

## Manual Prueba Tecnica

### Requerimientos en maquina:

- Composer
- Base de datos Mysql
- PHP

La prueba esta echa en php con el framework de laravel, por lo que es necesario instalar composer.

Para instalarlo solo basta con ir a la pagina de Composer, descargar el ejecutable y seguir los pasos que ahí se indiquen.



### A Dependency Manager for PHP

Latest: **2.4.2** ([changelog](#))

[Getting Started](#)[Download](#)[Documentation](#)[Browse Packages](#)[Issues](#)[GitHub](#)

### Download Composer Latest: v2.4.2

#### 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.

#### Command-line installation

To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use [the guide on installing Composer programmatically](#).

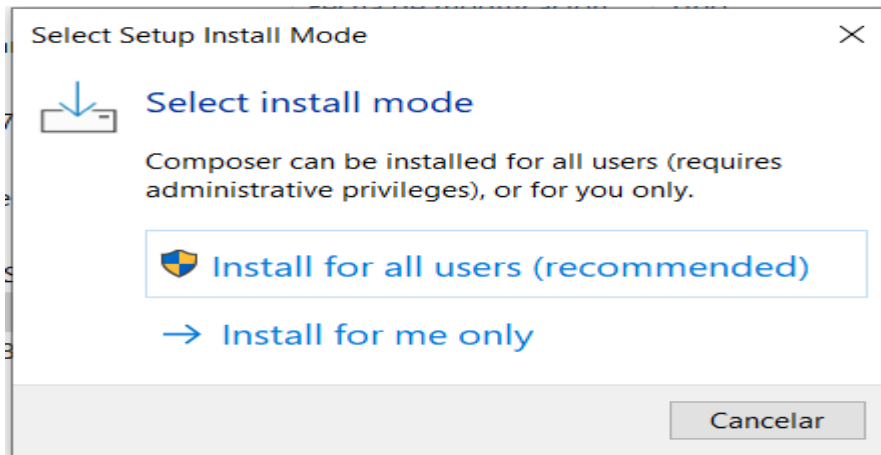
```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === '55ce33d7678c5a611085589f1f3dd8b3c52d662cd01d4ba75c66d4307e07627507') {echo 'sha384';} else {echo 'sha512';}"
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

This installer script will simply check some `php.ini` settings, warn you if they are set incorrectly, and then download the latest `composer.phar` in the current directory. The 4 lines above will, in order:

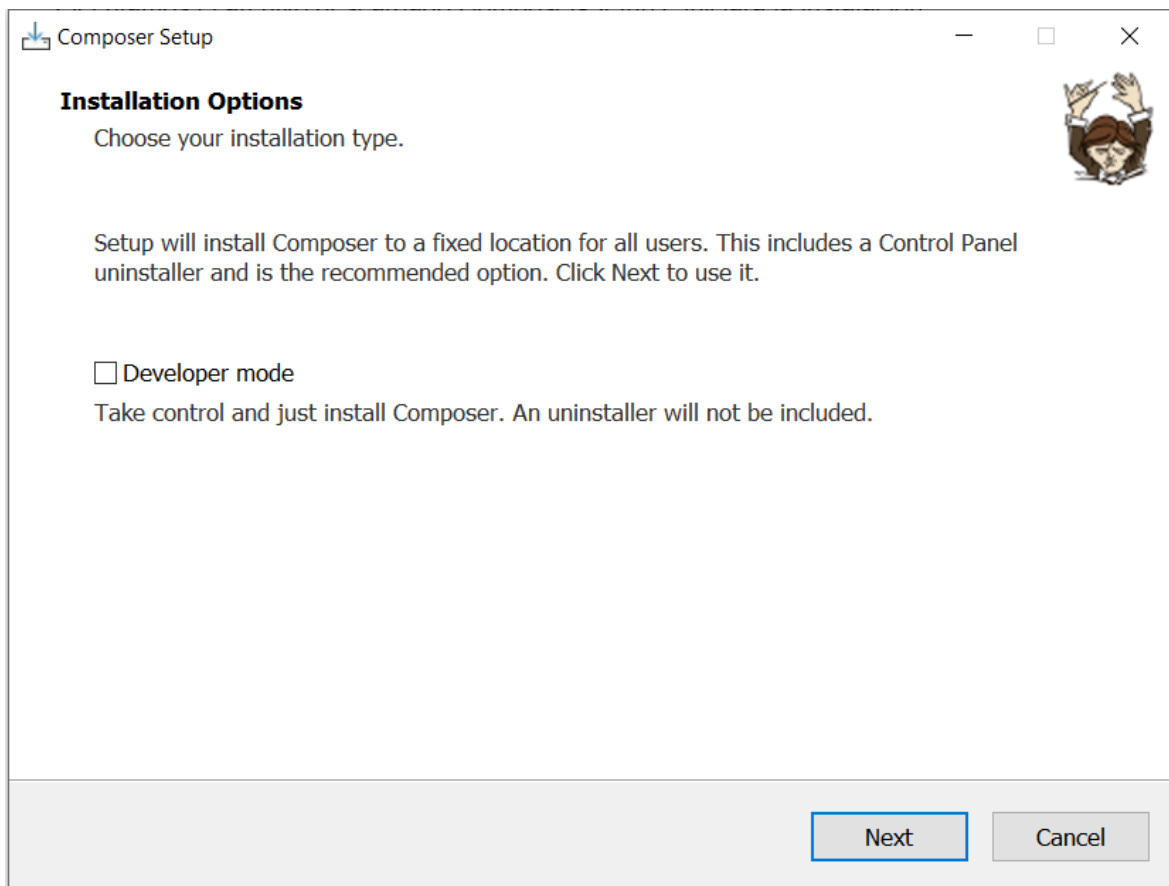
- Download the installer to the current directory
- Verify the installer SHA-384, which you can also [cross-check here](#)
- Run the installer
- Remove the installer

Most likely, you want to put the `composer.phar` into a directory on your PATH, so you can simply call `composer` from any directory (*Global install*), using for example:

Ejecutamos el archivo descargado Composer-Setup e iniciara la instalación de Composer. En caso que tu equipo cuente con más de un usuario se nos visualiza el modo de instalación para los usuarios del equipo, le damos clic en el recomendado.

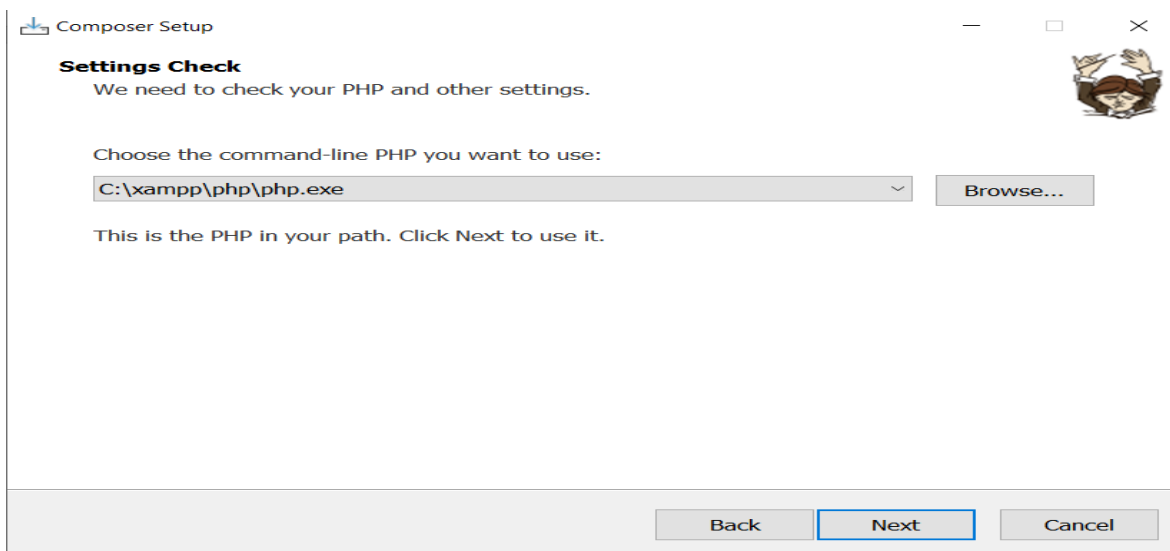


En esta ventana el asistente de instalación nos indica si queremos instalar Composer en modo desarrollador, lo dejamos por defecto no seleccionado, le damos clic Next.

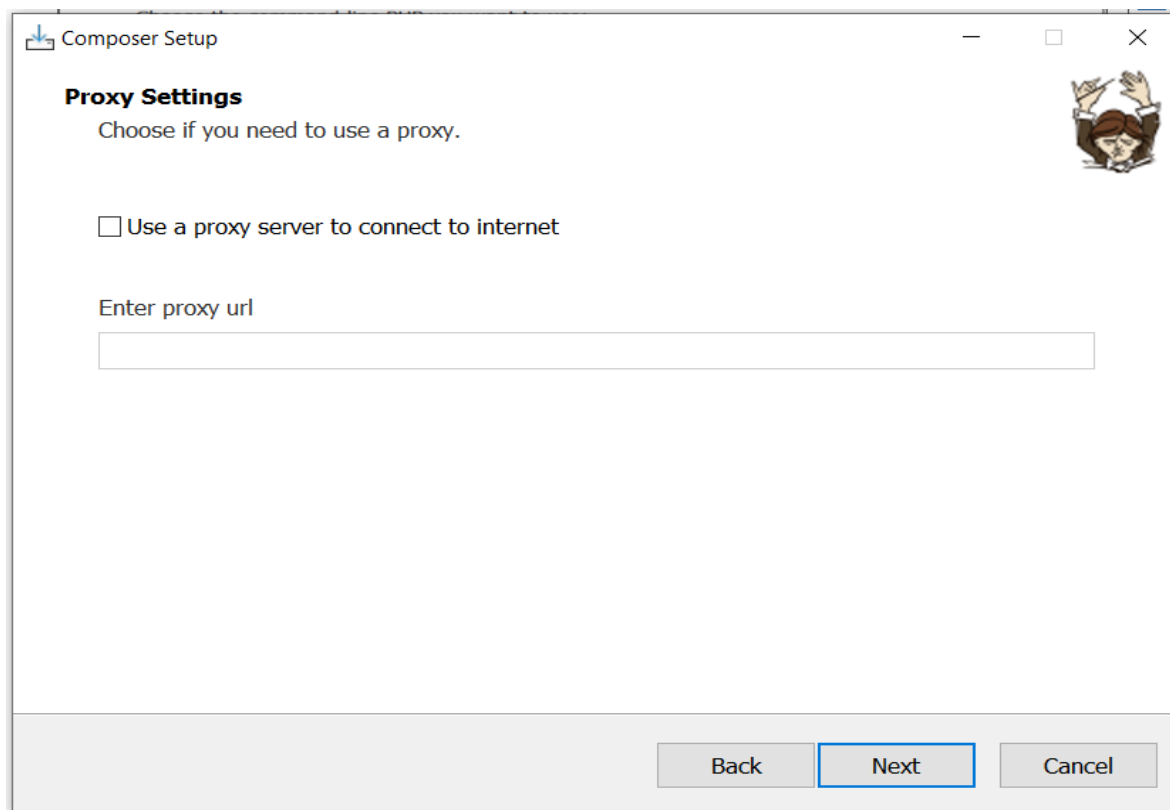


El asistente de instalación por defecto reconoce que tenemos una instalación de Xampp en nuestro equipo, en caso de no contar con este te pedirá la ruta de la línea de comandos de PHP.

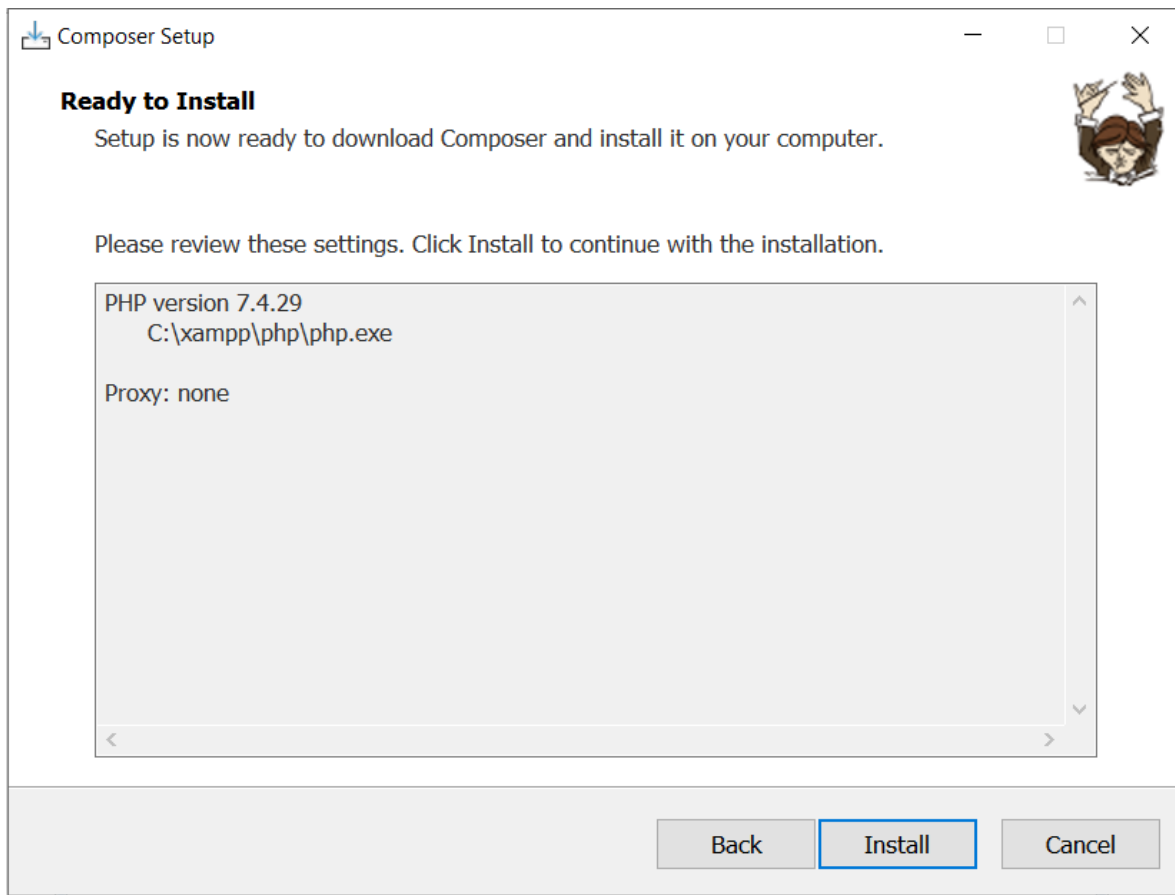
Seleccionamos la opción de añadir al path de PHP, le damos clic Next.



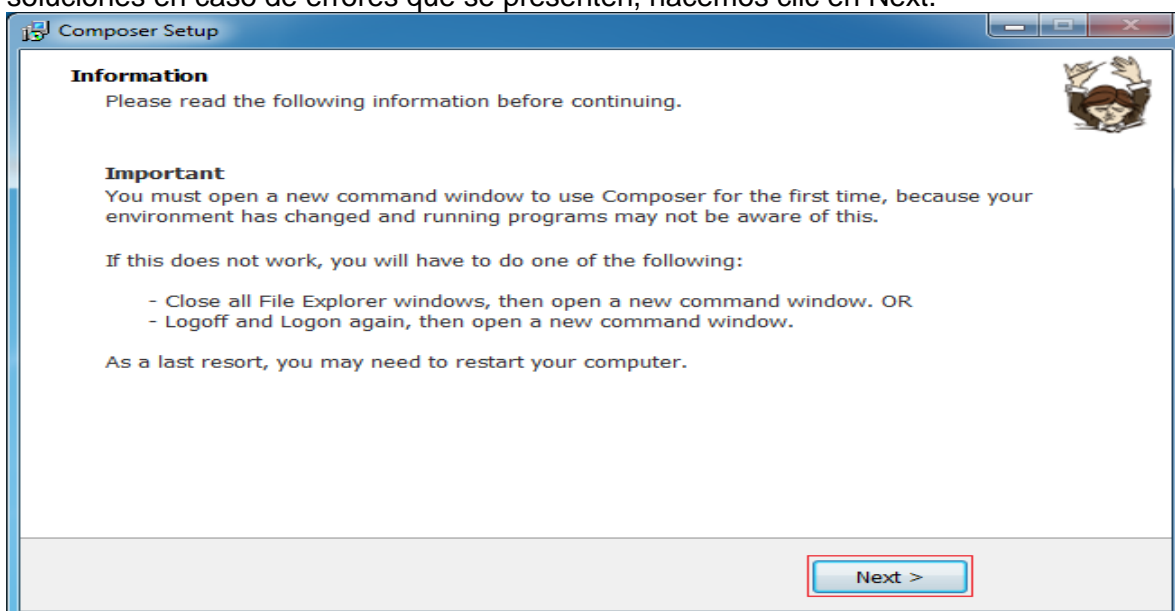
En esta ventana te solicita la configuración del proxy, Deja la casilla por defecto no seleccionada, le damos clic Next.



Una vez completada las configuraciones, ya estamos listo para empezar la instalación de Composer en nuestra máquina local, le damos clic en Install.



Una vez finalizada la instalación de Composer, nos brinda más información de uso y soluciones en caso de errores que se presenten, hacemos clic en Next.

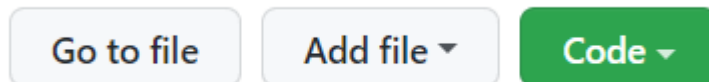


En esta ventana nos muestra que la instalación ha sido completada.

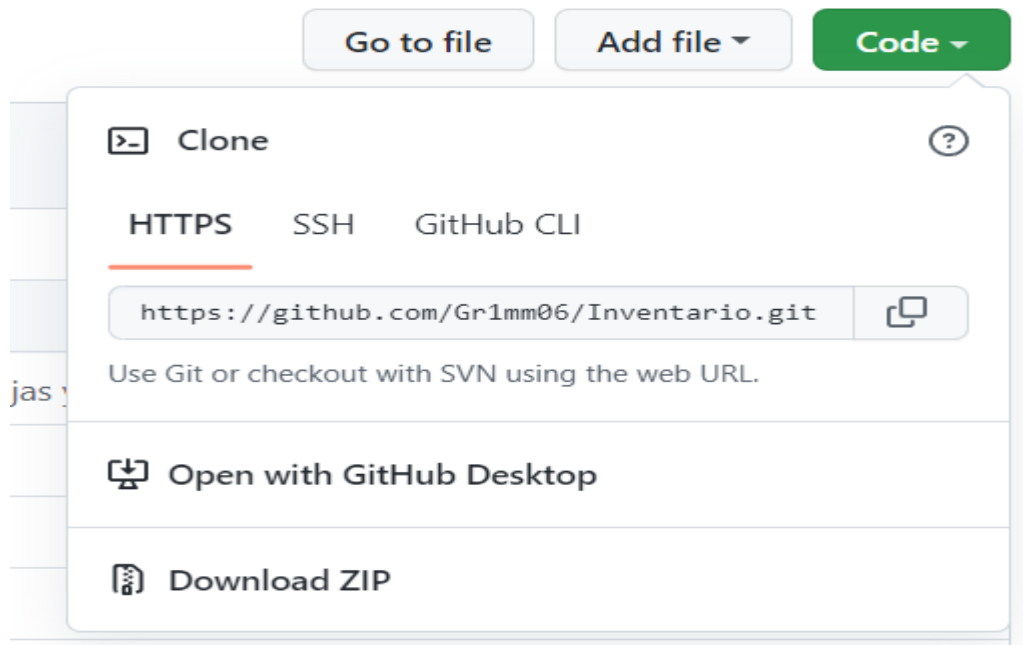


Con composer instalado en nuestra maquina ya nos permitiría ejecutar nuestro proyecto. Ahora necesitamos clonar nuestro proyecto desde el repositorio de GitHub, para eso tendremos que seguir unos cuantos pasos:

- En la pagina de GitHub tendremos que ir a la pagina principal del repositorio y daremos clic en “codigo”



- Copiaremos HTTPS para clonarlo desde ahí



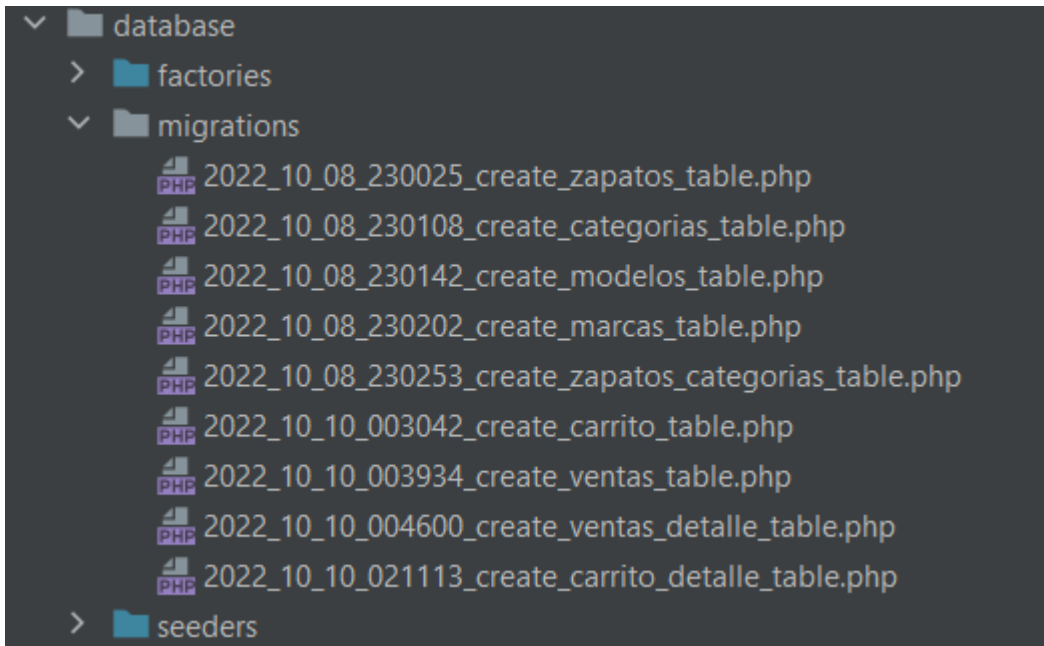
- Abra Git Bash.
- Cambia el directorio de trabajo actual a la ubicación en donde quieres clonar el directorio.
- Escriba `git clone` y pegue la dirección URL que copio.

Ahora que ya tenemos nuestro proyecto es necesario seguir unos pasos para configurar la base de datos.

Antes que nada el nombre de la base de datos es “inventario” por lo que se necesita ser creada, las credenciales son `user = root` y `password = ""`, en el caso de que se requiera cambiar lo pueden hacer en el archivo **database.php** dentro de la carpeta **config**, en el siguiente bloque.

```
'mysql' => [  
    'driver' => 'mysql',  
    'url' => env( key: 'DATABASE_URL' ),  
    'host' => env( key: 'DB_HOST', default: '127.0.0.1' ),  
    'port' => env( key: 'DB_PORT', default: '3306' ),  
    'database' => env( key: 'DB_DATABASE', default: 'inventario' ),  
    'username' => env( key: 'DB_USERNAME', default: 'root' ),  
    'password' => env( key: 'DB_PASSWORD', default: '' ),  
    'unix_socket' => env( key: 'DB_SOCKET', default: '' ),  
    'charset' => 'utf8mb4',  
    'collation' => 'utf8mb4_unicode_ci',  
    'prefix' => '',  
    'prefix_indexes' => true,  
    'strict' => true,  
    'engine' => null,  
    'options' => extension_loaded( 'pdo_mysql' ) ? array_filter([  
        PDO::MYSQL_ATTR_SSL_CA => env( key: 'MYSQL_ATTR_SSL_CA' ),  
    ]) : [],
```

Una vez que ya tengamos configurada la conexión, procedemos a crear las tablas en la base de datos, laravel nos permite crearlas sin necesidad de entrar directamente al gestor de base de datos, estos archivos se llaman **migraciones** y se encuentran en la ruta **database/migrations**



Para el correcto funcionamiento se tienen que correr ciertas migraciones primero, ya que algunas tablas están relacionadas entre si, es necesario crear primero unas que otras

El orden adecuado seria:

1. create\_categorias\_table.php
2. create\_modelos\_table.php
3. create\_marcas\_table.php
4. create\_zapatos\_table.php
5. create\_zapatos\_categorias\_table.php
6. create\_carrito\_table.php
7. create\_carrito\_detalle\_table.php
8. create\_ventas\_table.php
9. create\_ventas\_detalle\_table.php

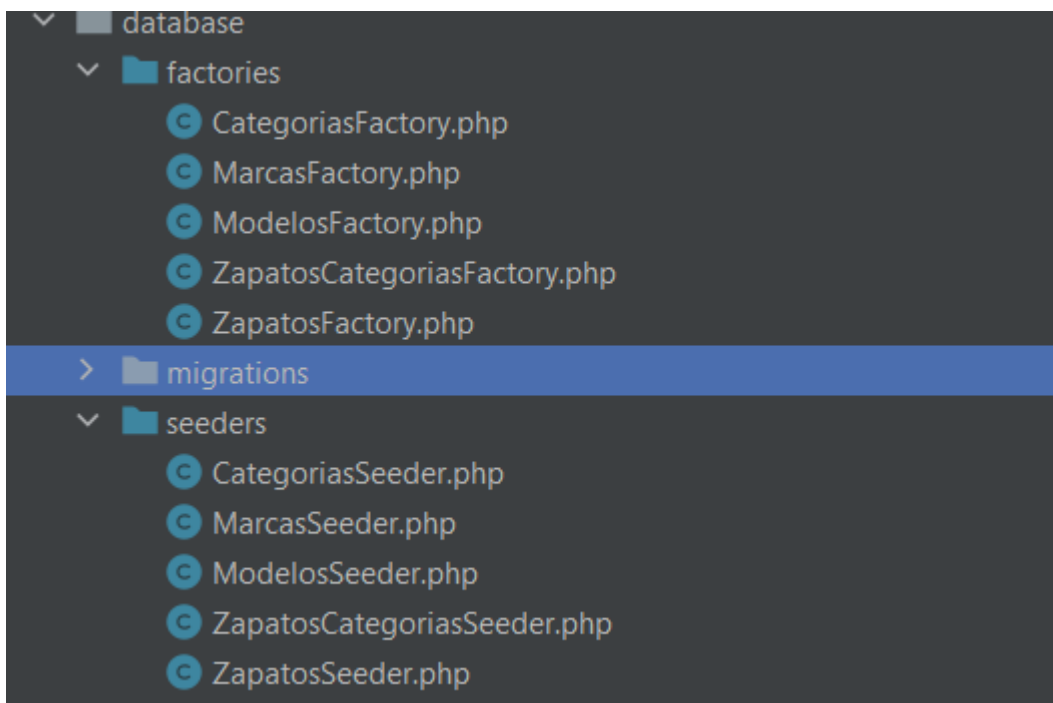
Para correr la migración en específico es necesario ejecutar la siguiente línea de comando desde la consola(esto dentro de la carpeta del proyecto)

```
php artisan migrate --path=/database/migrations/ 2022_10_08_230108_create_categorias_table.php
php artisan migrate --path=/database/migrations/ 2022_10_08_230142_create_modelos_table.php
php artisan migrate --path=/database/migrations/ 2022_10_08_230142_create_modelos_table.php
php artisan migrate --path=/database/migrations/ 2022_10_08_230202_create_marcas_table.php
php artisan migrate --path=/database/migrations/ 2022_10_08_230025_create_zapatos_table.php
php artisan migrate --path=/database/migrations/ 2022_10_08_230253_create_zapatos_categorias_table.php
php artisan migrate --path=/database/migrations/ 2022_10_10_003042_create_carrito_table.php
php artisan migrate --path=/database/migrations/ 2022_10_10_021113_create_carrito_detalle_table.php
php artisan migrate --path=/database/migrations/ 2022_10_10_003934_create_ventas_table.php
```

```
php artisan migrate --path=/database/migrations/ 2022_10_10_004600_create_ventas_detalle_table.php
```

Ya contamos con nuestras tablas en la base de datos, otra funcionalidad de laravel es que nos permite insertar datos de prueba para realizar pruebas nosotros mismos, estos son los **seeder** y **factories**

Estos archivos se encuentran en **database/seeder**s y **database/factories** respectivamente



Los archivos ya están configurados solo es necesario correr algunas líneas de comando

```
php artisan db:seed --class=CategoriasSeeder.php
php artisan db:seed --class=MarcasSeeder.php
php artisan db:seed --class=ModelosSeeder.php
php artisan db:seed --class=ZapatosSeeder.php
php artisan db:seed --class=ZapatosCategoriasSeeder.php
```

El único comando que se ejecutaría 2 veces sería el

```
php artisan db:seed --class=ModelosSeeder.php
```

Ya que en el ModelosFactory tendríamos que cambiar un valor



```

use Illuminate\Database\Eloquent\Factories\Factory;
use Illuminate\Support\Str;

class ModelosFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array
     */
    public function definition()
    {
        return [
            'codigo_modelo' => Str::random(10),
            'descripcion' => 'Dama',
        ];
    }
}

```

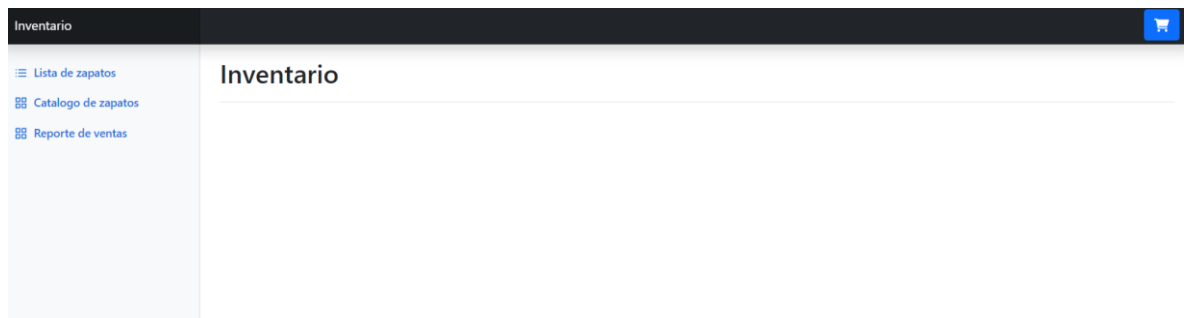
Si lo ejecutamos primero con **Dama** la siguiente vez que se ejecute tendríamos que cambiar el valor a **Caballero**

Con esto ya tendríamos lista nuestra base de datos y podríamos correr nuestra aplicación, para mostrar la aplicación es necesario abrir la consola de comandos sobre nuestra carpeta de proyectos y correr el siguiente comando

```
php artisan serve
```

se le puede incluir **-host=** con esto indicamos la dirección por la cual queremos que se muestre y **-port=** con esto indicamos el puerto que escuchará

Una vez desplegada la aplicación, se mostrará la siguiente pantalla



Cada opción despliega información diferente

### 1. Lista de zapatos: despliega la siguiente infomacion

Inventario

Lista de zapatos

Catalogo de zapatos

Reporte de ventas

Inventario

Lista de Zapatos

Modelo





















Categorías

Buscar

Agregar

Mostrar 10 Resultados

Busqueda Global:

Descripcion	Modelo	Codigo del Modelo	Marca	Cantidad	Numero	Categorías	Precio	Acciones
asperiores	Caballero	stk2mjuWw	molestiae	5	18	ullam	\$ 7.74	 
at	Caballero	stk2mjuWw	ut	1	25	nemo	\$ 34.40	 
aut	Caballero	b25lX5Z1Du	in	2	27	corporis.corporis	\$ 32.36	 
aut	Dama	YPopAn7yn0	molestiae	4	12	excepturi	\$ 30.09	 
aut	Dama	40BEqzz30A	in	3	17	corporis.excepturi	\$ 29.13	 
corrupti	Caballero	RzZ2R724lU	molestiae	3	11	nemo.nemo.excepturi	\$ 18.44	 
cum	Dama	40BEqzz30A	temporibus	0	6	dolorem	\$ 74.56	 
ddd	Caballero	RzZ2R724lU	molestiae	3	3		\$ 12.00	 
ddd	Caballero	RzZ2R724lU	molestiae	3	3	excepturi.ullam	\$ 12.00	 
dolor	Dama	fCEhQsdf	molestiae	4	19		\$ 82.99	 

Mostrando 1 de 10 en 53 Resultados

Anterior

1

2

3

4

5

6

Siguiente

Podemos ver una lista con todos los zapatos, con su respectiva infomacion; modelo,marca,cantidad, numero y categorías, tambien acciones para editar o eliminar el zapato.

El botón **agregar** crea un nuevo registro en la base de datos y despliega la siguiente vista

Inventario

Lista de zapatos

Catalogo de zapatos

Reporte de ventas

Inventario

Crear Zapato

Agregar

Cancelar

Informacion del Zapato

Descripcion


Modelo

Modelo

Marca

Marca

Foto



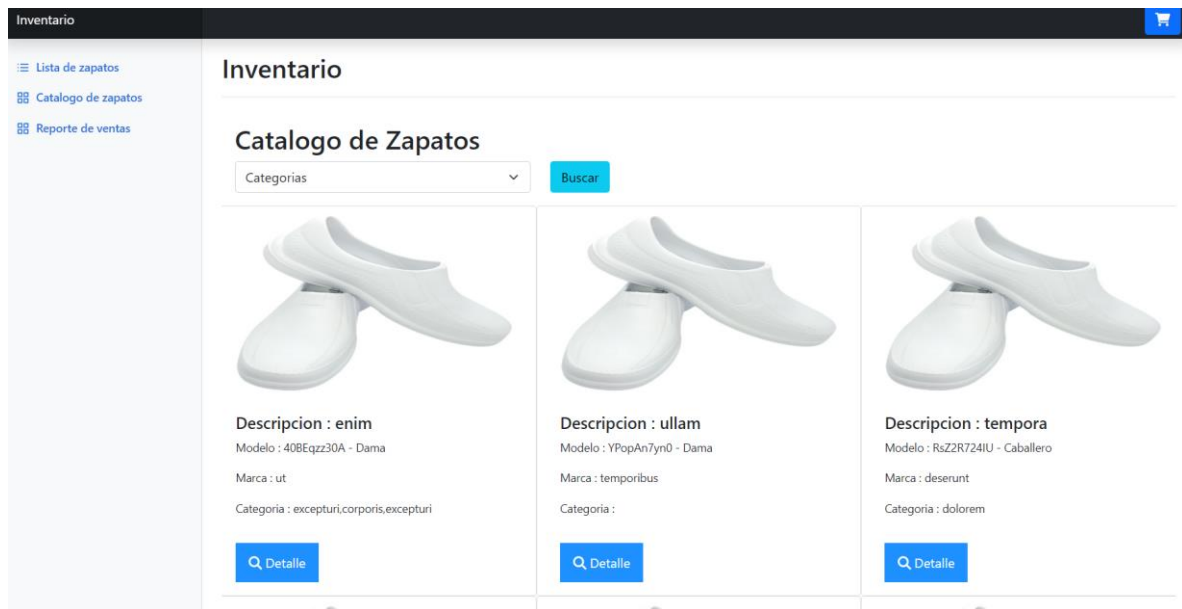
Cargar Imagen

Cantidad

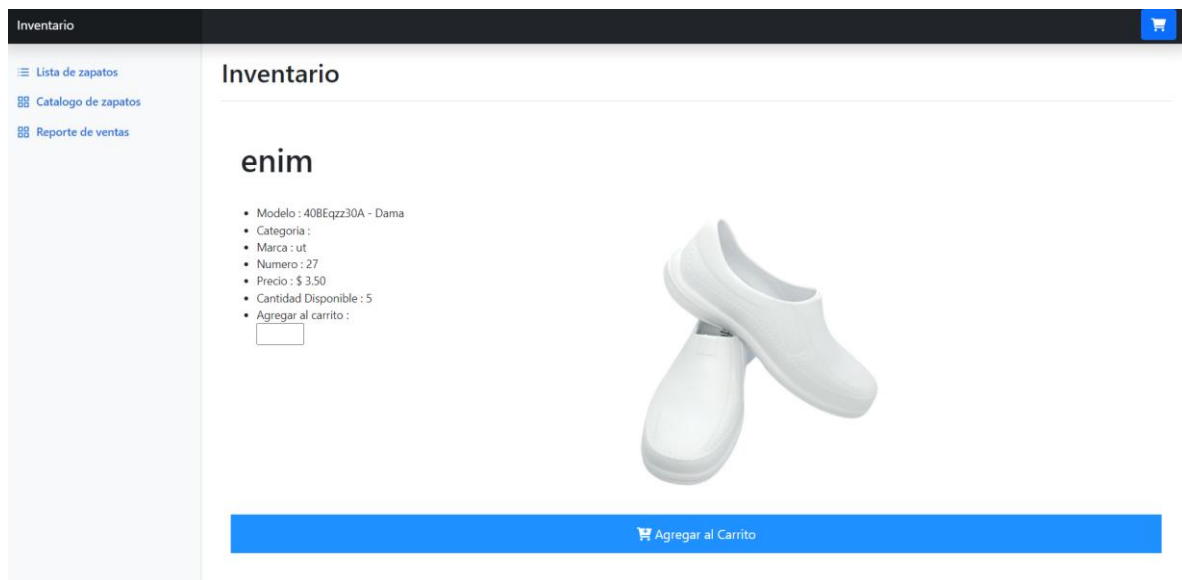
Numero

Solo se debe de llenar la información correspondiente y dar click en **Agregar**

## 2. Catalogo de zapatos: Mostrara la siguiente vista

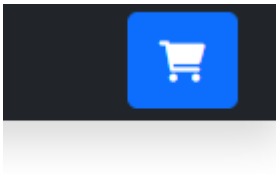


Mostrara las características principales del calzado y un botón de **Detalle** esto mostrara mas a profundidad el producto

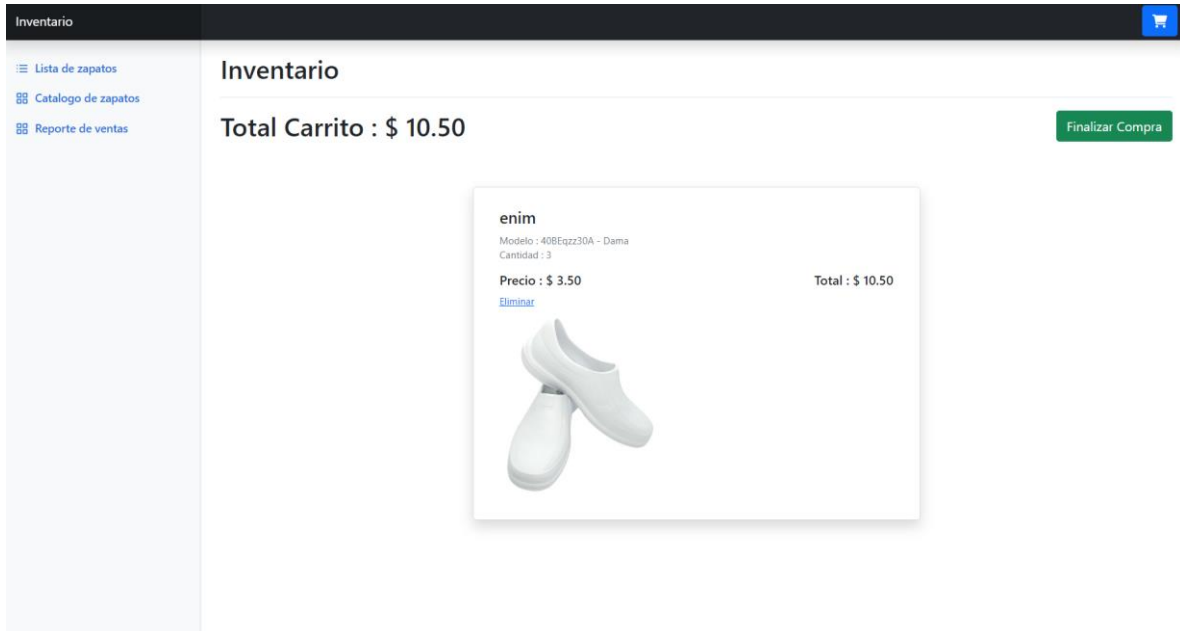


Aquí podremos seleccionar cuantas piezas queremos del calzado y agregarlas al carrito con el botón **Agregar al Carrito**

3. Carrito: para acceder al carrito tendremos que dar click en la siguiente sección

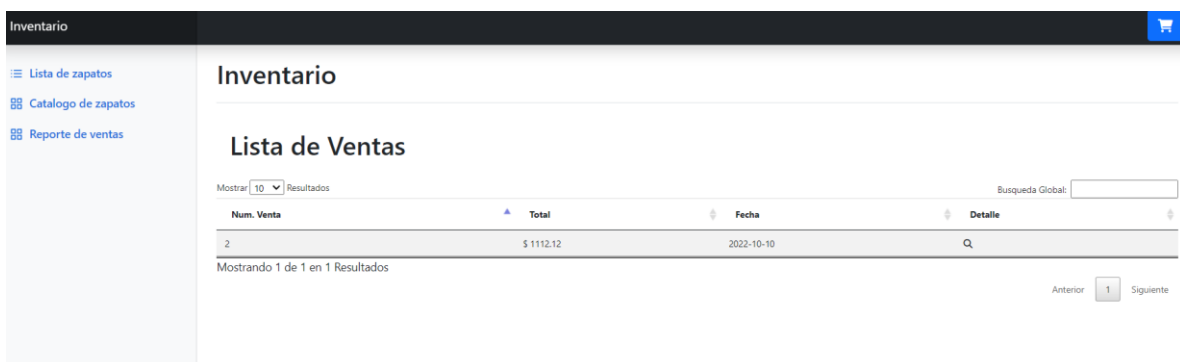


La pantalla de se mostrara es la siguiente



En esta se muestra el resumen, el total del carrito, el calzado y numero de piezas y el total en relación a la cantidad y precio del calzado

4. Reporte de ventas: Esta opción es para ver las ventas que se han realizado y muestra la siguiente información



Nos muestra el total de la venta y la fecha en la que se realizo, también con una opción de detalle para ver a profundidad la venta

-  [Lista de zapatos](#)
-  [Catalogo de zapatos](#)
-  [Reporte de ventas](#)

## Inventario

# Venta : 2

Total Venta : \$ 1112.12

[Regresar](#)

### ipsam

Modelo : 408Eqz30A - Dama  
Cantidad : 5

Total : \$ 169.80



### Zapatilla

Modelo : RAQ1JJW18Y - Dama  
Cantidad : 16

Total : \$ 910.40