

Manual del programador

Indice:

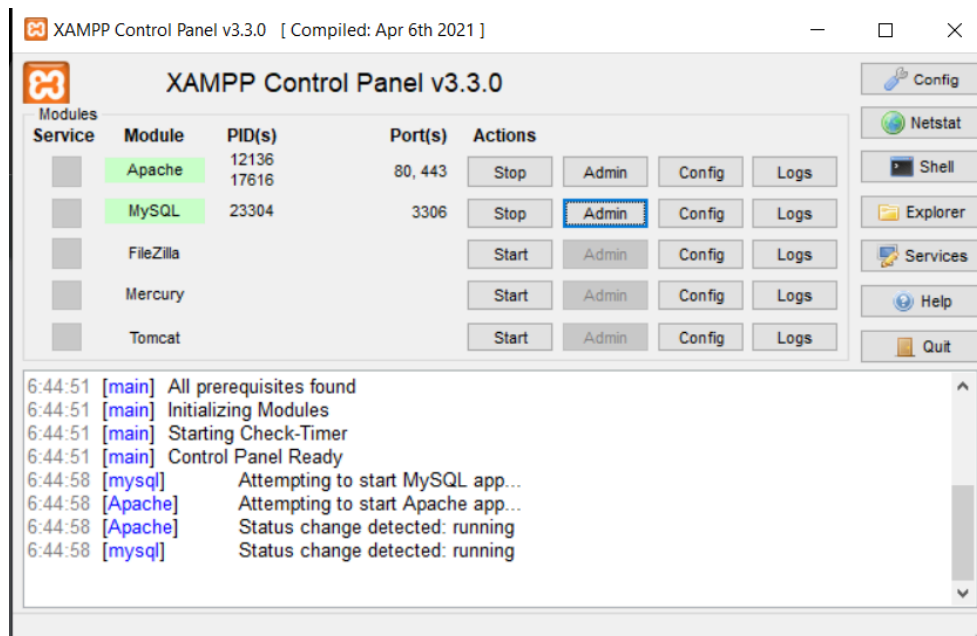
1. Requisitos.....	2
2. Creación del proyecto.....	2
3. Gestión de la base de datos y migraciones.....	3
4. Autenticación.....	6
5. Creación de los CRUDs.....	8
6. Acceso a los CRUDs.....	10
7. Creación de los enlaces a Juegos y Consolas.....	10
8. Acceder a los datos de Consolas desde Juegos.....	11
9. Creación de un select para las Consolas.....	12
10. Modificación del nombre del encabezado Consolas de mi tabla.....	13
11. Crear un seeder.....	14
12. Cambiar el idioma al español.....	15
13. Ejecución del proyecto.....	17

1. Requisitos

Para trabajar con laravel vamos a necesitar instalar Xampp(servidor local), Composer(Gestión de dependencias PHP) y Node.js(paquete de librerías en javascript).

2. Creación del proyecto

Antes de nada debemos abrir Xampp y arrancar tanto Apache como MySQL.



Después debemos posicionarnos en la carpeta htdocs de la carpeta xampp en una terminal y ejecutar el siguiente comando:

```
htdocs>composer create-project laravel/laravel proyecto-laravel
```

```

- Installing sebastian/diff (5.1.0): Extracting archive
- Installing sebastian/comparator (5.0.1): Extracting archive
- Installing sebastian/code-unit (2.0.0): Extracting archive
- Installing sebastian/cli-parser (2.0.0): Extracting archive
- Installing phpunit/php-timer (6.0.0): Extracting archive
- Installing phpunit/php-text-template (3.0.1): Extracting archive
- Installing phpunit/php-invoker (4.0.0): Extracting archive
- Installing phpunit/php-file-iterator (4.1.0): Extracting archive
- Installing theseer/tokenizer (1.2.2): Extracting archive
- Installing sebastian/lines-of-code (2.0.2): Extracting archive
- Installing sebastian/complexity (3.2.0): Extracting archive
- Installing sebastian/code-unit-reverse-lookup (3.0.0): Extracting archive
- Installing phpunit/php-code-coverage (10.1.11): Extracting archive
- Installing phar-io/version (3.2.1): Extracting archive
- Installing phar-io/manifest (2.0.3): Extracting archive
- Installing myclabs/deep-copy (1.11.1): Extracting archive
- Installing phpunit/phpunit (10.5.10): Extracting archive
- Installing spatie/backtrace (1.5.3): Extracting archive
- Installing spatie/flare-client-php (1.4.4): Extracting archive
- Installing spatie/ignition (1.12.0): Extracting archive
- Installing spatie/laravel-ignition (2.4.2): Extracting archive
60 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

83 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

 INFO  No publishable resources for tag [laravel-assets].

No security vulnerability advisories found.
> @php artisan key:generate --ansi

 INFO  Application key set successfully.

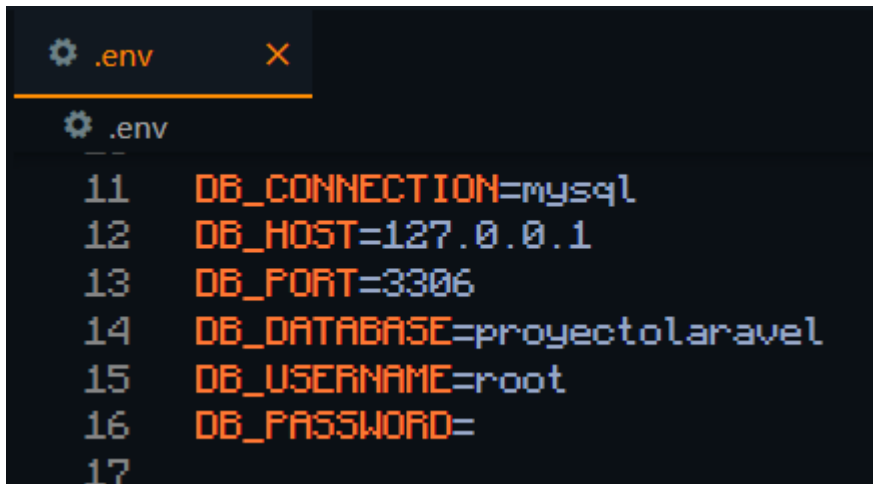
```

Nos posicionamos en el proyecto creado con `htdocs>cd proyecto-laravel` y lo abrimos en Visual Studio Code con `proyecto-laravel> code .`

3. Gestión de la base de datos y migraciones

Lo primero que tenemos que hacer es crear la base de datos “proyectolaravel”. Para ello debemos abrir la aplicación Xampp y pulsar el botón Admin que está en la sección de MySQL. De esta forma abriremos phpMyAdmin y crearemos la base de datos “proyectolaravel”.

Una vez creada, en el VSC accedemos a `proyecto-laravel/.env` y cambiamos el nombre de la base de datos a `proyectolaravel`.



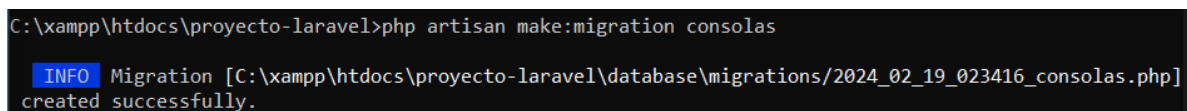
```
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=proyectolaravel
15 DB_USERNAME=root
16 DB_PASSWORD=
17
```

Para crear las migraciones es muy importante crearlas por orden(primero las tablas que no contengan un campo relacional) ya que si las creamos desordenadas pueden llegar a dar problemas a la larga.

Vamos a crear las migraciones de la tabla Consolas y la tabla Juegos. Como Juegos depende de Consolas, crearemos Consolas primero.

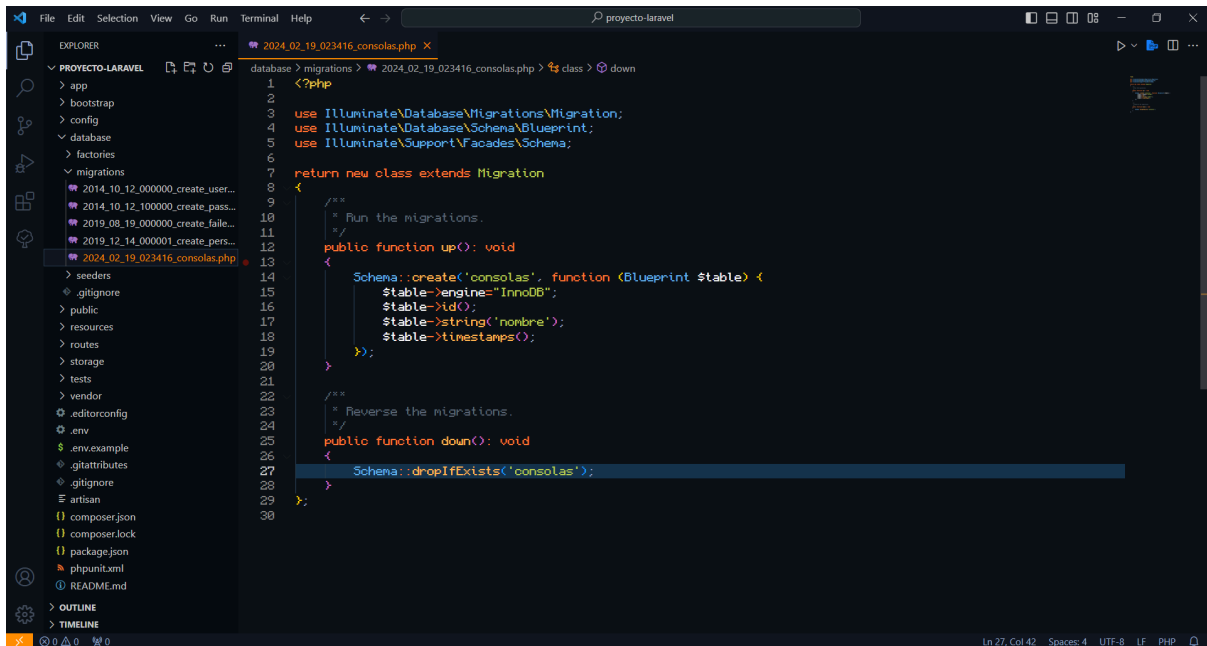
Consolas:

Ejecutamos `proyecto-laravel>php artisan make:migration consolas`



```
C:\xampp\htdocs\proyecto-laravel>php artisan make:migration consolas
INFO Migration [C:\xampp\htdocs\proyecto-laravel\database\Migrations\2024_02_19_023416_consolas.php]
created successfully.
```

Vamos a la migración de consolas en `proyecto-laravel/database/migrations/` y rellenamos las funciones `up()` y `down()` como indica la imagen.

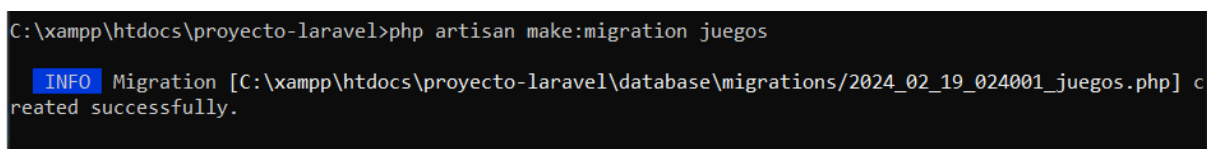


Ejecutamos `proyecto-laravel>php artisan migrate` para crear las tablas.

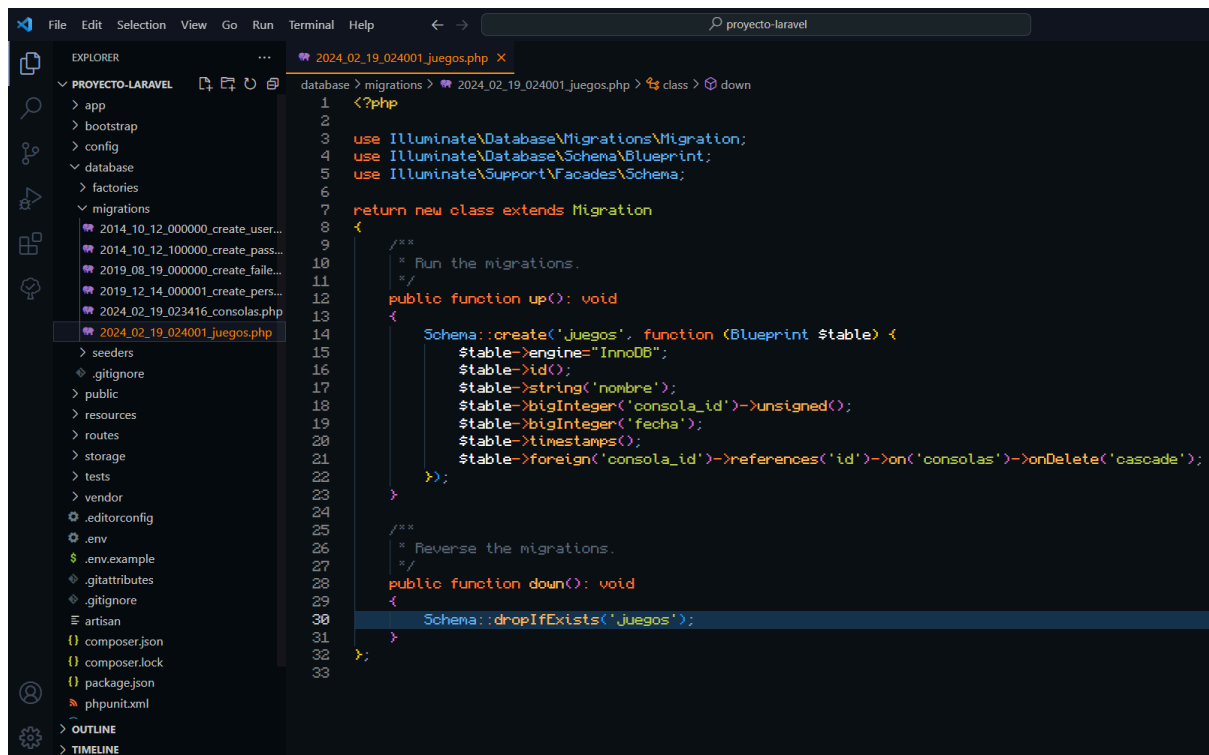


Juegos:

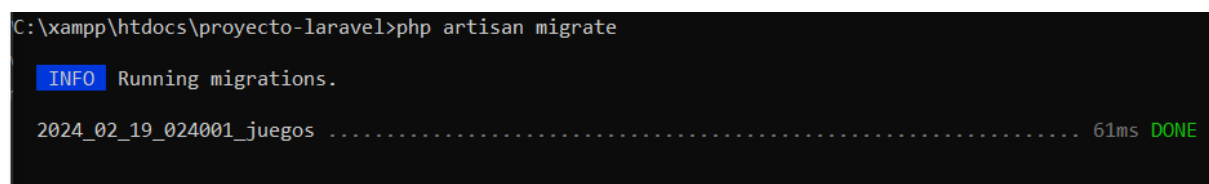
Ejecutamos `proyecto-laravel>php artisan make:migration juegos`



Vamos a la migración de juegos en `proyecto-laravel/database/migrations/` y rellenamos las funciones `up()` y `down()` como indica la imagen.



Ejecutamos `proyecto-laravel>php artisan migrate` para crear la tabla Juegos.



4. Autenticación

Para este proyecto vamos a utilizar ui con bootstrap. Para ello debemos de ejecutar los siguientes comandos en consola:

- `proyecto-laravel>composer require laravel/ui`

```

C:\xampp\htdocs\proyecto-laravel>composer require laravel/ui
./composer.json has been updated
Running composer update laravel/ui
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
  - Locking laravel/ui (v4.4.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Installing laravel/ui (v4.4.0): Extracting archive
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
laravel/ui ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

83 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

 INFO  No publishable resources for tag [laravel-assets].

No security vulnerability advisories found.
Using version ^4.4 for laravel/ui

```

- proyecto-laravel>php artisan ui bootstrap --auth

```

C:\xampp\htdocs\proyecto-laravel>php artisan ui bootstrap --auth

The [Controller.php] file already exists. Do you want to replace it? (yes/no) [no]
y yes

 INFO  Authentication scaffolding generated successfully.

 INFO  Bootstrap scaffolding installed successfully.

 WARN  Please run [npm install && npm run dev] to compile your fresh scaffolding.

```

- proyecto-laravel>npm install

```

C:\xampp\htdocs\proyecto-laravel>npm install

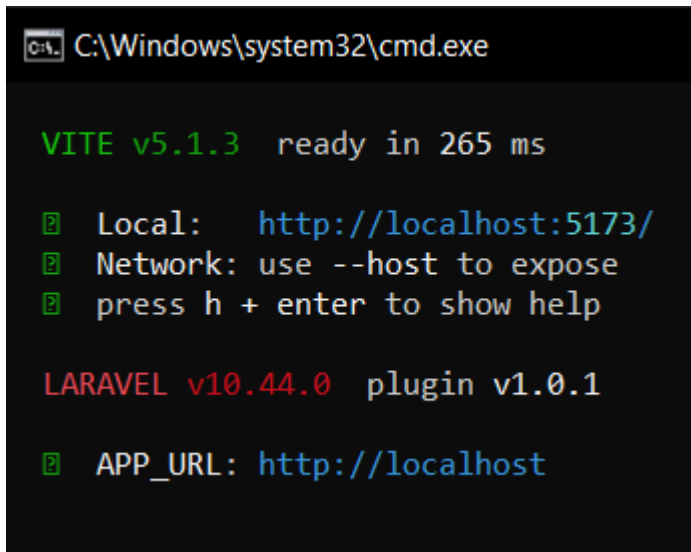
added 39 packages, and audited 40 packages in 15s

8 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

```


- proyecto-laravel>npm run dev (este último se debe dejar ejecutado en otro terminal y tenerlo ejecutado siempre que ejecutemos el proyecto)



```
C:\Windows\system32\cmd.exe

VITE v5.1.3 ready in 265 ms

  Local:   http://localhost:5173/
  Network: use --host to expose
  press h + enter to show help

LARAVEL v10.44.0 plugin v1.0.1

  APP_URL: http://localhost
```

Como esta terminal va a estar encargada de mantener el comando `npm run dev`, debemos abrir otra terminal y nos posicionamos en proyecto-laravel para seguir realizando la página.

5. Creación de los CRUDs

Para generar los CRUDs vamos a utilizar crud-generator. Para ello debemos de ejecutar los siguientes comandos:

- proyecto-laravel>composer require ibex/crud-generator --dev

```

C:\xampp\htdocs\proyecto-laravel>composer require ibex/crud-generator --dev
./composer.json has been updated
Running composer update ibex/crud-generator
Loading composer repositories with package information
Updating dependencies
Lock file operations: 2 installs, 0 updates, 0 removals
  - Locking ibex/crud-generator (v1.4.1)
  - Locking laravelcollective/html (v6.4.1)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 2 installs, 0 updates, 0 removals
  - Installing laravelcollective/html (v6.4.1): Extracting archive
  - Installing ibex/crud-generator (v1.4.1): Extracting archive
Package laravelcollective/html is abandoned, you should avoid using it. Use spatie/laravel-html instead.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

ibex/crud-generator ..... DONE
laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
laravel/ui ..... DONE
laravelcollective/html ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

84 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

 INFO  No publishable resources for tag [laravel-assets].

No security vulnerability advisories found.
Using version ^1.4 for ibex/crud-generator

```

- proyecto-laravel>php artisan vendor:publish --tag=crud
-

```

C:\xampp\htdocs\proyecto-laravel>php artisan vendor:publish --tag=crud

 INFO  Publishing [crud] assets.

Copying file [C:\xampp\htdocs\proyecto-laravel\vendor\ibex\crud-generator\src\config\crud.php] to [C:\xampp\htdocs\proyecto-laravel\config\crud.php] DONE

```

- proyecto-laravel>php artisan make:crud consolas

```

C:\xampp\htdocs\proyecto-laravel>php artisan make:crud consolas
Running Crud Generator ...
Creating Controller ...
Creating Model ...
Creating Views ...
Created Successfully.

```

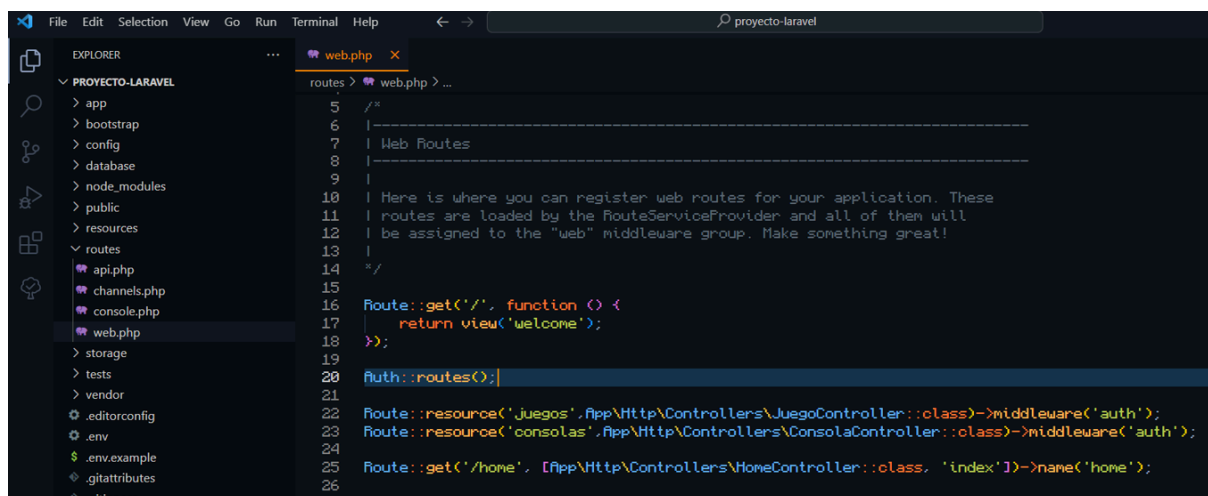
- proyecto-laravel>php artisan make:crud juegos

```
C:\xampp\htdocs\proyecto-laravel>php artisan make:crud juegos
Running Crud Generator ...
Creating Controller ...
Creating Model ...
Creating Views ...
Created Successfully.
```

De esta forma ya deberían de estar creados los modelos, controladores y vistas para el funcionamiento básico de nuestra página.

6. Acceso a los CRUDs

Para poder acceder a los CRUDs debemos de crear sus rutas de acceso. En VSC vamos a `proyecto-laravel/routes/web.php` donde vamos a añadir las dos rutas (`Route::resource`) y añadimos el middleware de autenticación para bloquear su acceso si no hay un usuario logueado.

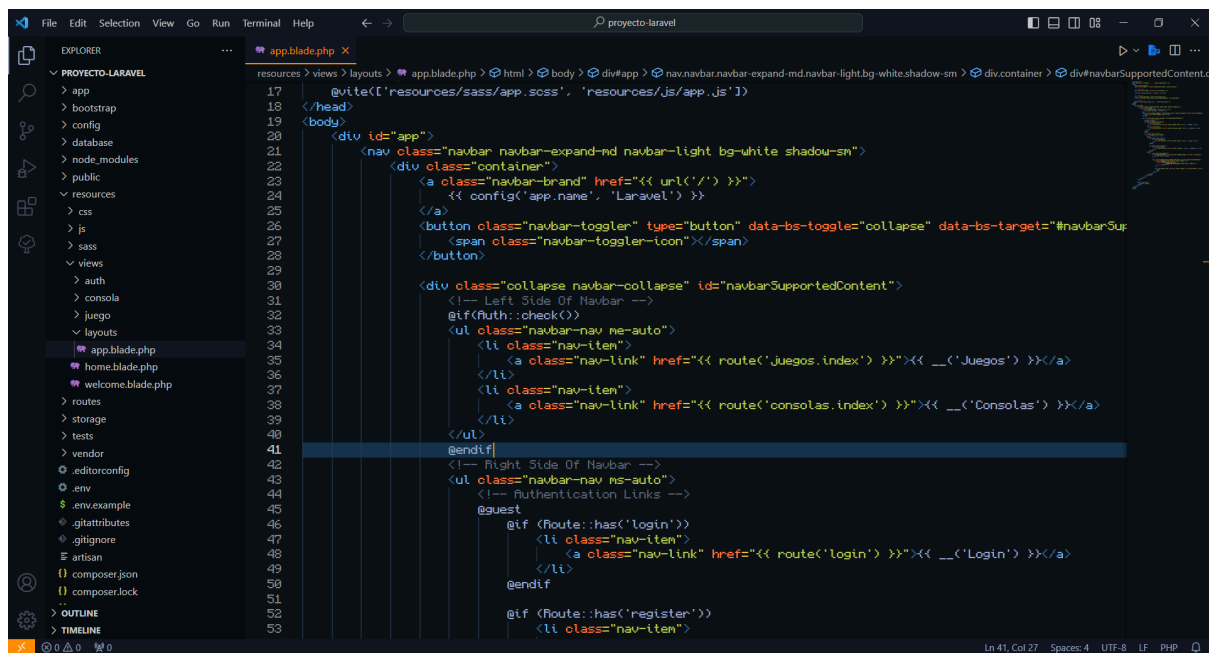


```
5  /*
6  |-----
7  | Web Routes
8  |-----
9  |
10 | Here is where you can register web routes for your application. These
11 | routes are loaded by the RouteServiceProvider and all of them will
12 | be assigned to the "web" middleware group. Make something great!
13 |
14 | */
15
16 Route::get('/', function () {
17     return view('welcome');
18 });
19
20 Auth::routes();
21
22 Route::resource('juegos', App\Http\Controllers\JuegoController::class)->middleware('auth');
23 Route::resource('consolas', App\Http\Controllers\ConsolaController::class)->middleware('auth');
24
25 Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])->name('home');
26
```

7. Creación de los enlaces a Juegos y Consolas

Vamos a crear un menu para acceder a juegos y consolas en `proyecto-laravel/resources/views/layouts/app.blade.php`, pero solo si el usuario está logueado.

Para ello cambiamos el código que hay entre el comentario "Left Side Of Navbar" y el comentario "Right Side Of Navbar" por el siguiente:



8. Acceder a los datos de Consolas desde Juegos

Para poder acceder a los datos de la tabla Consolas desde la tabla Juegos debemos añadir las siguientes líneas a

proyecto-laravel/app/Http/Controllers/JuegoController.php :

- use App\Models\Consola a los imports

```
use App\Models\Juego;
use Illuminate\Http\Request;
use App\Models\Consola;
```

- La variable \$consolas y añadir 'consolas' como parámetro a compact en la función create() y en la función edit()

```

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    $juego = new Juego();
    $consolas = Consola::pluck('nombre','id');
    return view('juego.create', compact('juego','consolas'));
}

```

```

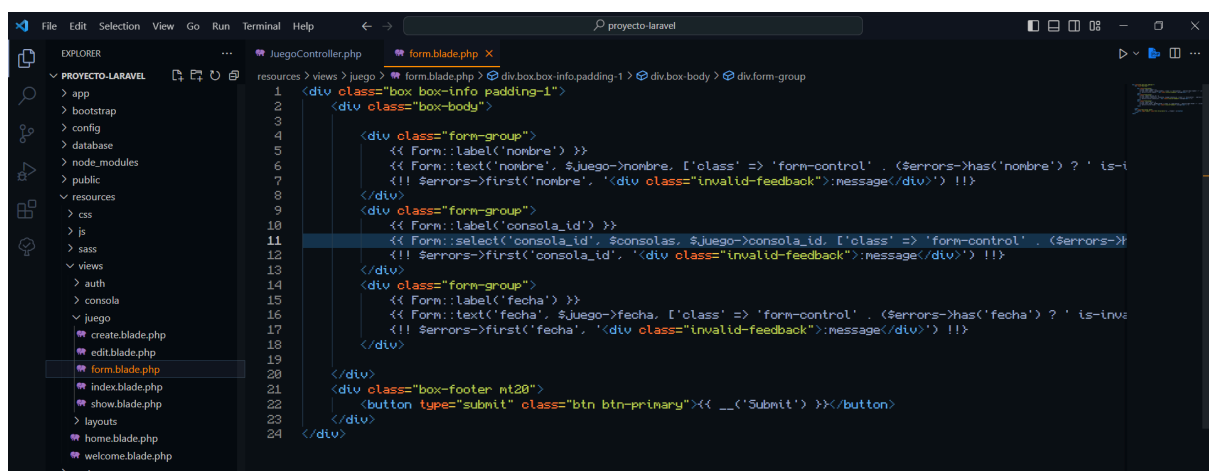
/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    $juego = Juego::find($id);
    $consolas = Consola::pluck('nombre','id');
    return view('juego.edit', compact('juego','consolas'));
}

```

9. Creación de un select para las Consolas

Creamos un select para elegir consolas al crear juegos nuevos en `proyecto-laravel/resources/views/juego/form.blade.php`

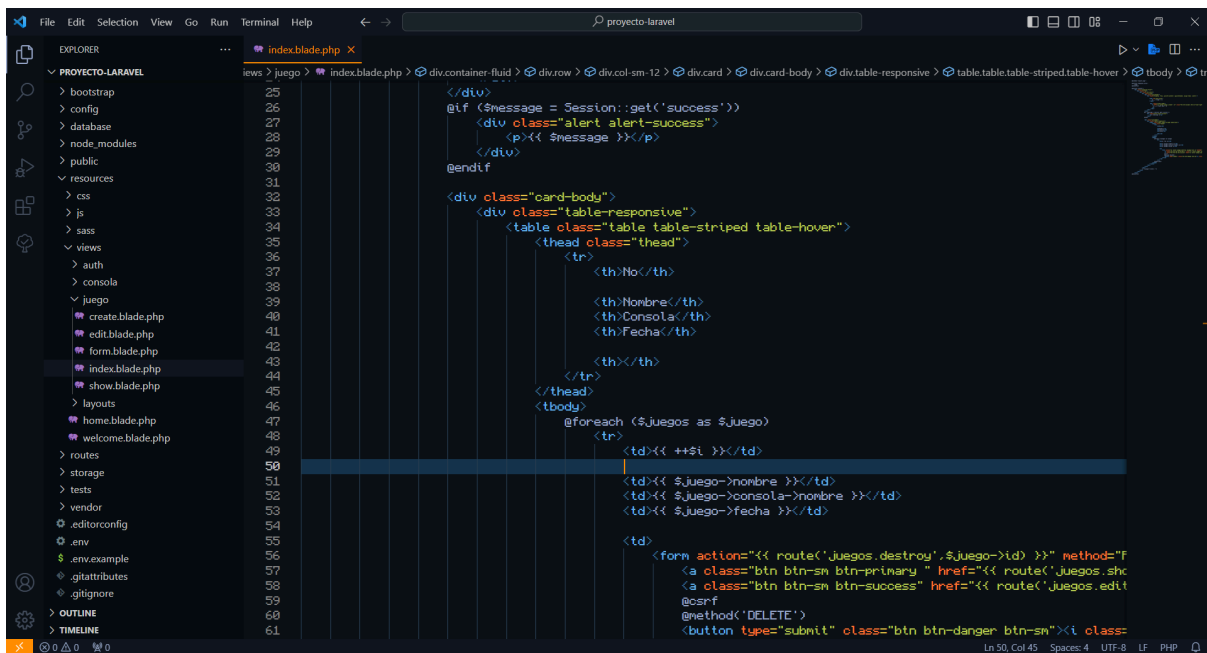
Para ello sustituimos en la línea seleccionada el `Form::text` por `Form::select` e introducimos el nuevo parámetro `$consolas`.



10. Modificación del nombre del encabezado Consolas de mi tabla

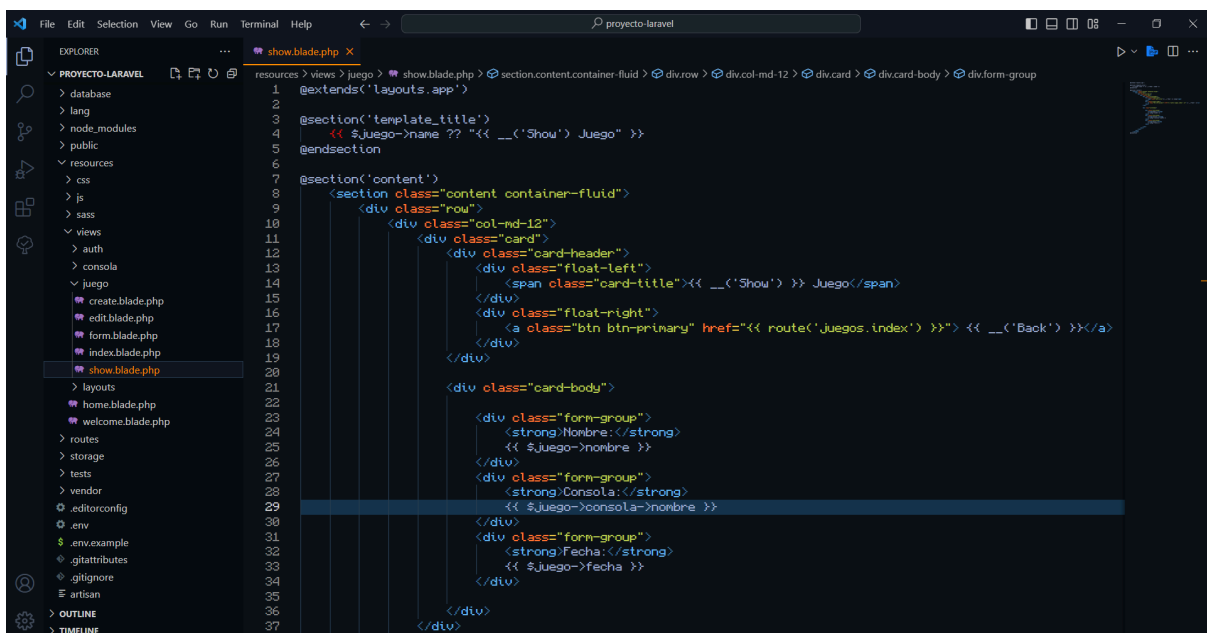
Vamos a `proyecto-laravel/resources/views/juego/index.blade.php` y cambiamos:

- "Consola Id" por "Consola".
- `$juego->consola_id` por `$juego->consola->nombre`



```
25 </div>
26 @if ($message = Session::get('success'))
27 <div class="alert alert-success">
28 <p>{{ $message }}</p>
29 </div>
30 @endif
31
32 <div class="card-body">
33 <div class="table-responsive">
34 <table class="table table-striped table-hover">
35 <thead class="thead">
36 <tr>
37 <th>No</th>
38 <th>Nombre</th>
39 <th>Consola</th>
40 <th>Fecha</th>
41 </tr>
42 </thead>
43 <tbody>
44 @foreach ($juegos as $juego)
45 <tr>
46 <td>{{ ++$i }}</td>
47 <td>{{ $juego->nombre }}</td>
48 <td>{{ $juego->consola_id }}</td>
49 <td>{{ $juego->fecha }}</td>
50 </tr>
51 @endforeach
52 </tbody>
53 </table>
54 </div>
55 </div>
56
57 <div class="form-group">
58 <strong>Nombre:</strong>
59 <input type="text" value="{{ $juego->nombre }}" />
60 </div>
61 <div class="form-group">
62 <strong>Consola:</strong>
63 <input type="text" value="{{ $juego->consola_id }}" />
64 </div>
65 <div class="form-group">
66 <strong>Fecha:</strong>
67 <input type="text" value="{{ $juego->fecha }}" />
68 </div>
69 <div class="form-group">
70 <button type="submit" class="btn btn-primary">Guardar</button>
71 </div>
```

También vamos a hacer lo mismo en `proyecto-laravel/resources/views/juego/show.blade.php`



```
1 @extends('layouts.app')
2
3 @section('template_title')
4 <strong>{{ $juego->nombre }}</strong>
5 @endsection
6
7 @section('content')
8 <div class="content container-fluid">
9 <div class="row">
10 <div class="col-md-12">
11 <div class="card">
12 <div class="card-header">
13 <div class="float-left">
14 <span class="card-title">{{ $juego->nombre }}</span>
15 </div>
16 <div class="float-right">
17 <a class="btn btn-primary" href="{{ route('juegos.index') }}">{{ __('Back') }}</a>
18 </div>
19 </div>
20 <div class="card-body">
21 <div class="form-group">
22 <strong>Nombre:</strong>
23 <input type="text" value="{{ $juego->nombre }}" />
24 </div>
25 <div class="form-group">
26 <strong>Consola:</strong>
27 <input type="text" value="{{ $juego->consola_id }}" />
28 </div>
29 <div class="form-group">
30 <strong>Fecha:</strong>
31 <input type="text" value="{{ $juego->fecha }}" />
32 </div>
33 </div>
34 </div>
35 </div>
36 </div>
37 </div>
```

11. Crear un seeder

Para crear nuestro primer usuario vamos a introducir sus datos en la tabla Users mediante un seeder:

Para ello debemos ejecutar `proyecto-laravel>php artisan make:seeder UserSeeder`

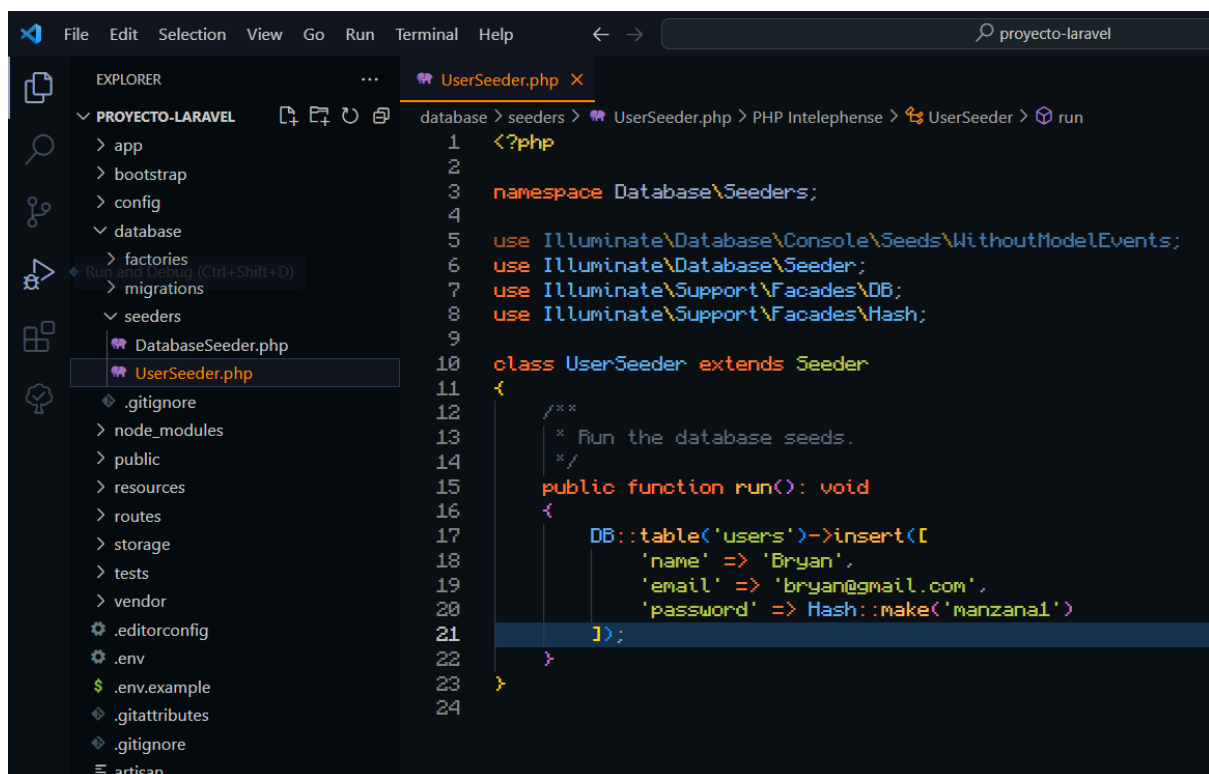
```
C:\xampp\htdocs\proyecto-laravel>php artisan make:seeder UserSeeder  
INFO Seeder [C:\xampp\htdocs\proyecto-laravel\database\seeders\UserSeeder.php] created successfully.
```

Una vez ejecutado vamos al archivo que acabamos de crear (`proyecto-laravel/database/seeders/UserSeeder.php`).

Tenemos que añadir en los imports:

```
use Illuminate\Support\Facades\DB;  
use Illuminate\Support\Facades\Hash;
```

y rellenar la función `run()`, quedaría así:



```
1 <?php  
2  
3 namespace Database\Seeders;  
4  
5 use Illuminate\Database\Console\Seeds\WithoutModelEvents;  
6 use Illuminate\Database\Seeder;  
7 use Illuminate\Support\Facades\DB;  
8 use Illuminate\Support\Facades\Hash;  
9  
10 class UserSeeder extends Seeder  
11 {  
12     /**  
13      * Run the database seeds.  
14      */  
15     public function run(): void  
16     {  
17         DB::table('users')->insert([  
18             'name' => 'Bryan',  
19             'email' => 'bryan@gmail.com',  
20             'password' => Hash::make('manzana1')  
21         ]);  
22     }  
23 }  
24
```

Una vez rellenado UserSeeder ejecutamos `proyecto-laravel>php artisan db:seed --class=UserSeeder` y ya debería estar el usuario insertado.

```
C:\xampp\htdocs\proyecto-laravel>php artisan db:seed --class=UserSeeder

INFO Seeding database.
```

12. Cambiar el idioma al español

Vamos a cambiar el idioma de la página a español, para hacerlo vamos a ejecutar los siguientes comandos:

- `proyecto-laravel>composer require laravel-lang/common --dev`

```
Símbolo del sistema
- Installing laravel-lang/native-currency-names (1.3.0): Extracting archive
- Installing laravel-lang/native-country-names (1.3.0): Extracting archive
- Installing archtechx/enums (v1.0.1): Extracting archive
- Installing laravel-lang/locale-list (1.3.0): Extracting archive
- Installing laravel-lang/locales (2.5.0): Extracting archive
- Installing dragon-code/pretty-array (v4.1.0): Extracting archive
- Installing composer/semver (3.4.0): Extracting archive
- Installing laravel-lang/publisher (16.2.1): Extracting archive
- Installing laravel-lang/lang (14.3.0): Extracting archive
- Installing laravel-lang/json-fallback (2.0.0): Extracting archive
- Installing laravel-lang/http-statuses (3.8.1): Extracting archive
- Installing laravel-lang/attributes (2.9.4): Extracting archive
- Installing laravel-lang/actions (1.5.1): Extracting archive
- Installing laravel-lang/common (6.1.0): Extracting archive
3 package suggestions were added by new dependencies, use `composer suggest` to see details.
Package laravelcollective/html is abandoned, you should avoid using it. Use spatie/laravel-html instead.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

INFO Discovering packages.

ibex/crud-generator ..... DONE
laravel-lang/actions ..... DONE
laravel-lang/attributes ..... DONE
laravel-lang/http-statuses ..... DONE
laravel-lang/lang ..... DONE
laravel-lang/locales ..... DONE
laravel-lang/publisher ..... DONE
laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
laravel/ui ..... DONE
laravelcollective/html ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

89 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

INFO No publishable resources for tag [laravel-assets].

No security vulnerability advisories found.
Using version ^6.1 for laravel-lang/common
```


- proyecto-laravel>php artisan lang:add es

```
C:\xampp\htdocs\proyecto-laravel>php artisan lang:add es

INFO  LaravelLang\Actions\Plugin.

Collect source ..... 4ms DONE
Collecting es ..... 2ms DONE

INFO  LaravelLang\Attributes\Plugin.

Collect source ..... 1ms DONE
Collecting es ..... 1ms DONE

INFO  LaravelLang\HttpStatuses\Plugin.

Collect source ..... 1ms DONE
Collecting es ..... 1ms DONE

INFO  LaravelLang\Lang\Plugin.

Collect source ..... 3ms DONE
Collecting es ..... 2ms DONE

INFO  Storing changes...

es.json ..... 6ms DONE
es/actions.php ..... 5ms DONE
es/validation.php ..... 8ms DONE
es/http-statuses.php ..... 3ms DONE
es/auth.php ..... 1ms DONE
es/pagination.php ..... 2ms DONE
es/passwords.php ..... 1ms DONE
```

- proyecto-laravel>php artisan lang:update

```
C:\xampp\htdocs\proyecto-laravel>php artisan lang:update

INFO  LaravelLang\Actions\Plugin.

Collect source ..... 3ms DONE
Collecting en ..... 1ms DONE
Collecting es ..... 1ms DONE

INFO  LaravelLang\Attributes\Plugin.

Collect source ..... 1ms DONE
Collecting en ..... 1ms DONE
Collecting es ..... 1ms DONE

INFO  LaravelLang\HttpStatuses\Plugin.

Collect source ..... 1ms DONE
Collecting en ..... 1ms DONE
Collecting es ..... 1ms DONE

INFO  LaravelLang\Lang\Plugin.

Collect source ..... 2ms DONE
Collecting en ..... 3ms DONE
Collecting es ..... 2ms DONE

INFO  Storing changes...

en.json ..... 5ms DONE
en/actions.php ..... 6ms DONE
en/validation.php ..... 7ms DONE
en/http-statuses.php ..... 3ms DONE
en/auth.php ..... 2ms DONE
en/pagination.php ..... 2ms DONE
en/passwords.php ..... 3ms DONE
es.json ..... 5ms DONE
es/actions.php ..... 4ms DONE
es/validation.php ..... 8ms DONE
es/http-statuses.php ..... 3ms DONE
es/auth.php ..... 1ms DONE
es/pagination.php ..... 2ms DONE
es/passwords.php ..... 2ms DONE
```

- Vamos a `proyecto-laravel/config/app.php` y en locale sustituimos 'en' por 'es'.

```
'locale' => 'es',
```

- En caso de querer corregir alguna traducción que no nos guste podemos ir a `proyecto-laravel/lang/es.json` y corregirla a mano.

13. Ejecución del proyecto

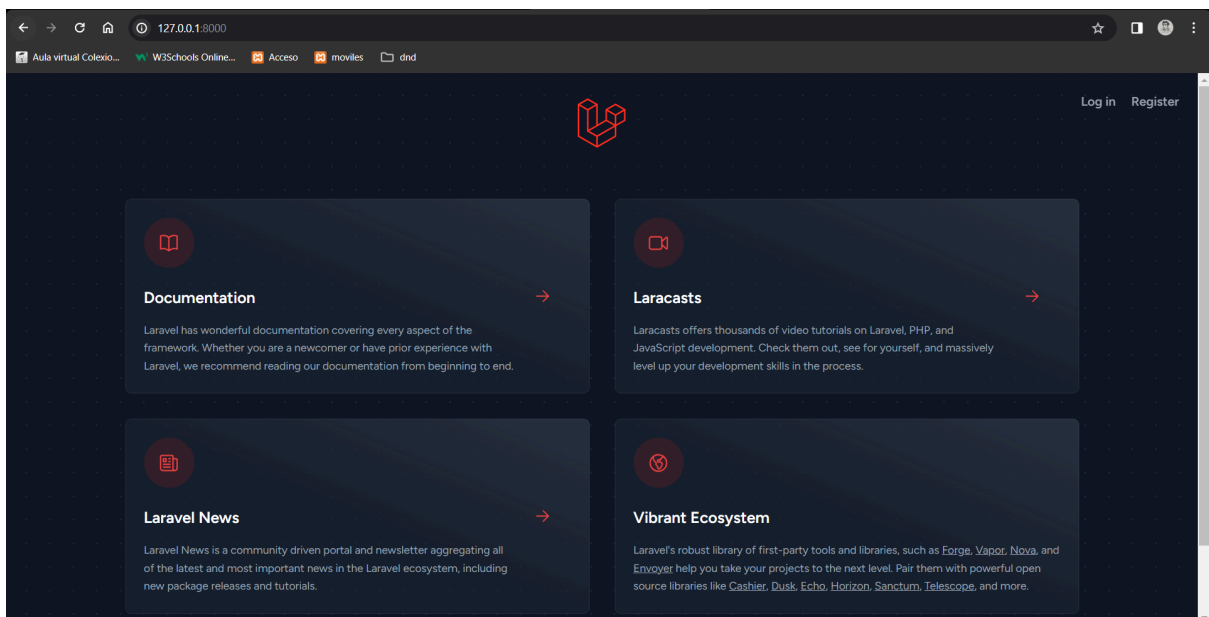
Por último para ejecutar el proyecto debemos ejecutar:

- `proyecto-laravel>npm run dev` en una consola
- `proyecto-laravel>php artisan serve` en otra. La dirección que aparece es la dirección que debes de poner en el navegador.

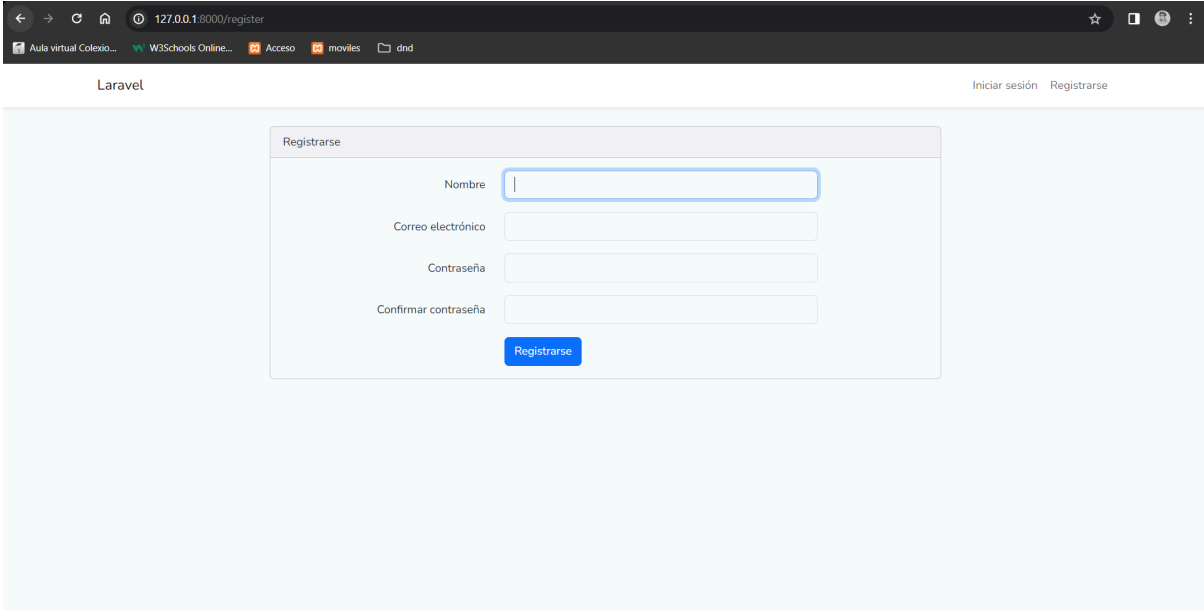
```
C:\xampp\htdocs\proyecto-laravel>php artisan serve  
  
INFO Server running on [http://127.0.0.1:8000].  
  
Press Ctrl+C to stop the server
```

Debería de quedar así:

- Inicio:

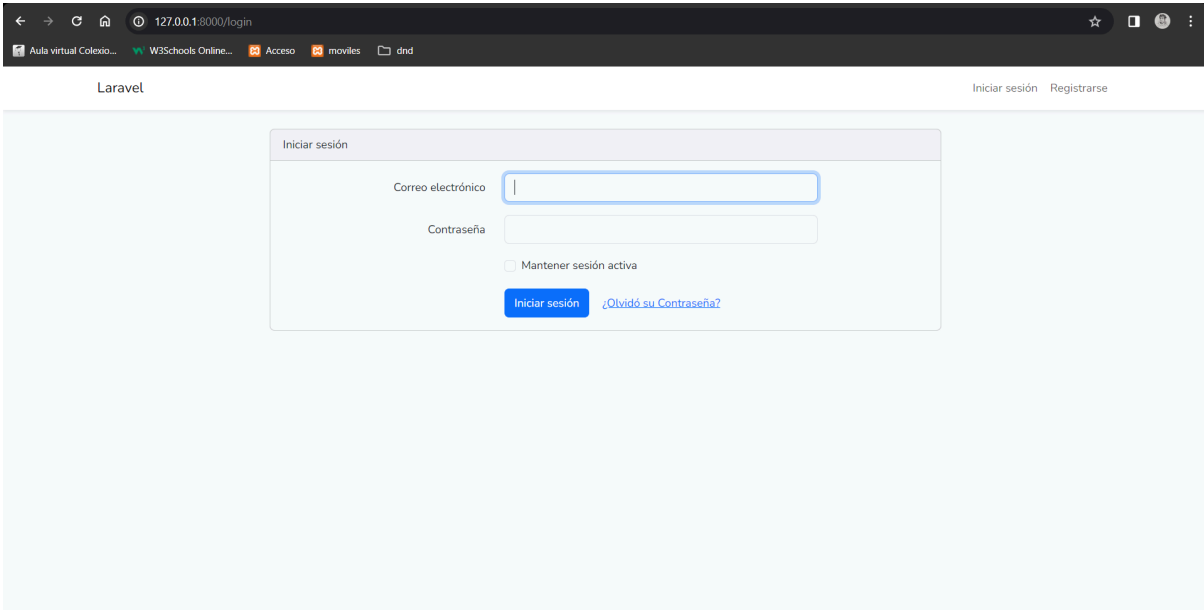


- Registrarse:



A screenshot of a web browser displaying the registration page of a Laravel application. The browser's address bar shows the URL "127.0.0.1:8000/register". The page has a light blue background. At the top, there is a navigation bar with the word "Laravel" on the left and links for "Iniciar sesión" and "Registrarse" on the right. The main content area features a registration form titled "Registrarse" in a light purple header. The form contains four input fields: "Nombre", "Correo electrónico", "Contraseña", and "Confirmar contraseña". Below these fields is a blue button labeled "Registrarse".

- Iniciar sesión:



A screenshot of a web browser displaying the login page of a Laravel application. The browser's address bar shows the URL "127.0.0.1:8000/login". The page has a light blue background. At the top, there is a navigation bar with the word "Laravel" on the left and links for "Iniciar sesión" and "Registrarse" on the right. The main content area features a login form titled "Iniciar sesión" in a light purple header. The form contains two input fields: "Correo electrónico" and "Contraseña". Below these fields is a checkbox labeled "Mantener sesión activa". At the bottom of the form is a blue button labeled "Iniciar sesión" and a link labeled "¿Olvidó su Contraseña?".

- Juegos:

Juego

Create New

Juego created successfully.

No	Nombre	Consola	Fecha	
1	Pokemon Escarlata	Switch	2022	Mostrar Editar Eliminar
2	Final Fantasy Crystal Chronicles: Ring of Fates	DS	2007	Mostrar Editar Eliminar

- Consolas:

Consola

Create New

No	Nombre	
1	Switch	Mostrar Editar Eliminar
2	PS5	Mostrar Editar Eliminar
3	3DS	Mostrar Editar Eliminar
4	DS	Mostrar Editar Eliminar