**ESCUELA POLITÉCNICA NACIONAL**

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DESARROLLO DE APLICACIONES WEB**

|  |  |
| --- | --- |
|  | |
| ASIGNATURA: | Desarrollo de Aplicaciones Web |
| PROFESOR: | Ing. Loarte Byron |
| PERÍODO ACADÉMICO: |  |
|  | |

**LABORATORIO - 06**

|  |  |
| --- | --- |
|  | |
| TÍTULO:  **LOGIN + CRUD + BDD** | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  |  |
|  |  |
| Como instalar Laravel 6 en mi servidor |  |
|  |  |
|  | |

**PROPÓSITO DE LA PRÁCTICA**

Familiarizar al estudiante con la estructura de directorios en Laravel y la implementación de paquetes de autenticación.

**OBJETIVO GENERAL**

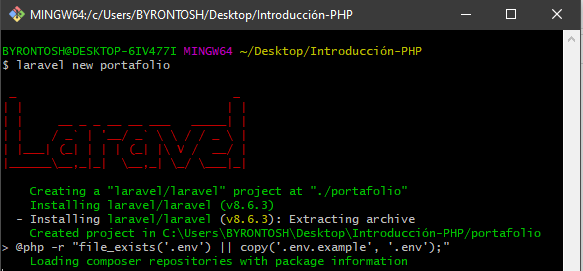
Determinar los principios en el desarrollo web con Laravel, el manejo de estructura de directorios y paquetes de autenticación.

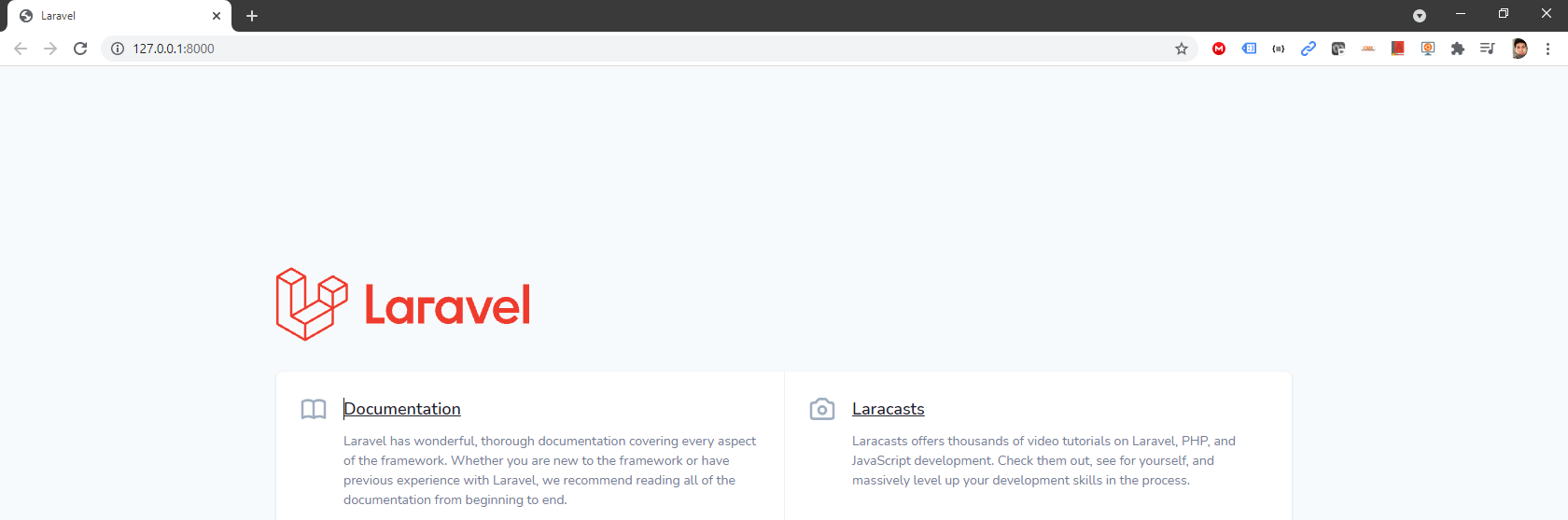
**OBJETIVOS ESPECÍFICOS**

* Configurar las herramientas para el laboratorio.
* Crear un proyecto en Laravel.
* Comprender los archivos y directorios más importantes en Laravel.
* Comprender el uso y manejo de rutas.
* Comprender el uso y manejo de controladores.
* Comprender el uso y manejo de vistas.
* Comprender el uso y manejo del motor de plantillas Blade.
* Comprender el uso de paquetes de autenticación.
* Visualizar los resultados obtenidos.

**INSTRUCCIONES**

1. Se procede con la creación de un nuevo proyecto en Laravel, para eso se debe ejecutar el siguiente comando en una terminal de Windows.

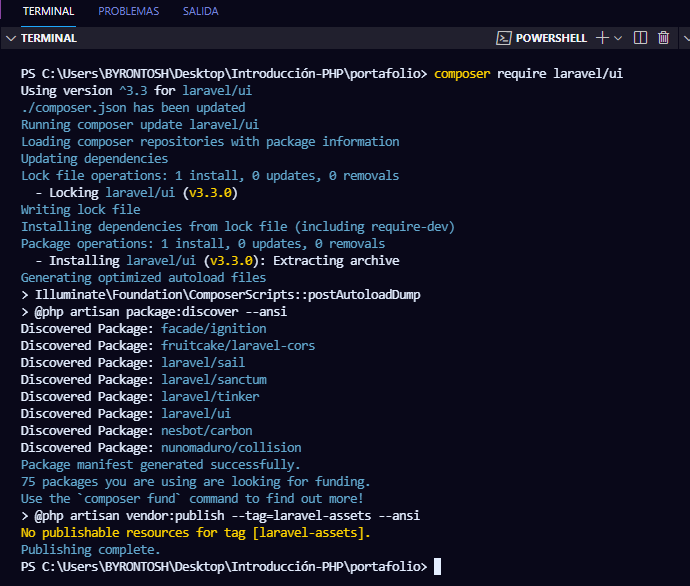




1. Ahora se procede con la importación de un paquete para el manejo de registro y autenticación, ejecutando el siguiente comando.

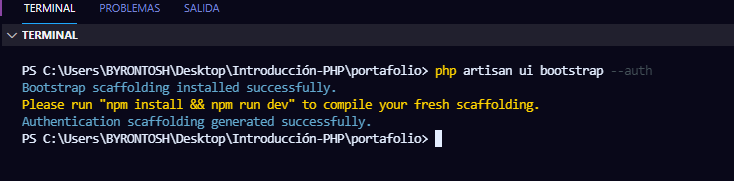
<https://github.com/laravel/ui>

**composer require laravel/ui**



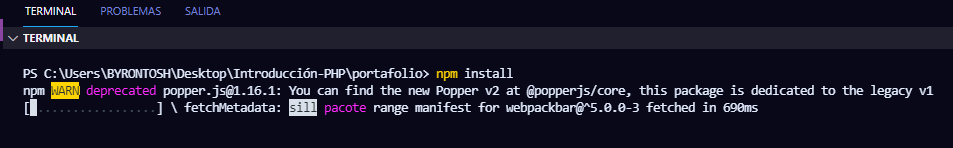
1. Posterior a ello ejecutar el siguiente comando.

**php artisan ui bootstrap --auth**

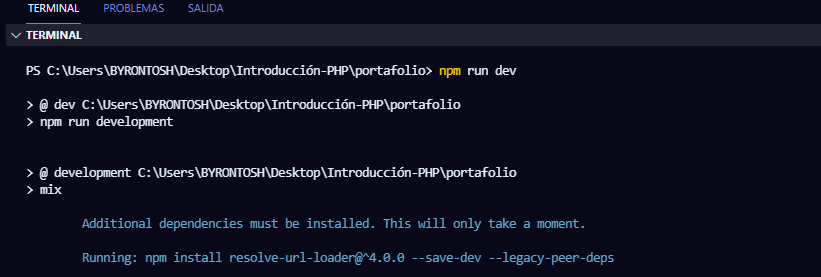


Por último, ejecutar los dos comandos solicitados. No obstante, hay que tener en cuenta que se debe tener instalado **NODE** para ejecutar los siguientes comandos.

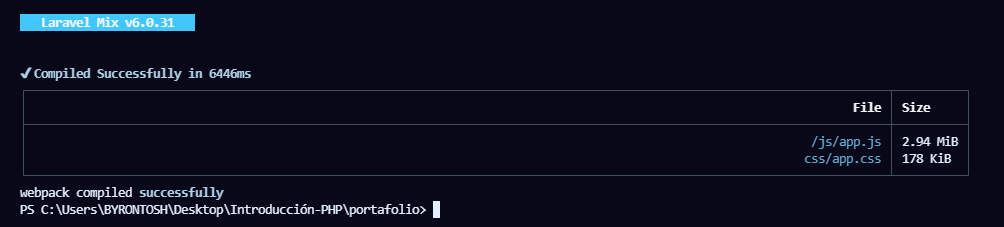
**npm install**



**npm run dev**



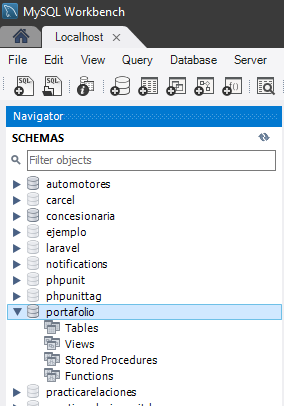
Volver a ejecutar el comando, si presenta algún error al ejecutarlo.



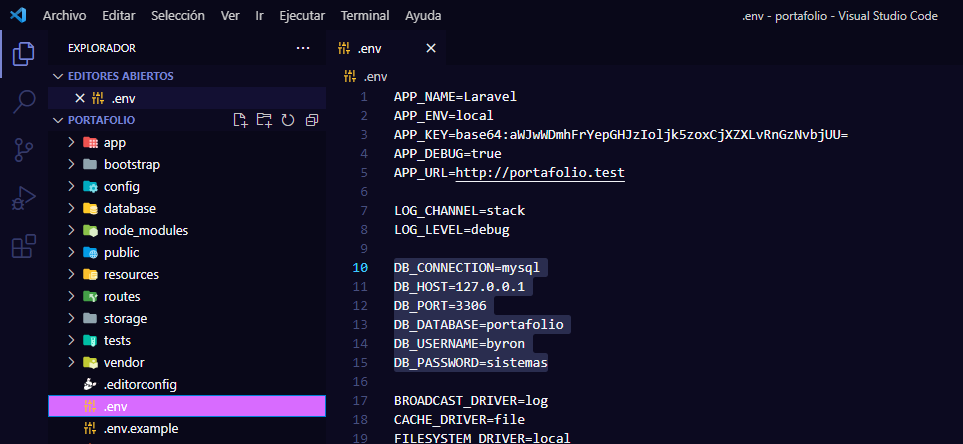
1. Verificar nuevamente con la ejecución del proyecto, y se puede apreciar la ejecución del proyecto con un sistema de inicio de sesión y registro.



1. Posterior a ello se procede con la creación de una Base de Datos llamada portafolio.



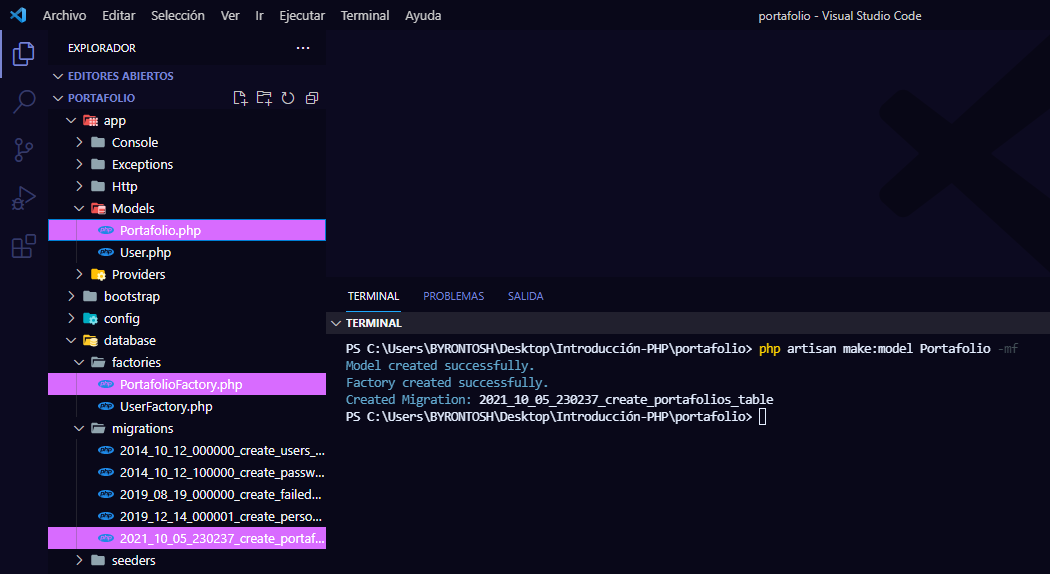
1. Ahora se debe proceder con la configuración de la Base de Datos en los archivos de configuración del proyecto.



1. Ahora se procede a trabajar con la creación de un modelo y una migración ejecutando el siguiente comando.

**php artisan make:model Portafolio –mf**

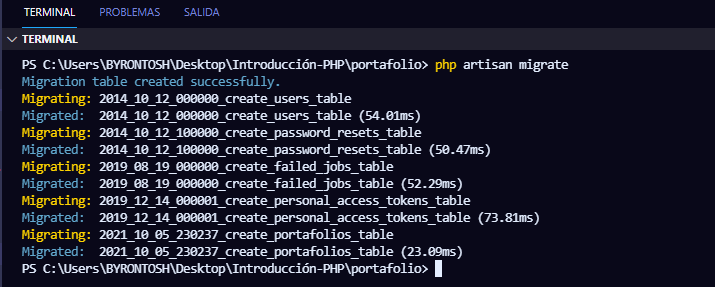
<https://laravel.com/docs/9.x/eloquent#generating-model-classes>

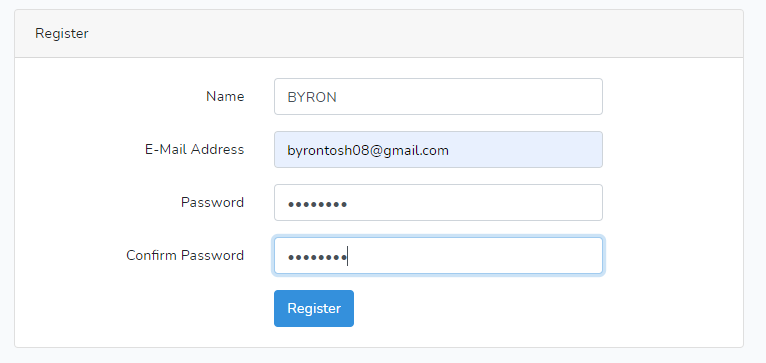


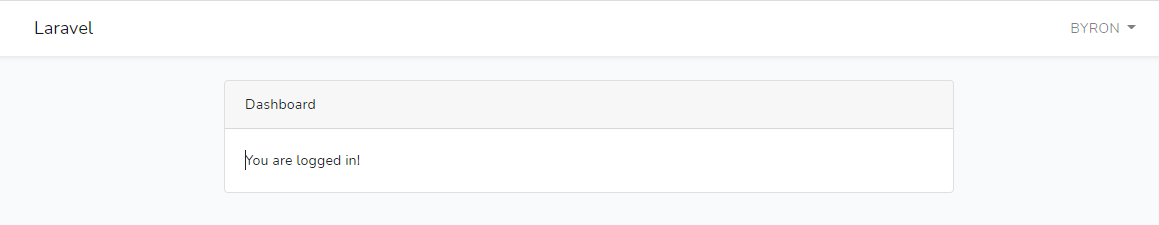
1. Para probar el paquete agregado anteriormente se debe ejecutar las migraciones y luego verificar

**php artisan migrate**

<https://laravel.com/docs/9.x/migrations#running-migrations>







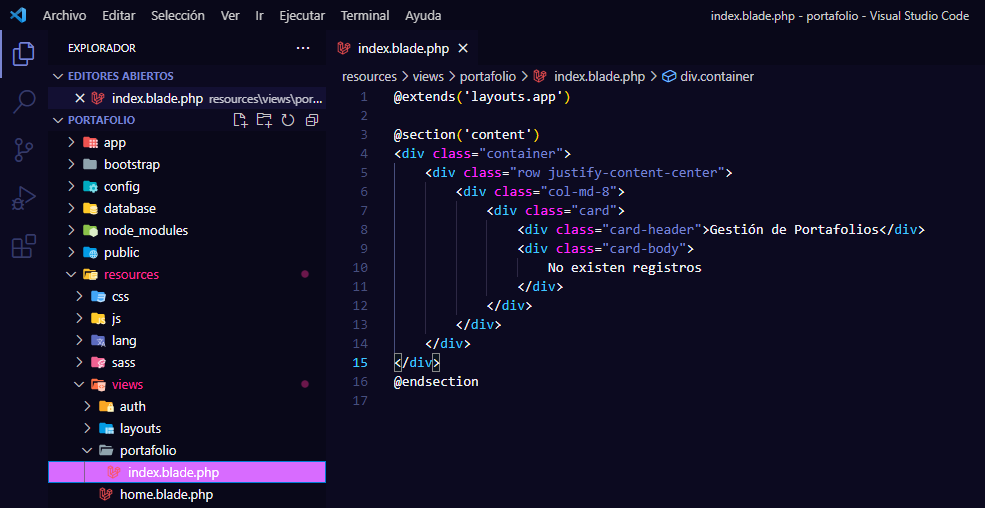


1. Ahora se procede a crear lo siguiente:

* Ruta de portafolio
* Vista de portafolio
* Enlace en la plantilla



*Route*::view('portafolio','portafolio.index')->name('portafolio');



@extends('layouts.app')

@section('content')

    <div class="container">

        <div class="row justify-content-center">

            <div class="col-md-12">

                <div class="card">

                    <div class=" card-header card-body d-flex justify-content-between align-items-center">

                        Gestión de Portafolios

                    </div>

                    <div class="card-body">

                        <p>No Existen proyectos</p>

                    </div>

                </div>

            </div>

        </div>

        <br>

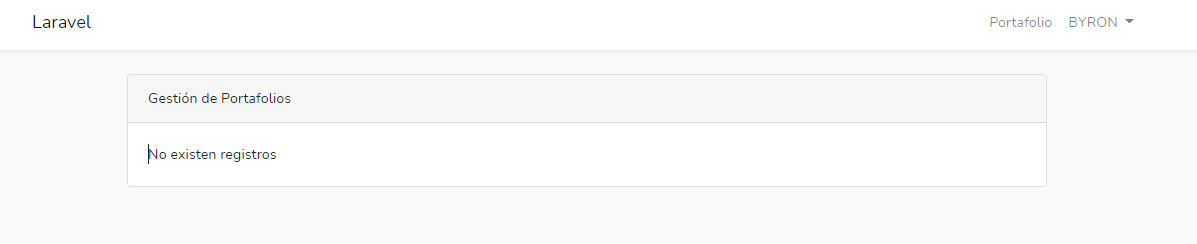
    @endsection



                        <li class="nav-item">

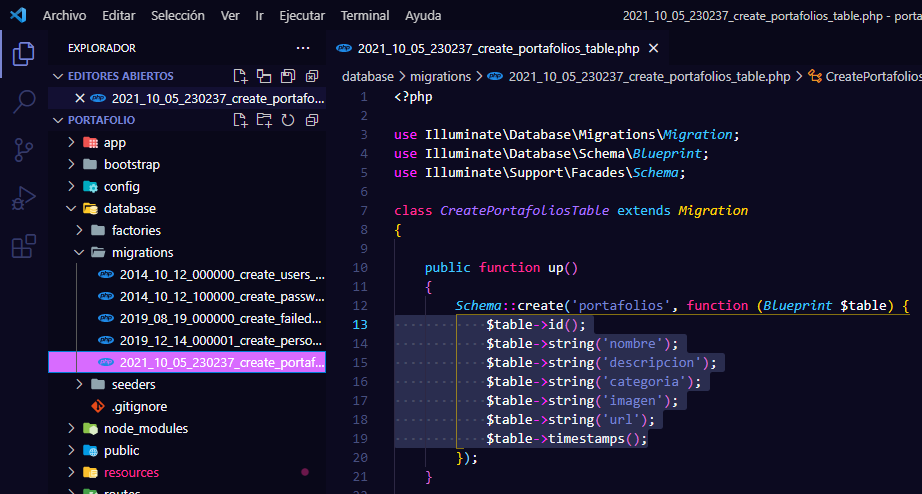
                            <a class="nav-link" href="{{ route('portafolio') }}">Portafolio</a>

                        </li>



1. Ahora se procede a definir los campos para el modelo creado anteriormente, cabe recalcar que estos campos se deben definir en la migración.

<https://laravel.com/docs/9.x/migrations#tables>



            $table->id();

            $table->string('nombre');

            $table->string('descripcion');

            $table->string('categoria');

            $table->string('imagen');

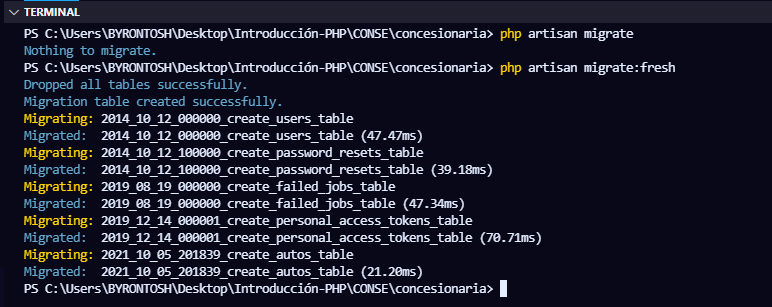
            $table->string('url');

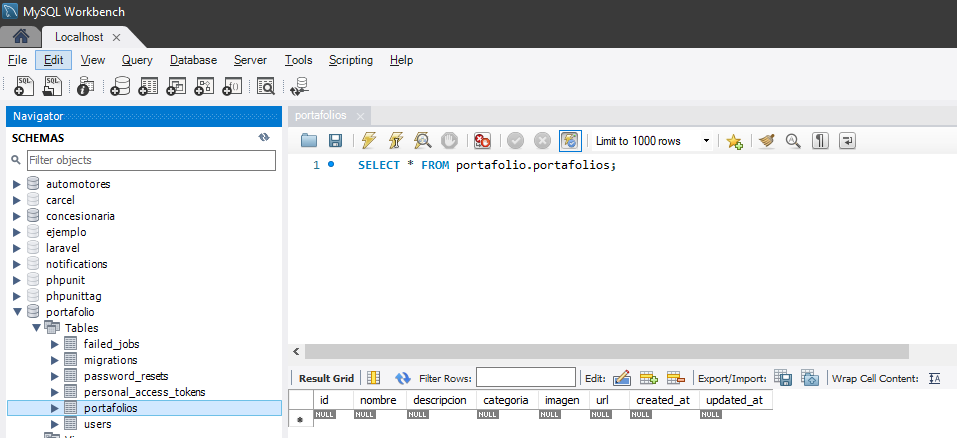
            $table->timestamps();

1. Para ver los cambios aplicados, se debe ejecutar el siguiente comando y ver la tabla de la base de datos.

**php artisan migrate**

**php artisan migrate:fresh**

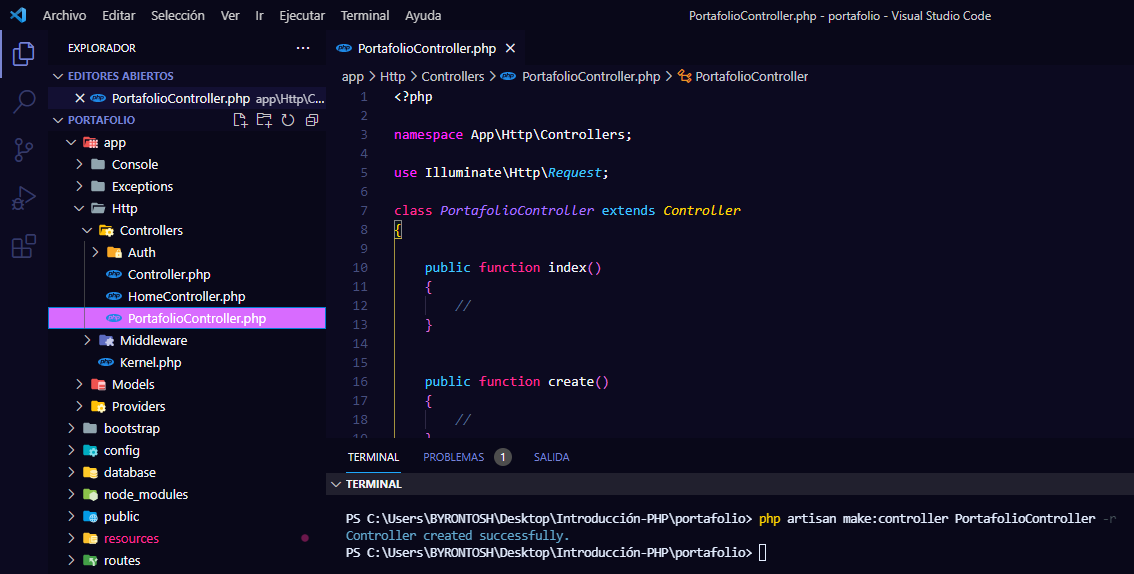




1. Ahora se procede con la creación de un controlador para la gestión de la información, para lo cual se procede a ejecutar el siguiente comando.

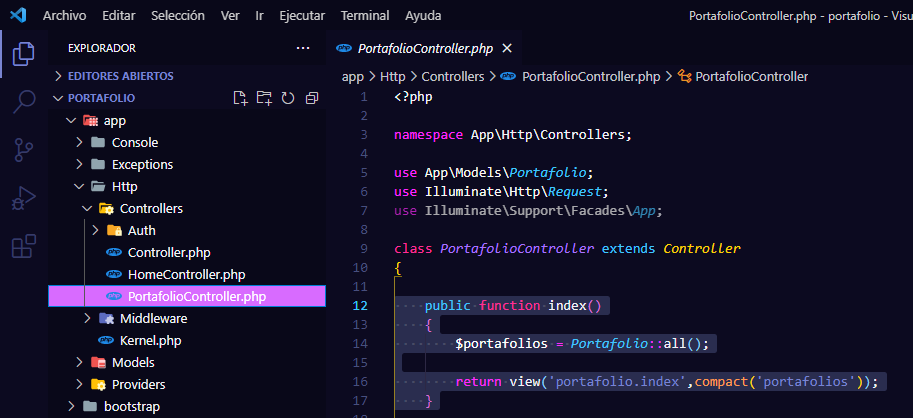
**php artisan make:controller PortafolioController -r**

<https://laravel.com/docs/9.x/controllers#resource-controllers>



1. Se procede a trabajar en la primera función del controlador

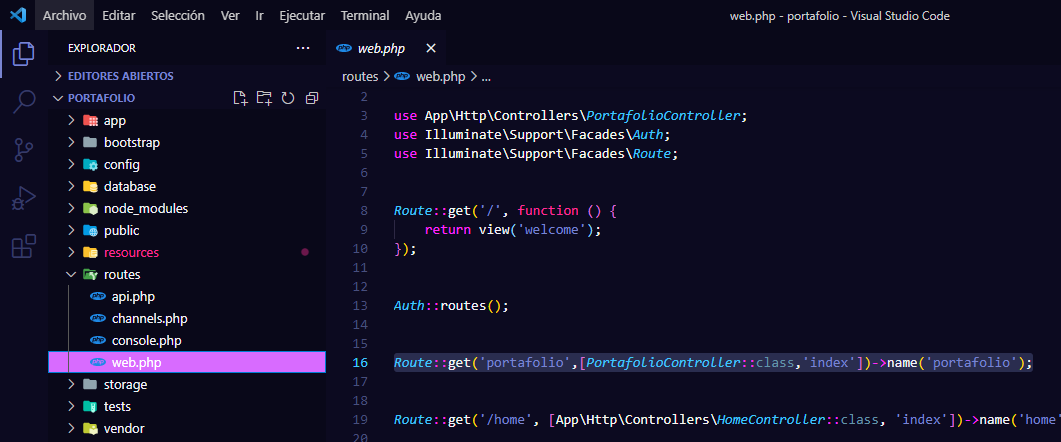
<https://laravel.com/docs/9.x/eloquent#retrieving-models>



        $portafolios = *Portafolio*::all();

        return view('portafolio.index',compact('portafolios'));

1. Luego se procede a trabajar en la ruta para llamar al controlador sin descuidarse de importar el controlador y luego trabajar en la vista correspondiente.



*Route*::get('portafolio',[*PortafolioController*::class,'index'])->name('portafolio');

<https://laravel.com/docs/9.x/blade#loops>



@extends('layouts.app')

@section('content')

<div class="container">

    <div class="row justify-content-center">

        <div class="col-md-12">

            <div class="card">

                <div class=" card-header card-body d-flex justify-content-between align-items-center">

                    Gestión de Portafolios

                    <a  class="btn btn-primary btn-sm">Crear</a>

                </div>

            </div>

        </div>

    </div>

    <br>

    @forelse ($portafolios as $portafolio)

    <h5 class="card-title">Nombre: {{ $portafolio->nombre }}</h5>

    <h5 class="card-title">Descripción: {{ $portafolio->descripcion }}</h5>

    <h5 class="card-title">Categoría: {{ $portafolio->categoria }}</h5>

    <h5 class="card-title">Imagen: {{ $portafolio->imagen }}</h5>

    <h5 class="card-title">Vídeo: {{ $portafolio->url }}</h5>

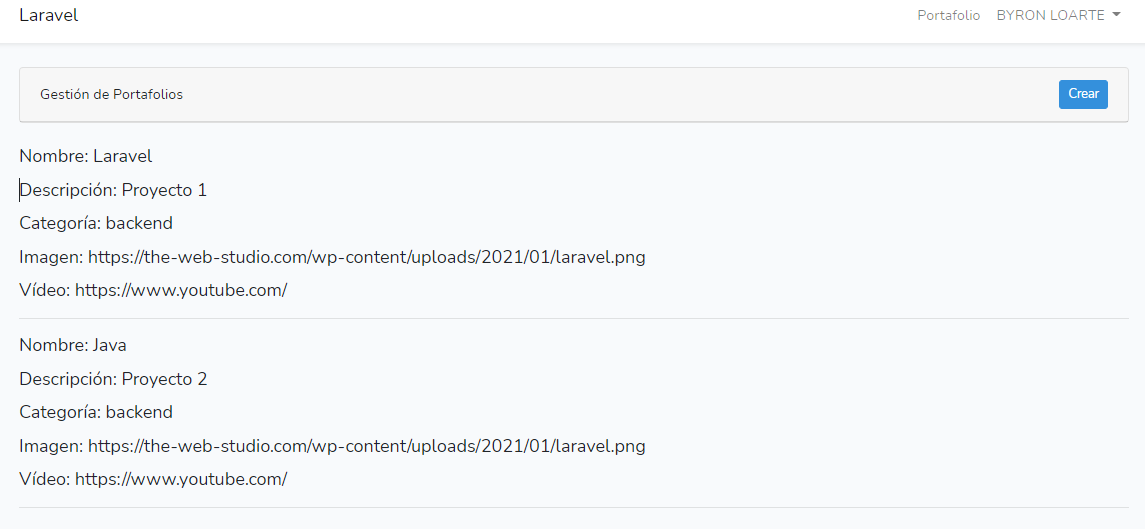
    <hr>

    @empty

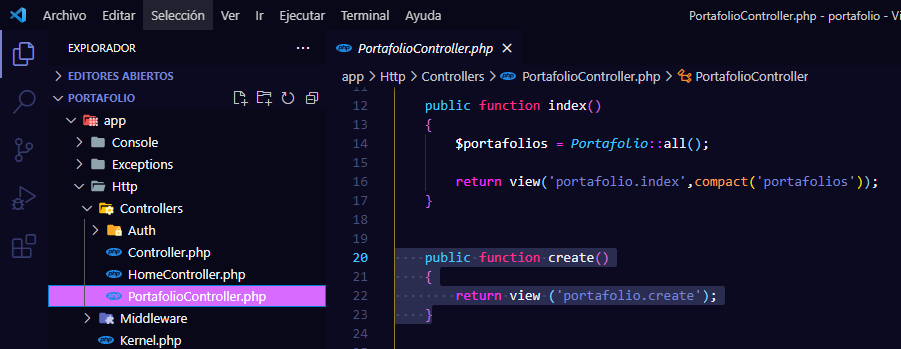
    <p>No Existen proyectos</p>

    @endforelse

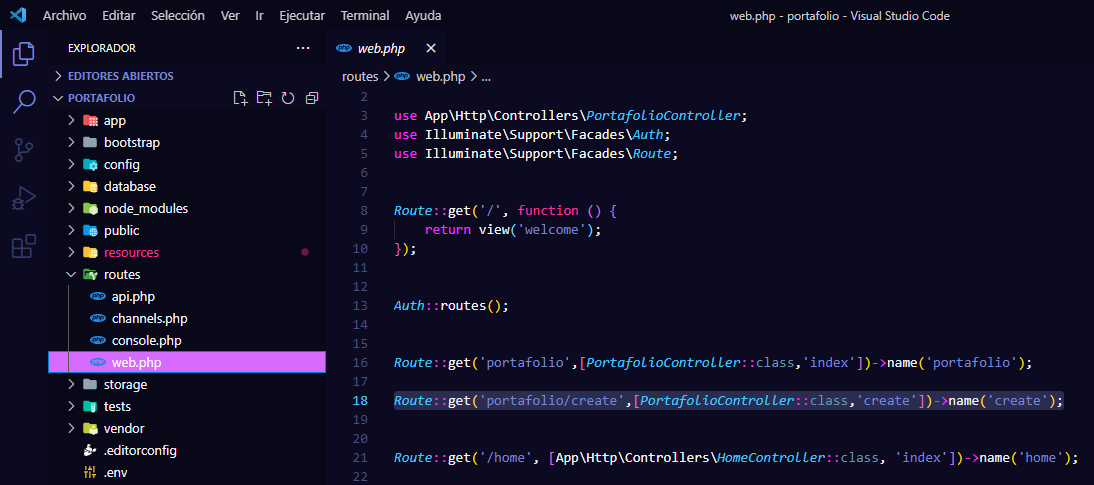
    @endsection



1. Ahora se procede a crear el segundo método para el controlador, la ruta y la vista correspondiente.



        return view ('portafolio.create');



*Route*::get('portafolio/create',[*PortafolioController*::class,'create'])->name('create');



@extends('layouts.app')

@section('content')

<div class="container">

    <div class="row justify-content-center">

        <div class="col-md-12">

            <div class="card">

            <div class=" card-header card-body d-flex justify-content-between align-items-center">

                    Creación de un Portafolio

                    <a href="{{route('portafolio')}}" class="btn btn-danger btn-sm">Regresar</a>

                </div>

                <div class="card-body">

                    <form  method="POST">

                        @csrf

                        <div class="form-group row mb-3">

                            <label for="colFormLabelSm" class="col-sm-2 col-form-label col-form-label-sm">Nombre</label>

                            <div class="col-sm-10">

                                <input type="text" name="nombre" class="form-control form-control-sm" id="colFormLabelSm" placeholder="Ingrese información">

                            </div>

                        </div>

                        <div class="form-group row mb-3">

                            <label for="colFormLabelSm" class="col-sm-2 col-form-label col-form-label-sm">Descripción</label>

                            <div class="col-sm-10">

                            <textarea class="form-control" name="descripcion" id="exampleFormControlTextarea1" rows="3"></textarea>

                            </div>

                        </div>

                        <div class="form-group row mb-3">

                            <label for="colFormLabelSm" class="col-sm-2 col-form-label col-form-label-sm">Categoría</label>

                            <div class="col-sm-10">

                                <input type="text" name="categoria" class="form-control form-control-sm" id="colFormLabelSm" placeholder="Ingrese información">

                            </div>

                        </div>

                        <div class="form-group row mb-3">

                            <label for="colFormLabelSm" class="col-sm-2 col-form-label col-form-label-sm">Imagen</label>

                            <div class="col-sm-10">

                                <input type="text" name="imagen" class="form-control form-control-sm" id="colFormLabelSm" placeholder="Ingrese información">

                            </div>

                        </div>

                        <div class="form-group row mb-3">

                            <label for="colFormLabelSm" class="col-sm-2 col-form-label col-form-label-sm">Vídeo</label>

                            <div class="col-sm-10">

                                <input type="text" name="video"  class="form-control form-control-sm" id="colFormLabelSm" placeholder="Ingrese información">

                            </div>

                        </div>

                        <button type="submit" class="btn btn-primary w-100">Registrar</button>

                    </form>

                </div>

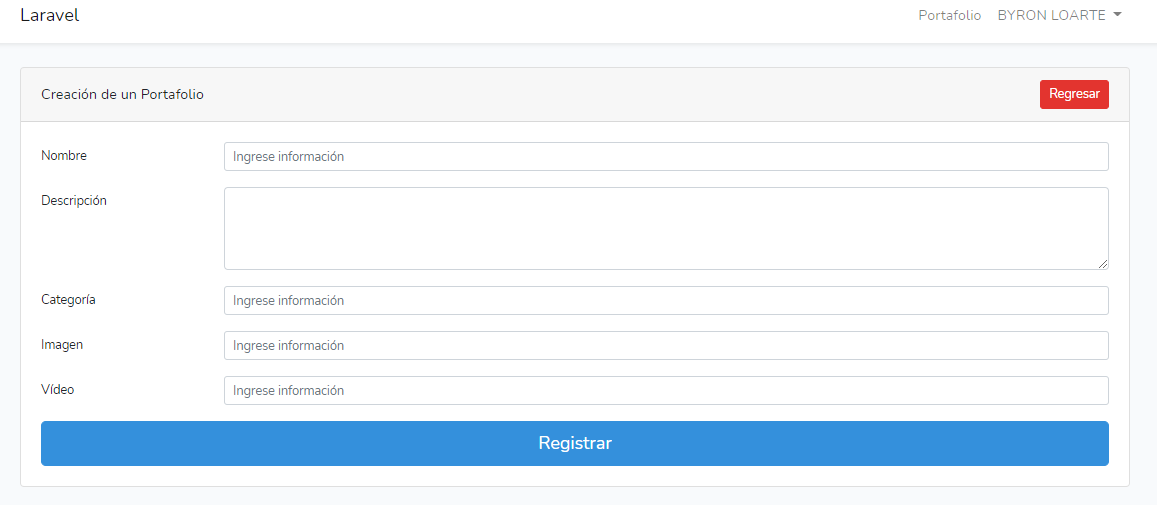
            </div>

        </div>

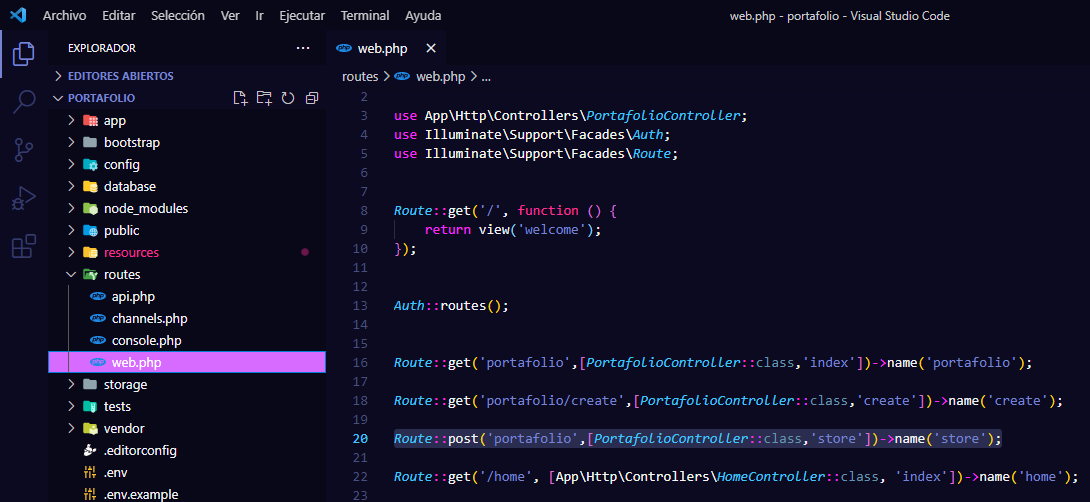
    </div>

    <br>

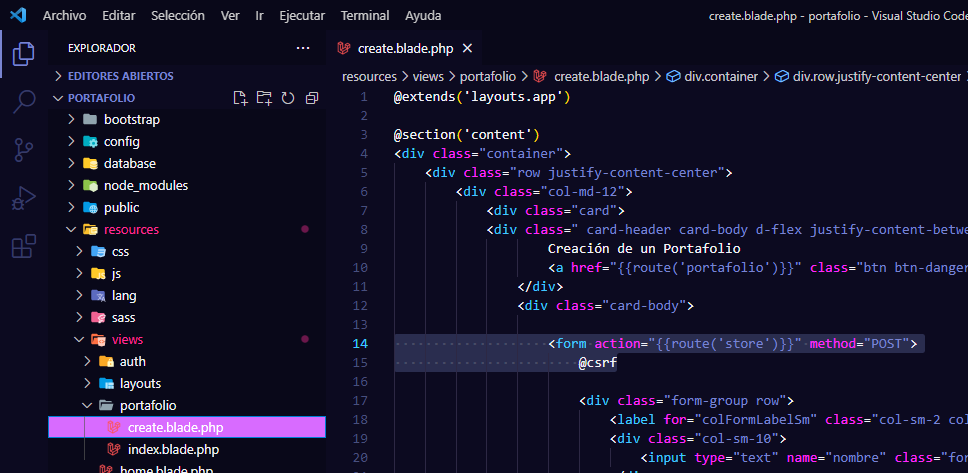
    @endsection



1. Ahora se procede a crear el tercer método para el controlador

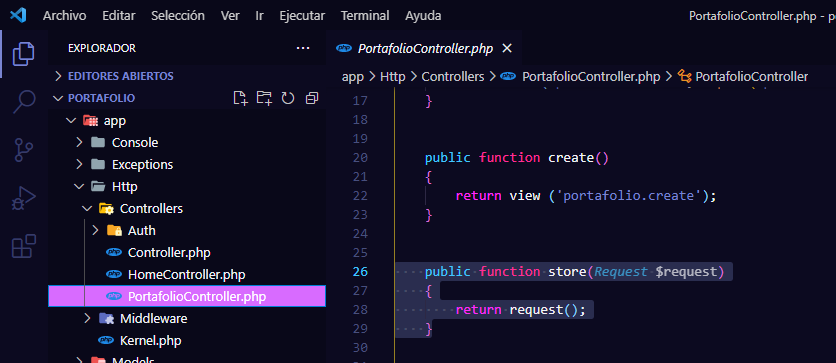


*Route*::post('portafolio',[*PortafolioController*::class,'store'])->name('store');

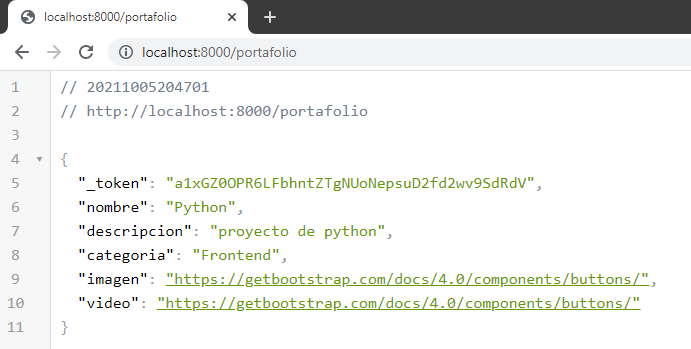


                    <form action="{{route('store')}}" method="POST">

                        @csrf

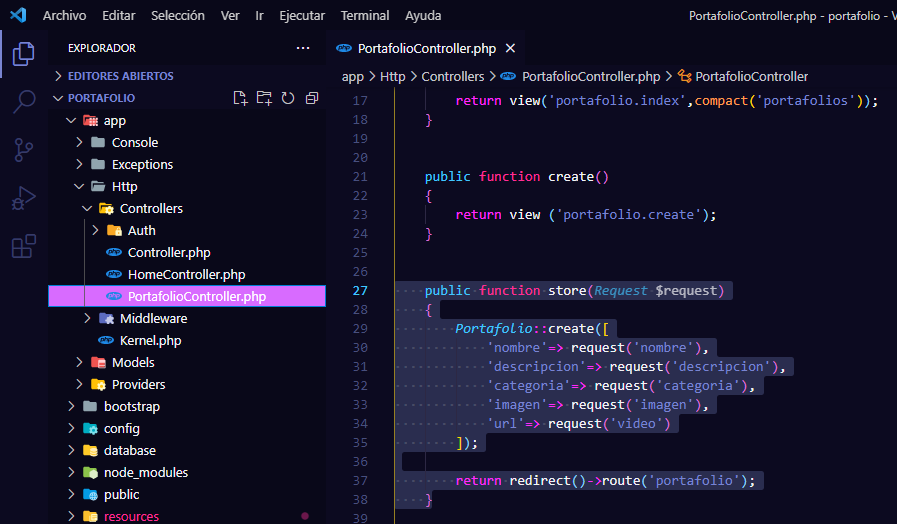


        return request();



<https://laravel.com/docs/9.x/eloquent#inserts>

<https://laravel.com/docs/9.x/responses#redirects>



*Portafolio*::create([

            'nombre'=> request('nombre'),

            'descripcion'=> request('descripcion'),

            'categoria'=> request('categoria'),

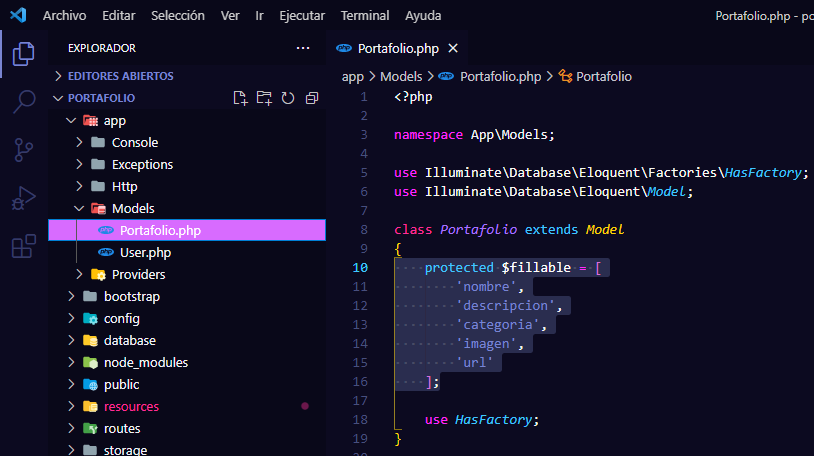
            'imagen'=> request('imagen'),

            'url'=> request('video')

        ]);

        return redirect()->route('portafolio');

<https://laravel.com/docs/9.x/eloquent#mass-assignment>



    protected $fillable = [

        'nombre',

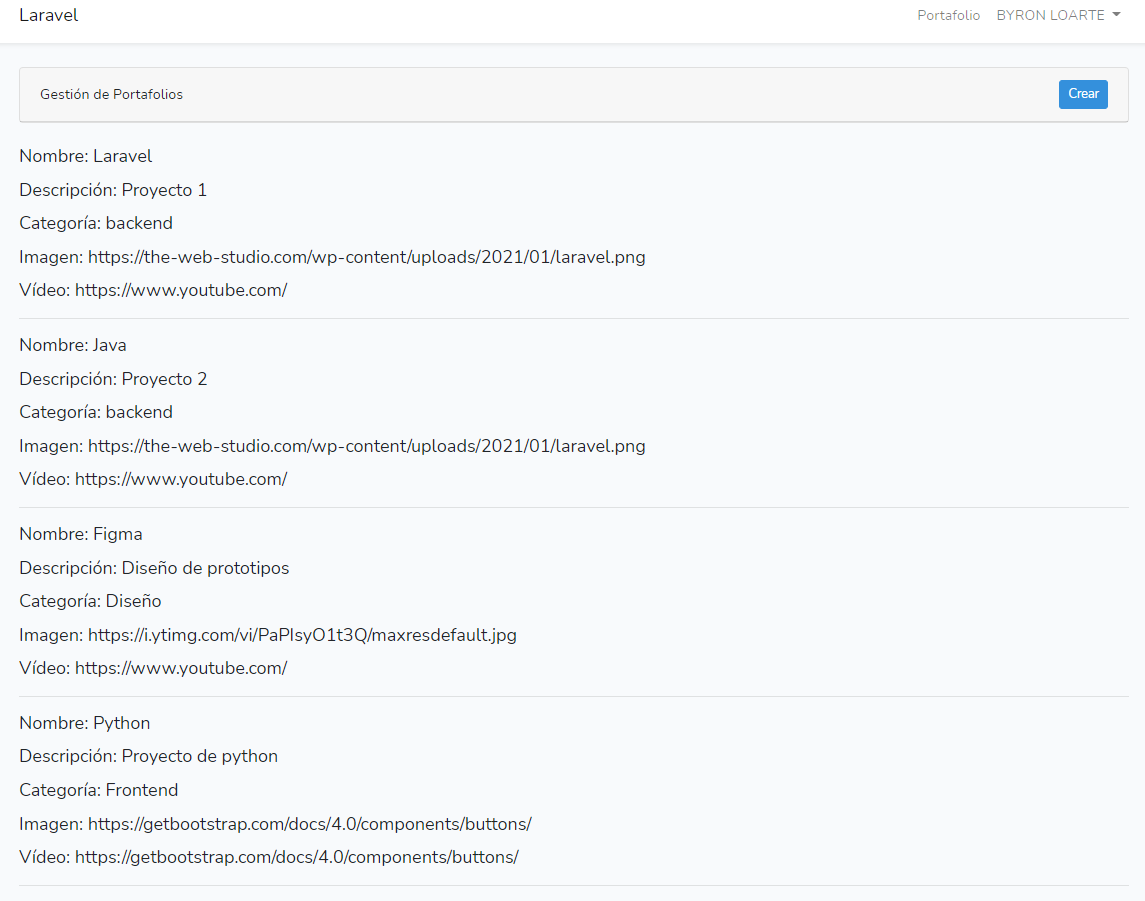
        'descripcion',

        'categoria',

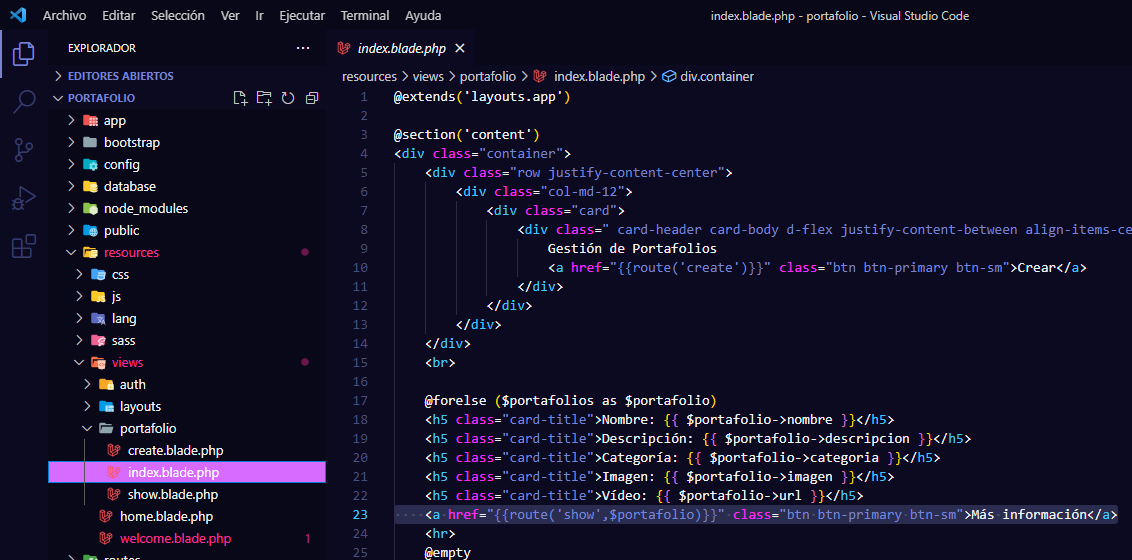
        'imagen',

        'url'

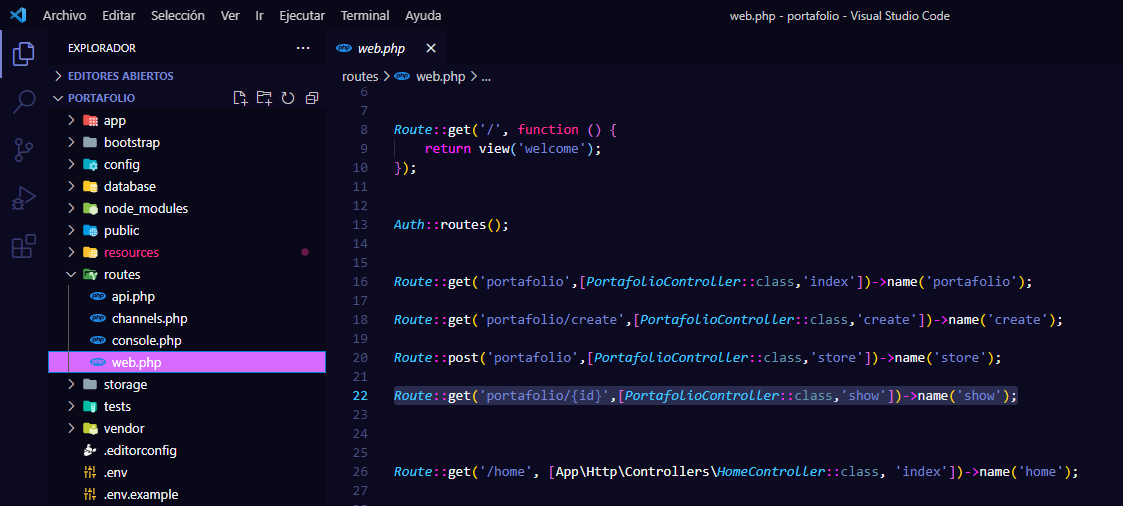
    ];



1. Ahora se procede a crear el tercer método del controlador, la ruta y la vista respectiva

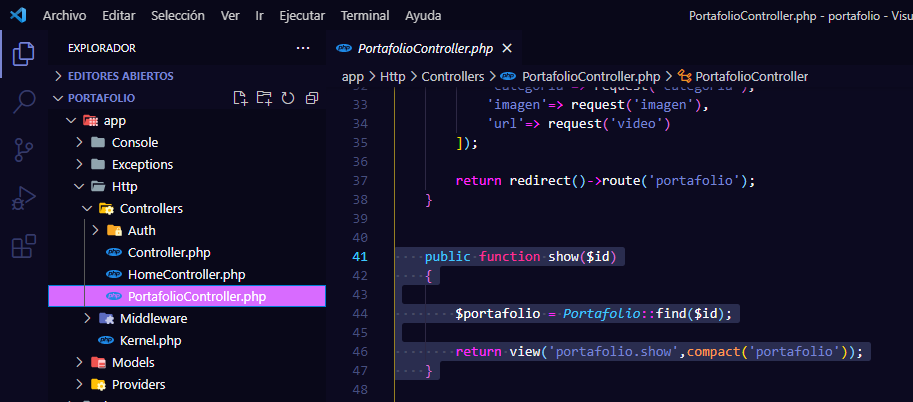


    <a href="{{route('show',$portafolio)}}" class="btn btn-primary btn-sm">Más información</a>



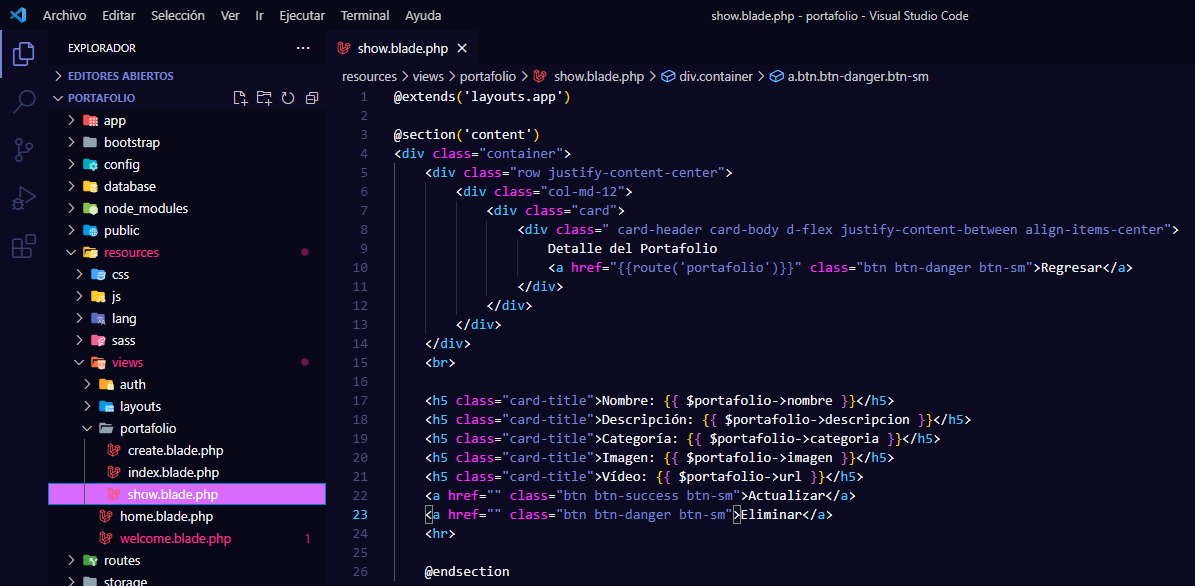
*Route*::get('portafolio/{id}',[*PortafolioController*::class,'show'])->name('show');

<https://laravel.com/docs/9.x/eloquent#retrieving-single-models>



        $portafolio = *Portafolio*::find($id);

        return view('portafolio.show',compact('portafolio'));



@extends('layouts.app')

@section('content')

<div class="container">

    <div class="row justify-content-center">

        <div class="col-md-12">

            <div class="card">

                <div class=" card-header card-body d-flex justify-content-between align-items-center">

                    Detalle del Portafolio

                    <a href="{{route('portafolio')}}" class="btn btn-danger btn-sm">Regresar</a>

                </div>

            </div>

        </div>

    </div>

    <br>

    <h5 class="card-title">Nombre: {{ $portafolio->nombre }}</h5>

    <h5 class="card-title">Descripción: {{ $portafolio->descripcion }}</h5>

    <h5 class="card-title">Categoría: {{ $portafolio->categoria }}</h5>

    <h5 class="card-title">Imagen: {{ $portafolio->imagen }}</h5>

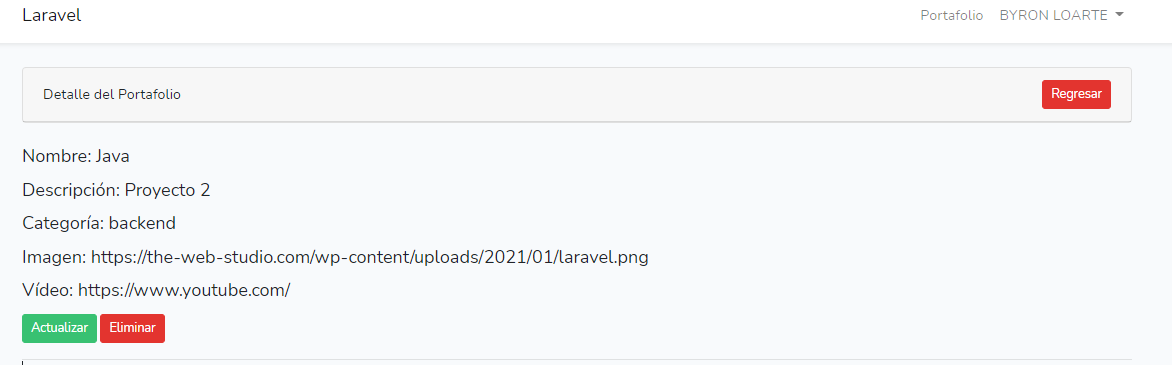
    <h5 class="card-title">Vídeo: {{ $portafolio->url }}</h5>

    <a href="" class="btn btn-success btn-sm" >Actualizar</a>

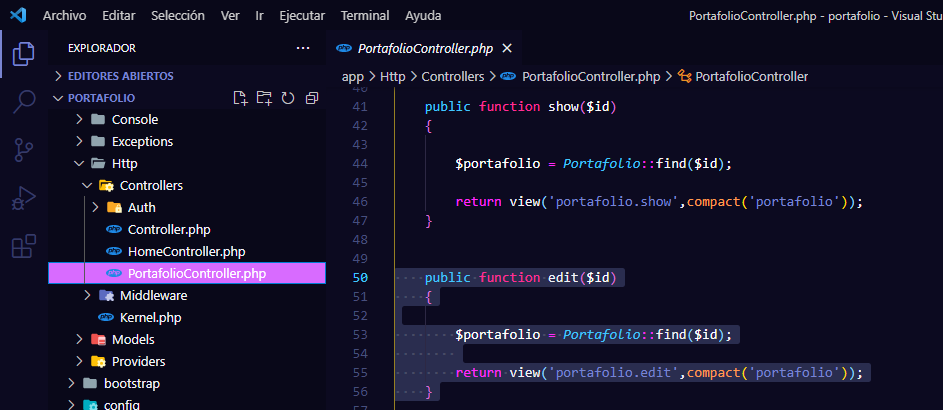
    <a href="" class="btn btn-danger btn-sm" >Eliminar</a>

    <hr>

    @endsection

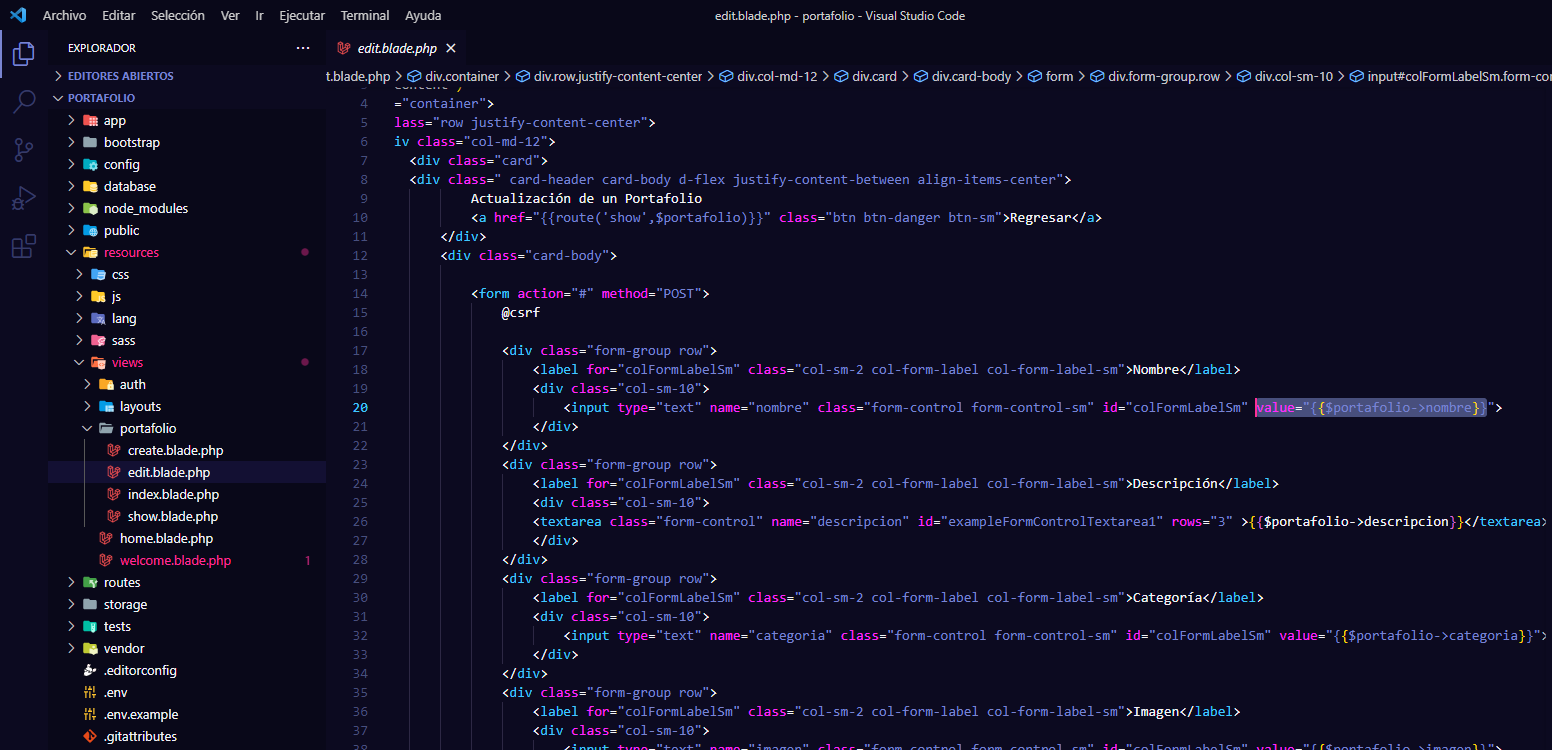


1. Ahora se procede a crear el cuarto método del controlador, la ruta y la vista respectiva



        $portafolio = *Portafolio*::find($id);

        return view('portafolio.edit',compact('portafolio'));



@extends('layouts.app')

@section('content')

<div class="container">

    <div class="row justify-content-center">

        <div class="col-md-12">

            <div class="card">

            <div class=" card-header card-body d-flex justify-content-between align-items-center">

                    Actualización de un Portafolio

                    <a href="{{route('show',$portafolio)}}" class="btn btn-danger btn-sm">Regresar</a>

                </div>

                <div class="card-body">

                    <form action="#" method="POST">

                        @csrf

                        @method('PUT')

                        <div class="form-group row mb-3">

                            <label for="colFormLabelSm" class="col-sm-2 col-form-label col-form-label-sm">Nombre</label>

                            <div class="col-sm-10">

                                <input type="text" name="nombre" class="form-control form-control-sm" id="colFormLabelSm" value="{{$portafolio->nombre}}">

                            </div>

                        </div>

                        <div class="form-group row mb-3">

                            <label for="colFormLabelSm" class="col-sm-2 col-form-label col-form-label-sm">Descripción</label>

                            <div class="col-sm-10">

                            <textarea class="form-control" name="descripcion" id="exampleFormControlTextarea1" rows="3" >{{$portafolio->descripcion}}</textarea>

                            </div>

                        </div>

                        <div class="form-group row mb-3">

                            <label for="colFormLabelSm" class="col-sm-2 col-form-label col-form-label-sm">Categoría</label>

                            <div class="col-sm-10">

                                <input type="text" name="categoria" class="form-control form-control-sm" id="colFormLabelSm" value="{{$portafolio->categoria}}">

                            </div>

                        </div>

                        <div class="form-group row mb-3">

                            <label for="colFormLabelSm" class="col-sm-2 col-form-label col-form-label-sm">Imagen</label>

                            <div class="col-sm-10">

                                <input type="text" name="imagen" class="form-control form-control-sm" id="colFormLabelSm" value="{{$portafolio->imagen}}">

                            </div>

                        </div>

                        <div class="form-group row mb-3">

                            <label for="colFormLabelSm" class="col-sm-2 col-form-label col-form-label-sm">Vídeo</label>

                            <div class="col-sm-10">

                                <input type="text" name="video"  class="form-control form-control-sm" id="colFormLabelSm" value="{{$portafolio->url}}">

                            </div>

                        </div>

                        <button type="submit" class="btn btn-primary w-100">Actualizar</button>

                    </form>

                </div>

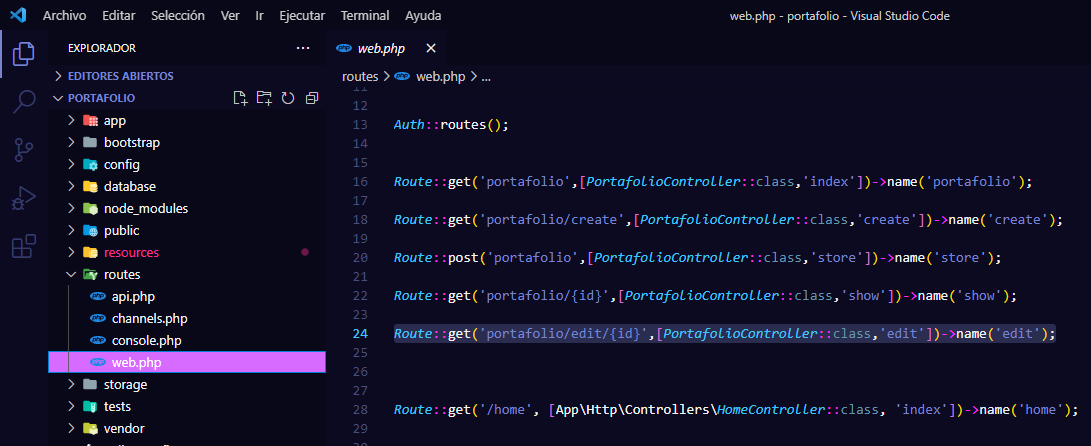
            </div>

        </div>

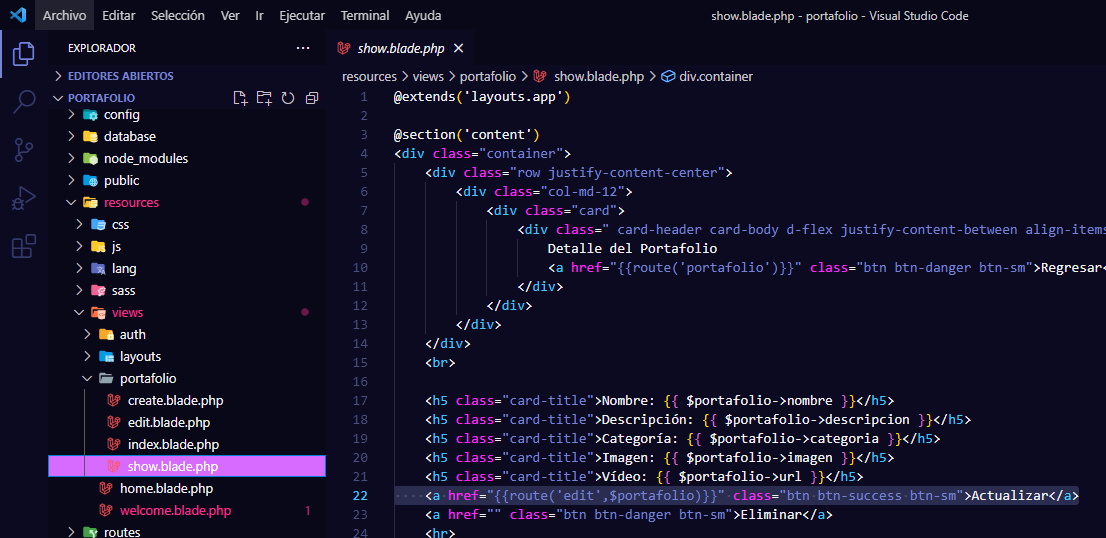
    </div>

    <br>

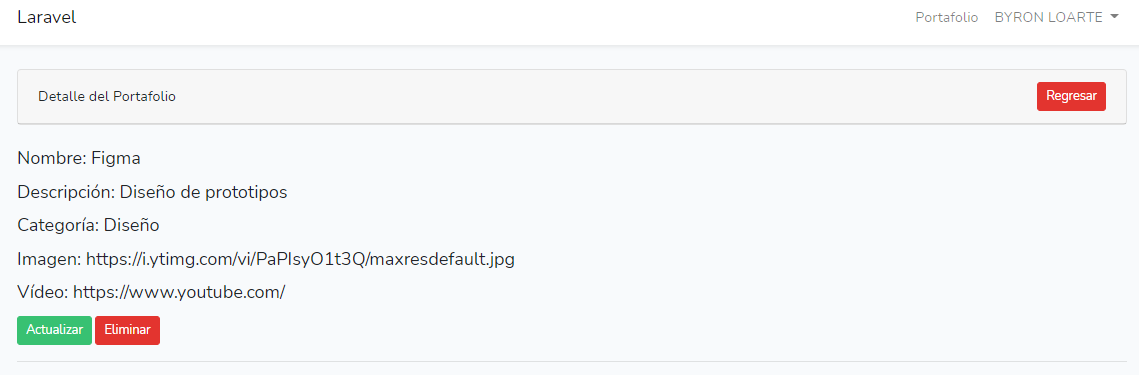
    @endsection

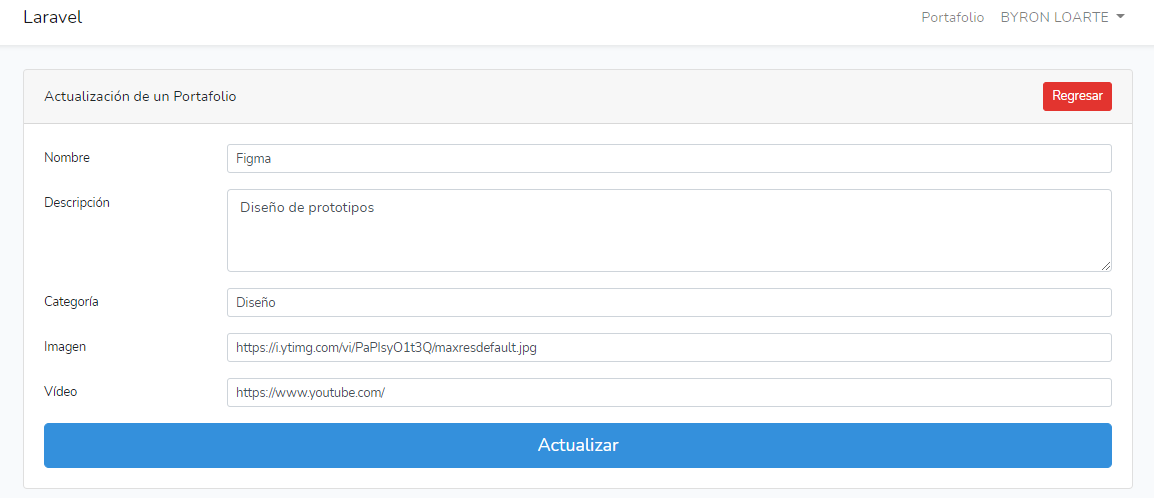


*Route*::get('portafolio/edit/{id}',[*PortafolioController*::class,'edit'])->name('edit');



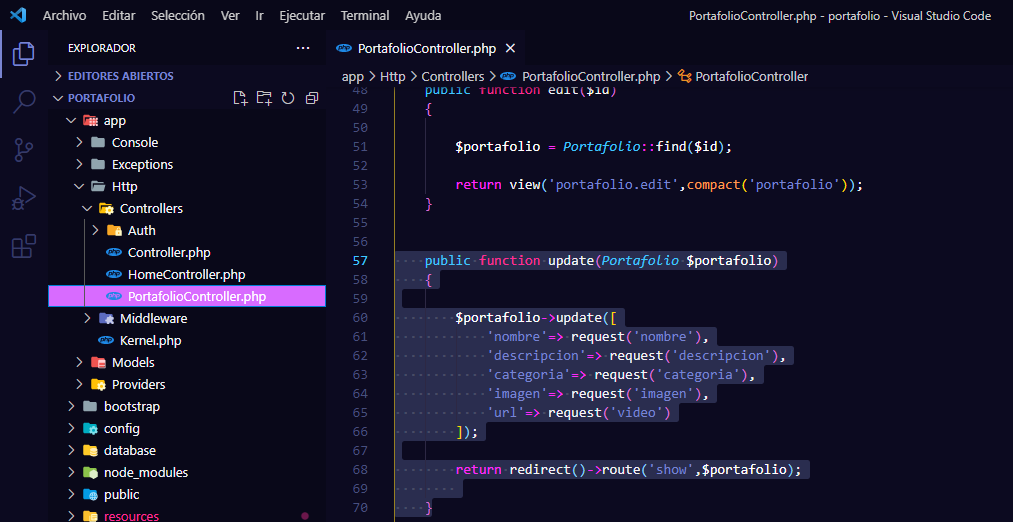
    <a href="{{route('edit',$portafolio)}}" class="btn btn-success btn-sm" >Actualizar</a>





1. Ahora se procede a crear el quinto método del controlador, la ruta y la vista respectiva.

<https://laravel.com/docs/9.x/eloquent#updates>



    public function update(*Portafolio* $portafolio)

    {

        $portafolio->update([

            'nombre'=> request('nombre'),

            'descripcion'=> request('descripcion'),

            'categoria'=> request('categoria'),

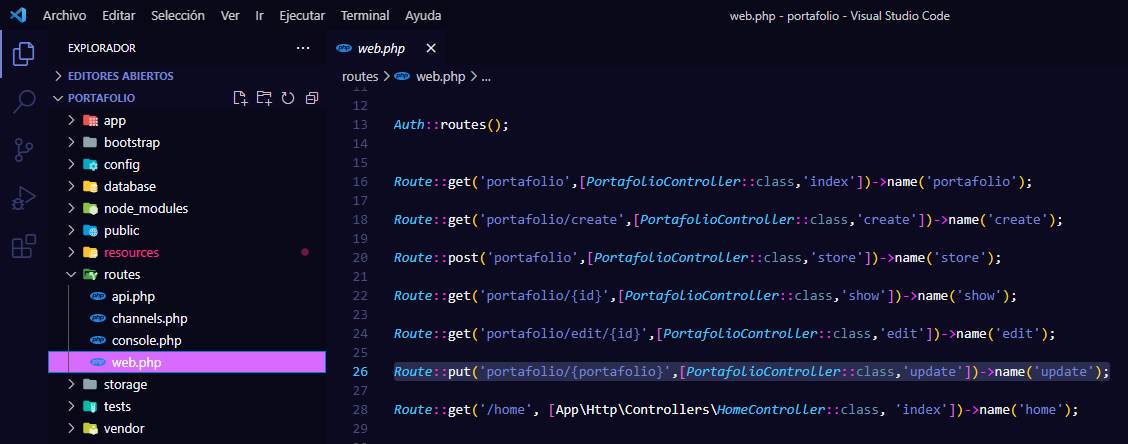
            'imagen'=> request('imagen'),

            'url'=> request('video')

        ]);

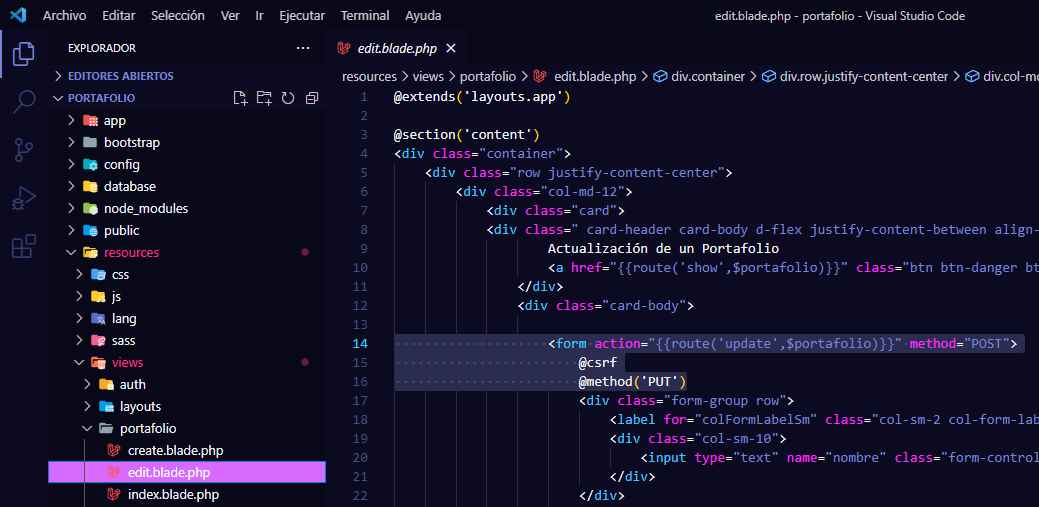
        return redirect()->route('show',$portafolio);

    }



*Route*::put('portafolio/{portafolio}',[*PortafolioController*::class,'update'])->name('update');

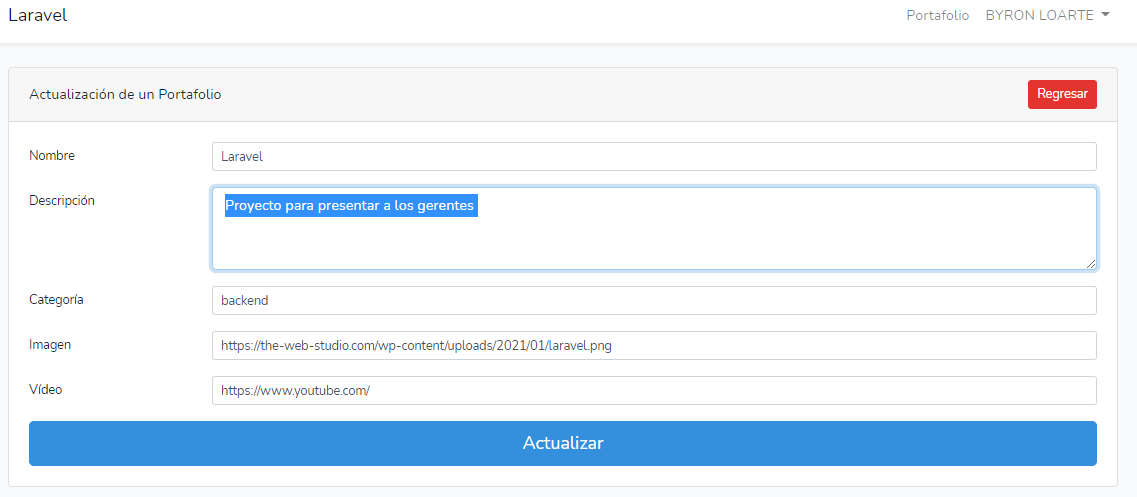
[*https://laravel.com/docs/9.x/csrf#preventing-csrf-requests*](https://laravel.com/docs/9.x/csrf#preventing-csrf-requests)

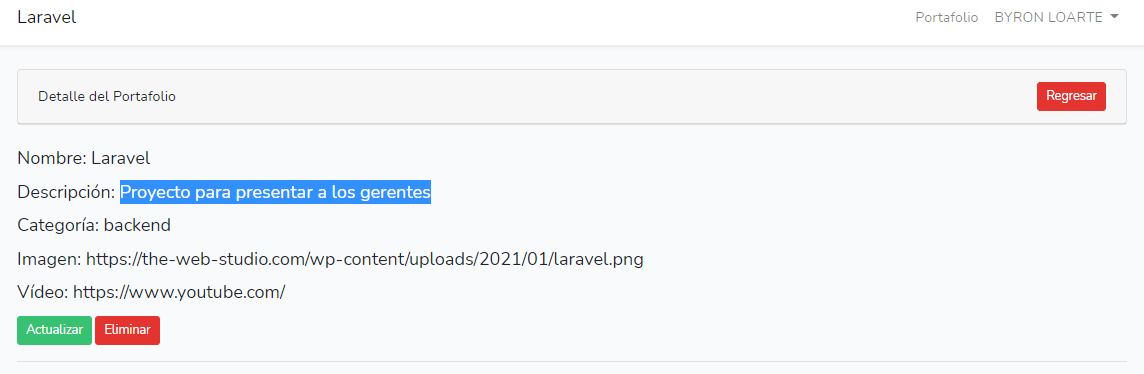


                    <form action="{{route('update',$portafolio)}}" method="POST">

                        @csrf

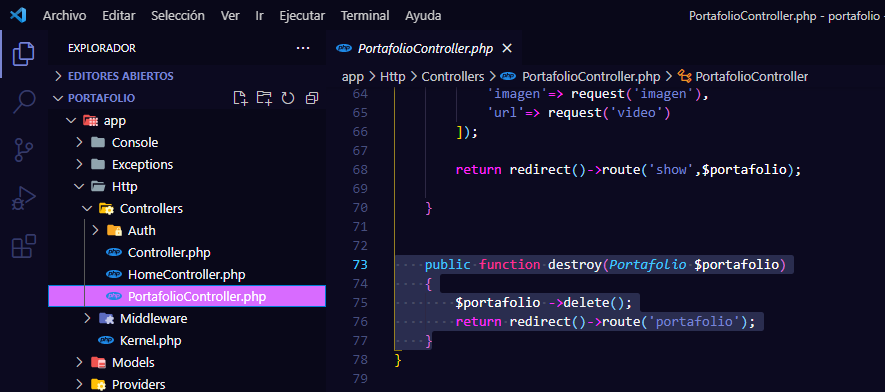
                        @method('PUT')





1. Ahora se procede a crear el sexto método del controlador, la ruta y la vista respectiva.

<https://laravel.com/docs/9.x/eloquent#deleting-models>



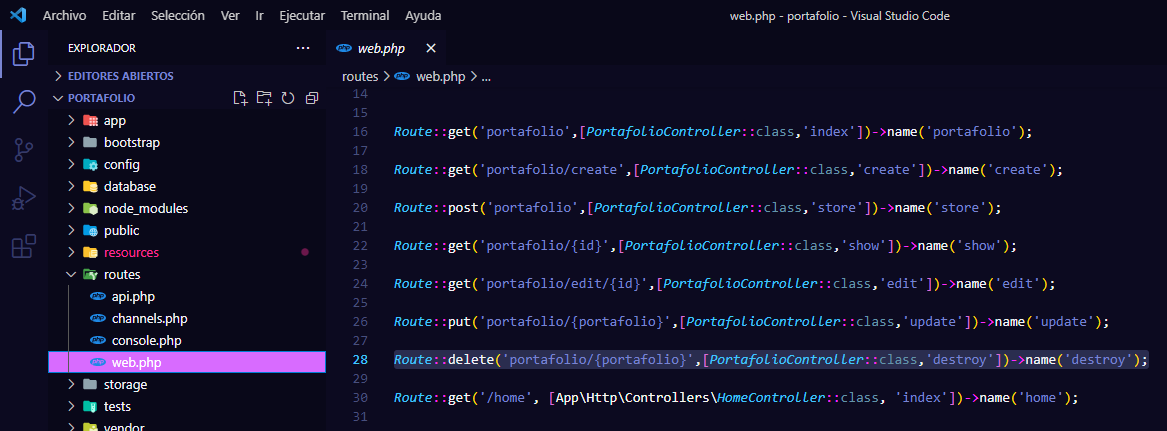
    public function destroy(*Portafolio* $portafolio)

    {

        $portafolio ->delete();

        return redirect()->route('portafolio');

    }



*Route*::delete('portafolio/{portafolio}',[*PortafolioController*::class,'destroy'])->name('destroy');

    <form action="{{route('destroy',$portafolio)}}" method="POST" style="display: inline;" onsubmit="return confirmation()">

        @csrf

        @method('DELETE')

        <button class="btn btn-danger btn-sm" >Eliminar</button>

        <script type="text/javascript">

            function confirmation()

            {

               if(confirm("¿Estás seguro de eliminar el registro?"))

           {

              return true;

           }

           else

           {

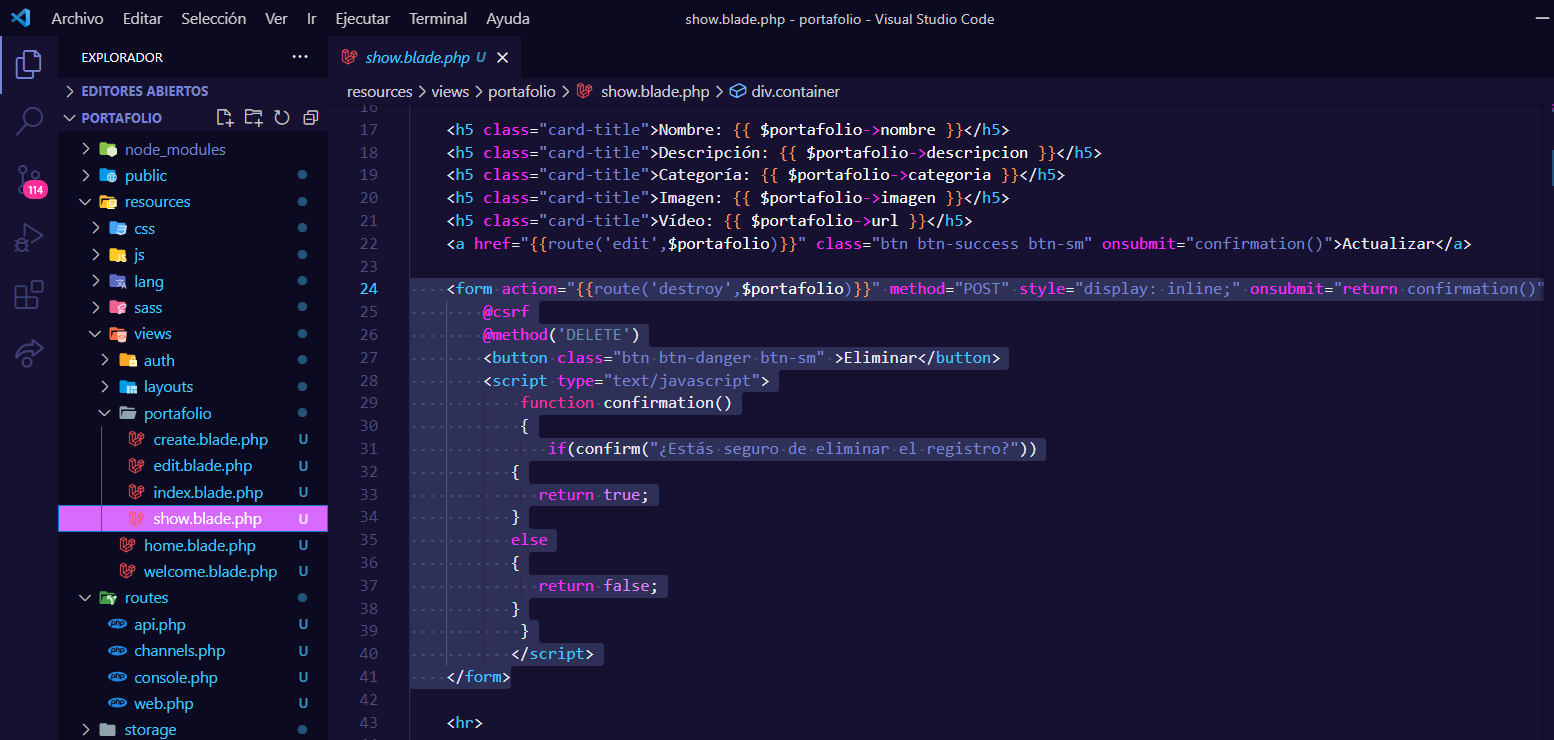
              return false;

           }

            }

           </script>

    </form>



    <form action="{{route('destroy',$portafolio)}}" method="POST" style="display: inline;" onsubmit="return confirmation()">

        @csrf

        @method('DELETE')

        <button class="btn btn-danger btn-sm" >Eliminar</button>

        <script type="text/javascript">

            function confirmation()

            {

               if(confirm("¿Estás seguro de eliminar el registro?"))

           {

              return true;

           }

           else

           {

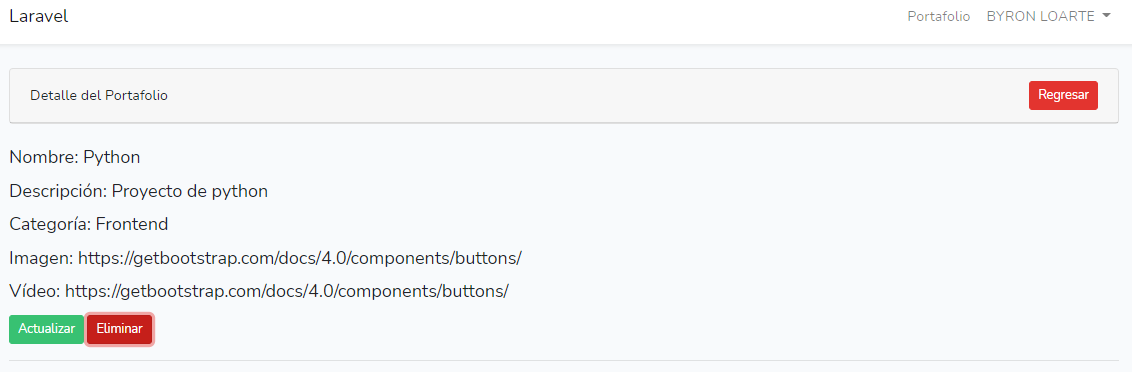
              return false;

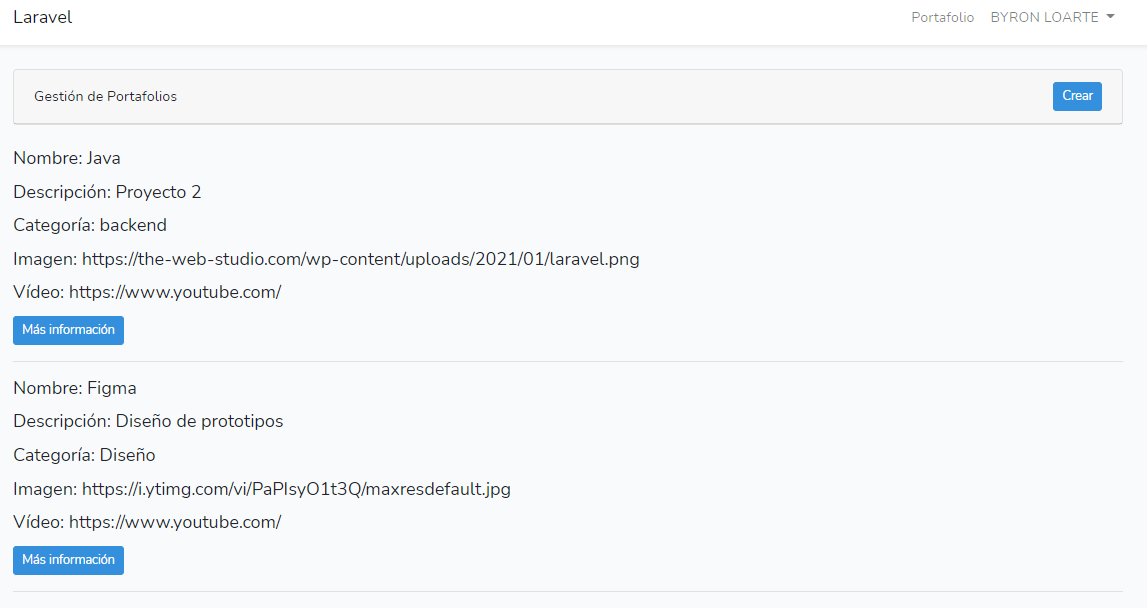
           }

            }

           </script>

    </form>





<https://github.com/Byrontosh/02_Portafolio>

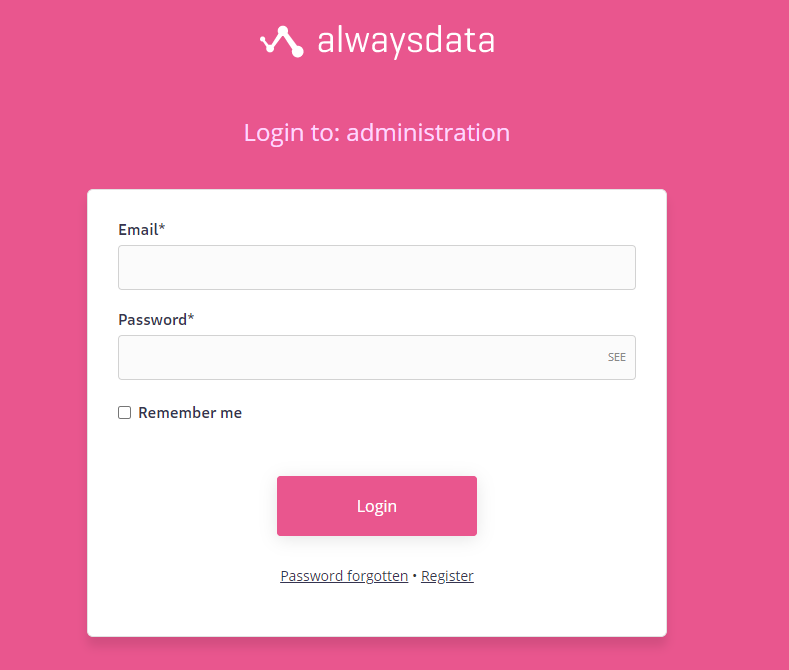
***TAREA DESAFÍO***



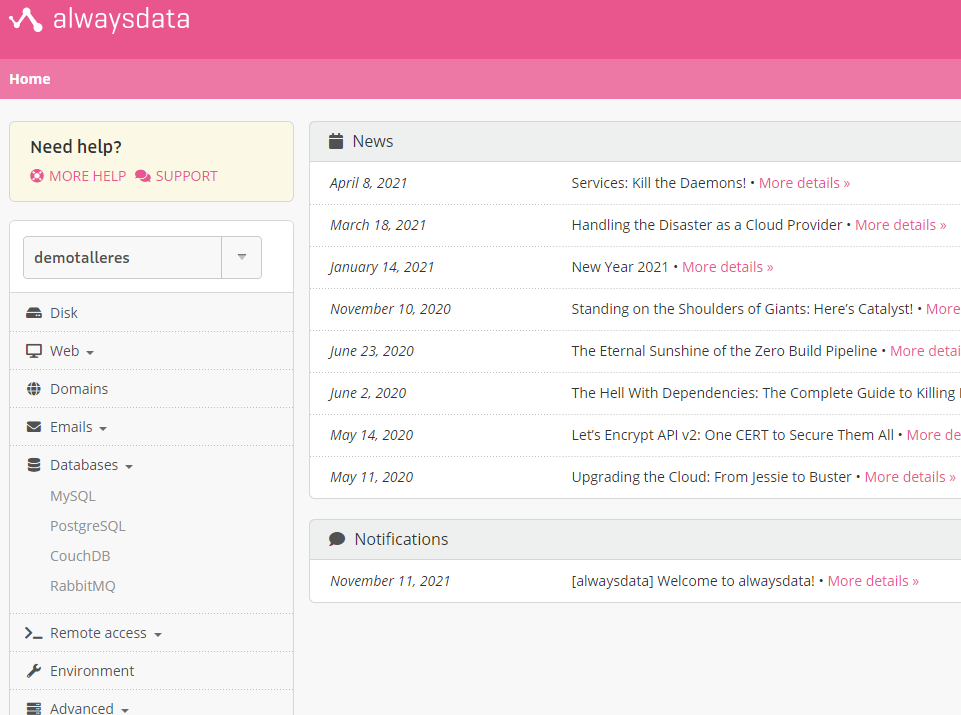
1. Realizar el despliegue a Heroku del sistema web y de la Base de Datos en alwaysdata, siguiendo los siguientes pasos que se describen a continuación.

Lo primero que se debe hacer es crear una Base de Datos en la plataforma alwaysdata ya que esta plataforma dispone de una SGBD MySQL, en cambio Heroku solo dispone de SGBD PostgreSQL. En ese sentido, se procede a crear una cuenta en alwaysdata e iniciar sesión.

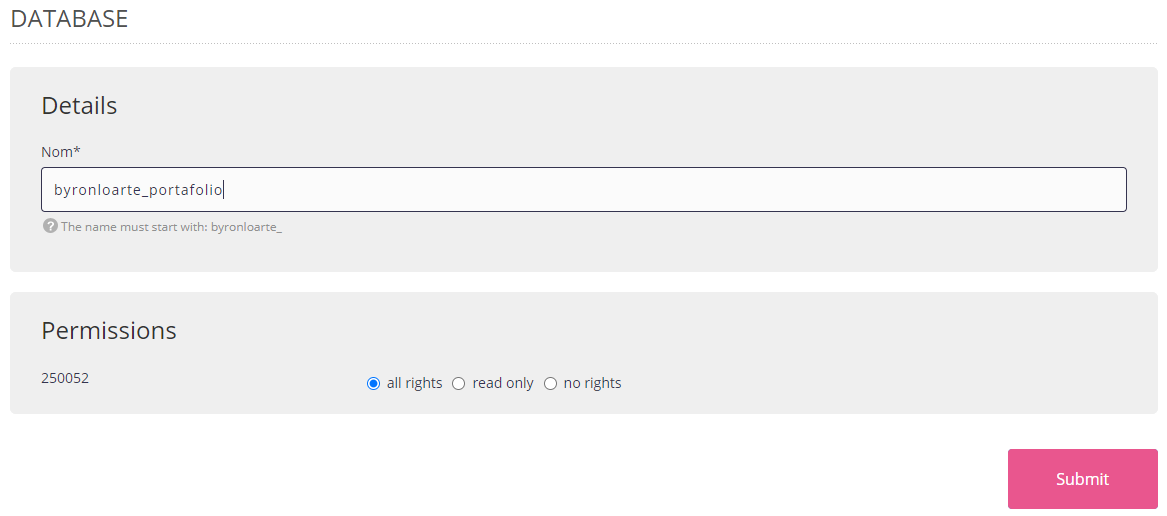
<https://www.alwaysdata.com/en/>



Luego dirigirse al apartado de Base de Datos y crear una nueva Base de Datos.



Recuerda que debe ser **nombredelproyecto\_nombrebdd**

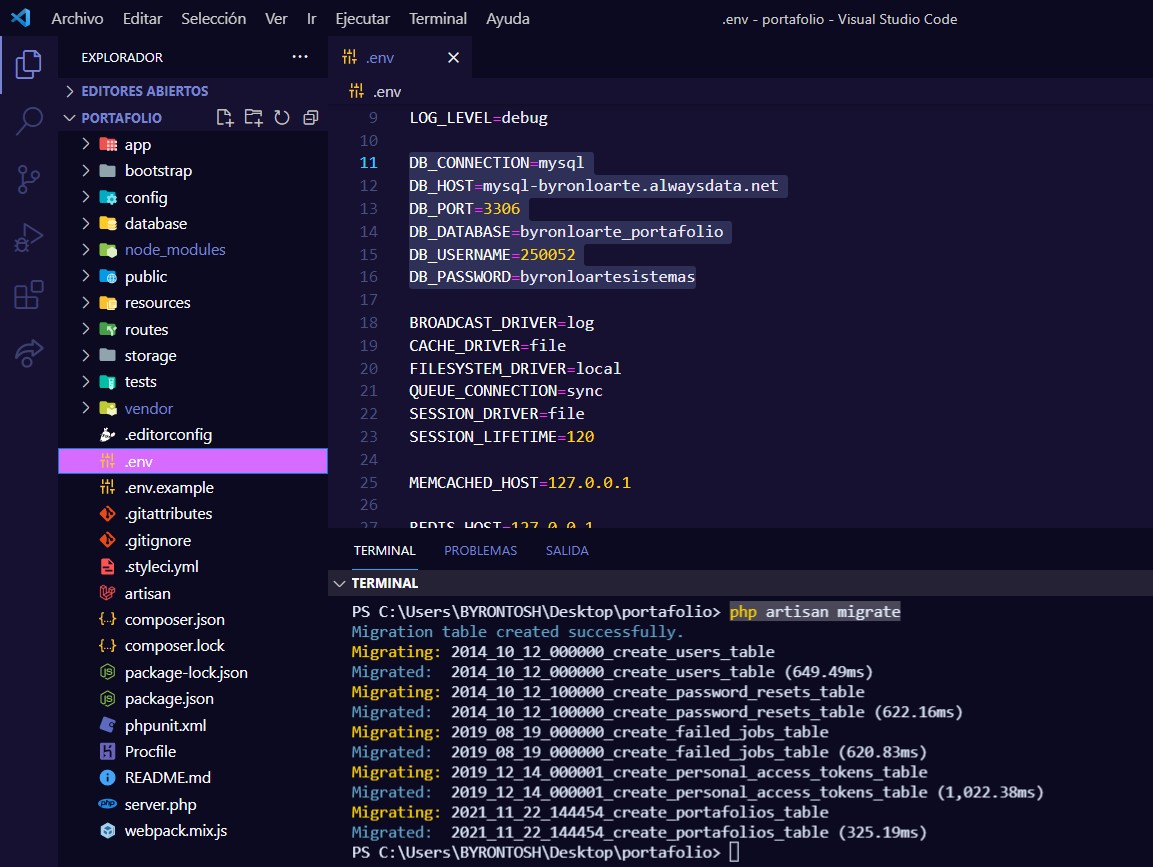


Luego se procede a verificar la creación de la base de datos y luego agregar las variables en la configuración de heroku.

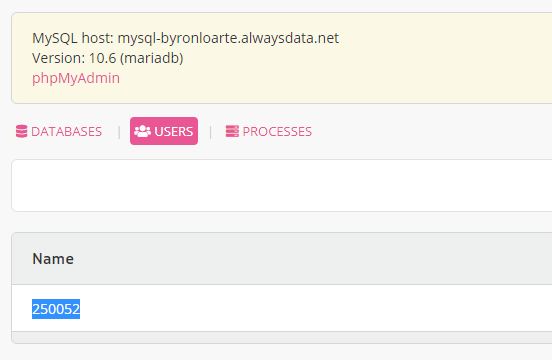


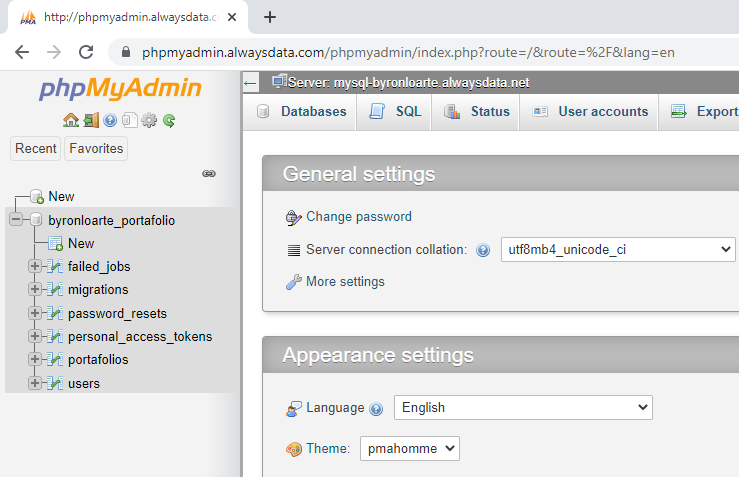
Colocar las credenciales en el proyecto y ejecutar el siguiente comando

**php artisan migrate**



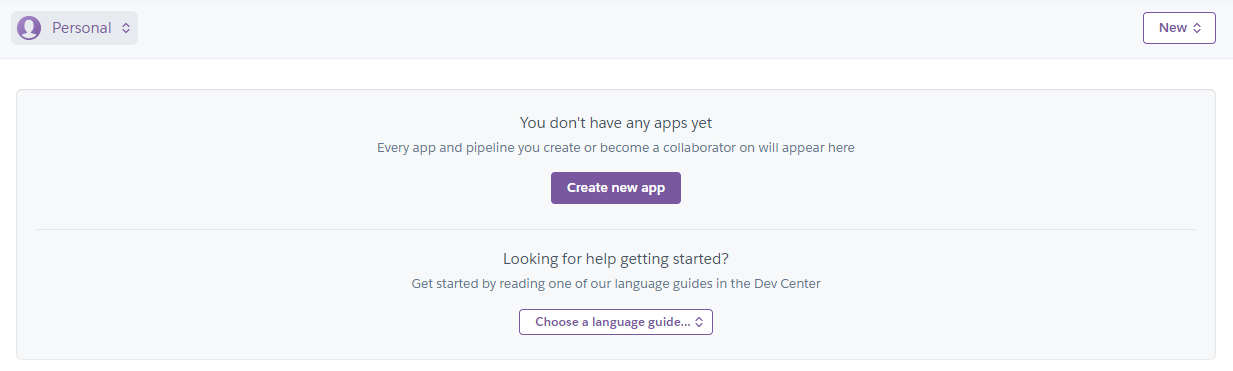
Se puede ver la creación exitosa, desde el panel de alwaystada

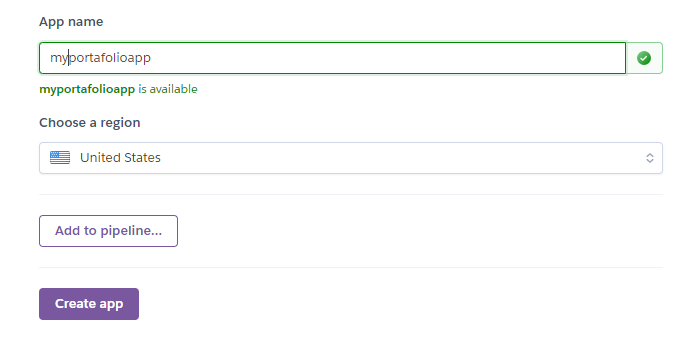




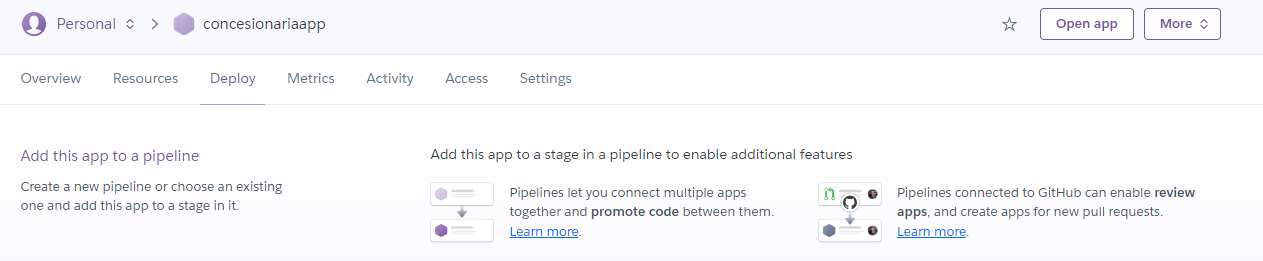
Ahora se procede a subir el proyecto en la plataforma de Heroku

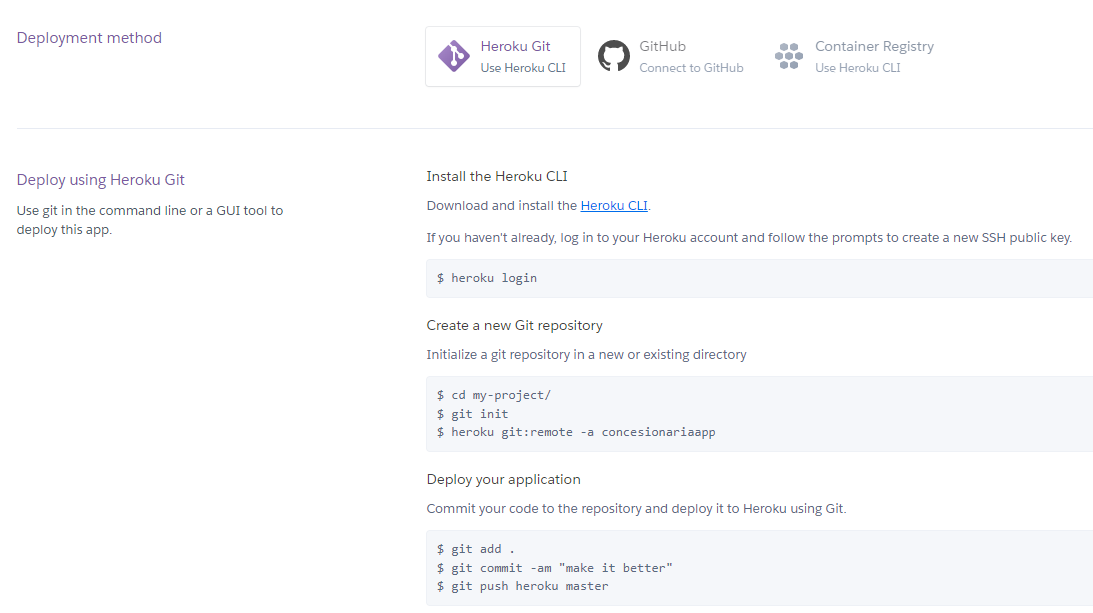
Crear una nueva aplicación

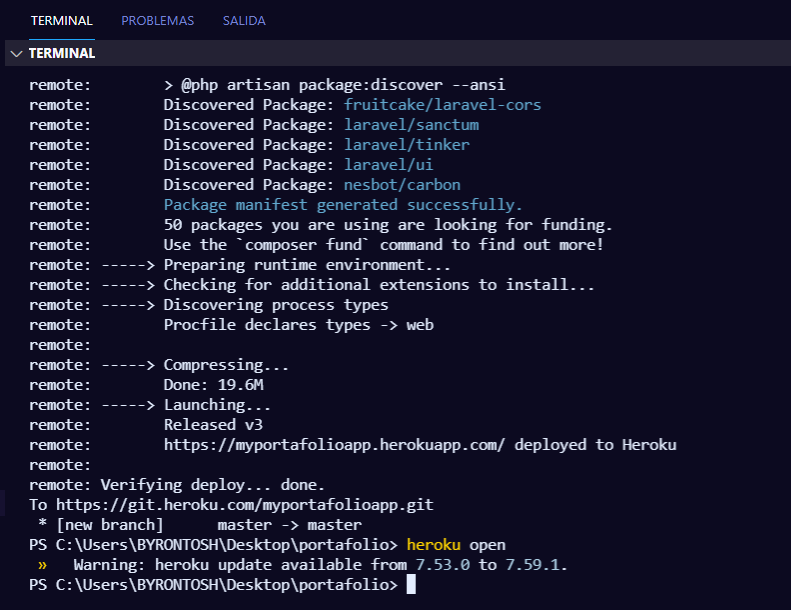




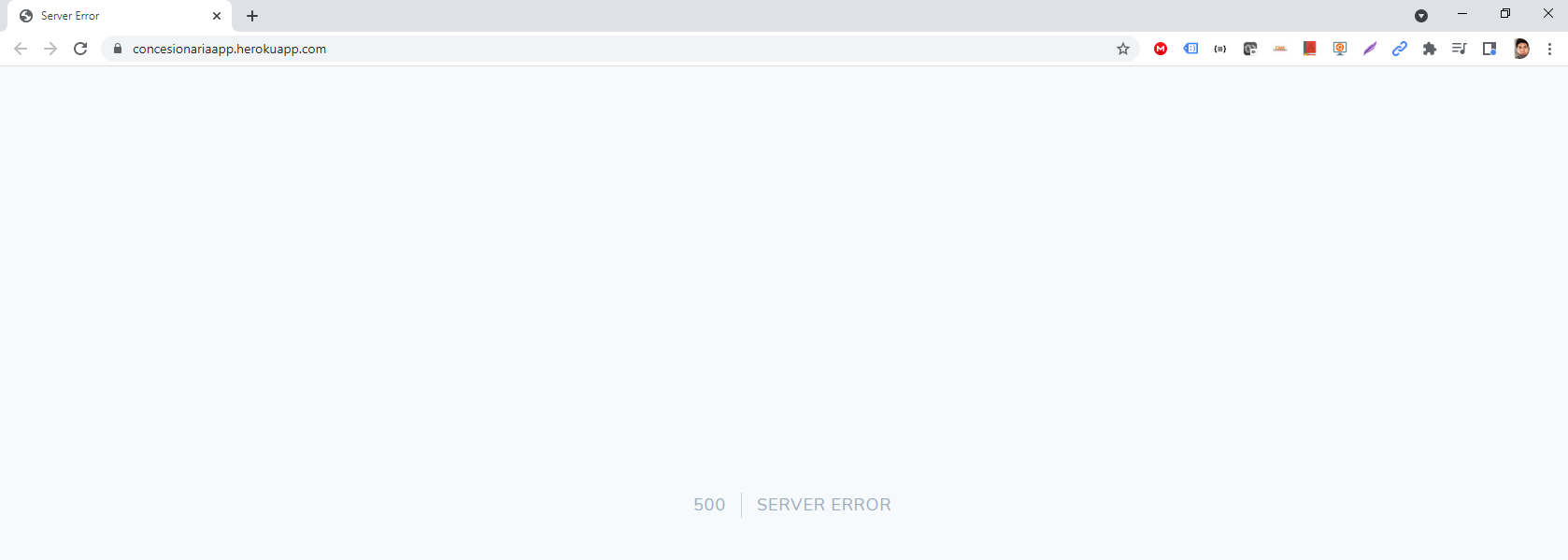
Luego dirigirse al apartado de Deploy y ejecutar los pasos requeridos



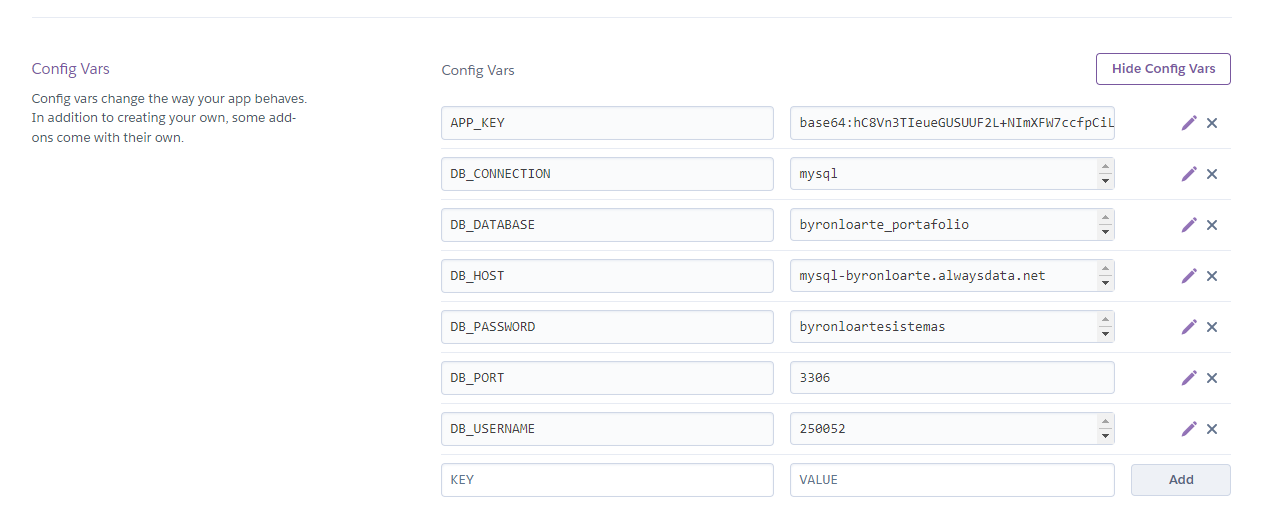




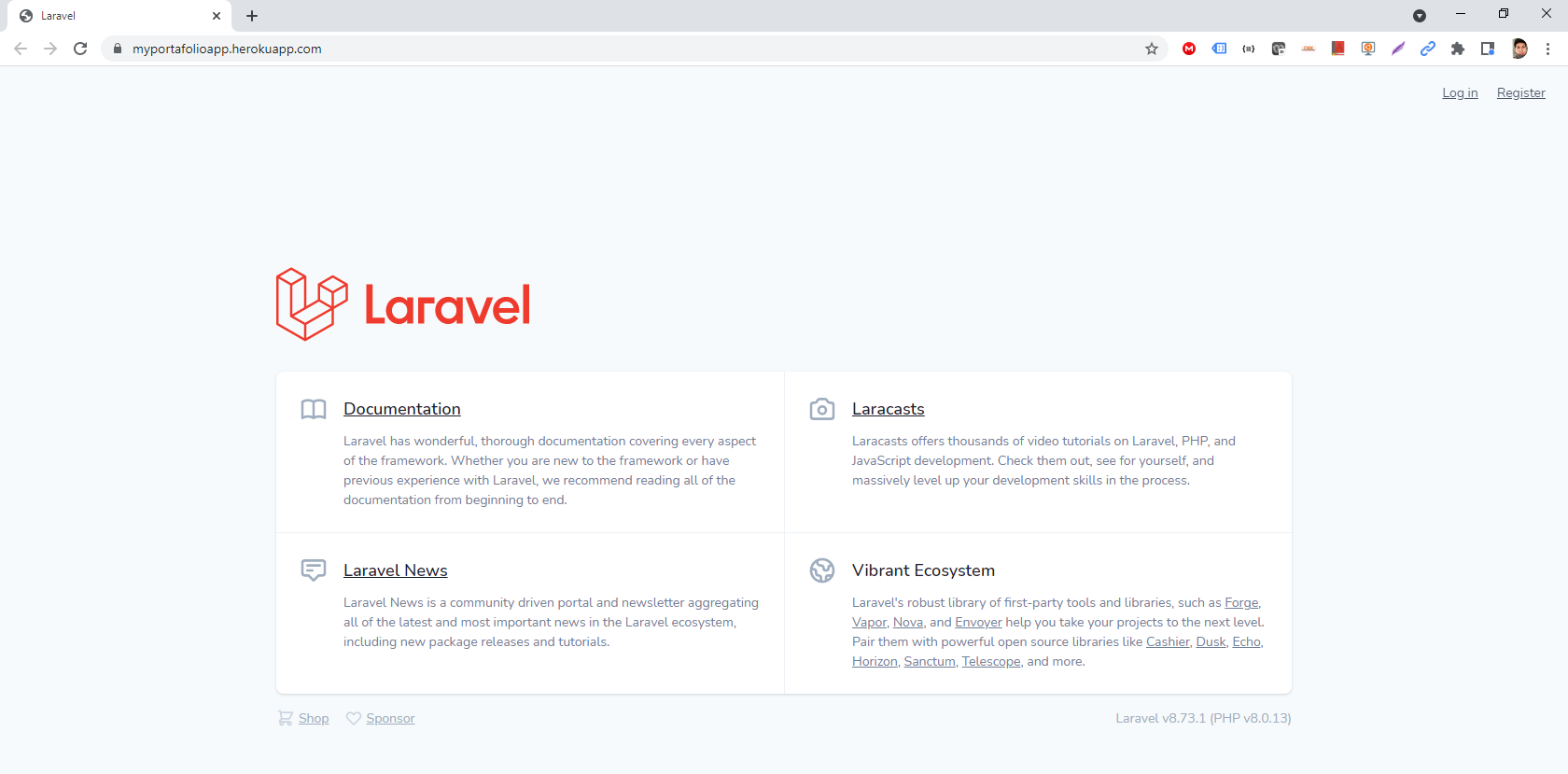
Verificar

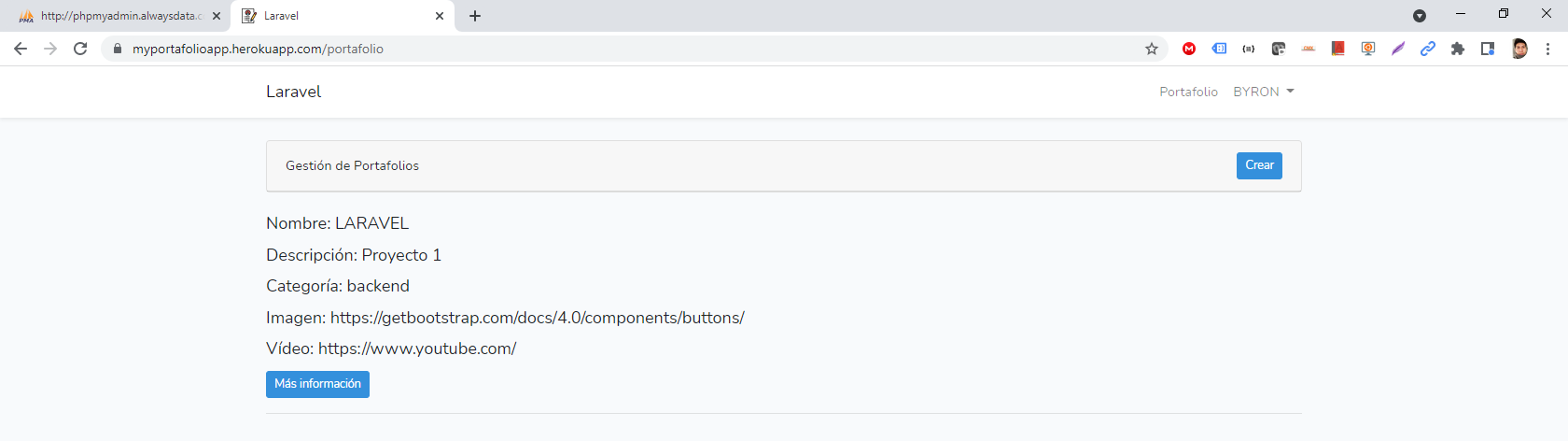


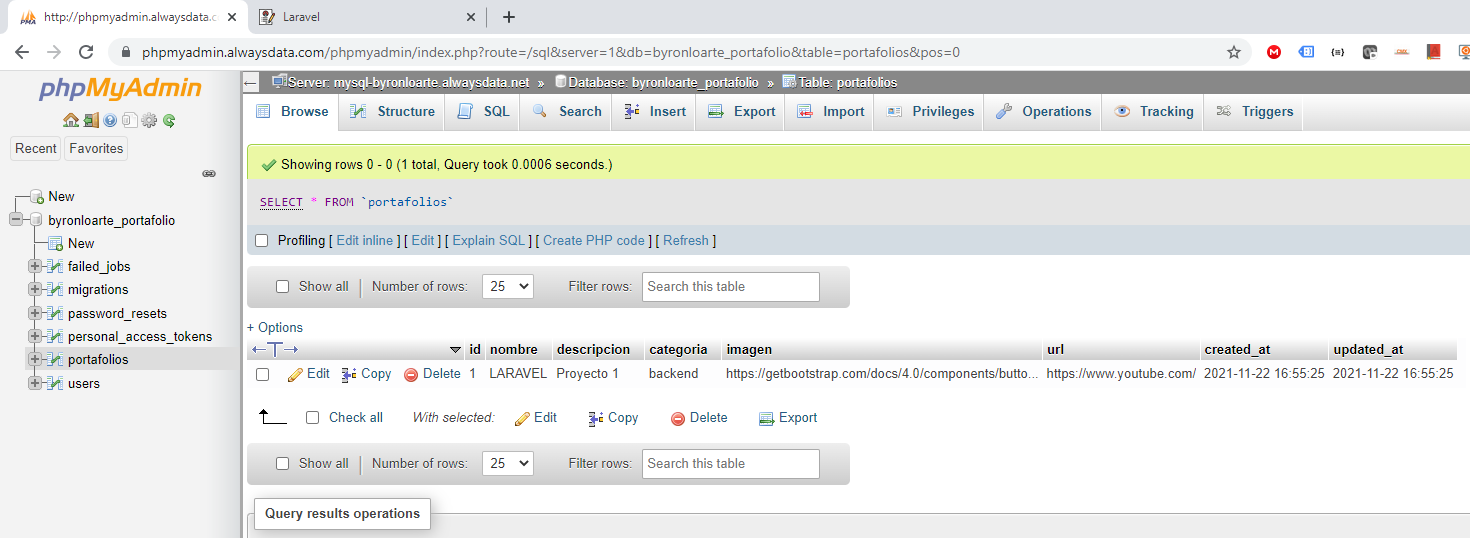
Ahora en la pestaña settings de heroku se procede a establecer la variable de entorno, esta mismas variables son las mismas del archivo .env del proyecto.



Verificar nuevamente con un F5 a la página.







<https://portafoliowebesfot.herokuapp.com/>

Referencias bibliográficas

* <https://dev.to/jsf00/deploy-laravel-application-with-database-to-heroku-l50>
* <https://stackoverflow.com/questions/31467871/setting-up-https-redirects-on-heroku-laravel-instance>