**Documentación del Proyecto**

**Grupo 8 - Estudiantes:**

* Meneses Solorzano, Brandon Luis - U21300784
* Velez de Villa Arista Jorge Brian - U20207294
* Mendoza Tinco, Tatiana - U0831381
* Barrantes Burga, César Jeanpieer - U21218945
* Lopez Ordinola, Miguel - U19312642

**Introducción**

Este proyecto es una aplicación que permite realizar operaciones matemáticas básicas y trabajar con derivadas de polinomios de grado 3. Está compuesto por un **frontend** desarrollado en Angular y un **backend** en Spring Boot.

**Funcionalidades Principales**

1. **Operaciones Matemáticas Básicas**:
   * Cálculo del **máximo**, **mínimo**, **suma**, **promedio**, y **ordenar** números.
2. **Derivadas de Polinomios**:
   * Calcular la **derivada** de un polinomio de grado 3.
   * Evaluar la derivada en un valor específico de x.

**Arquitectura del Proyecto**

El proyecto sigue una arquitectura de **cliente-servidor**:

* **Frontend**: Angular para la interfaz de usuario.
* **Backend**: Spring Boot para la lógica de negocio y cálculo de operaciones.
* **Comunicación**: Las operaciones se realizan mediante peticiones HTTP (RESTful APIs).

**Estructura del Proyecto**

**Frontend (Angular)**

* **src/app/polynomial-calculator.component.ts**: Componente principal que maneja las operaciones matemáticas y la lógica de derivadas.
* **src/app/polynomial-calculator.component.html**: Plantilla que renderiza los formularios de entrada y muestra los resultados de las operaciones.
* **src/app/services**: Servicios encargados de realizar las peticiones HTTP al backend.

**Backend (Spring Boot)**

* **src/main/java/com/example/controller/MathController.java**: Controlador que expone las APIs para realizar las operaciones matemáticas y el cálculo de derivadas.
* **src/main/resources/application.properties**: Archivo de configuración para el entorno de Spring Boot.
* **src/main/java/com/example/service/MathService.java**: Servicio que contiene la lógica de negocio para realizar los cálculos.

**Endpoints de la API**

**Operaciones Básicas**

1. **Máximo**:
   * **Método**: GET
   * **Endpoint**: /max-number
   * **Parámetros**: num1, num2, num3, num4
   * **Descripción**: Devuelve el número máximo entre los cuatro números ingresados.

Ejemplo:

GET http://localhost:8080/max-number?num1=4&num2=3&num3=2&num4=1

1. **Mínimo**:
   * **Método**: GET
   * **Endpoint**: /min-number
   * **Parámetros**: num1, num2, num3, num4
   * **Descripción**: Devuelve el número mínimo entre los cuatro números ingresados.
2. **Suma**:
   * **Método**: GET
   * **Endpoint**: /sum
   * **Parámetros**: num1, num2, num3, num4
   * **Descripción**: Devuelve la suma de los cuatro números.
3. **Promedio**:
   * **Método**: GET
   * **Endpoint**: /average
   * **Parámetros**: num1, num2, num3, num4
   * **Descripción**: Devuelve el promedio de los cuatro números.
4. **Ordenar**:
   * **Método**: GET
   * **Endpoint**: /sort-numbers
   * **Parámetros**: num1, num2, num3, num4, order (asc o desc)
   * **Descripción**: Devuelve los números ordenados ascendente o descendentemente.

**Operaciones con Derivadas**

1. **Calcular Derivada**:
   * **Método**: GET
   * **Endpoint**: /dynamic-derivative
   * **Parámetros**: num1, num2, num3, num4
   * **Descripción**: Devuelve la derivada del polinomio representado por los números ingresados.

Ejemplo:

GET http://localhost:8080/dynamic-derivative?num1=4&num2=3&num3=2&num4=1

1. **Evaluar Derivada**:
   * **Método**: GET
   * **Endpoint**: /evaluate-dynamic-derivative
   * **Parámetros**: num1, num2, num3, num4, x
   * **Descripción**: Evalúa la derivada del polinomio en el valor x ingresado.

Ejemplo:

GET

http://localhost:8080/evaluate-dynamic-derivative?num1=4&num2=3&num3=2&num4=1&x=2

**Descripción de Componentes Importantes**

**Frontend (Angular)**

1. **PolynomialCalculatorComponent.ts**:
   * Este componente gestiona el flujo de entrada de datos para las operaciones matemáticas.
   * Contiene métodos para enviar peticiones al backend y mostrar los resultados de las operaciones y derivadas.
2. **PolynomialCalculatorComponent.html**:
   * La interfaz gráfica que permite a los usuarios ingresar los números y realizar cálculos.
   * Renderiza los resultados de las operaciones y derivadas en la parte inferior.

**Backend (Spring Boot)**

1. **MathController.java**:
   * El controlador que expone los endpoints para las operaciones matemáticas y derivadas.
   * Gestiona las peticiones y envía la respuesta al frontend.
2. **MathService.java**:
   * Implementa la lógica de negocio para realizar los cálculos (máximo, mínimo, derivadas, etc.).

**Pruebas con Postman**

Puedes probar los diferentes endpoints usando **Postman**:

1. **Máximo**:

GET

http://localhost:8080/max-number?num1=4&num2=3&num3=2&num4=1

1. **Derivada**:

GET

http://localhost:8080/dynamic-derivative?num1=4&num2=3&num3=2&num4=1

1. **Evaluar Derivada**:

GET

<http://localhost:8080/evaluate-dynamic-derivative?num1=4&num2=3&num3=2&num4=1&x=2>

**Valor agregado:**

Además de cumplir con la tarea original de crear una **API** y un **cliente** para la **suma**, se ha añadido valor con las siguientes mejoras:

1. **Operaciones Matemáticas Ampliadas**: Además de la suma, se implementaron las funciones de **máximo**, **mínimo**, **promedio** y **ordenamiento** tanto ascendente como descendente de números, proporcionando más flexibilidad en las operaciones.
2. **Cálculo de Derivadas**: Se añadió la capacidad de **calcular derivadas** de polinomios de grado 3 y de **evaluar** la derivada en un punto específico. Esta funcionalidad extiende el uso de la API a aplicaciones matemáticas avanzadas.
3. **Interfaz Visual con DaisyUI y Angular**: Se implementó una interfaz atractiva y moderna usando **DaisyUI** junto a **Angular**, con soporte para **modo claro y oscuro** automático. Esto mejora la experiencia de usuario y facilita el cambio de modo según las preferencias o condiciones de luz.
4. **Estructura Optimizada con Spring Boot**: Gracias a **Spring Boot**, la estructura del proyecto se optimizó al separar la lógica de negocio en un servicio (MathService), mejorando la mantenibilidad, escalabilidad y permitiendo futuras ampliaciones sin comprometer la claridad del código.

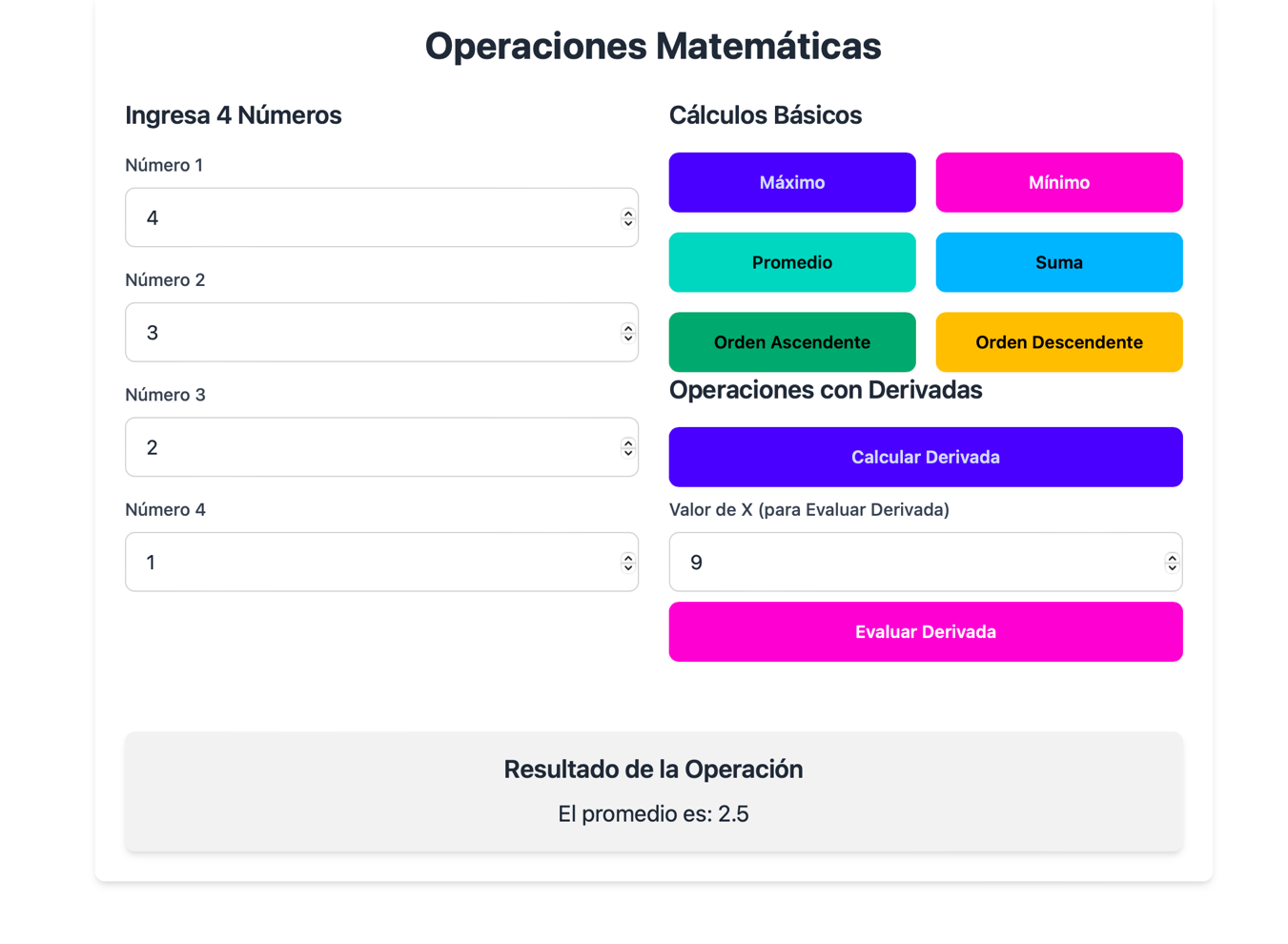
Estas mejoras hacen que la aplicación sea más funcional y adaptable, ofreciendo una experiencia mucho más completa que la especificada inicialmente.

**Capturas de Pantalla:**









**Captura Postman:**

