

Desarrollo de Aplicaciones Mobiles 1

Institucion : Universidad Autónoma de Tamaulipas
Nombre del Alumno : Mendez Padron Gustavo Emanuel
Nombre del Profesor: Gonzale Bravo Eder Jahir
Materia : Desarrollo de Aplicaciones Web 1
Fecha : 17/11/2025 Grupo: I
Carrera: Ingenieria en Sistemas Computacionales

I. DESCRIPCIÓN DEL PROYECTO

Este proyecto tiene como objetivo demostrar la implementación de rutas estáticas y dinámicas en Laravel, utilizando controladores y vistas Blade para crear una aplicación web funcional.

I.1 Objetivo General

Aplicar los conceptos fundamentales de Laravel relacionados con:

- Definición de rutas estáticas y dinámicas
- Creación y uso de controladores
- Implementación de vistas con Blade Templates
- Paso de parámetros dinámicos a través de URLs

I.2 Tecnologías Utilizadas

- **Framework:** Laravel 11.x
- **Lenguaje:** PHP 8.2+
- **Motor de Plantillas:** Blade
- **Frontend:** HTML5, CSS3, JavaScript (Vanilla)
- **Control de Versiones:** Git/GitHub

2. FUNCIONALIDADES IMPLEMENTADAS

2.1 Ruta Estática: `/bienvenida`

Página de bienvenida que presenta información general del proyecto. Esta ruta no requiere parámetros y siempre muestra el mismo contenido.

Características:

- URL fija: `http://localhost:8000/bienvenida`
- Contenido estático
- Diseño responsive
- Navegación hacia otras secciones

2.2 Ruta Dinámica: `/saludo/{nombre}`

Página que captura un parámetro dinámico (nombre) desde la URL y personaliza el contenido en función de este valor.

Características:

- URL dinámica: `http://localhost:8000/saludo/[nombre]`
- Captura de parámetro desde la URL
- Personalización del contenido
- Formato del nombre (primera letra mayúscula)
- Estadísticas en tiempo real (caracteres, hora, fecha)

Ejemplos de uso:

- `/saludo/Carlos` → Muestra "¡Hola, Carlos!"
- `/saludo/María` → Muestra "¡Hola, María!"
- `/saludo/San Jinwoong` → Muestra "¡Hola, San jinwoong!"

3. ESTRUCTURA DEL CÓDIGO

3.1 Controlador: `PaginaController.php`

```
``php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PaginaController extends Controller
{
    public function bienvenida()
    {
        return view('bienvenida');
    }

    public function saludo($nombre)
    {
        $nombreFormateado = ucfirst(strtolower($nombre));
        return view('saludo', ['nombre' => $nombreFormateado]);
    }
}
``
```

Explicación:

- `bienvenida()`: Retorna la vista estática
- `saludo(\$nombre)`: Captura el parámetro, lo formatea y lo pasa a la vista

3.2 Rutas: `web.php`

```
``php  
<?php
```

```
use Illuminate\Support\Facades\Route;  
use App\Http\Controllers\PaginaController;
```

```
Route::get('/', function () {  
    return redirect('/bienvenida');  
});
```

```
Route::get('/bienvenida', [PaginaController::class, 'bienvenida'])  
    ->name('bienvenida');
```

```
Route::get('/saludo/{nombre}', [PaginaController::class, 'saludo'])  
    ->name('saludo');  
``
```

Explicación:

- Primera ruta: Redirección del inicio hacia `/bienvenida`
- Segunda ruta: Ruta estática nombrada
- Tercera ruta: Ruta dinámica con parámetro `{nombre}`

3.3 Vistas Blade

Layout principal: `resources/views/layouts/app.blade.php`

- Estructura base con navbar y footer
- Estilos globales
- Secciones dinámicas con `@yield`

Vista estática: `resources/views/bienvenida.blade.php`

- Extiende el layout con `@extends`
- Define contenido con `@section`
- Información del proyecto

Vista dinámica: `resources/views/saludo.blade.php`

- Recibe variable `'\$nombre` desde el controlador
- Muestra saludo personalizado
- Estadísticas dinámicas

4. CAPTURAS DE PANTALLA

4.1 Página de Bienvenida



Descripción: Vista de la página estática `/bienvenida` mostrando el hero section, información del proyecto y botones de navegación.

4.2 Ruta Dinámica - Ejemplo 1



Descripción: Saludo personalizado para "San Jinwoong" mostrando avatar, estadísticas y mensaje personalizado.

5. PREGUNTAS DE REFLEXIÓN

5.1 ¿Cuál es la diferencia entre una ruta estática y una ruta dinámica en Laravel?

Respuesta:

Una **ruta estática** es aquella que no cambia y siempre apunta al mismo contenido. En Laravel se define con una URL fija, por ejemplo `'/bienvenida'`, y siempre ejecuta la misma lógica retornando el mismo resultado.

Por otro lado, una **ruta dinámica** contiene parámetros variables que se capturan desde la URL. En Laravel se definen usando llaves `{parametro}`, por ejemplo `'/saludo/{nombre}'`. El contenido mostrado depende del valor del parámetro recibido, permitiendo personalizar la respuesta según la entrada del usuario.

Ejemplo práctico del proyecto:

- **Estática:** `'/bienvenida'` siempre muestra la misma página de bienvenida
- **Dinámica:** `'/saludo/Carlos'` y `'/saludo/María'` usan la misma ruta pero muestran contenido diferente

5.2 ¿Qué función cumple el controlador en este proyecto?

Respuesta:

El controlador ('PaginaController') actúa como intermediario entre las rutas y las vistas, implementando el patrón MVC (Modelo-Vista-Controlador). Sus funciones principales son:

- 1. Separar la lógica de negocio:** El controlador contiene la lógica para procesar datos antes de enviarlos a la vista
- 2. Organizar el código:** Agrupa métodos relacionados en una sola clase
- 3. Facilitar el mantenimiento:** Centraliza la lógica, haciendo más fácil realizar cambios

En este proyecto específicamente:

- El método `'bienvenida()'` simplemente retorna una vista
- El método `'saludo(\$nombre)'` procesa el parámetro recibido (lo formatea con `ucfirst` y `strtolower`) antes de pasarla a la vista

Esto demuestra cómo el controlador puede realizar transformaciones de datos antes de presentarlos.

6. CONCLUSIONES

El proyecto demuestra exitosamente la implementación de rutas estáticas y dinámicas en Laravel, cumpliendo con todos los objetivos planteados:

- Se crearon rutas estáticas y dinámicas funcionales
- Se implementó un controlador para manejar la lógica
- Se utilizaron vistas Blade con herencia de plantillas
- Se logró pasar parámetros dinámicos desde la URL
- Se aplicaron buenas prácticas de desarrollo

Reflexión final:

Laravel demuestra ser un framework potente y elegante que simplifica el desarrollo web. Su sistema de rutas, controladores y vistas Blade permite crear aplicaciones escalables y mantenibles siguiendo el patrón MVC.

La experiencia de trabajar con rutas dinámicas abre la puerta a crear aplicaciones más interactivas y personalizadas, donde el contenido puede adaptarse según las necesidades del usuario.

7. REFERENCIAS

- Documentación oficial de Laravel: <https://laravel.com/docs>
- Laravel Routing: <https://laravel.com/docs/routing>
- Laravel Controllers: <https://laravel.com/docs/controllers>
- Blade Templates: <https://laravel.com/docs/blade>
- Repositorio del proyecto: <https://github.com/SanJinwoong/Practica-I>