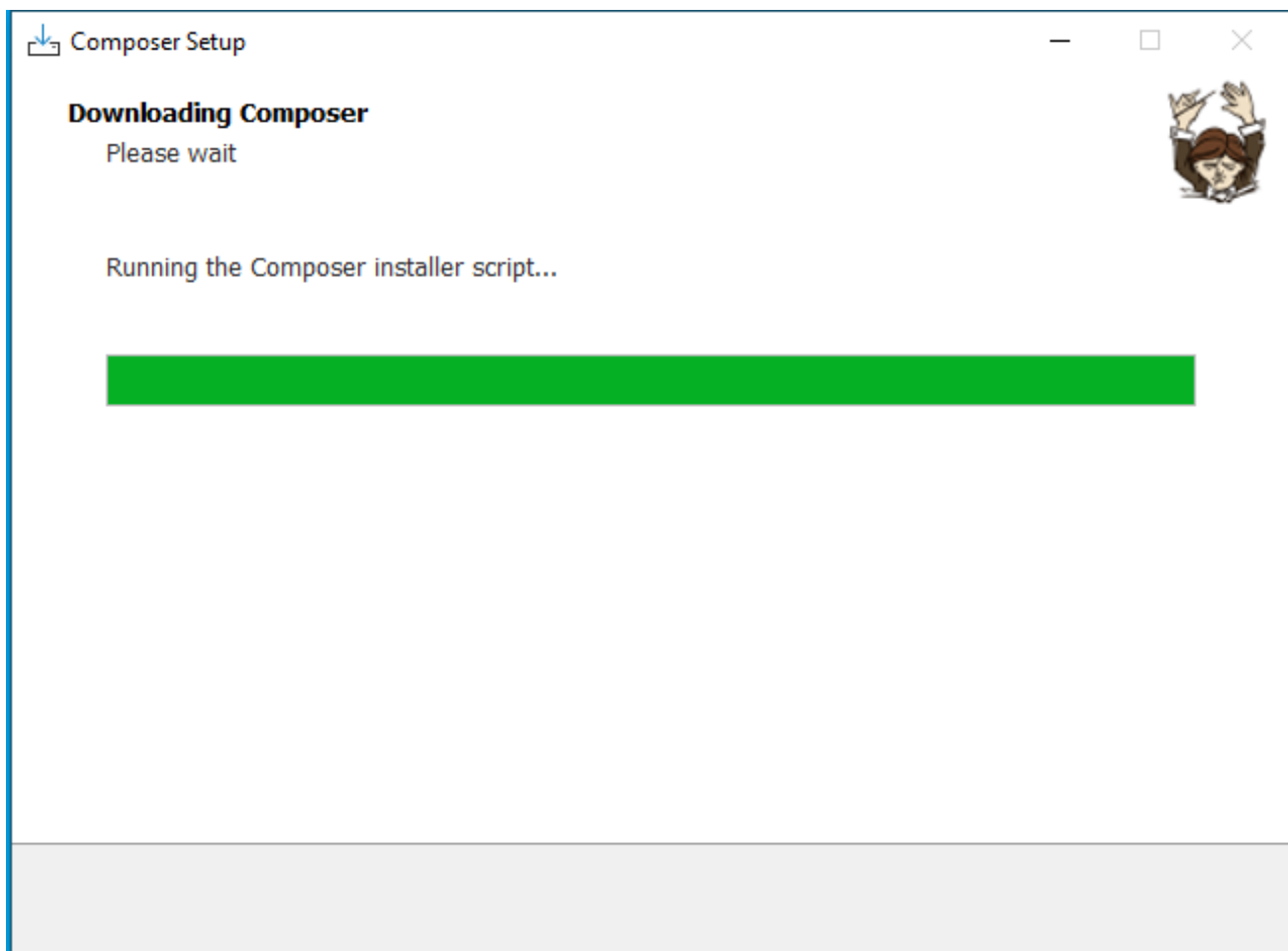
Proxy Settings

Choose if you need to use a proxy.

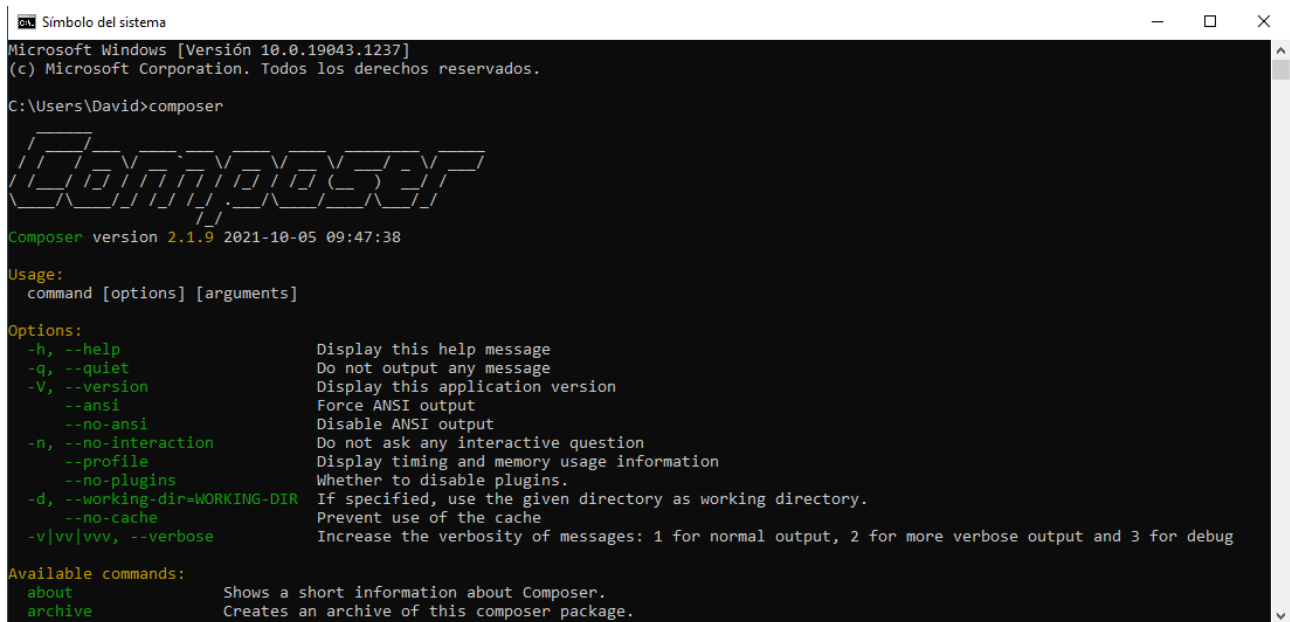


☐ Use a proxy server to connect to internet

Enter proxy url



Una vez terminada la instalación, podemos comprobar que funciona correctamente abriendo una consola de sistema, nos situamos en el directorio `\users` (que en este caso tiene el nombre *David*) y escribiremos el comando “composer”.



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19043.1237]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\David>composer

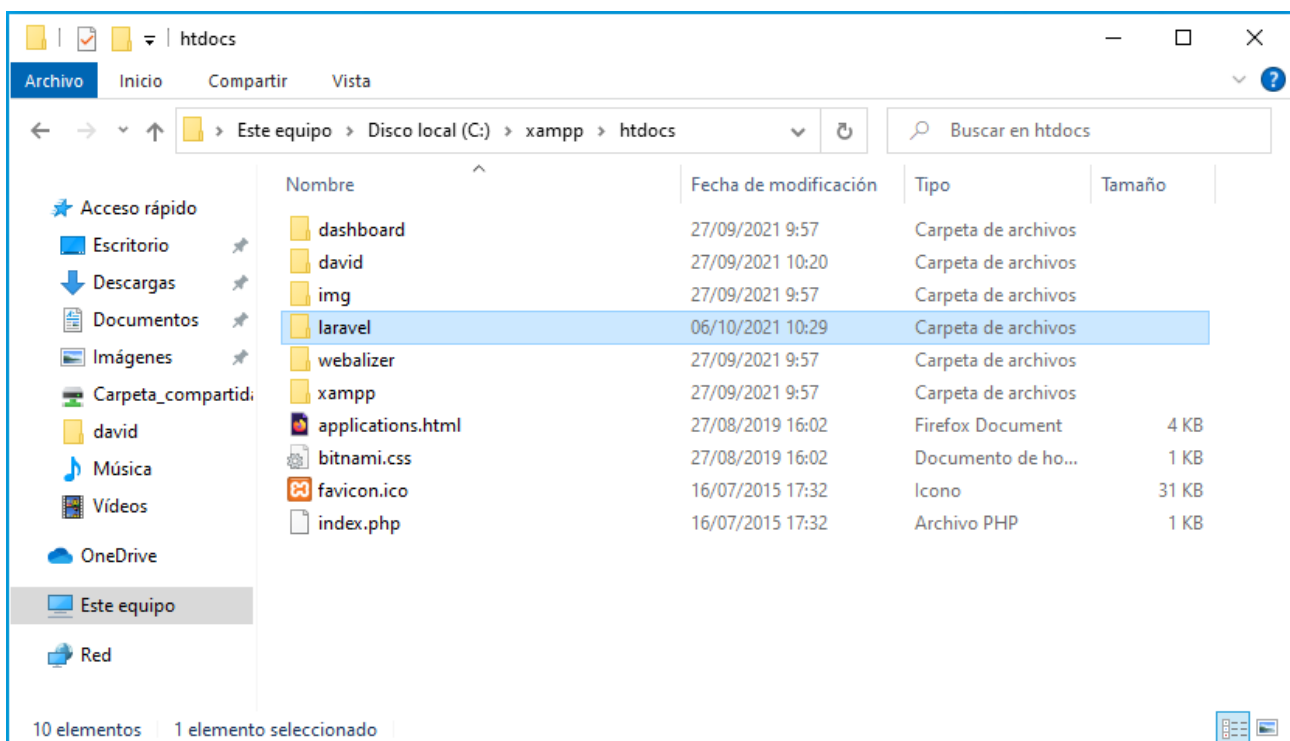
Composer
Composer version 2.1.9 2021-10-05 09:47:38

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet                Do not output any message
  -V, --version              Display this application version
      --ansi                 Force ANSI output
      --no-ansi              Disable ANSI output
  -n, --no-interaction       Do not ask any interactive question
      --profile              Display timing and memory usage information
      --no-plugins           Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
      --no-cache             Prevent use of the cache
  -v|vv|vvv, --verbose       Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  about      Shows a short information about Composer.
  archive    Creates an archive of this composer package.
```

En el siguiente paso nos dirigiremos a la carpeta *htdocs* dentro del directorio *xampp*, y dentro crearemos otra carpeta llamada *laravel*.



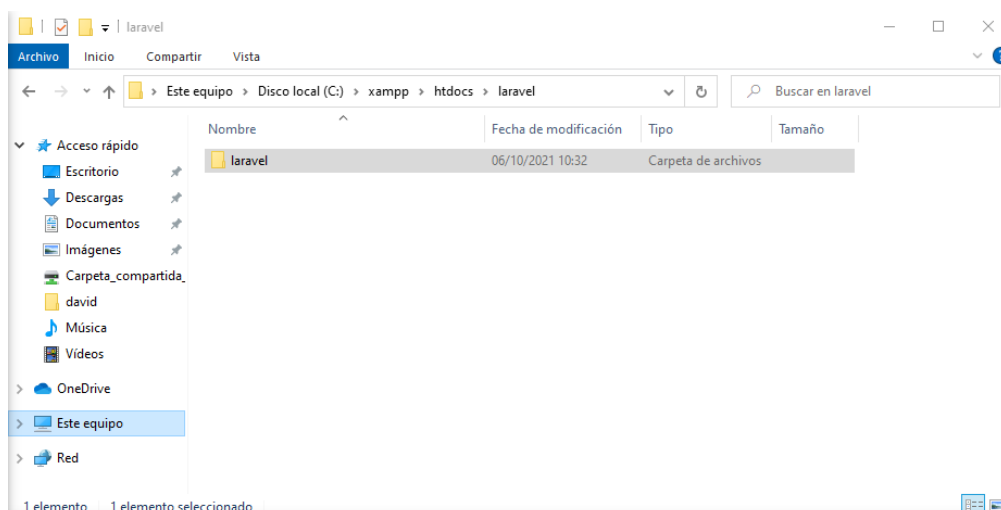
Ahora, desde la consola de sistema nos situamos en la carpeta *laravel* que acabamos de crear y ejecutamos el comando “composer create-project --prefer-dist laravel/laravel laravel”, esperando a que termine el proceso.

```
Símbolo del sistema
C:\>cd xampp\htdocs\laravel

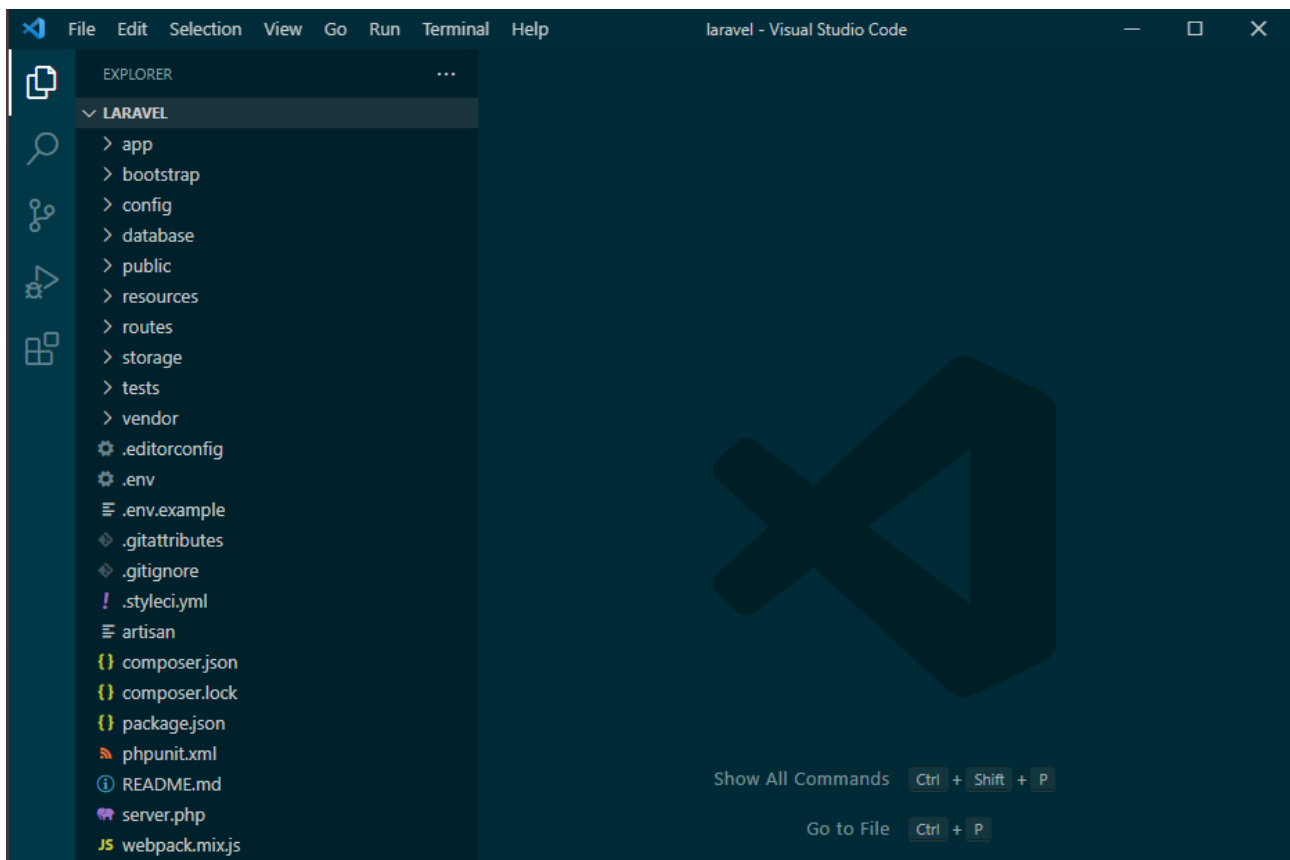
C:\xampp\htdocs\laravel>composer create-project --prefer-dist laravel/laravel laravel
Creating a "laravel/laravel" project at "./laravel"
Installing laravel/laravel (v8.6.3)
- Downloading laravel/laravel (v8.6.3)
- Installing laravel/laravel (v8.6.3): Extracting archive
Created project in C:\xampp\htdocs\laravel\laravel
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking asm89/stack-cors (v2.0.3)
- Locking brick/math (0.9.3)
- Locking dflydev/dot-access-data (v3.0.1)
- Locking doctrine/inflector (2.0.3)
- Locking doctrine/instantiator (1.4.0)
- Locking doctrine/lexer (1.2.1)
- Locking dragonmantank/cron-expression (v3.1.0)
- Locking egulias/email-validator (2.1.25)
- Locking facade/flare-client-php (1.9.1)
- Locking facade/ignition (2.14.0)
- Locking facade/ignition-contracts (1.0.2)
- Locking fakerphp/faker (v1.16.0)
- Locking filp/whoops (2.14.4)
- Locking fruitcake/laravel-cors (v2.0.4)
- Locking graham-campbell/result-type (v1.0.2)
- Locking guzzlehttp/guzzle (7.3.0)
- Locking guzzlehttp/promises (1.4.1)
- Locking guzzlehttp/psr7 (2.0.0)
- Locking hamcrest/hamcrest-php (v2.0.1)
```

```
Símbolo del sistema
83 package suggestions were added by new dependencies, use "composer suggest" to see details.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fruitcake/laravel-cors
Discovered Package: laravel/sail
Discovered Package: laravel/sanctum
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
76 packages you are using are looking for funding.
Use the "composer fund" command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi
No publishable resources for tag [laravel-assets].
Publishing complete.
> @php artisan key:generate --ansi
Application key set successfully.
C:\xampp\htdocs\laravel>
```

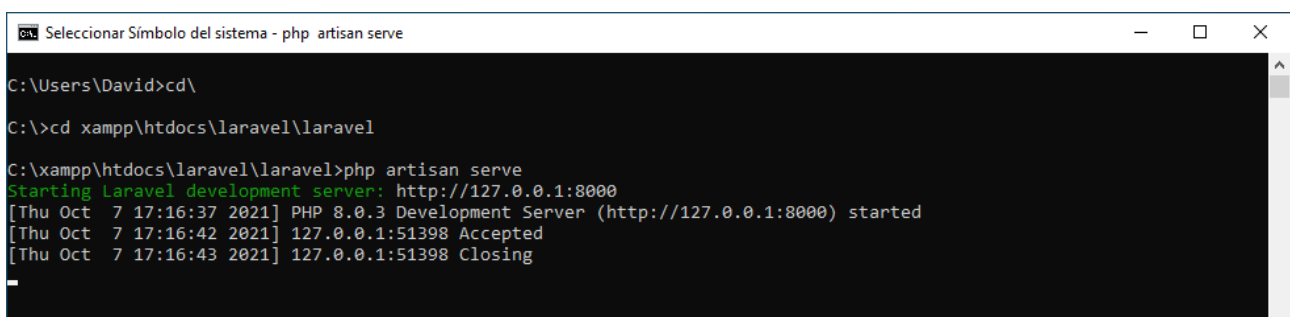
Vamos que la carpeta *laravel* se ha creado en el directorio donde estábamos anteriormente:



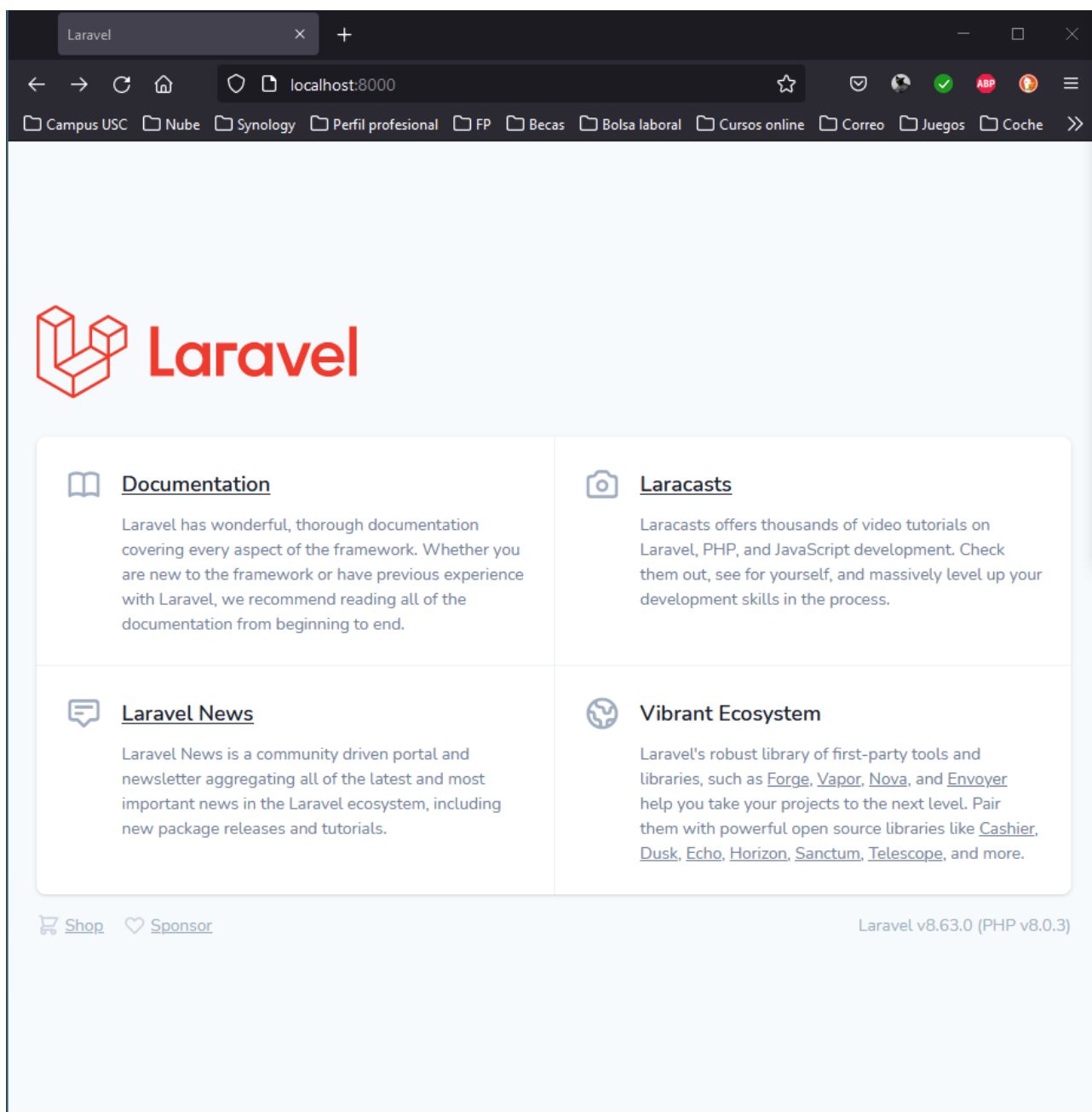
Con Visual Studio Code podemos comprobar la instalación de Laravel agregando la carpeta al IDE, donde verificaremos todos los archivos que contiene:



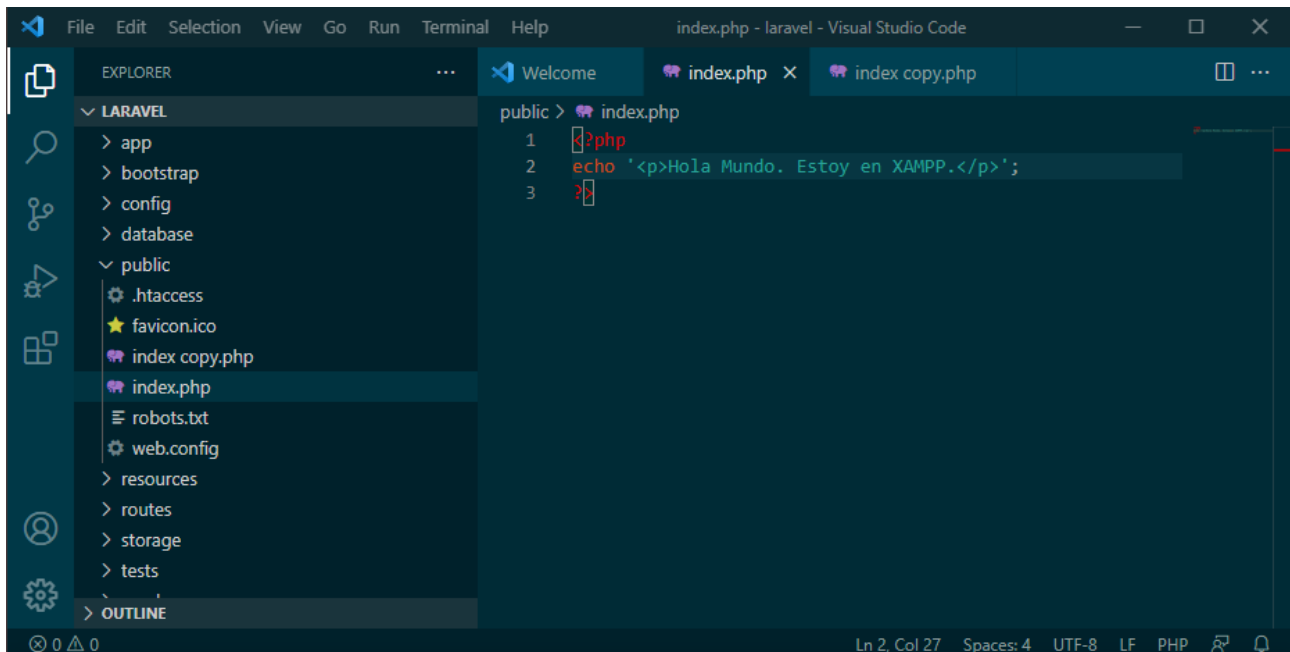
Seguidamente, estando ubicados en la carpeta *laravel*, debemos ejecutar el comando “php artisan serve” para hacer una prueba de conexión del servidor Laravel.



Podemos comprobar que la conexión funciona correctamente a través del navegador escribiendo la dirección “localhost:8000\index.php”.



El siguiente paso es editar un archivo de texto llamado *index.php* que se encuentra en la carpeta *public* dentro del directorio *laravel*. Este archivo tiene texto por defecto, pero para la prueba que necesitamos, lo editaremos cambiando su contenido por un mensaje “Hola Mundo. Estoy en XAMPP.” escrito en PHP.



```
File Edit Selection View Go Run Terminal Help
index.php - laravel - Visual Studio Code

EXPLORER
LARAVEL
  > app
  > bootstrap
  > config
  > database
  > public
    .htaccess
    favicon.ico
    index copy.php
    index.php
    robots.txt
    web.config
  > resources
  > routes
  > storage
  > tests
  > OUTLINE

public > index.php
1 <?php
2 echo '<p>Hola Mundo. Estoy en XAMPP.</p>';
3
```

Finalmente, podemos comprobar que todo ha funcionado correctamente abriendo el navegador y escribiendo la dirección “localhost:8000\index.php”.

