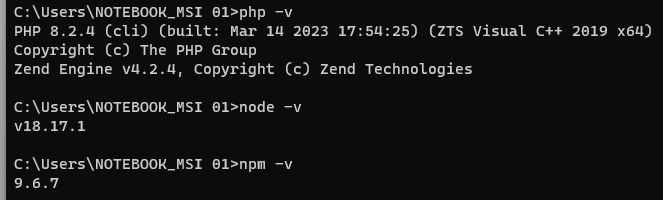
**CRUD LARAVEL 10**

Verificamos que tengamos todo para que laravel funcione

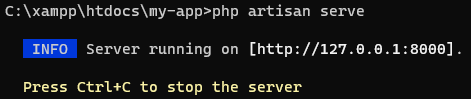
PHP, Composer, node, npm (desde una ventana de comando)

****

Posterior creamos la aplicación laravel

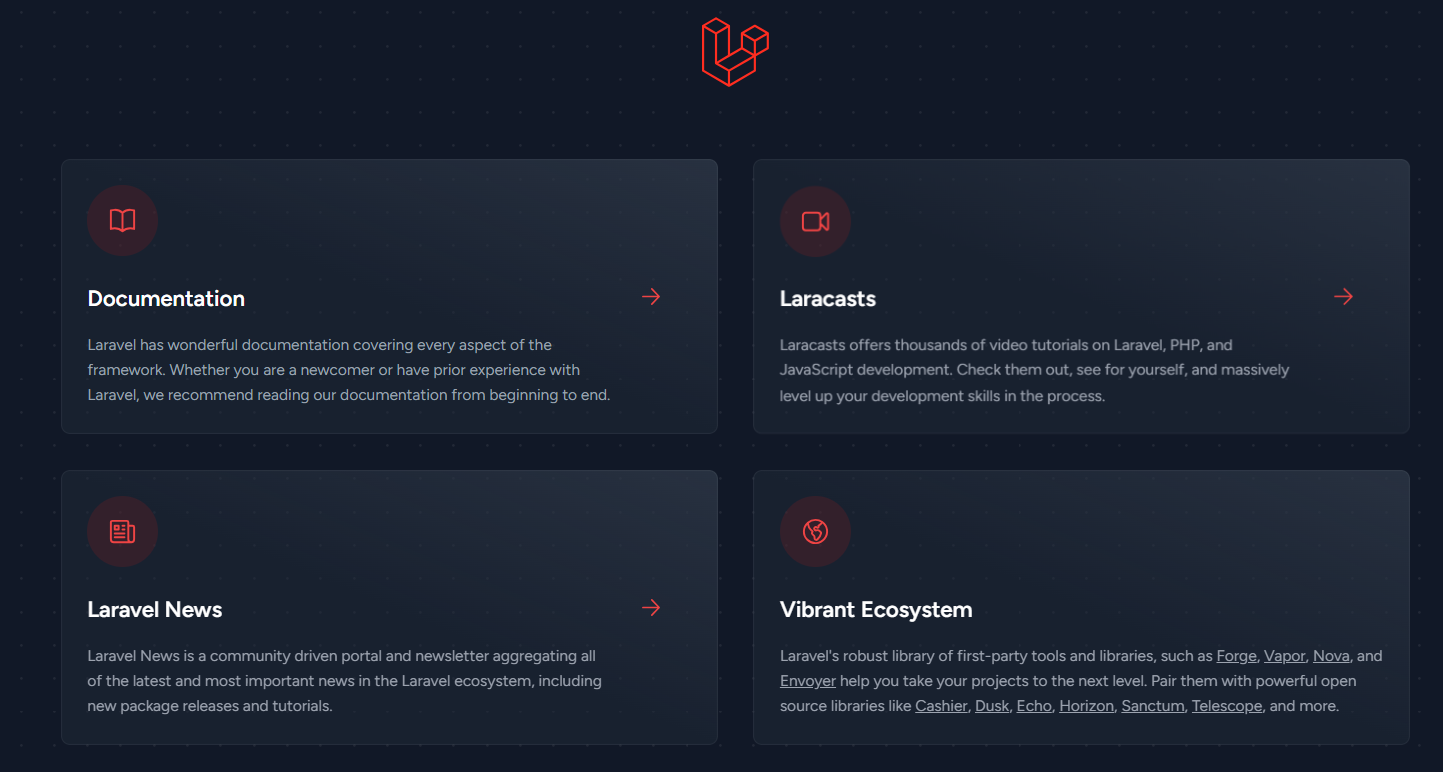
* composer create-project laravel/laravel my-app

nos vamos a esa carpeta creada y colocamos el siguiente código en una ventana de comando



* php artisan serve

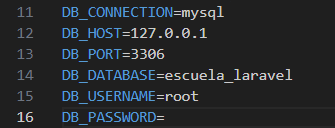
si todo sale bien se carga la aplicación laravel desde tu navegador web



Luego abrimos el proyecto desde visual studio

Ahora configuramos laravel para realizar el crud

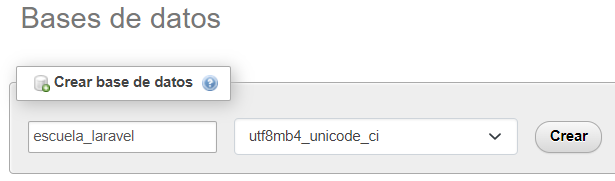
Nos vamos al archivo .env y configuramos lo siguiente



Nombre de la base de datos = escuela\_laravel

Usuario= root

Luego nos vamos a crear la base de datos phpadmin

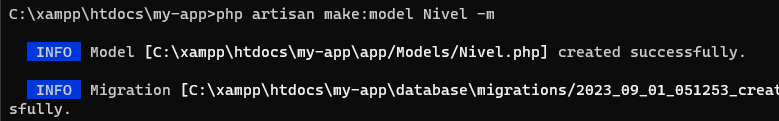


Nos vamos a visual y colocamos el siguiente código, en la terminal

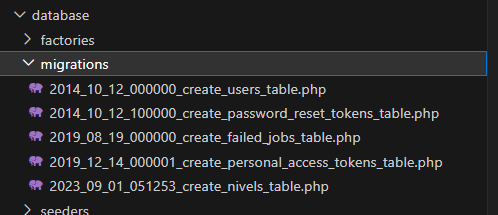
Vamos a realizar un crud de estudiantes con niveles

Creamos el modelo y la migración de la tabla nivel

* php artisan make:model Nivel -m



Esto se crea en la carpeta DataBase/Migraciones



Entramos en el archivo creado nivels

Creamos la tabla niveles, solo es colocar la letra e



Igual agregamos un nuevo campo nombre

Quedando de la siguiente forma

{

        Schema::create('niveles', function (Blueprint $table) {

            $table->bigIncrements('id');

            $table->string('nombre', 50);

            $table->timestamps();

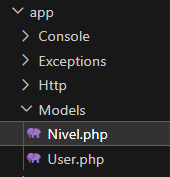
        });

    }

También le colocamos la letra e al schema

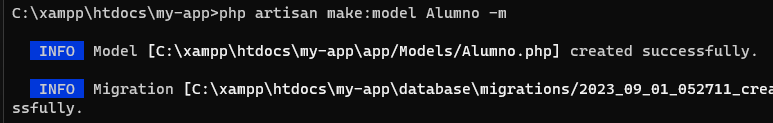


De igual forma vamos a la carpeta app/models y revisamos que este el modelo Nivel.php



Ahora creamos otra migración llamada alumno

* php artisan make:model Alumno -m



Nos vamos a la carpeta de DataBase/migraciones y creamos la tabla con los campos que necesitamos

{

        Schema::create('alumnos', function (Blueprint $table) {

            $table->id();

            $table->string('matricula',10)->unique();

            $table->string('nombre',120);

            $table->date('fecha\_nacimiento');

            $table->string('telefono', 20);

            $table->string('email', 50)->nullable();

            $table->unsignedBigInteger('nivel\_id');

            $table->timestamps();

            $table->foreign('nivel\_id')->references('id')->on('niveles');

        });

    }

Estos dos campos hacen referencia a la llave foránea de la otra tabla de Niveles creada con anterioridad, se está creando una referencia entre dos tablas para saber qué nivel tienen los alumnos

$table->unsignedBigInteger('nivel\_id');

$table->foreign('nivel\_id')->references('id')->on('niveles');

Nos fijamos que en el schema este alumnos

        Schema::dropIfExists('alumnos');

Ahora revisamos los modelos comenzamos con el modelo Nivel

Carpeta App/Model/Nivel.php

{

   protected $table = 'niveles';

}

Luego nos vamos a modelo alumno

class Alumno extends Model

{

    use HasFactory;

    public function nivel(){

    return $this->belongsTo(Nivel::class, 'nivel\_id', 'id');

}

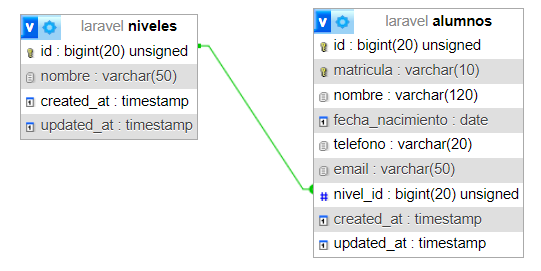
}

Todo esto esto de igual forma se puede realizar con PhpMyadmin, de igual forma nos sirve para revisar las migraciones y modelos.

Ahora ejecutamos las migraciones

php artisan migrate

nos damos cuenta en PhpMyadmin que se crearon las tablas, con las estructuras requeridas





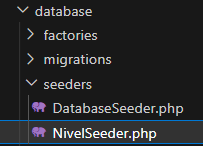
Ahora vamos a crear los seeders para agregar información a la tabla niveles

*“Los seeders, son un recurso que nos permite cargar información a nuestras tablas para probar de manera sencilla y rápida el funcionamiento de nuestra aplicación (paginación, filtros, entre otros)”*

Colocamo el siguiente código

* php artisan make:seeder NivelSeeder

nos vamos a la carpeta de seeders



Entramos a NivelSeeder y colocamos el siguiente código

 public function run(): void

    {

        $niveles = ['PRIMERO', 'SEGUNDO', 'TERCERO', 'CUARTO', 'QUINTO', 'SEXTO'];

        foreach($niveles as $nivel) {

            DB::table('niveles')->insert([

                'nombre' => $nivel

            ]);

        }

}

No olviden colocar en use en la parte de arriba

use Illuminate\Support\Facades\DB;

Esto para es para agregar los nombres en la tabla niveles. Luego ejecutamos el siguiente código en la terminal

* php artisan db:seed --class=nivelSeeder

luego revisamos la tabla de niveles



Con esto terminamos todo lo referente a bases de datos

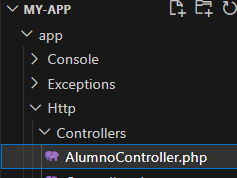
Ahora trabajaremos con los controladores

Colocamos en la terminal el siguiente código

* php artisan make:controller AlumnoController --model=Alumno –resource

creamos un controlador llamado alumno conectado al modelo alumno y se crea con todo lo necesario para realizar el CRUD

nos vamos a la ruta del controlador



Nos damos cuenta que el controlador ya está listo para realizar el CRUD

Ahora nos vamos a las rutas web.php

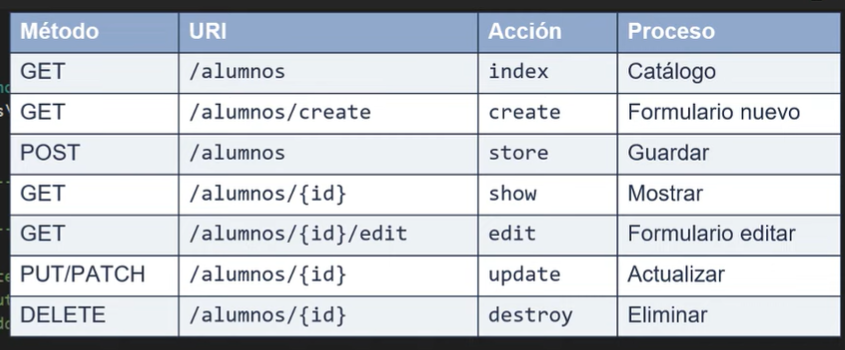
Lo primero es colocar el use en la parte superior

use App\Http\Controllers\AlumnoController;

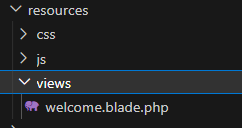
en la parte inferior colocamos la ruta

Route::resource('/alumnos', AlumnoController::class);

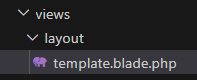
Con el resource nos podemos ir a cada una de las funciones del CRUD



Ahora nos vamos a las vistas



Creamos una capeta layout y un archivo llamado template



Dentro de template creamos la estructura básica de Html 5

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Document</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">

</head>

<body>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>

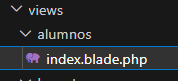
</body>

</html>

Luego colocamos el @yield(‘content’) dentro del body igualmente en el title

 <title>@yield('title')</title>

Luego creamos una carpeta en vista que se llame alumnos dentro un archivo index



Dentro de index colocamos el siguiente codigo

@extends('layout/template')

@section('title', 'Alumnos | Escuela')

@section('contenido')

<main>

     <div class="container py-4">

       <h2>Listado de alumnos </h2>

     </div>

</main>

En el controlador Alumno en la parte de index colocamos lo siguiente,

  public function index()

    {

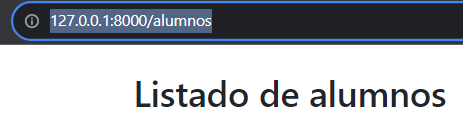
        return view('alumnos.index');

    }

Código para retornar la vista

Levantamos el servidor

Y colocamos el nombre <http://127.0.0.1:8000/alumnos>



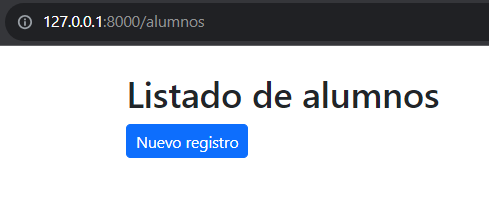
Ahora vamos a crear un registro con una ruta en laravel

Dentro de index colocamos el siguiente código, debajo de

<h2>Listado de alumnos </h2>

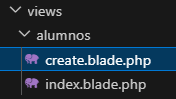
<a href="{{ url('alumnos/create') }}" class="btn btn-primary btn-sm">Nuevo registro</a>

Listo lo ejecutamos y nos muestra esto



El botón creado pero al darle click, saldrá página en blanco porque en el controlador en el método de créate no tenemos nada en el momento

Dentro la carpeta alumno creamos nuestra vista



Dentro de créate colocamos en siguiente codigo

<main>

<div class="container py-4">

    <h2>Registrar Alumno</h2>

</div>

</main>

Ahora cambiamos la función créate dentro del controlador y retornamos la vista creada

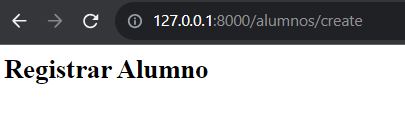
 public function create()

    {

        return view('alumnos.create');

    }

Al ejecutar y darle click al botón nos mostrara la vista



Ahora agregamos plantilla a créate que hereda del padre

Quedando de la siguiente forma

@extends('layout/template')

@section('title', 'ARegistrar | Escuela')

@section('contenido')

<main>

<div class="container py-4">

    <h2>Registrar Alumno</h2>

</div>

</main>

Ahora creamos un formulario dentro de créate de tipo post

<form action="{{ url('alumnos') }}" method="post">

        @csrf

           <div class="md-3 row">

              <label for="matricula" class="col-sm-2 col-form-label">Matricula:</label>

              <div class="col-sm-5">

                   <input type="text" class="form-control"  name="matricula"  id="matricula" value="{{ old('matricula') }}" required>

                </div>

            </div>

            <div class="md-3 row">

                <label for="nombre" class="col-sm-2 col-form-label">Nombre completo:</label>

                <div class="col-sm-5">

                    <input type="text" class="form-control"  name="nombre"  id="nombre" value="{{ old('nombre') }}" required>

                </div>

            </div>

            <div class="md-3 row">

                <label for="fecha" class="col-sm-2 col-form-label">Fecha de Nacimiento:</label>

                    <div class="col-sm-5">

                        <input type="date" class="form-control"  name="fecha"  id="fecha" value="{{ old('fecha') }}" required>

                    </div>

                </div>

                    <div class="md-3 row">

                        <label for="telefono" class="col-sm-2 col-form-label">Telefono:</label>

                        <div class="col-sm-5">

                            <input type="text" class="form-control"  name="telefono"  id="telefono" value="{{ old('telefono') }}" required>

                        </div>

                    </div>

                        <div class="md-3 row">

                            <label for="email" class="col-sm-2 col-form-label">Email:</label>

                            <div class="col-sm-5">

                                <input type="email" class="form-control"  name="email"  id="email" value="{{ old('email') }}" required>

                            </div>

                        </div>

                            </form>

Quedando de la siguiente forma



Ahora crearemos un ítem para nivel, pero cargado desde la base de datos

Creamos la primera en blanco

<div class="md-3 row">

                            <label for="nivel" class="col-sm-2 col-form-label">Nivel:</label>

                            <div class="col-sm-5">

                            <select name="nivel" id="nivel" class="form-select" required>

                                <option value="">Seleccionar nivel</option>

                            </div>

                        </div>

Para eso nos vamos al controlador en la función créate

 public function create()

    {

        return view('alumnos.create', ['niveles' =>Nivel::all()]);

    }

['niveles' =>Nivel::all()]);

Vista niveles le enviamos todo lo que esta en la tabla niveles

Ahora vamos al formulario para mostrar la información de la BD

 <div class="md-3 row">

                            <label for="nivel" class="col-sm-2 col-form-label">Nivel:</label>

                            <div class="col-sm-5">

                                <select name="nivel" id="nivel" class="form-select" required>

                                   <option value="">Seleccionar nivel</option>

                      @foreach ($niveles  as $nivel )

                                   <option value="{{  $nivel->id }}">{{ $nivel->nombre }}</option>"

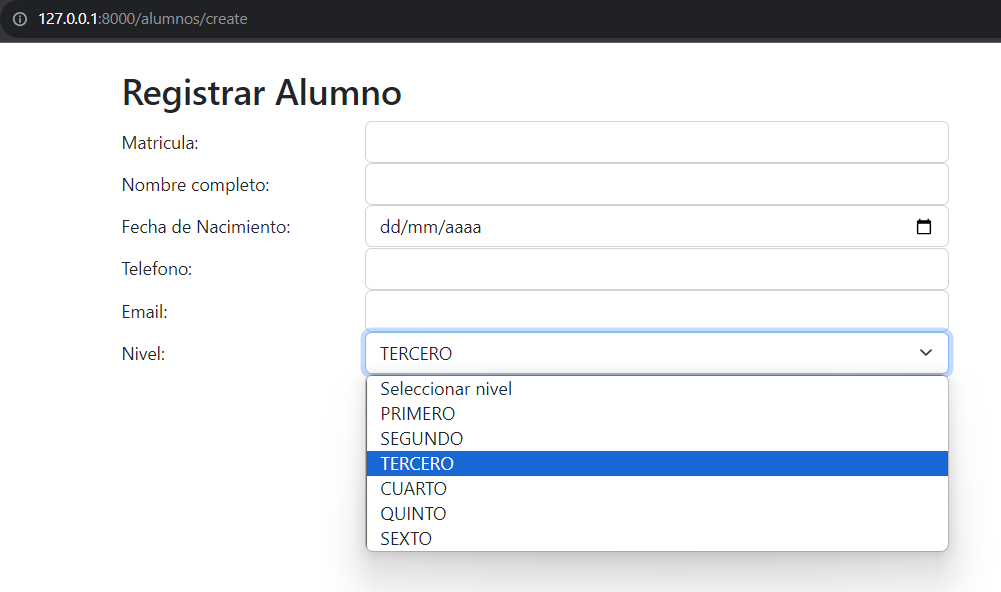
                                   @endforeach

                                </select>

                            </div>

                        </div>

Ahora cargamos y revisamos

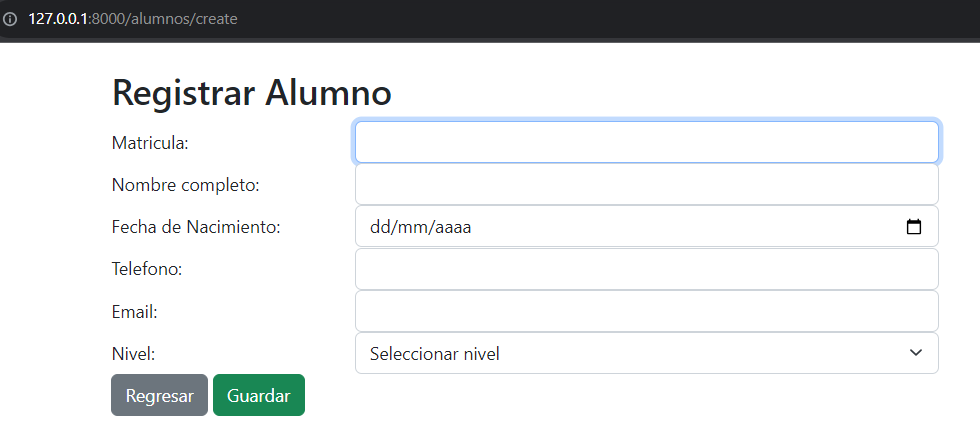


Ahora guardamos los datos del formulario en bases de datos

Agregamos dos botones

                        <a href="{{ url('alumnos') }}"  class="btn btn-secondary">Regresar</a>

                        <button type="sumit" class="btn btn-success">Guardar</button>



Ahora vamos a guardar los datos verificamos el action del formulario

Vemos que apunta a alumnos por el método post

  <form action="{{ url('alumnos') }}" method="post">

Eso quiere decir que cuando estemos usando re resource por el método post eso apunta al store dentro del controlador, dentro de este validaremos lo que cargue

Realizamos las reglas para validar

 public function store(Request $request)

    {

        $request->validate([

            'matricula' => 'required|unique:alumnos|max:10',

            'nombre' => 'required|max:255',

            'fecha' => 'required|date',

            'telefono' => 'required',

            'email' => 'required|max:255',

            'nivel' => 'required',

        ]);

    }

Ahora mostraremos los errores si los hay antes del formulario

<div class="alert alert-warning alert-dismissible fade show" role="alert">

        <ul>

            @foreach($errors->all() as $error)

            <li>{{$error}}</li>

            @endforeach

        </ul>

        <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

    </div>

    @endif

Por último colocamos en código para guardar

Nos vamos al controlador al store al final colocamos el siguiente código

public function store(Request $request)

    {

        $request->validate([

            'matricula' => 'required|unique:alumnos|max:10',

            'nombre' => 'required|max:255',

            'fecha' => 'required|date',

            'telefono' => 'required',

            'email' => 'required|max:255',

            'nivel' => 'required',

        ]);

         /\*\*

     \*  $alumno->CAMPOS DE LA BD = $request->input('CAMPOS DEL FORMULARIO');

     \*/

        $alumno = new Alumno();

        $alumno->matricula = $request->input('matricula');

        $alumno->nombre = $request->input('nombre');

        $alumno->fecha\_nacimiento = $request->input('fecha');

        $alumno->telefono = $request->input('telefono');

        $alumno->email = $request->input('email');

        $alumno->nivel\_id = $request->input('nivel');

        $alumno->save();

                 /\*\*

     \* creamos una vista para comentar que se guardo la informacion

     \*/

        return view("alumnos.message", ['msg' => "Registro Guardado"]);

    }

Luego creamos una vita llamada message

@extends('layout/template')

@section('title', 'Registrar Alumno| Escuela')

@section('contenido')

<main>

    <div class="container py-4">

       <h2>{{ $msg }}/h2>

       <a href="{{ url('alumnos') }}" class="btn btn-secondary">Regresar</a>

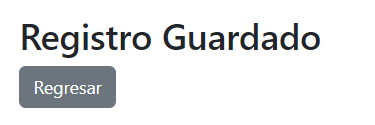
     </div>

</main>

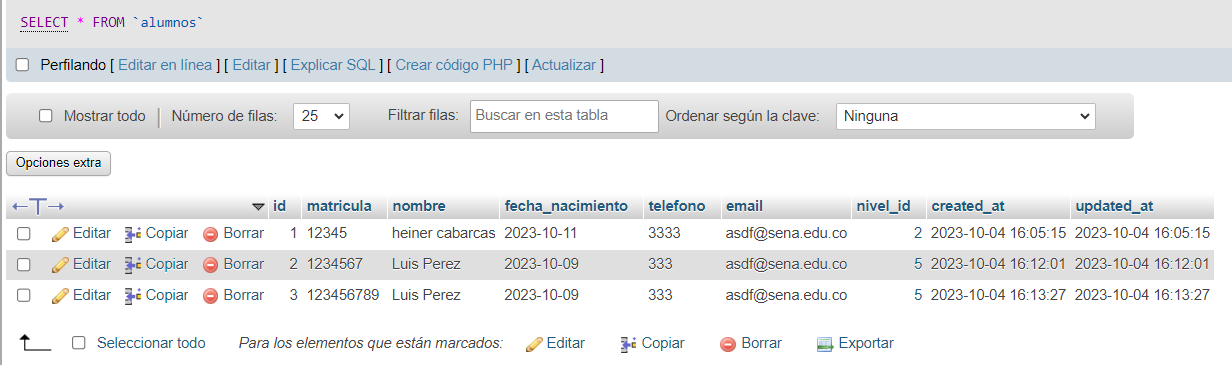
Ahora probamos



Si todo sale bien carga la vista creada



Consultamos la bd y podemos mirar que guarda los archivos



**MOSTRAR REGISTROS**

Nos vamos a index.blade.php

Y creamos una tabla dentro del div después de href

<table class="table table-hover">

        <thead>

            <tr>

                <th>#</th>

                <th>Matricula</th>

                <th>Nombre</th>

                <th>Fecha Nacimiento</th>

                <th>Telefono</th>

                <th>Email</th>

                <th>Nivel</th>

                <th></th>

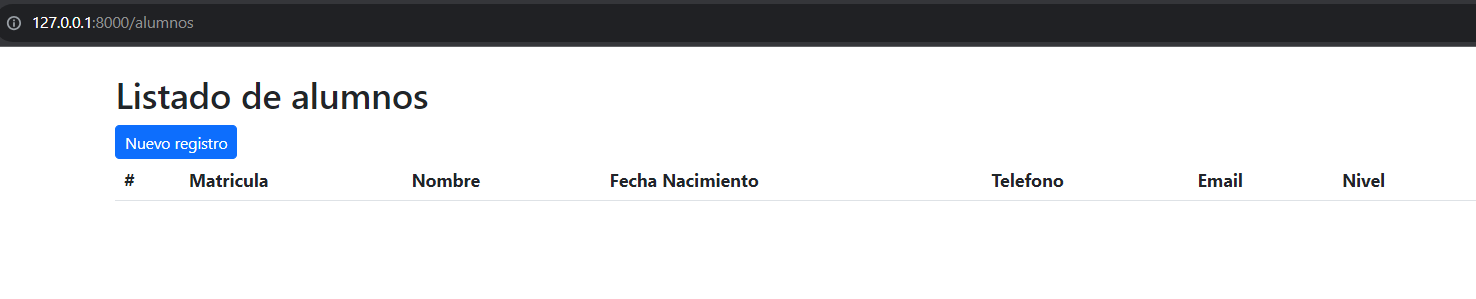
                <th></th>

            </tr>

         </thead>

       </table>

Si cargamos la pagina ya nos aparece la tabla



Ahora le enviaremos valores de la bd de la tabla que se llama alumno

Para eso vamos al controlador y modificamos la función index

Quedando de la siguiente manera

public function index()

    {

        $alumnos = Alumno::all();

        return view('alumnos.index', ['alumnos' => $alumnos]);

    }

        $alumnos = Alumno::all();

Variable =método para obtener todo

        return view('alumnos.index', ['alumnos' => $alumnos]);

Lo que carga de la base de datos se le asigna a la variable alumno y se envía a la vista

Listo ahora vamos a la vista index

Y debajo de la tabla que creamos agregamos @foreach para que recorra la tabla, quedando todo de la siguiente forma

 <table class="table table-hover">

        <thead>

            <tr>

                <th>#</th>

                <th>Matricula</th>

                <th>Nombre</th>

                <th>Fecha Nacimiento</th>

                <th>Telefono</th>

                <th>Email</th>

                <th>Nivel</th>

                <th></th>

                <th></th>

            </tr>

         </thead>

       <tbody>

           @foreach ($alumnos as $alumno)

           <tr>

               <td>{{ $alumno->id }}</td>

               <td>{{ $alumno->matricula }}</td>

               <td>{{ $alumno->nombre}}</td>

               <td>{{ $alumno->fecha\_nacimiento }}</td>

               <td>{{ $alumno->telefono }}</td>

               <td>{{ $alumno->email }}</td>

               <td>{{ $alumno->nivel->nombre }}</td>

               <td></td>

               <td></td>

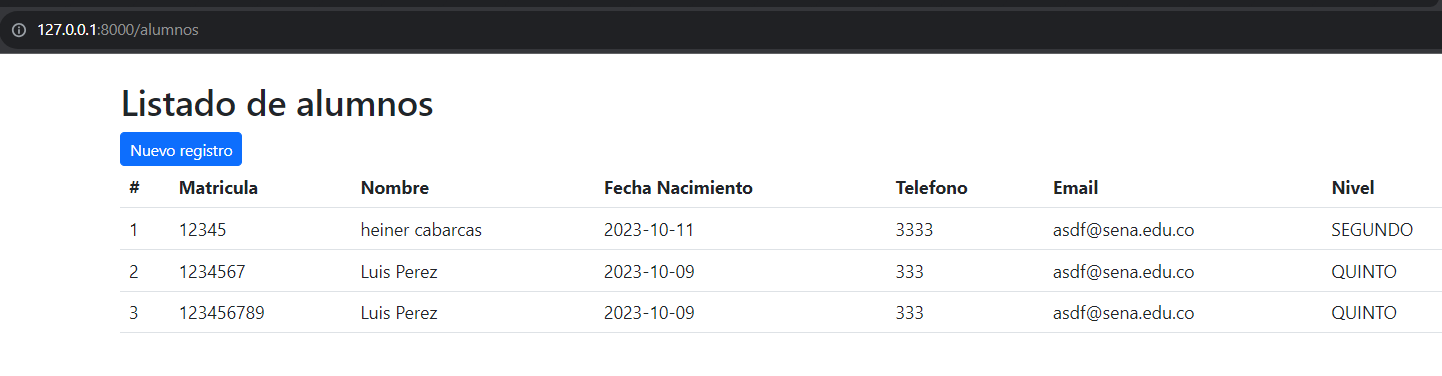
            </tr>

          @endforeach

       </tbody>

       </table>

Luego guardamos y refrescamos



**EDITAR CAMPOS**

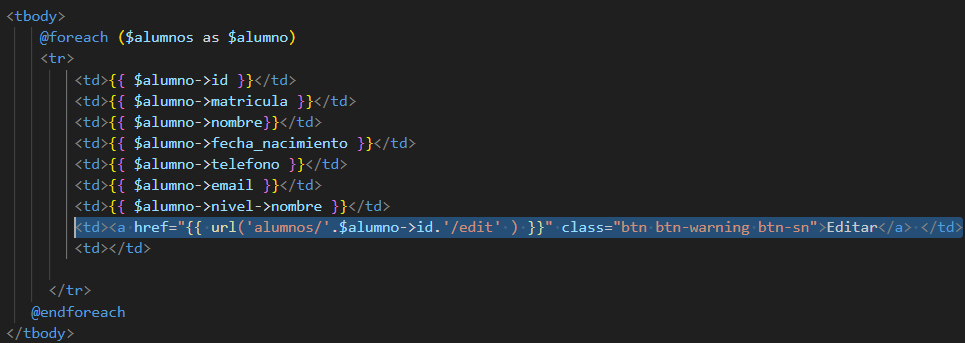
Necesitamos coloca en el archivo index un href, que nos lleve a función editar y cargue una nueva vista. Para eso debemos concatenar alumnos/id/edit

Nos vamos a la vista index

Colocamos el siguiente código

<td><a href="{{ url('alumnos/'.$alumno->id.'/edit' ) }}" class="btn btn-warning btn-sn">Editar</a> </td>

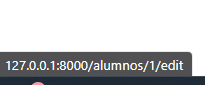
Dentro del @foreach



Guardamos y cargamos la pagina

****

Si colocas el mause en el link de editar, aparecerá en la parte inferior izquierda una dirección y te darás cuenta que la concatenación está bien.



Ahora creamos el formulario editar

Copiamos el archivo créate y lo pegamos. Luego cambiamos el nombre por edit.blade.php

Cambiamos los títulos de crear por editar en mi caso línea 3 y línea 10

El formulario se cambia, pero solo la ruta en mi caso es la línea 26 y agregamos método PUT

<form action="{{ url('alumnos/'.$alumno->id) }}" method="post">

        @method("PUT")

Ya tenemos el formulario debemos llamarlos desde el controler, recuerden que para editar hay que buscar primero

 public function edit($id)

    {

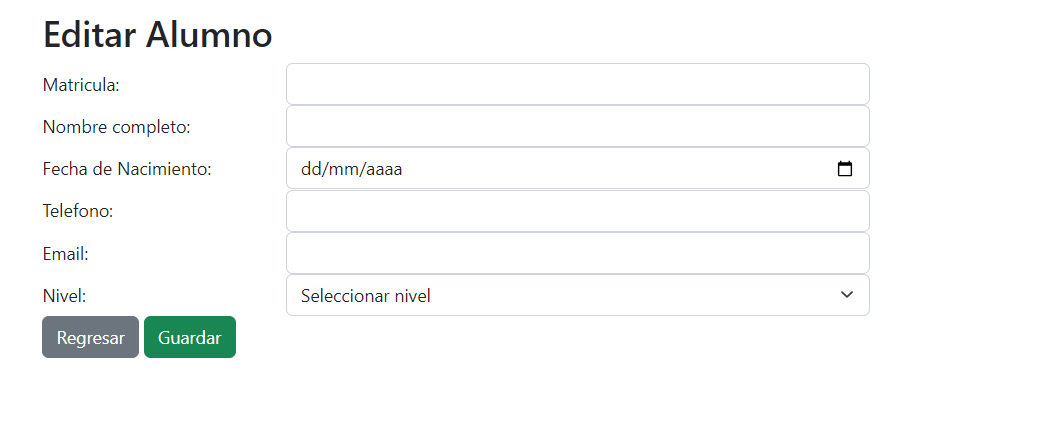
        $alumno = Alumno::find($id);

        return view('alumnos.edit', ['alumno' => $alumno, 'niveles' => Nivel::all()]);

    }

Luego cargamos la pagina podemos observar que el botón llama a la vista edit





Nos falta mostrar la información seleccionada

Para eso vamos al formulario edit y cabíamos lo siguiente (dentro de cada input cambiamos old por la variable $alumno con su respectivo valor

value="{{ old('matricula') }}

se cambia por

value="{{ $alumno->matricula }}

el formulario queda de la siguiente forma

<form action="{{ url('alumnos/'.$alumno->id) }}" method="post">

        @method("PUT")

        @csrf

           <div class="md-3 row">

              <label for="matricula" class="col-sm-2 col-form-label">Matricula:</label>

              <div class="col-sm-5">

                   <input type="text" class="form-control"  name="matricula"  id="matricula" value="{{ $alumno->matricula }}" required>

                </div>

            </div>

            <div class="md-3 row">

                <label for="nombre" class="col-sm-2 col-form-label">Nombre completo:</label>

                <div class="col-sm-5">

                    <input type="text" class="form-control"  name="nombre"  id="nombre" value="{{ $alumno->nombre }}" required>

                </div>

            </div>

            <div class="md-3 row">

                <label for="fecha" class="col-sm-2 col-form-label">Fecha de Nacimiento:</label>

                    <div class="col-sm-5">

                        <input type="date" class="form-control"  name="fecha"  id="fecha" value="{{ $alumno->fecha\_nacimiento }}" required>

                    </div>

                </div>

                    <div class="md-3 row">

                        <label for="telefono" class="col-sm-2 col-form-label">Telefono:</label>

                        <div class="col-sm-5">

                            <input type="text" class="form-control"  name="telefono"  id="telefono" value="{{ $alumno->telefono }}" required>

                        </div>

                    </div>

                        <div class="md-3 row">

                            <label for="email" class="col-sm-2 col-form-label">Email:</label>

                            <div class="col-sm-5">

                                <input type="email" class="form-control"  name="email"  id="email" value="{{ $alumno->email }}" required>

                            </div>

                        </div>

                        <div class="md-3 row">

                            <label for="nivel" class="col-sm-2 col-form-label">Nivel:</label>

                            <div class="col-sm-5">

                                <select name="nivel" id="nivel" class="form-select" required>

                                   <option value="">Seleccionar nivel</option>

                                    @foreach ($niveles  as $nivel )

                                   <option value="{{  $nivel->id }}"  @if ($nivel->id == $alumno->nivel\_id) {{ 'selected' }} @endif >{{ $nivel->nombre }}</option>"

                                   @endforeach

                                </select>

                            </div>

                        </div>

                        <a href="{{ url('alumnos') }}"  class="btn btn-secondary">Regresar</a>

                        <button type="sumit" class="btn btn-success">Guardar</button>

    </form>

Ya buscamos solo nos falta editar los campos en la base de datos

Para eso nos vamos a la función update en el controlador

 public function update(Request $request, $id)

    {

        $request->validate([

            'matricula' => 'required|max:10|unique:alumnos,matricula,' . $id,

            'nombre' => 'required|max:255',

            'fecha' => 'required|date',

            'telefono' => 'required',

            'email' => 'required|max:255',

            'nivel' => 'required',

        ]);

         /\*\*

     \*  $alumno->CAMPOS DE LA BD = $request->input('CAMPOS DEL FORMULARIO');

     \*/

        $alumno = Alumno::find($id);

        $alumno->matricula = $request->input('matricula');

        $alumno->nombre = $request->input('nombre');

        $alumno->fecha\_nacimiento = $request->input('fecha');

        $alumno->telefono = $request->input('telefono');

        $alumno->email = $request->input('email');

        $alumno->nivel\_id = $request->input('nivel');

        $alumno->save();

                   /\*\*

     \* creamos una vista para comentar que se guardo la informacion

     \*/

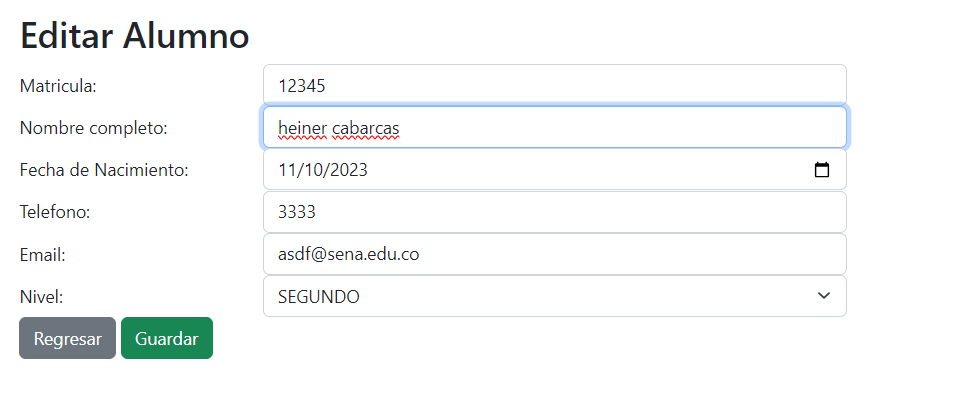
    return view("alumnos.message", ['msg' => "Registro Modificado"]);

    }

Luego cargamos la pagina

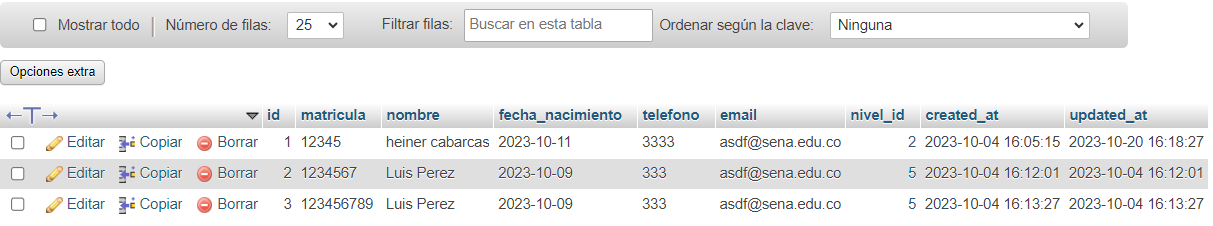


Modificaremos el primero Heiner ca





Vamos a la base de datos y verificamos



Nos damos cuenta que esta modificado

**ELIMINAR REGISTRO**

Para eliminar crearemos un formulario de tipo post, nos vamos a la vista index en el ultimo td agregamos el código

Dentro del @foreach

 <td>

                   <form action="{{ url('alumnos/'.$alumno->id) }}" method="post">

                       @method("DELETE")

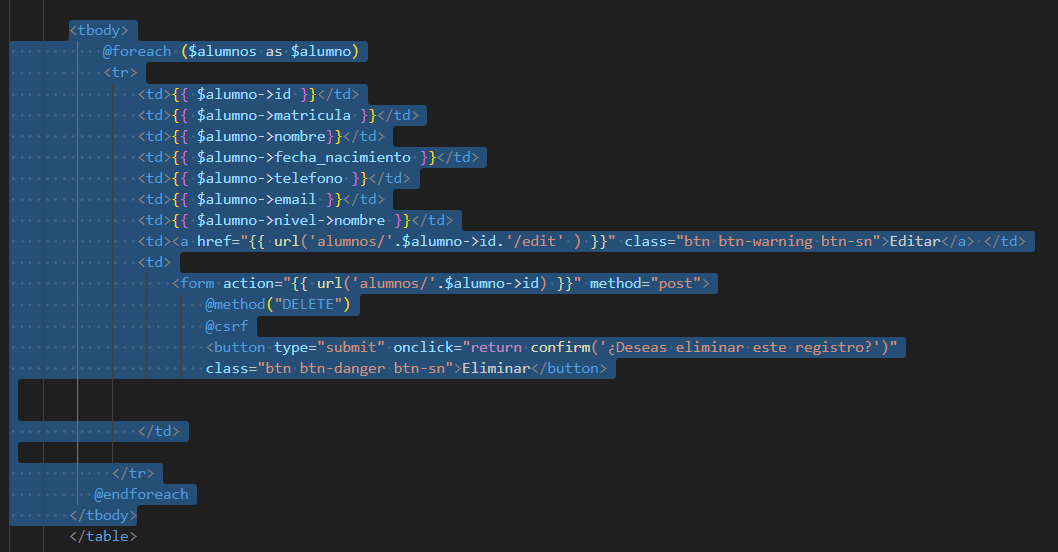
                       @csrf

                       <button type="submit" onclick="return confirm('¿Deseas eliminar este registro?')"

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

               </td>

Quedando todo de la siguiente forma



Ahora nos vamos al controlador función destroy, recuerden que para eliminar hay que primero buscar.

 public function destroy($id)

    {

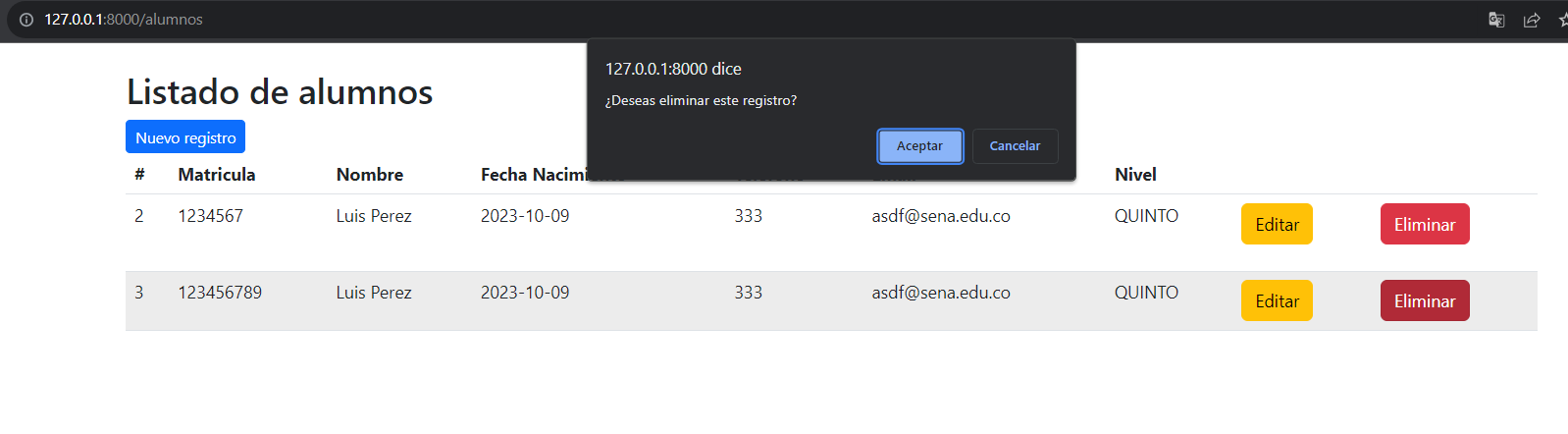
        $alumno = Alumno::find($id);

        $alumno->delete();

        return redirect("alumnos");

    }

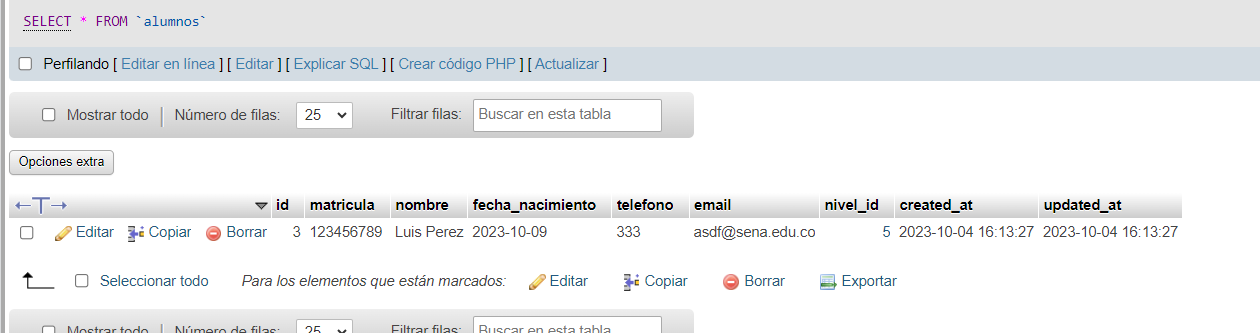
Ejecutamos la pagina y borramos un registro



Revisamos la pagina y la base de datos y nos damos cuenta que esta eliminado el registro



En Base de Datos



A practicar señores