



Unidad # 2

Desarrollo Back End con el Framework del lado del servidor y base de datos.

Programación Computacional IV

Gisela Yasmín García Espinoza



**UNIVERSIDAD
GERARDO BARRIOS**
Líderes en Gestión del Conocimiento





Competencia de la asignatura

Aplicar los lenguajes de programación y tecnologías para la Web con el propósito de utilizarlos en el desarrollo de aplicaciones basándose en estándares internacionales de desarrollo de software trabajando de forma individual y colaborativa.

Contenidos a desarrollar

- CRUD con Laravel
- Formularios con Laravel



¿Cómo crear un CRUD con Laravel?



Crear tabla

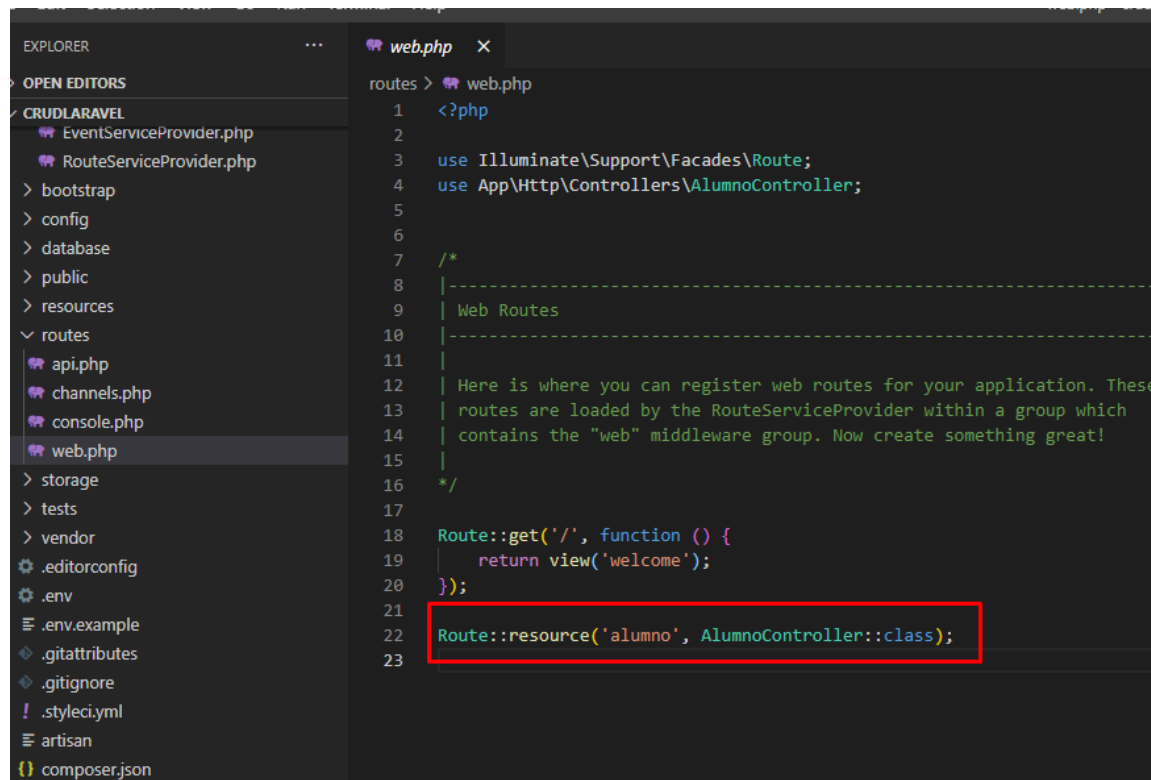
Se crea la tabla para la cual se creara el CRUD, seguido se realizara el proceso de migración

```
PS C:\Users\Gisela\Desktop\crudLaravel> php artisan make:migration alumno --table=alumno --create
Created Migration: 2021_01_11_170428_alumno
PS C:\Users\Gisela\Desktop\crudLaravel> 
```

```
PS C:\Users\Gisela\Desktop\crudLaravel> php artisan migrate
Migrating: 2021_01_11_170428_alumno
Migrated: 2021_01_11_170428_alumno (748.61ms)
```

Fuente: Docente Autor de Contenidos

Agregar ruta de recursos



```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\AlumnoController;
5
6
7 /*
8 |-----
9 | Web Routes
10 |-----
11 |
12 | Here is where you can register web routes for your application. These
13 | routes are loaded by the RouteServiceProvider within a group which
14 | contains the "web" middleware group. Now create something great!
15 |
16 */
17
18 Route::get('/', function () {
19     return view('welcome');
20 });
21
22 Route::resource('alumno', AlumnoController::class);
23
```

Agregar rutas para las operaciones CRUD, Laravel proporciona una ruta de recursos e se encargará del CRUD, que es una ruta para Crear, otra ruta para Recuperar, una ruta separada para Actualizar y finalmente una ruta para Eliminar.

Fuente: Docente Autor de Contenidos

Crear controlador y modelo

Anteriormente, en el paso 4, el segundo parámetro para el recurso es AlumnoController, por lo que se debe crear el controlador y especificar el modelo de recurso agregando --model

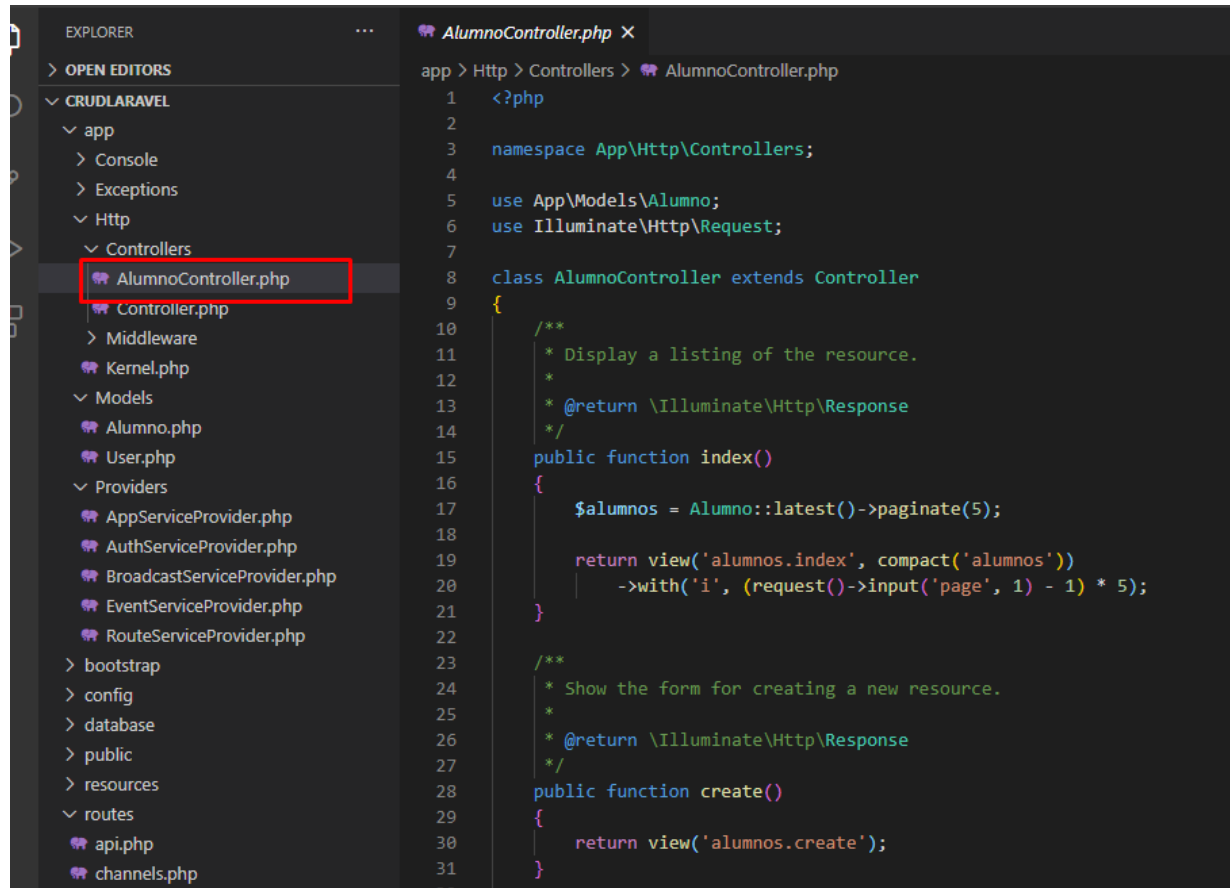
```
PS C:\Users\Gisela\Desktop\crudLaravel> php artisan make:controller AlumnoController --resource --model=Alumno

A App\Models\Alumno model does not exist. Do you want to generate it? (yes/no) [yes]:
> yes

Model created successfully.
Controller created successfully.
```

Fuente: Docente Autor de Contenidos

Editar Controlador



```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Alumno;
6  use Illuminate\Http\Request;
7
8  class AlumnoController extends Controller
9  {
10     /**
11      * Display a listing of the resource.
12      *
13      * @return \Illuminate\Http\Response
14      */
15     public function index()
16     {
17         $alumnos = Alumno::latest()->paginate(5);
18
19         return view('alumnos.index', compact('alumnos'))
20             ->with('i', (request()->input('page', 1) - 1) * 5);
21     }
22
23     /**
24      * Show the form for creating a new resource.
25      *
26      * @return \Illuminate\Http\Response
27      */
28     public function create()
29     {
30         return view('alumnos.create');
31     }
32 }
```

```
public function store(Request $request)
{
    $request->validate([
        'nombre' => 'required',
        'apellido' => 'required'
    ]);

    Alumno::create($request->all());
    $alumnos = Alumno::latest()->paginate(5);

    return view('alumnos.index', compact('alumnos'))
        ->with('i', (request()->input('page', 1) - 1) * 5);
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return Response
 */
public function show($id)
{
    $alumno = Alumno::find($id);
    return view('alumnos.show', compact('alumno'));
}

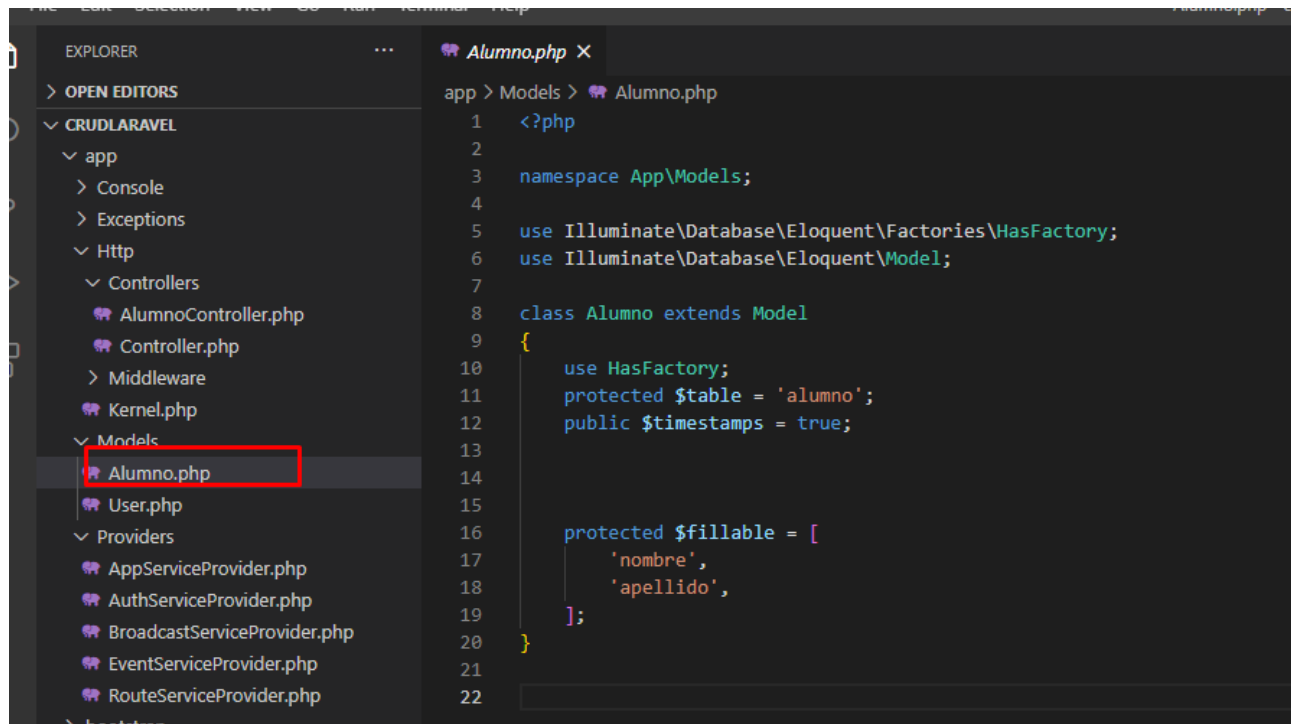
/**
 * Display the specified resource.
 *
 * @param int $id
 * @return Response
 */
public function edit($id)
{
    $alumno = Alumno::find($id);
    return view('alumnos.edit', compact('alumno'));
}

/**
```

Fuente: Docente Autor de Contenidos

Editar Modelo

En el archivo de deben especificar los campos de la tabla.

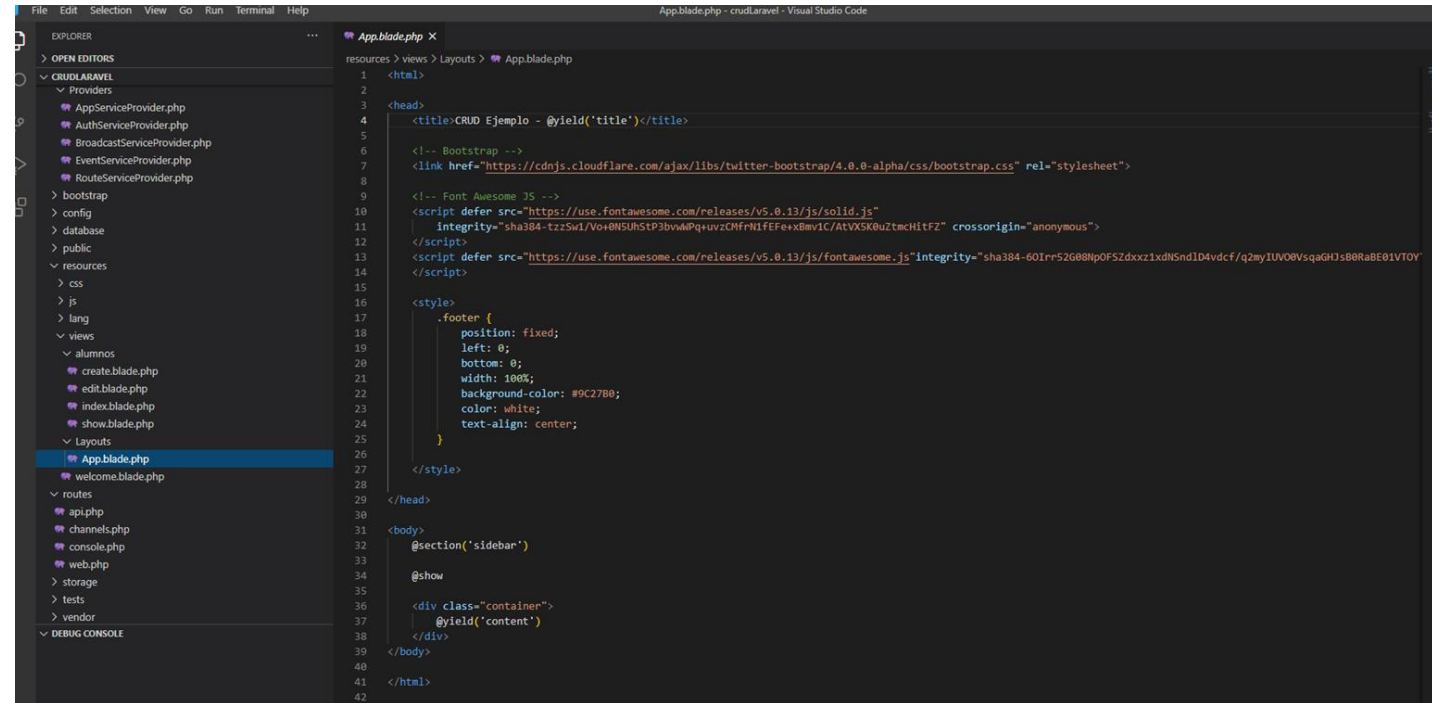
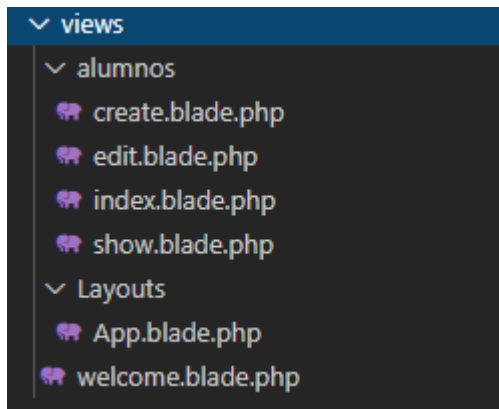


```
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Alumno extends Model
9  {
10     use HasFactory;
11     protected $table = 'alumno';
12     public $timestamps = true;
13
14
15
16     protected $fillable = [
17         'nombre',
18         'apellido',
19     ];
20 }
21
22
```

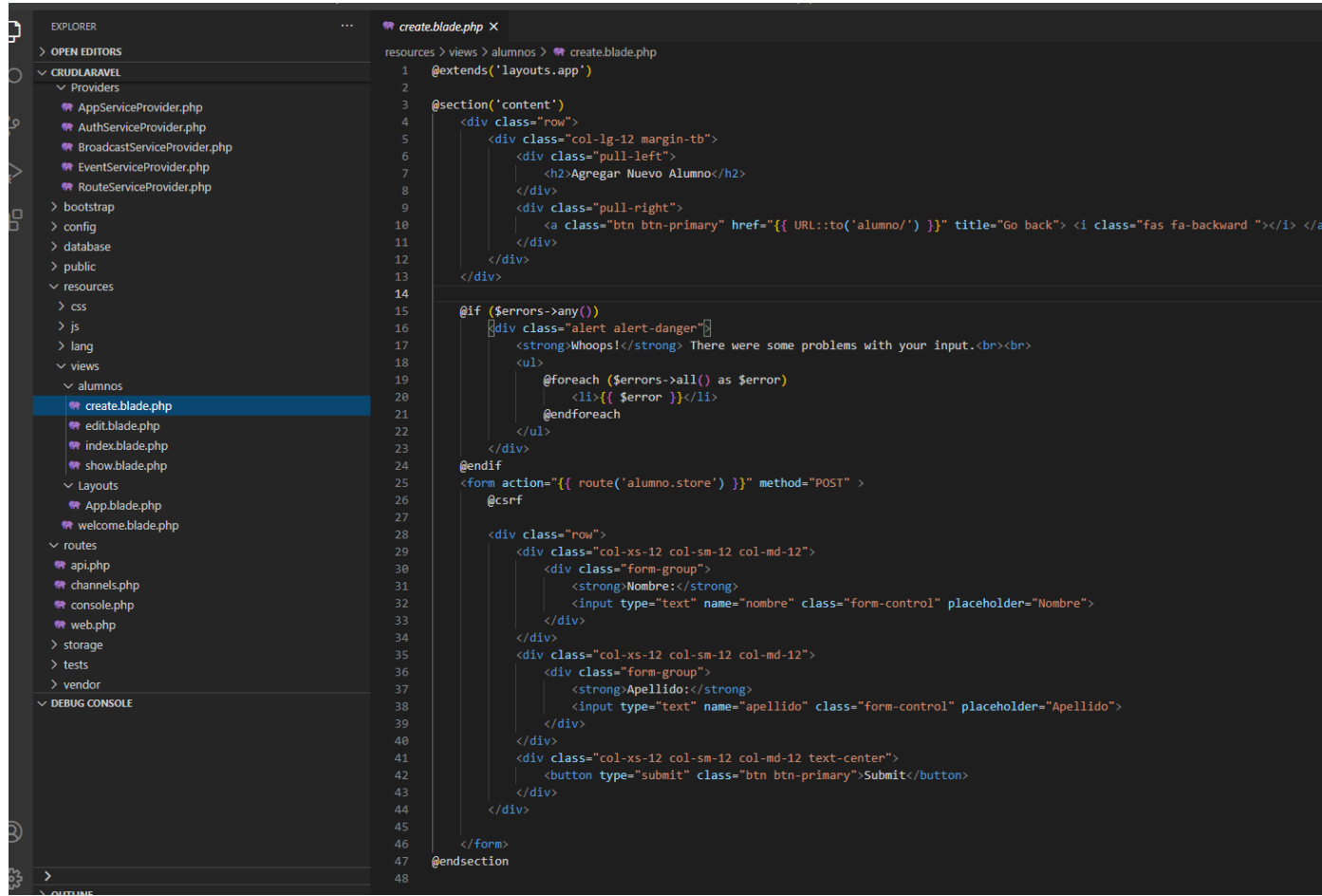
Fuente: Docente Autor de Contenidos

Agregar las vistas

Laravel hace uso del sistema de plantillas de hoja para las vistas, dentro de la carpeta vistas se crearan dos carpetas Layouts y Alumno.



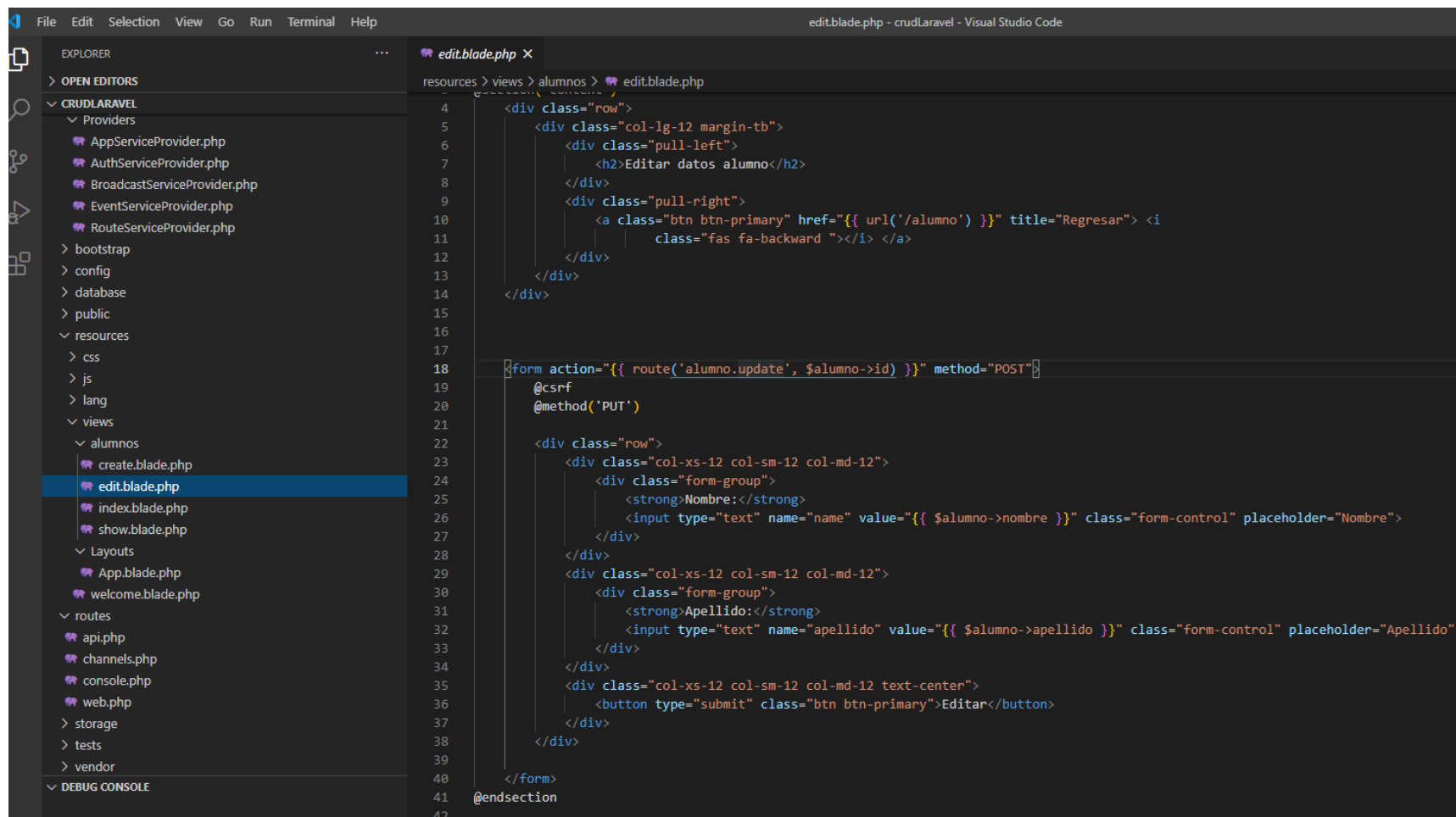
Agregar las vistas | Create



```
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="row">
5         <div class="col-lg-12 margin-tb">
6             <div class="pull-left">
7                 <h2>Agregar Nuevo Alumno</h2>
8             </div>
9             <div class="pull-right">
10                 <a class="btn btn-primary" href="{{ URL::to('alumno/') }}" title="Go back"> <i class="fas fa-backward"></i> </a>
11             </div>
12         </div>
13     </div>
14
15     @if ($errors->any())
16         <div class="alert alert-danger">
17             <strong>Whoops!</strong> There were some problems with your input.<br><br>
18             <ul>
19                 @foreach ($errors->all() as $error)
20                     <li>{{ $error }}</li>
21                 @endforeach
22             </ul>
23         </div>
24     @endif
25     <form action="{{ route('alumno.store') }}" method="POST" >
26         @csrf
27
28         <div class="row">
29             <div class="col-xs-12 col-sm-12 col-md-12">
30                 <div class="form-group">
31                     <strong>Nombre:</strong>
32                     <input type="text" name="nombre" class="form-control" placeholder="Nombre">
33                 </div>
34             </div>
35             <div class="col-xs-12 col-sm-12 col-md-12">
36                 <div class="form-group">
37                     <strong>Apellido:</strong>
38                     <input type="text" name="apellido" class="form-control" placeholder="Apellido">
39                 </div>
40             </div>
41             <div class="col-xs-12 col-sm-12 col-md-12 text-center">
42                 <button type="submit" class="btn btn-primary">Submit</button>
43             </div>
44         </div>
45
46     </form>
47 @endsection
48
```

Fuente: Docente Autor de Contenidos

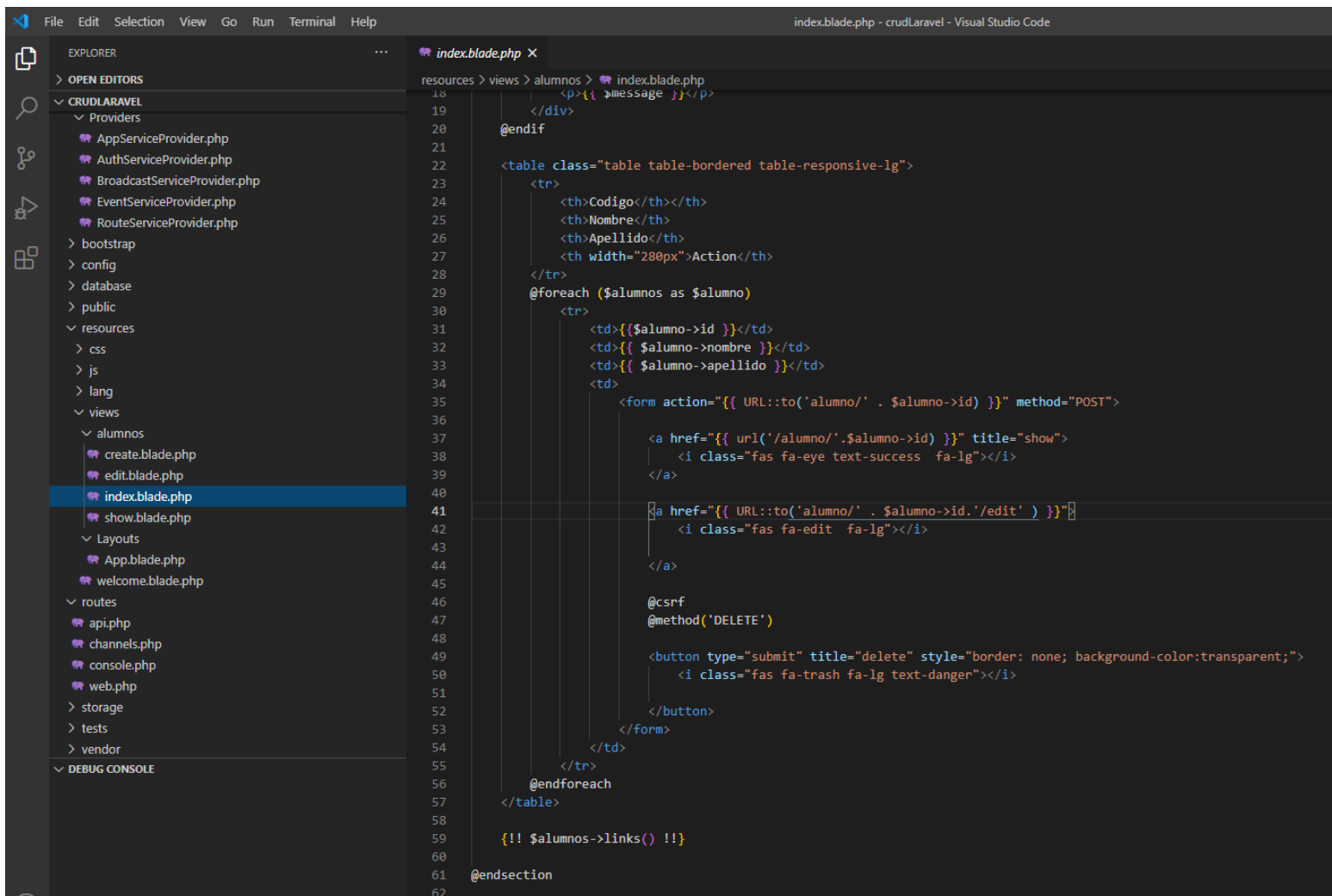
Agregar las vistas | Edit



```
4 <div class="row">
5 <div class="col-lg-12 margin-tb">
6 <div class="pull-left">
7 <h2>Editar datos alumno</h2>
8 </div>
9 <div class="pull-right">
10 <a class="btn btn-primary" href="{{ url('/alumno') }}" title="Regresar"> <i
11 class="fas fa-backward "></i> </a>
12 </div>
13 </div>
14 </div>
15
16
17
18 <form action="{{ route('alumno.update', $alumno->id) }}" method="POST">
19 @csrf
20 @method('PUT')
21
22 <div class="row">
23 <div class="col-xs-12 col-sm-12 col-md-12">
24 <div class="form-group">
25 <strong>Nombre:</strong>
26 <input type="text" name="name" value="{{ $alumno->nombre }}" class="form-control" placeholder="Nombre">
27 </div>
28 </div>
29 <div class="col-xs-12 col-sm-12 col-md-12">
30 <div class="form-group">
31 <strong>Apellido:</strong>
32 <input type="text" name="apellido" value="{{ $alumno->apellido }}" class="form-control" placeholder="Apellido">
33 </div>
34 </div>
35 <div class="col-xs-12 col-sm-12 col-md-12 text-center">
36 <button type="submit" class="btn btn-primary">Editar</button>
37 </div>
38 </div>
39 </form>
40
41 @endsection
42
```

Fuente: Docente Autor de Contenidos

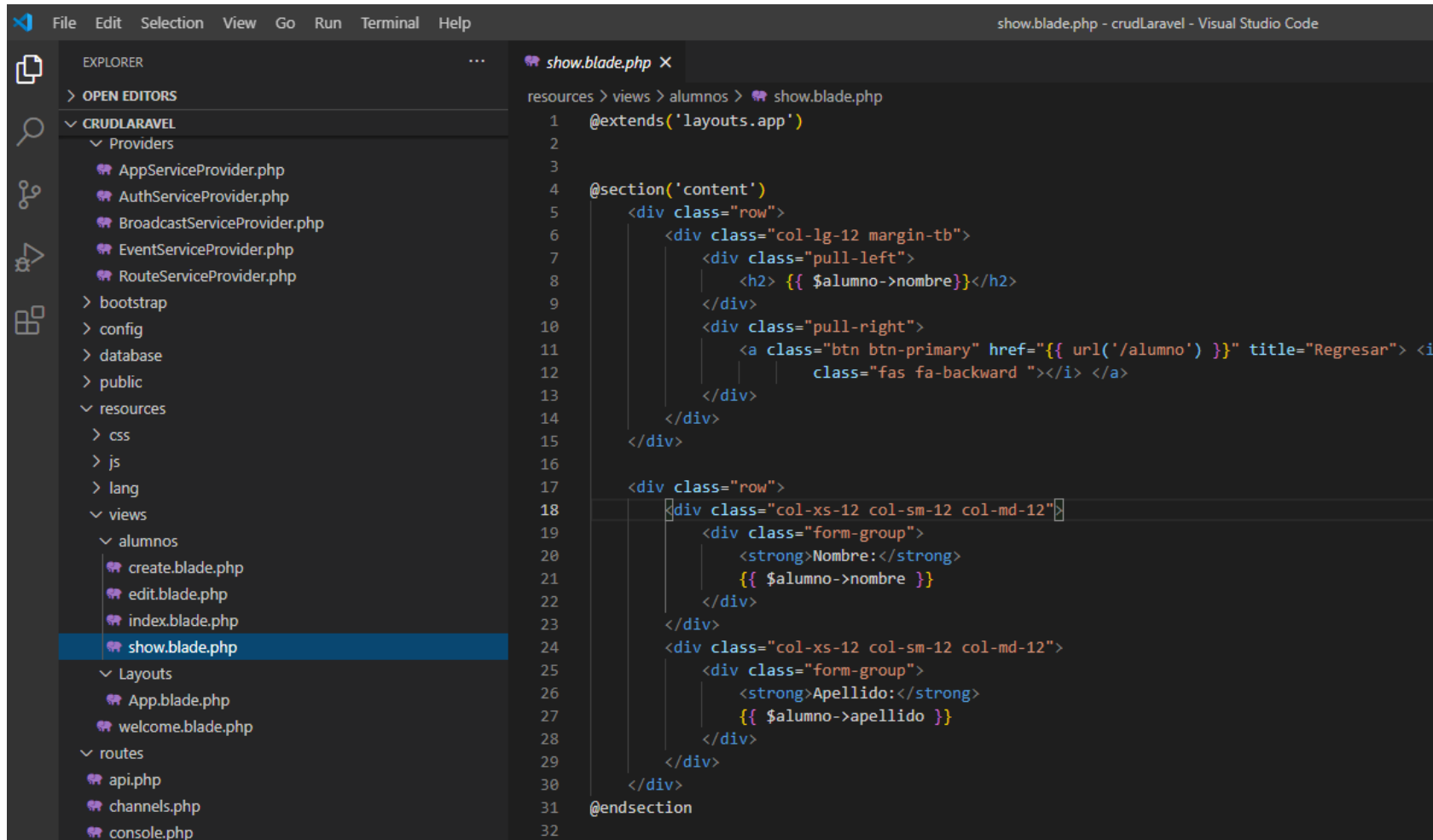
Agregar las vistas | Index



```
18 <p>{{ message }}</p>
19 </div>
20 @endif
21
22 <table class="table table-bordered table-responsive-lg">
23   <tr>
24     <th>Codigo</th></th>
25     <th>Nombre</th>
26     <th>Apellido</th>
27     <th width="280px">Action</th>
28   </tr>
29   @foreach ($alumnos as $alumno)
30     <tr>
31       <td>{{ $alumno->id }}</td>
32       <td>{{ $alumno->nombre }}</td>
33       <td>{{ $alumno->apellido }}</td>
34       <td>
35         <form action="{{ URL::to('alumno/' . $alumno->id) }}" method="POST">
36
37           <a href="{{ url('alumno/' . $alumno->id) }}" title="show">
38             <i class="fas fa-eye text-success fa-lg"></i>
39           </a>
40
41           <a href="{{ URL::to('alumno/' . $alumno->id . '/edit' ) }}">
42             <i class="fas fa-edit fa-lg"></i>
43
44           </a>
45
46           @csrf
47           @method('DELETE')
48
49           <button type="submit" title="delete" style="border: none; background-color:transparent;">
50             <i class="fas fa-trash fa-lg text-danger"></i>
51
52           </button>
53         </form>
54       </td>
55     </tr>
56   @endforeach
57 </table>
58
59 {!! $alumnos->links() !!}
60
61 @endsection
62
```

Fuente: Docente Autor de Contenidos

Agregar las vistas | Show



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure for 'crudLaravel'. The 'views' directory is expanded, showing 'alumnos' with files 'create.blade.php', 'edit.blade.php', 'index.blade.php', and 'show.blade.php'. The 'show.blade.php' file is selected and its content is displayed in the main editor. The code is a Blade template that extends 'layouts.app' and uses the '@section' directive to define the content area. It includes a Bootstrap row with two columns: one for the student's name and another for a 'Regresar' button. Below this, there are two form groups for displaying the student's name and last name, each using a Bootstrap column class.

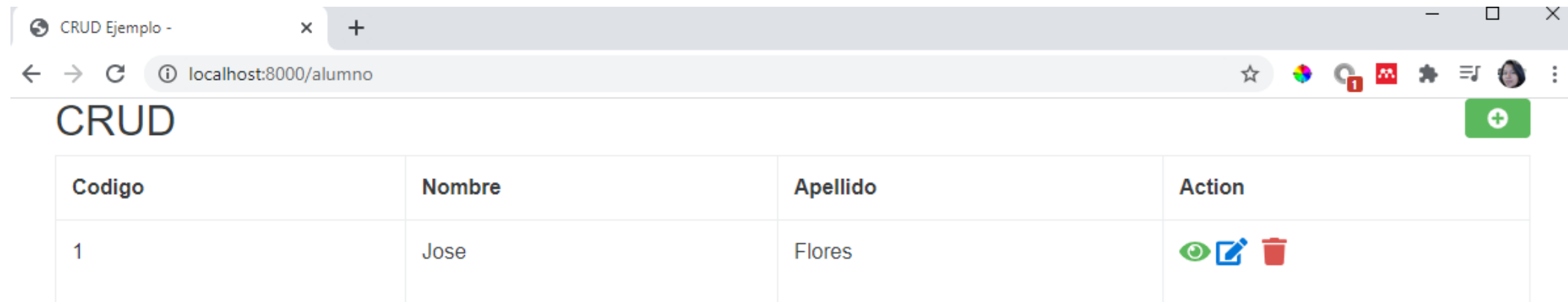
```
1 @extends('layouts.app')
2
3
4 @section('content')
5     <div class="row">
6         <div class="col-lg-12 margin-tb">
7             <div class="pull-left">
8                 <h2> {{ $alumno->nombre }}</h2>
9             </div>
10            <div class="pull-right">
11                <a class="btn btn-primary" href="{{ url('/alumno') }}" title="Regresar"> <i
12                    class="fas fa-backward "></i> </a>
13            </div>
14        </div>
15    </div>
16
17    <div class="row">
18        <div class="col-xs-12 col-sm-12 col-md-12">
19            <div class="form-group">
20                <strong>Nombre:</strong>
21                {{ $alumno->nombre }}
22            </div>
23        </div>
24        <div class="col-xs-12 col-sm-12 col-md-12">
25            <div class="form-group">
26                <strong>Apellido:</strong>
27                {{ $alumno->apellido }}
28            </div>
29        </div>
30    </div>
31 @endsection
32
```

Fuente: Docente Autor de Contenidos

Prueba

Finalmente ejecutar la aplicación, y visualizar la salida en el navegador

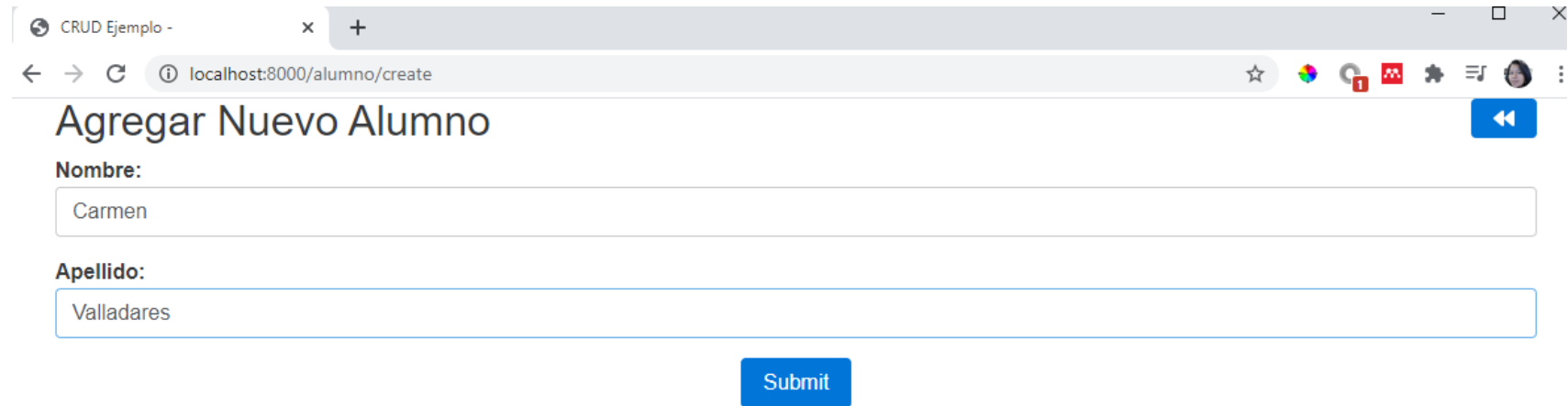
```
PS C:\Users\Gisela\Desktop\crudLaravel> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
[Mon Jan 11 14:34:46 2021] PHP 8.0.0 Development Server (http://127.0.0.1:8000) started
```



Fuente: Docente Autor de Contenidos

Prueba

Nuevo registro



A screenshot of a web browser window with the title "CRUD Ejemplo -". The address bar shows "localhost:8000/alumno/create". The page title is "Agregar Nuevo Alumno". There are two input fields: "Nombre:" with the value "Carmen" and "Apellido:" with the value "Valladares". A blue "Submit" button is at the bottom right.

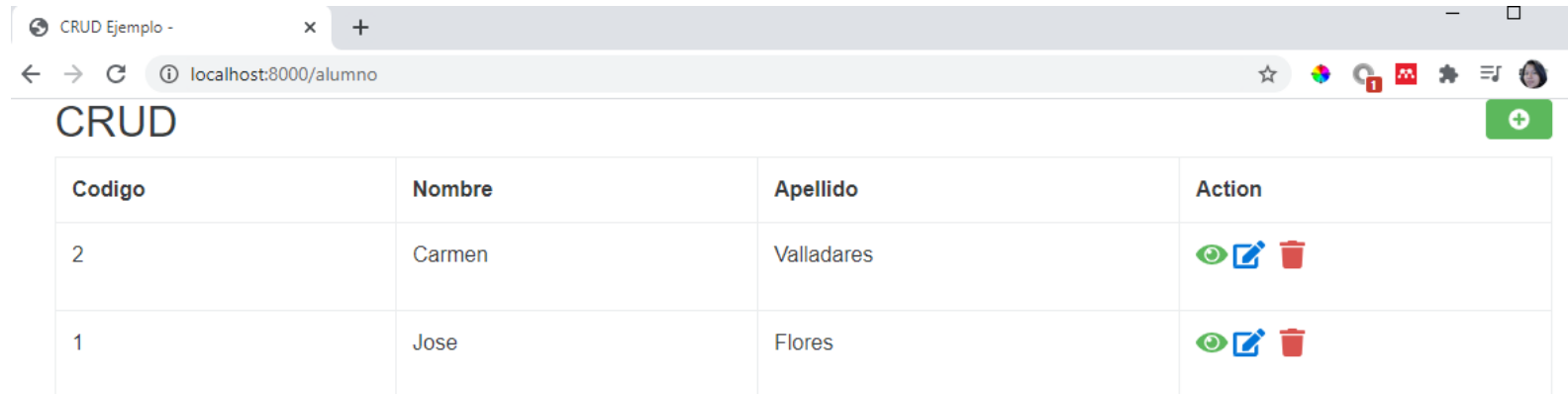
Nombre:

Carmen







Apellido:

Valladares

Submit



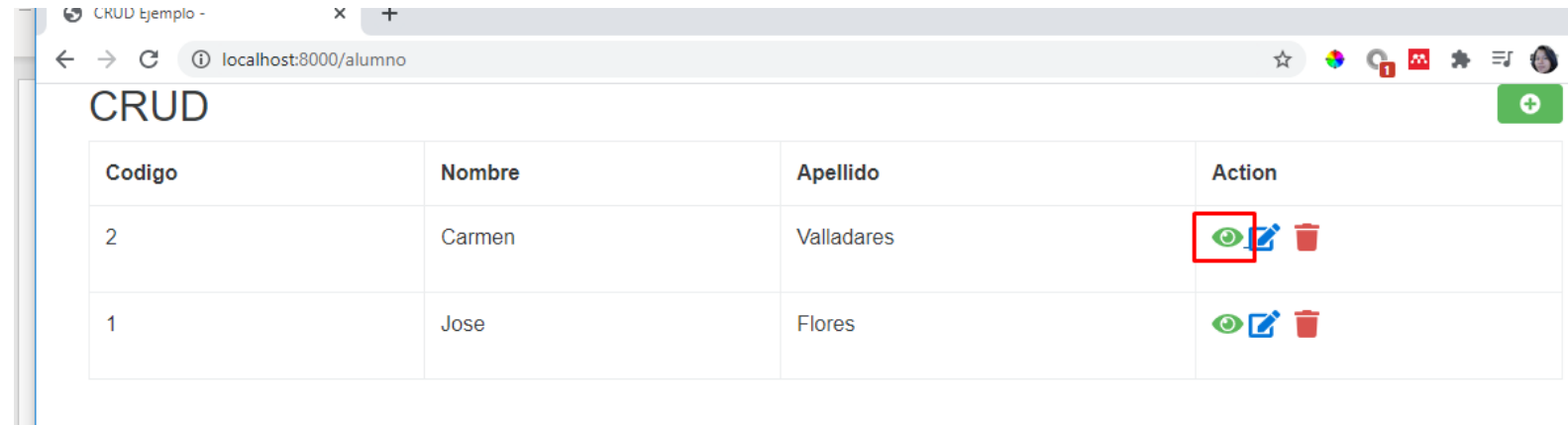
A screenshot of a web browser window with the title "CRUD Ejemplo -". The address bar shows "localhost:8000/alumno". The page title is "CRUD". There is a green "+" button in the top right corner. Below the title is a table with 4 columns: "Codigo", "Nombre", "Apellido", and "Action". The table contains two rows of data.







Codigo	Nombre	Apellido	Action
2	Carmen	Valladares	  
1	Jose	Flores	  

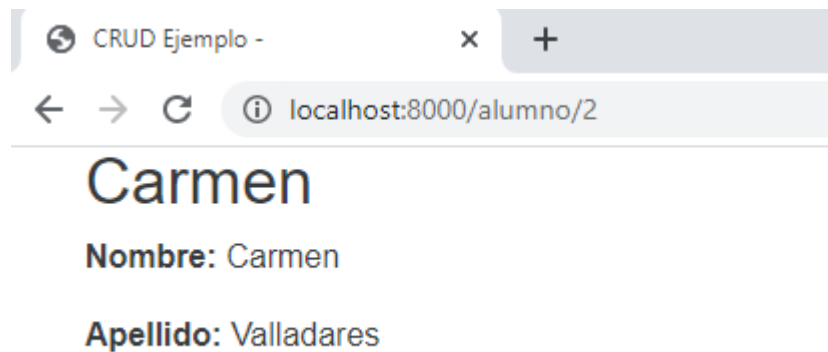
Fuente: Docente Autor de Contenidos

Prueba

Visualizar registro



Codigo	Nombre	Apellido	Action
2	Carmen	Valladares	  
1	Jose	Flores	  



Carmen

Nombre: Carmen

Apellido: Valladares

Tabla de actividades

Nombre de la actividad	CRUD Laravel
Tipo de actividad	Tarea
Tipo de participación	En parejas
Competencias específica de la asignatura	Integrar tecnologías y arquitecturas web, paradigmas de programación y gestión de datos con DBMS para desarrollar aplicaciones web mediante el lenguaje de programación PHP trabajando con un enfoque lógico y aplicando las buenas prácticas de programación de manera individual y colaborativa.
Instrucciones	Después de haber dado lectura a los contenidos de la semana y participar activamente en la video conferencia, realice un CRUD para la base de datos asignada por docente. Finalmente deberá compartir la url del repositorio con la solución.
Fecha de entrega	La fecha límite de participación será el domingo al final de la semana , a las 11:59 p.m.
Instrumento de evaluación	Rúbrica
Ponderación	50% - Laboratorio I

Recursos Complementarios

Recurso	Título	Cita Referencial
Sitio Web	CRUD Laravel	https://www.digitalocean.com/community/tutorials/simple-laravel-crud-with-resource-controllers
Sitio Web	Tutorial de Laravel 8 CRUD por ejemplo	https://www.techiediaries.com/laravel-8-crud-tutorial/

¿Preguntas?

¡Muchas gracias!