



MANUAL DEL PROGRAMADOR



Jose Manuel Delgado Chapela
2ºDAM Gestión Empresarial

- Migraciones: Creo las migraciones empezando por categorías y por ultimo películas que es la que contiene el campo relacional (comando php artisan make:migration)

```
return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('categorias',function(Blueprint $table){
            $table->engine="InnoDB";
            $table->id();
            $table->string('nombre');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('categorias');
    }
};
```

```
return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('peliculas',function(Blueprint $table){
            $table->engine="InnoDB";
            $table->id();
            $table->bigInteger('categoria_id')->unsigned();
            $table->string('nombre');
            $table->string('descripcion');
            $table->bigInteger('duracion');
            $table->bigInteger('year');
            $table->string('imagen');
            $table->timestamps();
            $table->foreign('categoria_id')->references('id')->on('categorias')->onDelete('cascade');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('libros');
    }
};
```

- Establecer rutas (web.php)

```
Comment Code
Route::get('/', function () {
    return view('welcome');
});

Auth::routes();

Route::resource('peliculas', App\Http\Controllers\PeliculaController::class)->middleware('auth');
Route::resource('categorias', App\Http\Controllers\CategoriaController::class)->middleware('auth');

Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])->name('home');
```

- Creación del menú de la navbar para el apartado de Películas y de Categorías

```
<!-- Left Side Of Navbar -->
@if (Auth::check())

<ul class="navbar-nav me-auto">
    <li class="nav-item">
        <a href="{{ route('peliculas.index') }}">{{ __('Peliculas') }}</a>
    </li>
    <li class="nav-item">
        <a href="{{ route('categorias.index') }}">{{ __('Categorias') }}</a>
    </li>
</ul>

@endif
<!-- Right Side Of Navbar -->
```

- Acceder a los datos de Categorías desde Películas

```
0 references | 0 overrides
public function create()
{
    $pelicula = new Pelicula();
    $categorias = Categoria::pluck('nombre','id');
    return view('pelicula.create', compact('pelicula','categorias'));
}
```

```
0 references | 0 overrides
public function edit($id)
{
    $pelicula = Pelicula::find($id);

    $categorias = Categoria::pluck('nombre','id');
    return view('pelicula.create', compact('pelicula','categorias'));
}
```

- Creación de un select para las Categorías en el formulario de Películas

```
<div class="form-group">
    {{ Form::label('categoria_id') }}
    {{ Form::select('categoria_id', $categorias,$pelicula->categoria_id, ['
    {!! $errors->first('categoria_id', '<div class="invalid-feedback">:mess
</div>
```

- Modificación de la página principal por defecto de Laravel

```
<div class=" mx-auto w-20 ml-6">
  <div class="flex justify-center pt-0 justify-start sm:pt-0">
    <h1 class="mt-2 dark:text-red text-xl fs-1" style="color: red">Laravel Films</h1>
  </div>

  <div class="mt-8 bg-white dark:bg-gray-800 overflow-hidden shadow sm:rounded-lg w-20">

    <div class="p-6 ml-4 max-w-20">
      <div class="flex items-center ml-6">
        <div class="ml-4 text-lg leading-7 font-semibold underline text-gray-900 dark:text-white">
          Bienvenido
        </div>
      </div>

      <div class="ml-4">
        <div class="mt-2 text-gray-600 dark:text-gray-400 text-sm">
          Esto es un CRUD de un catálogo de películas de ejemplo hecho con Laravel.
        </div>
      </div>
    </div>
  </div>
</div>
```

- Modificación de la pantalla home que se muestra cuando inicias sesión (recoger el nombre del usuario y escribe un saludo)

```
<div class="card-body">
  @if (session('status'))
    <div class="alert alert-success" role="alert">
      {{ session('status') }}
    </div>
  @endif
  <h1>
    {{ __('¡Bienvenido, ') }} {{ Auth::user()->name }} {{ __('!') }}
  </h1>
  <a class="btn btn-primary mt-5" href="{{ route('peliculas.index') }}">{{ __('Explorar el catálogo') }}</a>
</div>
```

- Modificación de la pantalla index de Películas donde se muestran las películas en un GridView con Cards y los botones de acción debajo.

```
<div class="row row-cols-1 row-cols-md-5 g-4 text-center">
  @foreach ($peliculas as $pelicula)
    <div class="col d-flex justify-content-center">
      <div class="card text-center h-100" style="width:200px">
        
        <div class="card-body">
          <h5 class="card-title">{{ $pelicula->nombre }}</h5>
          <h6 class="card-subtitle">{{ $pelicula->year }} {{ $pelicula->categoria->nombre }}
            <br>Duración: {{ $pelicula->duracion }}</h6>
          <p class="card-text">
            {{ Str::substr($pelicula->descripcion, 0, 40) }}...
          </p>
          <form action="{{ route('peliculas.destroy', $pelicula->id) }}" method="POST">
            <a class="btn btn-sm btn-primary"
              href="{{ route('peliculas.show', $pelicula->id) }}"><i
                class="fa fa-fw fa-eye"></i> {{ __('Ver') }}</a>
            <a class="btn btn-sm btn-success"
              href="{{ route('peliculas.edit', $pelicula->id) }}"><i
                class="fa fa-fw fa-edit"></i> {{ __('Editar') }}</a>
            @csrf
            @method('DELETE')
            <button type="submit" class="btn btn-danger btn-sm"><i
              class="fa fa-fw fa-trash"></i> {{ __('Borrar') }}</button>
          </form>
        </div>
      </div>
    </div>
  </div>
@endforeach
```

- Modificación de la pantalla Show de Películas para que cuando se visualicen los detalles de una película aparezcan con un diseño más adecuado para este caso (**CSS extra en app.blade**)

```
<div class="movie-ticket">
  <div>
    <!-- Item -->
    <div class="ui-item">
      <!-- Head -->
      <div class="ui-head">
        <!-- Background Image -->
        
        <!-- Transparent Background -->
        <div class="ui-trans clearfix">
          <!-- Image -->
          
          <!-- Details -->
          <div class="ui-details clearfix">
            <!-- Movie Name -->
            <h2><a href="#">{{ $pelicula->nombre }}</a></h2>
            <!-- Labels -->
            <a href="#" class="label">{{ $pelicula->year }}</a>

            <a href="#" class="label"> {{ $pelicula->duracion }} min</a>
            <a href="#" class="label"> {{ $pelicula->categoria->nombre }}</a>

            <!-- Paragraph -->
            <p class="pcard">{{ $pelicula->descripcion }}</p>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

- Creación de un seeder para insertar una cuenta de admin en la base de datos (incluye la inserción de contraseña como Hash)

```
class AdminSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        //
        DB::table("users") -> insert([
            "name" => "admin",
            "email" => "admin@admin.com",
            "password" => Hash::make("admin"),
        ]);
    }
}
```

- Establecer unas reglas para la validación de formulario en la clase Película para los campos duración y year para asegurar que sean números enteros

```
class Pelicula extends Model
{
    2 references
    static $rules = [
        'categoria_id' => 'required',
        'nombre' => 'required',
        'descripcion' => 'required',
        'duracion' => 'required|numeric|integer',
        'year' => 'required|numeric|integer',
        'imagen' => 'required',
    ];
}
```

- Cambiar idioma en app.php

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