Semana 2: Domina las Rutas y Controladores en Laravel 12 desde Cero

Nombre del Proyecto

"Navegando TechCorp: Gestión de Empleados con Rutas y Controladores"

© Objetivos de la Semana

- 1. Aprender a crear rutas simples y dinámicas en Laravel 12.
- 2. Comprender cómo estructurar controladores y conectar rutas con vistas.
- 3. Integrar la lógica de rutas y controladores con una base de datos utilizando Eloquent.

Requisitos Previos

- Laravel 12 instalado en XAMPP (htdocs/TechCorpPortal)
- Visual Studio Code configurado
- Base de datos creada en phpMyAdmin
- Extensión ZIP habilitada en php.ini
- Git, Composer y PHP en las variables de entorno

Aplicaciones Necesarias

HERRAMIENTA	FUNCIÓN
XAMPP	Servidor web (Apache, MySQL)
Composer	Gestor de dependencias de Laravel
Git	Control de versiones
VS Code	Editor de código fuente
phpMyAdmin	Administración visual de MySQL





Caso de Estudio: TechCorp Portal- Módulo de Empleados

TechCorp, una empresa tecnológica, necesita desarrollar un módulo para listar empleados y mostrar sus detalles al hacer clic sobre su nombre. El equipo de desarrollo debe usar rutas dinámicas, controladores y bases de datos para resolver esta necesidad.

- 🕝 Fase de Preparación: Crear Base de Datos y Conectarla a Laravel
- Paso 1: Crear la Base de Datos en phpMyAdmin
 - 1. Abre http://localhost/phpmyadmin desde tu navegador.
 - 2. Haz clic en "Nueva" y nombra tu base de datos: techcorp db.
 - 3. Deja el cotejamiento por defecto (utf8mb4_unicode_ci) y presiona "Crear".
- Paso 2: Conectar Laravel con la Base de Datos
 - Abre tu proyecto en Visual Studio Code: C:\xampp\htdocs\TechCorpPortal.
 - 2. Edita el archivo .env:

DB CONNECTION=mysql DB HOST=127.0.0.1 DB PORT=3306 DB DATABASE=techcorp db DB_USERNAME=root DB PASSWORD=

- 1. Guarda los cambios y asegúrate de reiniciar el servidor de Laravel si estaba corriendo.
- Paso 3: Crear la Migración de Empleados

php artisan make:migration create empleados table





En el archivo generado (database/migrations/xxxx_create_empleados_table.php), define las columnas:

```
Schema::create('empleados', function (Blueprint $table) {
    $table->id();
    $table->string('nombre');
    $table->string('email')->unique();
    $table->string('cargo');
    $table->timestamps();
});
```

Ejecuta la migración:

php artisan migrate

Paso 4: Crear el Modelo Empleado y Controlador

```
php artisan make:model Empleado
php artisan make:controller EmpleadoController
```

- 🗱 Fase de Desarrollo Paso a Paso
- **☑** Paso 5: Rutas Web para Empleados

Edita routes/web.php:

```
use App\Http\Controllers\EmpleadoController;

Route::get('/empleados', [EmpleadoController::class, 'index']);

Route::get('/empleados/{id}', [EmpleadoController::class, 'show']);
```

✓ Paso 6: Programar el Controlador

```
app/Http/Controllers/EmpleadoController.php:
```

```
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Empleado;

class EmpleadoController extends Controller
{
```



```
public function index()
{
    $\perp{\text{sempleados} = Empleado::all();}
    return view('empleados.index', compact('empleados'));}
}

public function show($id)
{
    $\text{sempleado} = Empleado::findOrFail($id);}
    return view('empleados.show', compact('empleado'));}
}
```

Paso 7: Crear las Vistas Blade

Crear la carpeta: resources/views/empleados/

• index.blade.php:

show.blade.php:

```
<h1>Detalle del Empleado</h1>
<strong>Nombre:</strong> {{ $empleado->nombre }}
<strong>Email:</strong> {{ $empleado->email }}
<strong>Cargo:</strong> {{ $empleado->cargo }}
<a href="/empleados"> \leftarrow Volver al listado</a>
```



S Conclusión de la Semana

Esta semana te permitió:

- Entender cómo funcionan las rutas en Laravel.
- ✓ Crear controladores y conectar datos desde una base de datos real.
- Construir un módulo funcional que puedes expandir.

