Actividad p2Web1 en Laravel

Descripción del Proyecto

En esta actividad trabajé con Laravel para conectar mi proyecto a una base de datos usando SQLite. El objetivo fue dejar de usar datos escritos directamente en el código y empezar a guardarlos de forma permanente en una base de datos real. Para lograr esto, creé una tabla llamada *productos*, donde puedo guardar información como el nombre de un producto, su descripción y su precio.

También creé un modelo llamado **Producto**, que es básicamente la forma en la que Laravel se comunica con la tabla en la base de datos. Gracias a esto, el sistema puede guardar datos, buscarlos y mostrarlos sin tener que escribir muchas consultas manualmente.

Después de eso, generé un seeder para insertar datos de ejemplo en la tabla. Esto me ayudó a llenar la base de datos rápidamente con 3 productos de prueba para comprobar que todo funcionara bien.

Finalmente, creé una ruta llamada **/productos**, que se conecta con un controlador. El controlador se encarga de pedirle al modelo todos los productos guardados en la base de datos y luego enviarlos a una vista. En la vista utilicé un ciclo @foreach para mostrar cada producto en pantalla.

Con todo esto, mi proyecto ya puede leer información real desde una base de datos y mostrarla al usuario de forma dinámica.

2. REFLEXIÓN SOBRE MIGRACIONES Y ELOQUENT

Reflexión

Durante esta actividad aprendí cómo funciona el proceso de crear y manejar una base de datos dentro de un proyecto de Laravel. Uno de los desafíos principales fue entender cómo se relacionan las migraciones, los seeders, el modelo y las vistas.

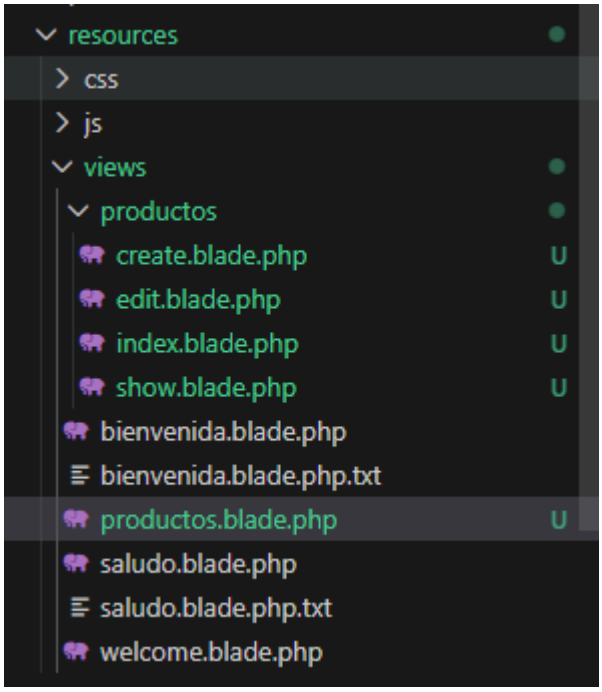
Las migraciones me parecieron útiles porque permiten definir la estructura de la base de datos sin tener que entrar a un programa externo. Fue un poco confuso al principio entender que si me equivoco en la estructura tengo que borrar la base de datos o rehacer la migración, pero ya con la práctica es más sencillo.

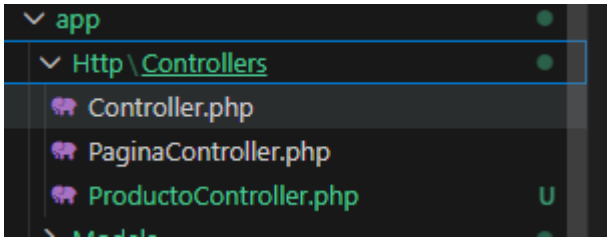
Con Eloquent entendí que Laravel hace que trabajar con bases de datos sea mucho más fácil, porque ya no tengo que escribir las consultas manualmente. Solo con escribir Producto::all() ya puedo obtener todos los registros. Esto hace que el código sea más limpio y también más fácil de entender.

Otro desafío fue organizar bien los archivos: el modelo, el controlador, las rutas y la vista. Pero cuando entendí que cada uno tiene su función y que todo se conecta de forma ordenada, el proyecto empezó a tener sentido.

En general, esta actividad me ayudó mucho a entender cómo funciona el proceso completo: crear la base de datos, llenarla con datos, pedir esos datos desde el controlador y mostrarlos en pantalla. Ahora siento que puedo hacer proyectos que ya no dependan de datos inventados, sino que usen información real.

Captura de prueba:





ProductoController.php U

app > Http > Controllers > ProductoController.php

```php
<?php

namespace App\Http\Controllers;

use App\Models\Producto;

class ProductoController extends Controller
{

    public function index()
    {
        // Obtener todos los productos desde la base de datos
        $productos = Producto::all();

        // Enviar los productos a la vista productos.blade.php
        return view('productos', ['productos' => $productos]);
    }
}

```

ProductoController.php U ● index.blade.php U ×

resources > views > productos > index.blade.php

```blade
<!DOCTYPE html>
<html>
<head>
    <title>Lista de Productos</title>
</head>
<body>
    <h1>Productos</h1>

    <a href="{{ route('productos.create') }}">Crear Producto</a>

    <table border="1" cellpadding="10" style="margin-top: 20px;">
        <thead>
            <tr>
                <th>ID</th>
                <th>Nombre</th>
                <th>Precio</th>
                <th>Stock</th>
                <th>Acciones</th>
            </tr>
        </thead>
        <tbody>
            @foreach($productos as $producto)
                <tr>
                    <td>{{ $producto->id }}</td>
                    <td>{{ $producto->nombre }}</td>
                    <td>{{ $producto->precio }}</td>
                    <td>{{ $producto->stock }}</td>
                    <td>
                        <a href="{{ route('productos.edit', $producto->id) }}">Editar</a>
                        <form action="{{ route('productos.destroy', $producto->id) }}" method="POST" style=
                            @csrf
                            @method('DELETE')
```

```blade
h($productos as $producto)
>
    <td>{{ $producto->id }}</td>
    <td>{{ $producto->nombre }}</td>
    <td>{{ $producto->precio }}</td>
    <td>{{ $producto->stock }}</td>
    <td>
        <a href="{{ route('productos.edit', $producto->id) }}">Editar</a>
        <form action="{{ route('productos.destroy', $producto->id) }}" method="POST" style="display:inline;">
            @csrf
            @method('DELETE')
            <button type="submit">Eliminar</button>
        </form>
    </td>
r>
each
```