Documentación API Mutantes – Arquitectura y Secuencia

Introducción:

Este proyecto consiste en una API que verifica si un ADN es mutante o humano, con la capacidad de almacenar estos datos en una base de datos H2 y devolver estadísticas.
La API cuenta con dos endpoints principales: uno para verificar el ADN y otro para obtener estadísticas de las verificaciones.

Arquitectura del sistema:

La arquitectura consta de los siguientes componentes:

1. **Cliente**: Puede ser Postman, cURL, o cualquier cliente que realice peticiones HTTP.
2. **API**: Implementada en Spring Boot, con endpoints para la verificación de ADN y estadísticas de ADN verificados.
3. **Base de Datos H2**: Se utiliza para almacenar los registros de ADN verificados.
4. **Flujo de comunicación**:

    - Cliente → API (solicitud con secuencia de ADN).
    - API → Base de Datos (almacenamiento de ADN).
    - API → Cliente (respuesta con el resultado de la verificación).
    - Cliente → API (solicitud de estadísticas).

Diagrama de Secuencia:

El flujo de trabajo en la API consta de los siguientes pasos:

1. **POST /mutant/**: El cliente envía una solicitud POST a /mutant/ con un JSON que contiene la secuencia de ADN.
2. **Validación del ADN**: La API procesa el JSON y verifica si la secuencia de ADN pertenece a un mutante o a un humano.
3. **MutantService**: Este servicio contiene la lógica de validación y se encarga de determinar si la secuencia es mutante o no.
4. **Registro en la base de datos**: Si el ADN es mutante o humano, se guarda un único registro en la base de datos, con su resultado correspondiente.
5. **Respuesta al cliente**: La API responde al cliente con el resultado de la verificación (mutante o humano).

6. **GET /stats/**: Para obtener estadísticas, el cliente puede realizar una solicitud GET a /stats/, y la API devuelve un JSON con las estadísticas de las verificaciones de ADN.

Detalles Técnicos de los Endpoints:

1. **POST /mutant/:**
   Recibe un JSON con la secuencia de ADN y devuelve:
   - 200 OK si el ADN es mutante.
   - 403 Forbidden si el ADN es humano.

Ejemplo de Request:

```
{
    "dna": ["ATGCGA", "CAGTGC", "TTATTT", "AGACGG", "GCGTCA", "TCACTG"]
}
```

Ejemplo de Respuesta:

- Mutante: 200 OK.
- Humano: 403 Forbidden.

2. **GET /stats/**:
   Devuelve estadísticas sobre los análisis de ADN en formato JSON, incluyendo:
   - count_mutant_dna: Número de secuencias mutantes detectadas.
   - count_human_dna: Número de secuencias humanas detectadas.
   - ratio: Proporción de secuencias mutantes respecto a secuencias humanas.

Ejemplo de Respuesta:

```
{
  "count_mutant_dna": 40,
  "count_human_dna": 100,
  "ratio": 0.4
}
```

Instrucciones para Ejecutar el Proyecto:

1. **Ejecutar localmente:**
    - Clonar el repositorio.
    - Ejecutar la aplicación Spring Boot.
2. **Crear la imagen Docker:**
    - Compilar la imagen usando el siguiente comando:
      docker build -t adnmutante .
    - Ejecutar el contenedor con la imagen creada:
      docker run -p 8080:8080 adnmutante
3. **Realizar peticiones a la API:**
    - Para verificar un ADN mutante usando cURL:
      curl -X POST http://localhost:8080/mutant/ -d '{"dna": ["ATGCGA", "CAGTGC", "TTATTT", "AGACGG", "GCGTCA", "TCACTG"]}' -H "Content-Type: application/json"
    - Para consultar las estadísticas usando cURL:
      curl -X GET http://localhost:8080/stats/