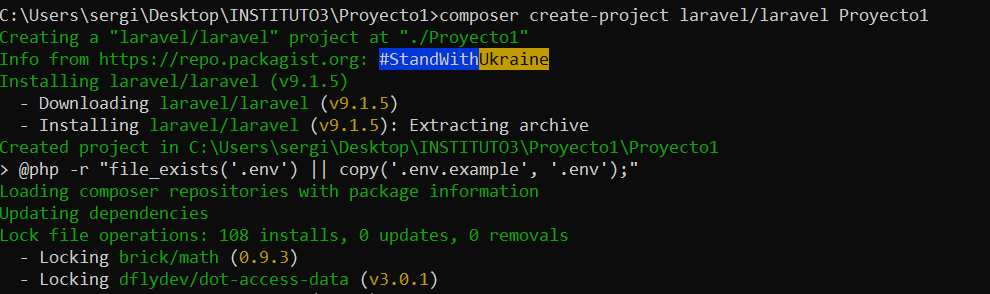
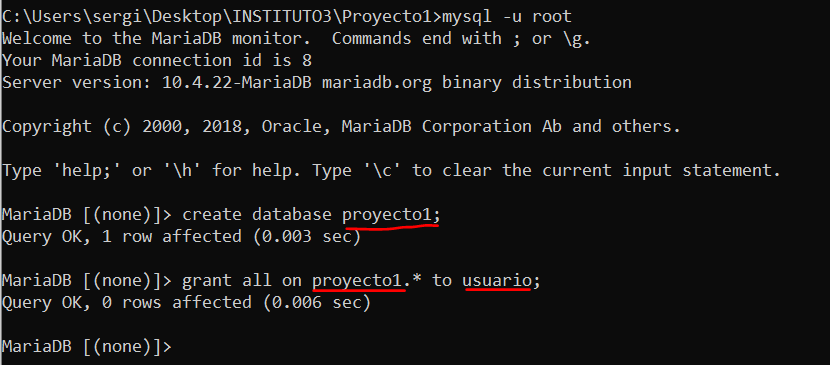
**DOCUMENTACIÓN PROYECTO FÚTBOL MASTER**

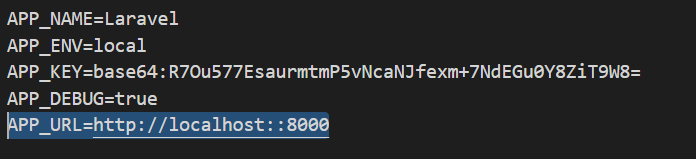
**Lo primero que haremos será crear nuestro proyecto situándonos en el directorio:**



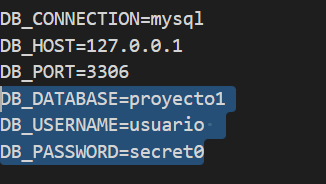
**Ahora crearemos la BBDD y le daremos sus permisos a un usuario:**



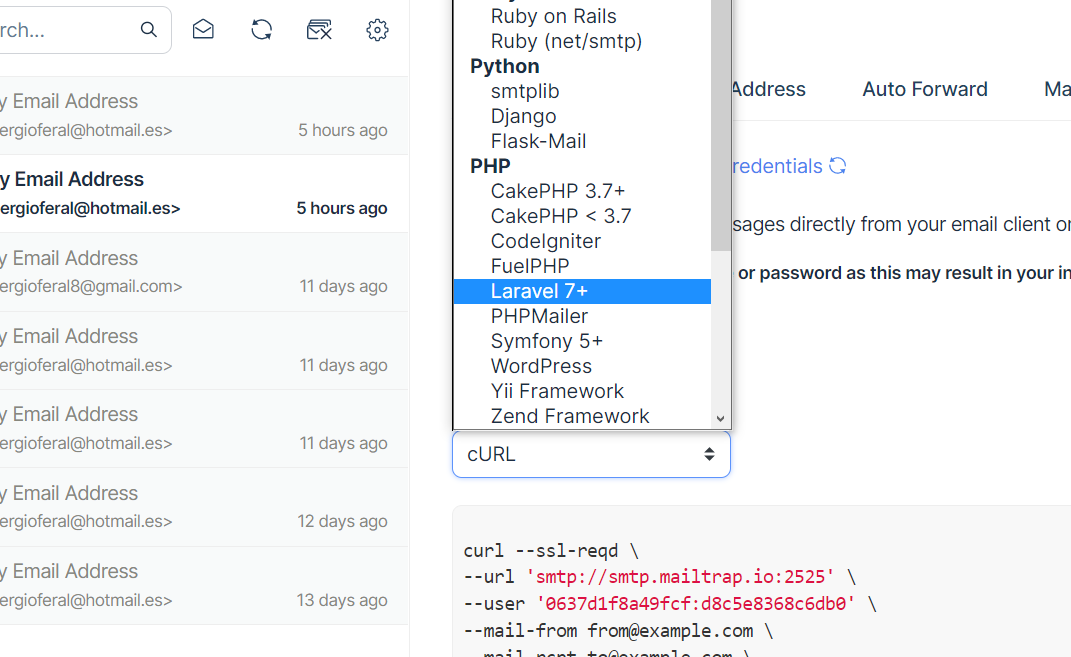
**Iniciaremos el proyecto en nuestro editor de código fuente que vayamos a utilizar como en mi caso es VSCode y empezaremos editando el archivo .env:**



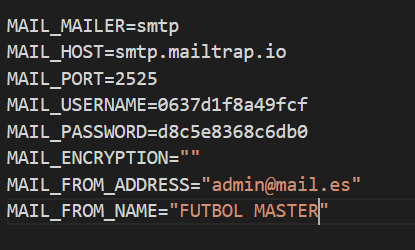
**Introduciremos el nombre de la base de datos del proyecto junto con el usuario que la administrará.**



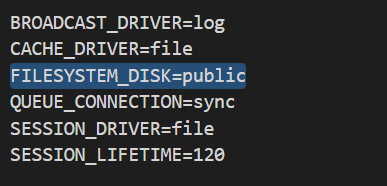
**Ahora nos dirigiremos al la pagina de mailtrap con la que copiaremos el trozo de parámetros para su función seleccionando la versión de Laravel 7+:**

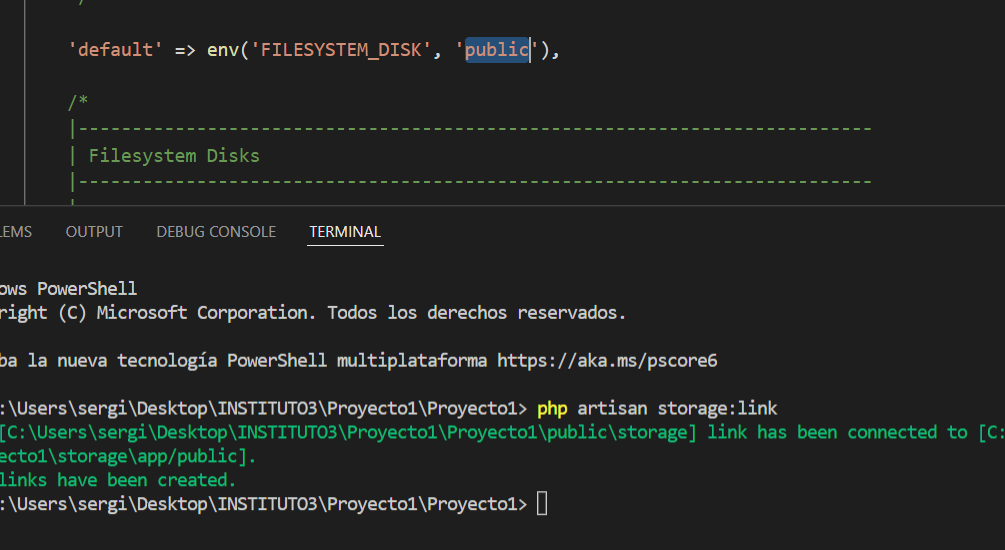


**Y lo incorporamos a nuestro archivo,ENCRYPTION lo dejaremos vacio para que no nos de errores con la verificación del email:**

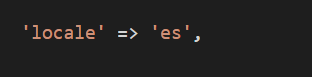


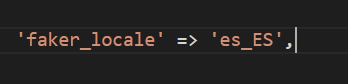
**Para poder habilitar los recursos de livewire y enlazar nuestra carpeta public del storage vamos a editar otra línea del .env y dentro de la carpeta /config->filesystems.php**



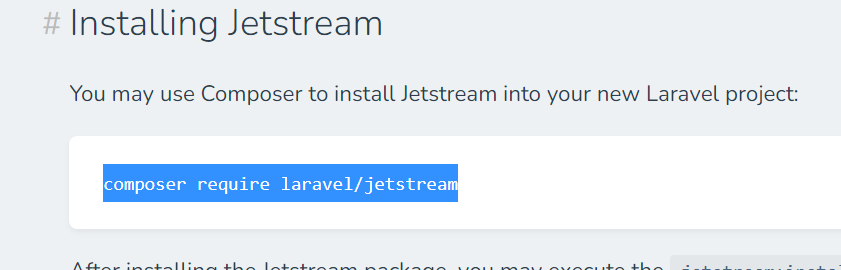


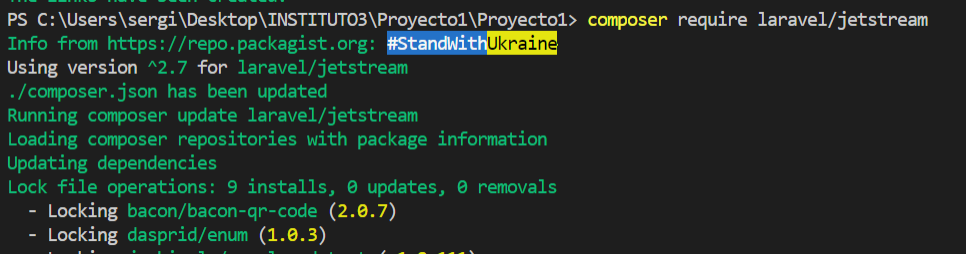
**Cambiaremos el Idioma en /config->app.php**

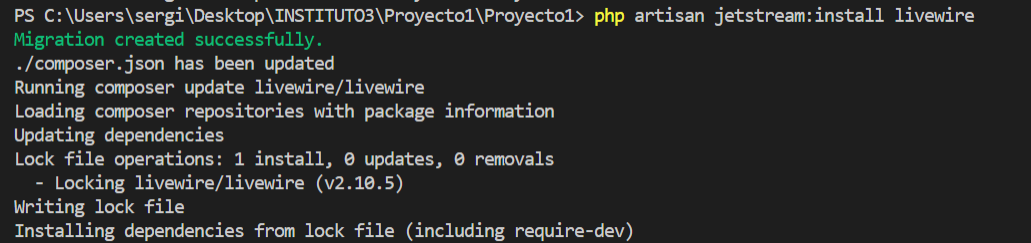


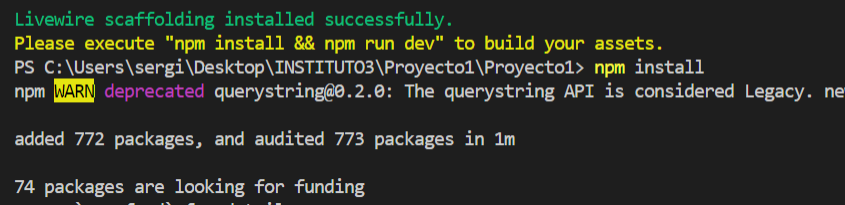


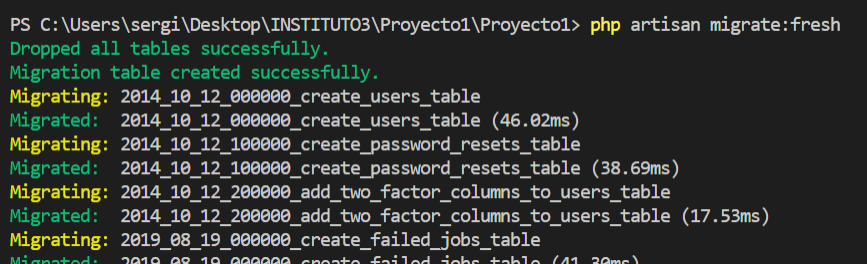
**Nos dirigimos a la pagina de jetstream de laravel para coger el comando de instalación y los siguiente comandos que ejecutaremos para cargar las plantillas y clases html:**



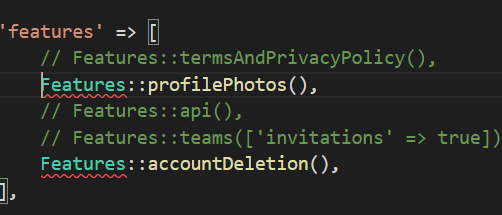




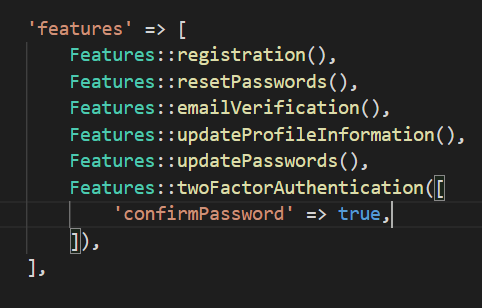




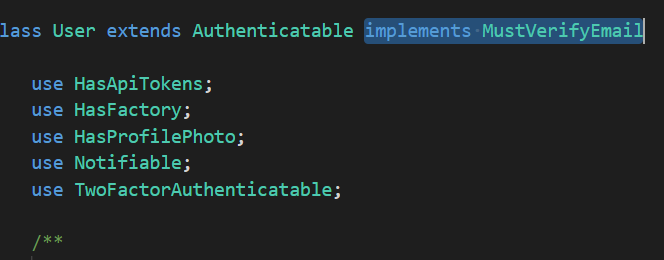
**Descomentaremos la línea de la foto de perfil de jetstream en jetstream.php:**



**Ahora activaremos la verificación por correo que vendrá desactivada por defecto de los recursos de jetstream en la clase fortify.php en /config:**



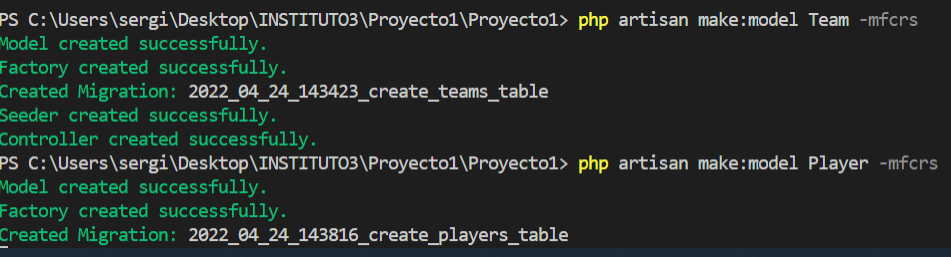
**Ahora nos iremos al modelo User.php y le implementaremos la interfaz MustVerifyEmail:**



**Ahora vamos a crear las migraciones,seeders,factories y controladores de cada tabla de datos que tendremos en nuestra aplicación aparte de la de usuarios que ya viene creada por defecto:**

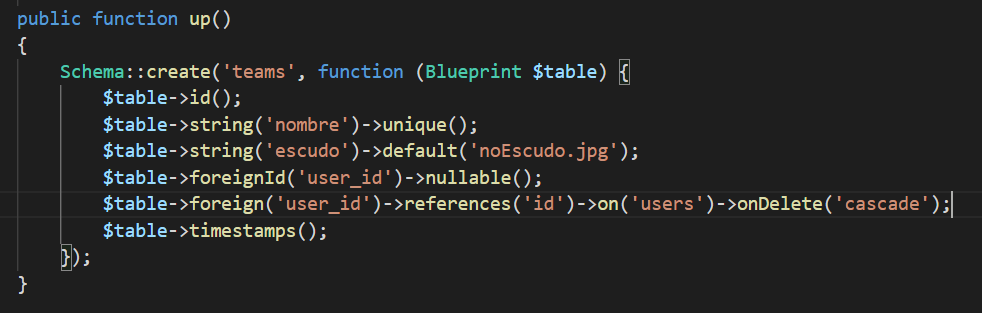
**Nota:** Esta parte de la documentación se ira modificando conforme se vayan añadiendo las tablas para menos dificultad a la hora de administrarse**.**

**Modelos Equipo y Jugador:**



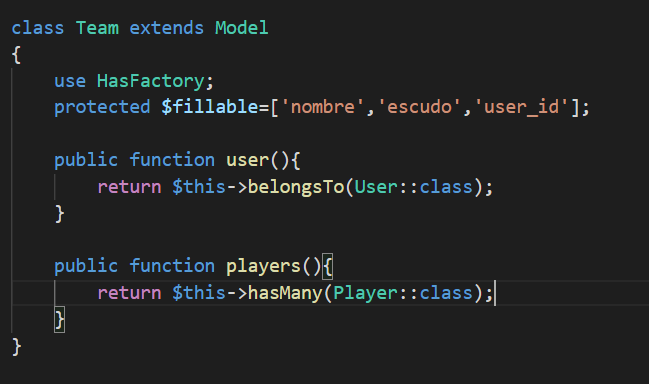
**Declaramos los datos y su tipo en la migración:**

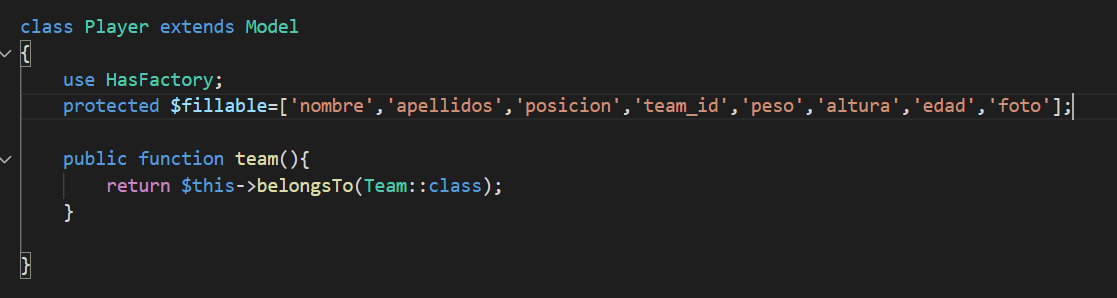
**De equipos:**



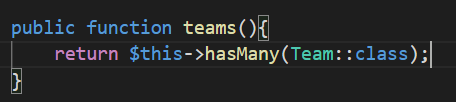
**De jugadores:**

**Ahora el protected fillable en el modelo de cada donde escribiremos los datos de cada tabla y la relación 1:n que les corresponde a cada uno:**





**A su vez en el modelo User añadiremos la relación con el equipo;**

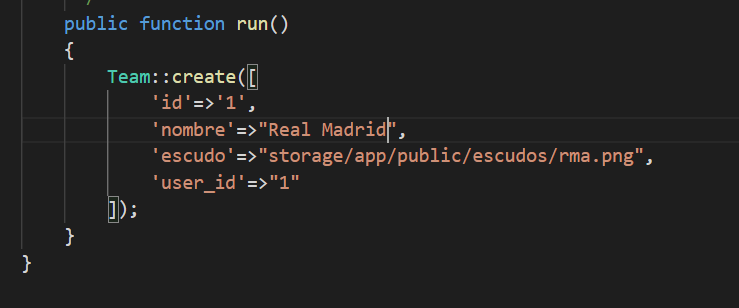


**Ahora añadiremos algunos datos de cada tabla creada que se le asignaran a un usuario creado ya previamente:**

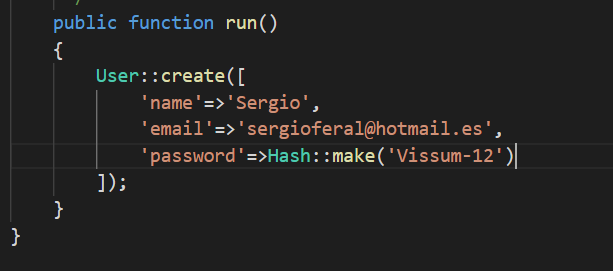
**Como son los jugadores:**



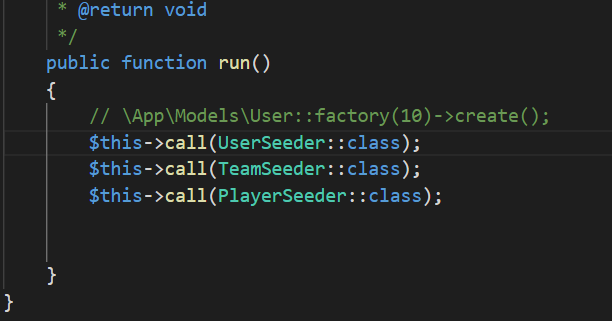
**Un equipo:**



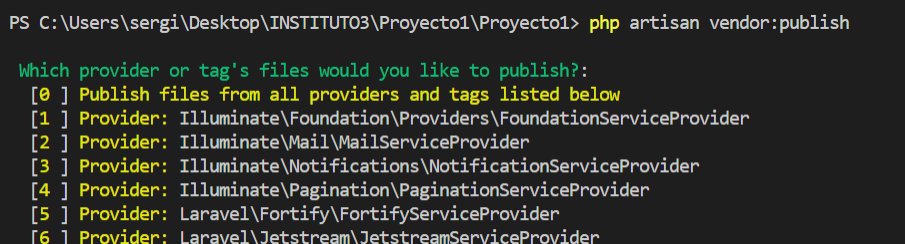
**Y el usuario:**



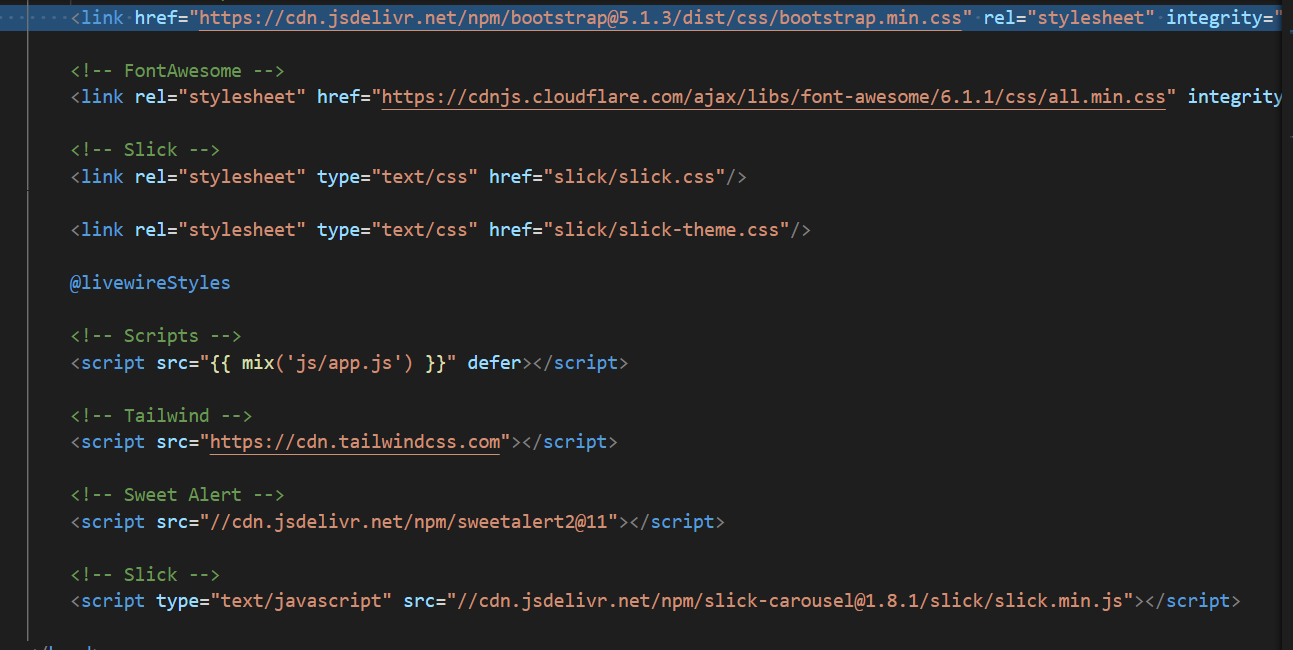
**Indicaremos que recoja los datos en el DatabaseSeeder:**



**Todas las imágenes que utilizaremos a lo largo del proyecto estarán almacenadas dentro de la carpeta storage/app/public:**

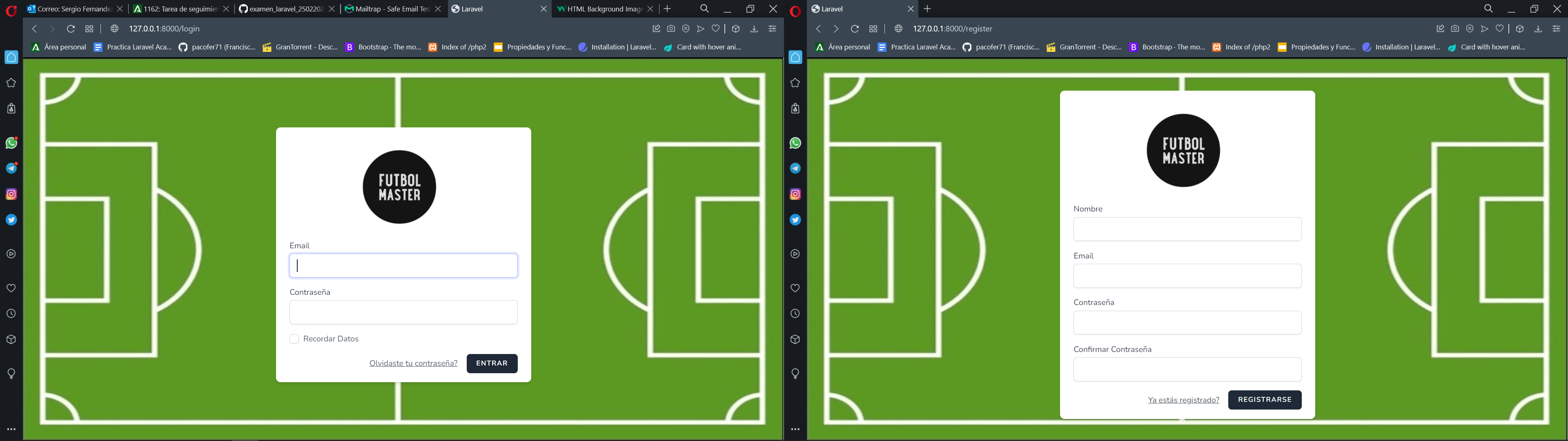


**Introduciremos en nuestra plantilla app.blade.php los cdn de algunas librerías que utilizaremos como es la de bootstrap,tailwind,sweetAlerts**

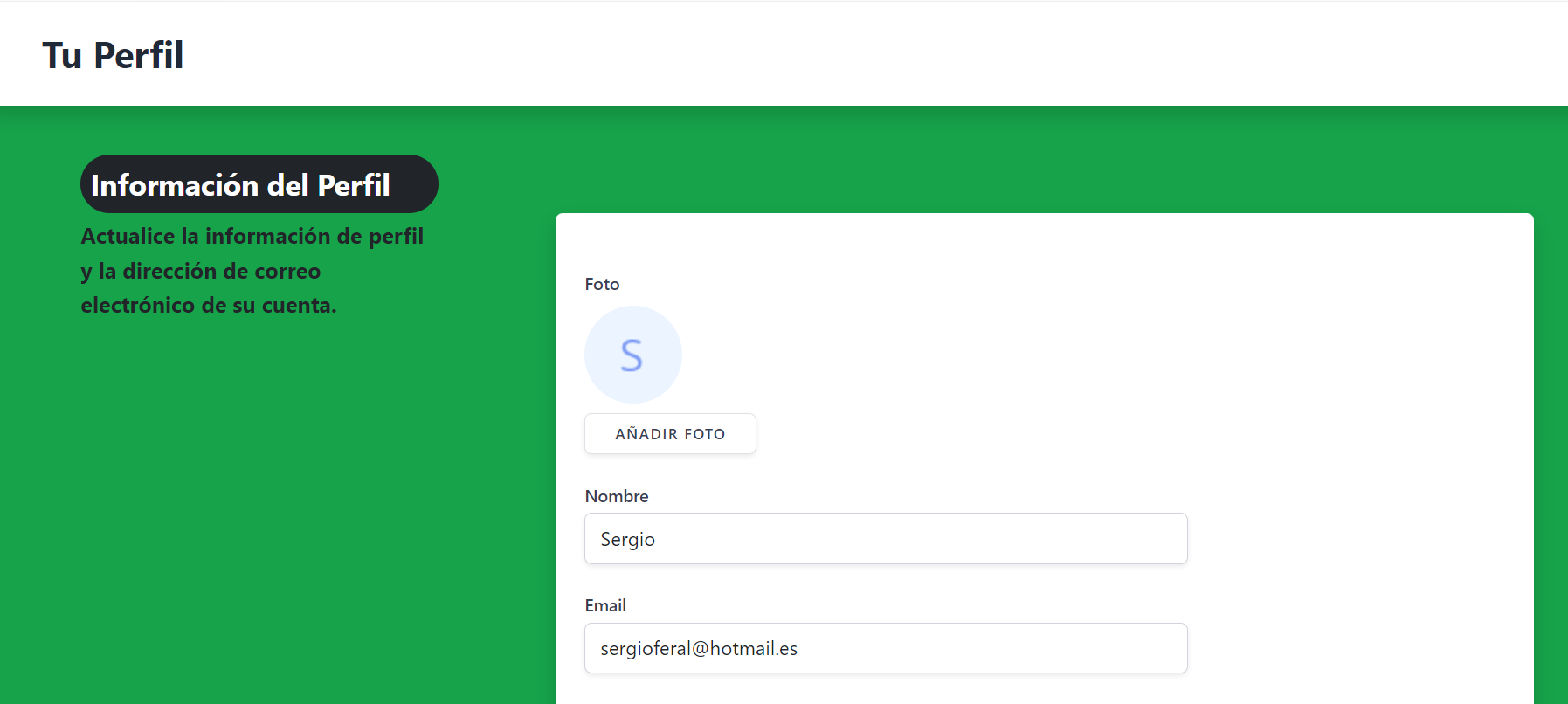


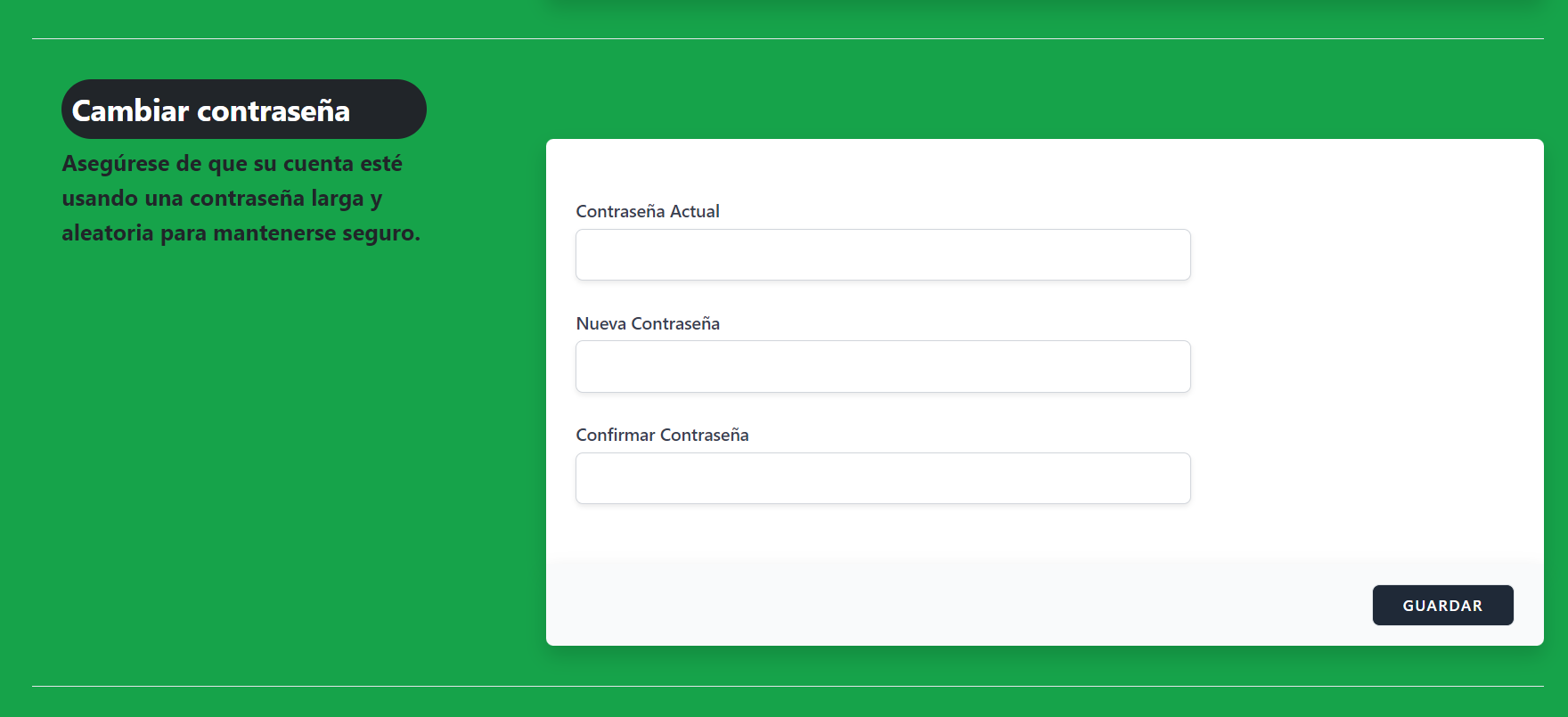
**Después he cambiado el estilo y traducido el contenido de todas las plantillas y paginas relacionadas con el usuario, la pagina de inicio, login y registro:**

**Login y register:**



**Configuración de perfil:**



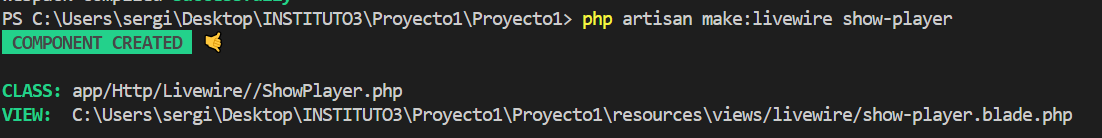


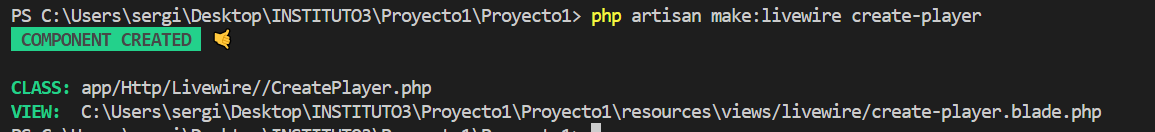
**Pagina de inicio(en proceso):**



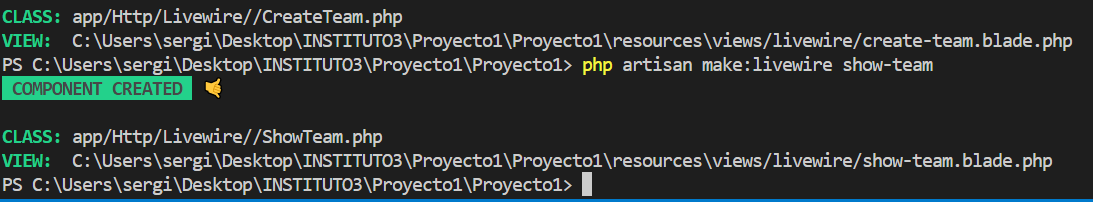


**Vamos a crear los componentes livewire repectivos de los Jugadores y Equipos:**



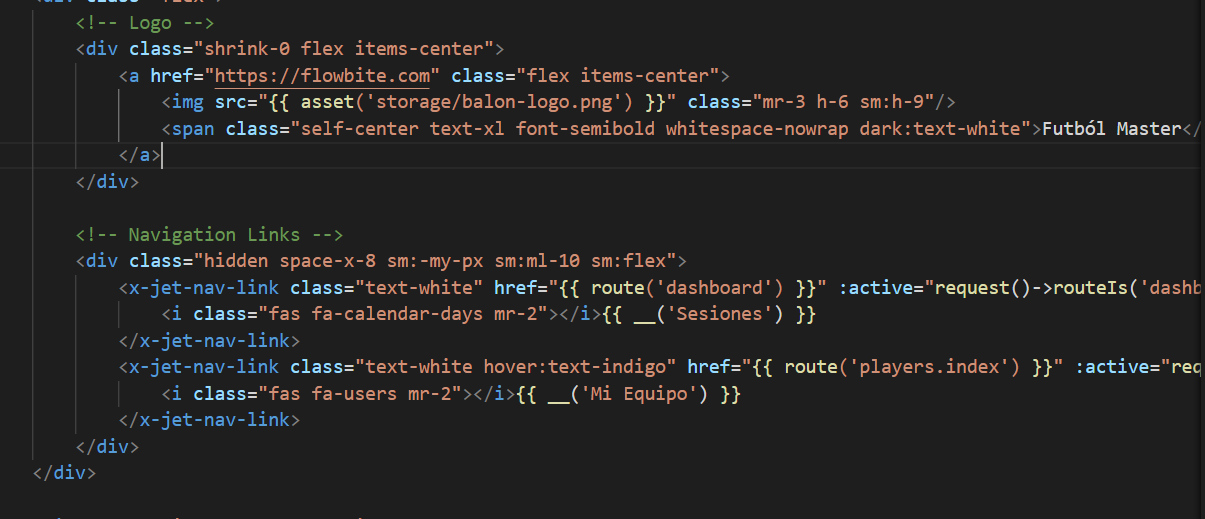


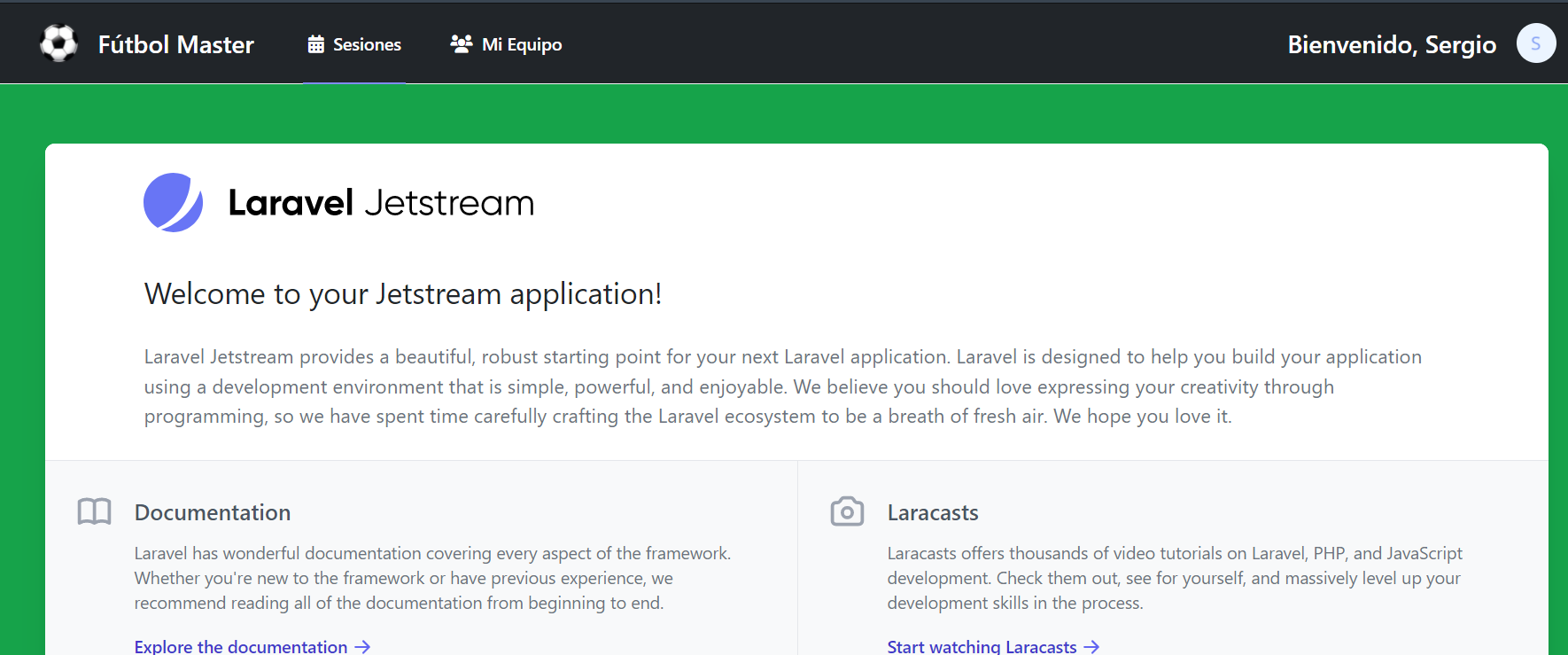
**Equipos:**



**En web.php le añadiremos la ruta para mostrarnos los equipos;**

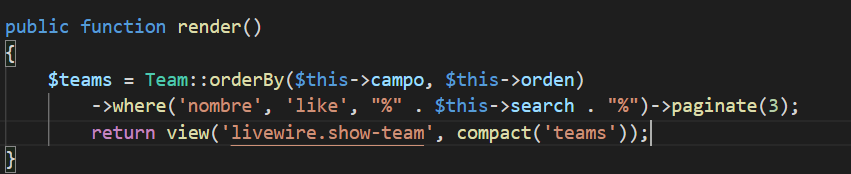
**Ahora tunearemos nuestra barra de navegación creada con jetstream automáticamente y le añadiremos la sección Player y mas adelante la de sesiones y demas:**



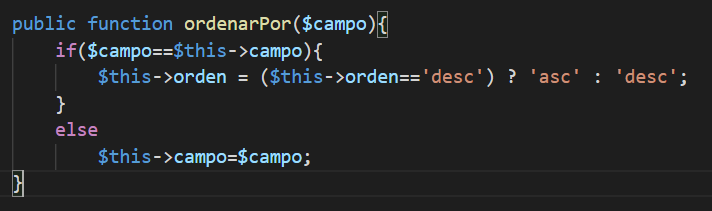


**Ahora iremos a la clase ShowTeams y añadiremos los métodos correspondientes a las funciones que queremos emplear:**

**Como recoger todos nuestros equipos creados y una forma de poder buscar el que queramos:**



**Poder ordenarlos:**



**Poder borrar y editar:**

public function borrar(Team $team){

        //1.- Borro el archivo de imagen

        Storage::delete($team->image);

        //Borro el team

        $team->delete();

        $this->emit('borrar', 'Equipo Borrado');

    }

    public function editar(Team $team){

        $this->team=$team;

        $this->isOpen=true;

    }

    public function update(){

        $this->validate([

            'team.nombre'=>['required', 'string', 'min:3', 'unique:teams,nombre,'.$this->team->id]

        ]);

        if($this->escudo){

            Storage::delete($this->team->escudo);

            $imagenNueva = $this->image->store('escudos');

            $this->team->image = $imagenNueva;

        }

        $this->team->save();

        $this->emit('info', "Equipo Editado.");

        $this->reset(['isOpen', 'image']);

    }

**Y a continuación la clase CreateTeam.php para poder crear nuevos equipos:**

use WithFileUploads;

    public $image;

    public $nombre;

    public $isOpen=false;

    protected $rules=[

        'nombre'=>['required', 'string', 'min:3', 'unique:teams,nombre'],

        'escudo'=>['required', 'image', 'max:1024']

    ];

    public function render()

    {

        return view('livewire.create-team');

    }

    public function guardar(){

        $this->validate();

        $imagen = $this->escudo->store('escudos');

        Team::create([

            'titulo'=>ucwords($this->nombre),

            'escudo'=>$imagen

        ]);

        $this->reset(['isOpen', 'nombre']);

        $this->emitTo('show-team', 'renderizarTeams');

        $this->emit('alerta', 'Equipo Creado con éxito');

    }

**Ahora procederemos a diseñar la pagina que visualizara los equipos:**

**(Aun tiene fallos en la estructura del diseño)**

<div class="mt-4 max-w-7xl mx-auto sm:px-6 lg:px-8">

    <div class="heading">

        <h1 style="font: bold">Selecciona un Equipo</h1>

    </div>

    <div class="my-2 flex mb-5">

        <div class="flex-1 w-80">

            <x-jet-input type="search" placeholder="Buscar..." wire:model="search" /><i class="fas fa-search"></i>

        </div>

        <div>

            @livewire('create-team')

        </div>

    </div>

    <div>

        @foreach ($teams as $item)

            <a href="#">

                <div class="card">

                    <div class="card-header">

                        <img src="{{ Storage::url('escudos/'.$item->escudo) }}">

                    </div>

                    <div class="card-body">

                        <p>

                            {{ $item->nombre }}

                        </p>

                        <div class="btn">

                          <button wire:click="edit({{ $item }})"

                            class="bg-yellow-500 hover:bg-yellow-700 text-white font-bold py-2 px-3 rounded">

                            <i class="fa-solid fa-edit"></i></button>

                        <button wire:click="borrar({{ $item }})"

                            class="bg-red-500 hover:bg-red-700 text-white font-bold py-2 px-3 rounded">

                            <i class="fa-solid fa-trash"></i></button>

                        </div>

                    </div>

                </div>

            </a>

        @endforeach

    </div>

</div>

**Y la ventana modal del create-team.blade.php:**

<div class="mt-4 max-w-7xl mx-auto sm:px-6 lg:px-8">

    <div class="heading">

        <h1 style="font: bold">Selecciona un Equipo</h1>

    </div>

    <div class="my-2 flex mb-5">

        <div class="flex-1 w-80">

            <x-jet-input type="search" placeholder="Buscar..." wire:model="search" /><i class="fas fa-search"></i>

        </div>

        <div>

            @livewire('create-team')

        </div>

    </div>

    <div>

        @foreach ($teams as $item)

            <a href="#">

                <div class="card">

                    <div class="card-header">

                        <img src="{{ Storage::url('escudos/'.$item->escudo) }}">

                    </div>

                    <div class="card-body">

                        <p>

                            {{ $item->nombre }}

                        </p>

                        <div class="btn">

                          <button wire:click="edit({{ $item }})"

                            class="bg-yellow-500 hover:bg-yellow-700 text-white font-bold py-2 px-3 rounded">

                            <i class="fa-solid fa-edit"></i></button>

                        <button wire:click="borrar({{ $item }})"

                            class="bg-red-500 hover:bg-red-700 text-white font-bold py-2 px-3 rounded">

                            <i class="fa-solid fa-trash"></i></button>

                        </div>

                    </div>

                </div>

            </a>

        @endforeach

    </div>

</div>

