



Actividad 04 FRAMEWORK

INTEGRANTES

Nombre: Callecusi Guarachi Jose Manuel

DOCENTE

ING. Gustavo Tantani Mamani

MATERIA: Programación Web I

CARRERA: Ingeniería de Sistemas



Profesionales + humanos

BOLIVIA-SCZ



INDICE

1	INTRODUCCIÓN	3
2	INVESTIGACIÓN DE FRAMEWORKS	3
2.1	Framework Laravel (El estándar de PHP)	3
2.1.1	Características Principales:.....	3
2.2	Framework Angular (La plataforma de Google).....	4
2.2.1	Características Principales:.....	4
3	DISEÑO Y FORMAS DE FUNCIONAMIENTO	5
3.1	Diseño Arquitectónico de Laravel: Patrón MVC.....	5
3.1.1	El Modelo (Model):	6
3.2	Diseño Arquitectónico de Angular: Basado en Componentes.....	7

1 INTRODUCCIÓN

En la actualidad, el desarrollo de software web ha evolucionado significativamente, pasando de páginas estáticas simples a aplicaciones complejas y dinámicas. Para gestionar esta complejidad, el uso de Frameworks se ha convertido en un estándar de la industria, permitiendo a los desarrolladores trabajar de manera más eficiente, segura y ordenada.

El presente trabajo de investigación se centra en el análisis de dos de las herramientas más potentes del mercado actual: Laravel, como solución robusta para el desarrollo del lado del servidor (Backend), y Angular, como plataforma líder para la construcción de interfaces interactivas en el lado del cliente (Frontend). A lo largo de este informe, se detallarán sus arquitecturas, formas de funcionamiento y las diferencias clave que definen su aplicación en el desarrollo web moderno.

2 INVESTIGACIÓN DE FRAMEWORKS

En el ecosistema del desarrollo web actual, la separación de responsabilidades es fundamental. Por ello, se investigan dos tecnologías líderes que dominan sus respectivos entornos: Laravel en el servidor (Backend) y Angular en el navegador (Frontend).

2.1 Framework Laravel (El estándar de PHP)

Definición: Laravel es un framework de código abierto para desarrollar aplicaciones web con PHP. Fue creado por Taylor Otwell con la filosofía de hacer el desarrollo "agradable y creativo". Sigue el patrón de arquitectura Modelo-Vista-Controlador (MVC). Es ampliamente reconocido por tener una sintaxis elegante, expresiva y fácil de leer, lo que reduce la complejidad de tareas comunes en el desarrollo web.

2.1.1 Características Principales:

- ✓ **Eloquent ORM (Object-Relational Mapping):** Es quizás la herramienta más potente de Laravel. Permite interactuar con la base de datos utilizando objetos de PHP en lugar de escribir largas consultas SQL complejas. Por ejemplo, para obtener todos los usuarios, basta con escribir User::all().

- ✓ **Motor de Plantillas Blade:** Un sistema potente pero ligero que permite mezclar código HTML con lógica de PHP de manera limpia, facilitando la creación de las vistas que se mostrarán al usuario.
- ✓ **Artisan Console:** Una interfaz de línea de comandos incluida en Laravel que automatiza tareas repetitivas, como crear controladores, modelos o gestionar migraciones de base de datos, ahorrando horas de programación manual.
- ✓ **Seguridad Integrada:** Laravel maneja automáticamente la seguridad contra inyecciones SQL, ataques XSS (Cross-Site Scripting) y CSRF, protegiendo la aplicación desde su instalación.
- ✓ **Aplicación en el Desarrollo Web (Backend):** En un entorno profesional, Laravel se utiliza para construir la API (Interfaz de Programación de Aplicaciones). Su función no es preocuparse por los colores o animaciones de la página, sino por la lógica y los datos: recibir peticiones, validar que el usuario tenga permisos, consultar la base de datos y devolver la información (generalmente en formato JSON) para que el Frontend la consuma.

2.2 Framework Angular (La plataforma de Google)

Definición: Angular es una plataforma y framework de desarrollo para construir aplicaciones web de una sola página (SPA) utilizando HTML y TypeScript. Es desarrollado y mantenido por Google. A diferencia de una librería simple, Angular es una solución completa ("todo incluido") que proporciona todas las herramientas necesarias para el desarrollo (enrutamiento, cliente HTTP, gestión de formularios) sin depender de terceros.

2.2.1 Características Principales:

Uso de TypeScript: Angular está construido sobre TypeScript, un superconjunto de JavaScript que añade tipos estáticos. Esto permite detectar errores mientras se escribe el código (antes de ejecutarlo) y mejora el autocompletado y la inteligencia en los editores de código.

- ✓ **Arquitectura Basada en Componentes:** La aplicación se divide en piezas pequeñas, independientes y reutilizables llamadas "componentes" (por ejemplo: HeaderComponent, ProductListComponent). Cada componente encapsula su propia lógica, vista y estilo.
- ✓ **Two-Way Data Binding (Enlace de datos bidireccional):** Permite una sincronización automática e inmediata entre el Modelo (la lógica en TypeScript) y la Vista (el HTML). Si el usuario cambia un dato en un formulario, la variable en el código se actualiza instantáneamente, y viceversa.
- ✓ **Inyección de Dependencias:** Un patrón de diseño integrado en el núcleo de Angular que hace que el código sea modular, fácil de probar y eficiente al gestionar cómo se crean y comparten los servicios dentro de la aplicación.
- ✓ **Aplicación en el Desarrollo Web (Frontend):** Angular se aplica para crear SPAs (Single Page Applications). A diferencia de las webs tradicionales que recargan toda la página cada vez que haces clic en un enlace, Angular carga la página una sola vez y luego solo actualiza las partes necesarias de la pantalla dinámicamente. Esto ofrece una experiencia de usuario extremadamente rápida y fluida, similar a una aplicación de escritorio o móvil.

3 DISEÑO Y FORMAS DE FUNCIONAMIENTO

Mientras Laravel estructura el código en el servidor, Angular estructura la interfaz en el cliente.

3.1 Diseño Arquitectónico de Laravel: Patrón MVC

Laravel fundamenta su diseño en el patrón de arquitectura de software MVC (Modelo-Vista-Controlador). Este diseño tiene como objetivo principal la "Separación de Responsabilidades", asegurando que la lógica de negocios esté desacoplada de la interfaz de usuario.

Los tres componentes fundamentales son:

El Modelo (Model):

- Es la capa encargada de gestionar los datos y la lógica de negocio.
- Interactúa directamente con la Base de Datos (MySQL, PostgreSQL, etc.).
- En Laravel, cada tabla de la base de datos tiene su correspondiente "Modelo" (ej. tabla users -> modelo User).

La Vista (View):

- Es la capa de presentación, lo que el usuario final ve en su pantalla.
- En Laravel, las vistas se construyen utilizando Blade, un motor de plantillas que genera HTML dinámico basado en los datos recibidos.

El Controlador (Controller):

- Actúa como intermediario o "director de orquesta".
- Recibe las peticiones del usuario (desde el navegador), decide qué hacer con ellas, solicita datos al Modelo y finalmente selecciona qué Vista mostrar.

Forma de Funcionamiento (Ciclo de Vida de una Petición en Laravel): El funcionamiento de Laravel es lineal y se ejecuta en el servidor cada vez que llega una solicitud:

- **Entrada (Request):** El usuario ingresa a una URL.
- **Enrutamiento (Routes):** El archivo de rutas (web.php o api.php) intercepta la URL y la dirige al Controlador correspondiente.
- **Procesamiento:** El Controlador ejecuta la lógica necesaria (ej. "Buscar al alumno con ID 5").
- **Consulta de Datos:** El Controlador llama al Modelo para obtener la información de la Base de Datos.
- **Respuesta (Response):** El Controlador inyecta los datos en una Vista y devuelve el HTML final al navegador del usuario.

3.2 Diseño Arquitectónico de Angular: Basado en Componentes

A diferencia del MVC tradicional del backend, Angular utiliza una arquitectura Basada en Componentes. Una aplicación Angular es, en esencia, un árbol de componentes anidados.

Los elementos clave de su diseño son:

Módulos (NgModules): Son contenedores que agrupan componentes, directivas y servicios relacionados. Cada aplicación tiene al menos un "Módulo Raíz" (AppModule) que arranca la aplicación.

Componentes: Son la unidad básica de construcción. Cada componente controla una parte específica de la pantalla (ej. BarraDeNavegacion, ListaDeEstudiantes). Un componente consta de tres archivos:

- **Clase (.ts):** Lógica y comportamiento en TypeScript.
- **Plantilla (.html):** Estructura visual.
- **Estilo (.css):** Diseño y apariencia.

Servicios e Inyección de Dependencias: Angular separa la lógica que no es visual (como pedir datos a un servidor o validar formularios) en "Servicios". Estos se inyectan en los componentes que los necesitan, promoviendo la reutilización de código.

Forma de Funcionamiento (SPA - Single Page Application): El funcionamiento de Angular ocurre enteramente en el navegador del cliente:

- ✓ **Carga Inicial:** Al entrar a la web, el servidor envía un único archivo index.html muy ligero y los scripts de JavaScript de Angular.
- ✓ **Renderizado Dinámico:** Angular toma el control del DOM (la estructura de la página) y renderiza los componentes necesarios según la URL actual.
- ✓ **Navegación sin Recarga:** Cuando el usuario cambia de sección, Angular no pide una nueva página al servidor; simplemente destruye el componente actual y dibuja el nuevo instantáneamente. Esto reduce el consumo de ancho de banda y mejora la velocidad.

