# Documentación del Proyecto: **MutantesProg3**

## Consigna del Proyecto

Magneto te ha contratado para desarrollar un sistema que detecte si un humano es mutante basándose en su secuencia de ADN. La secuencia está representada por una matriz NxN compuesta por las letras A, T, C y G. Debes implementar un programa que determine si hay más de una secuencia de cuatro letras idénticas consecutivas en sentido horizontal, vertical u oblicuo.

### Requerimientos Funcionales

1. **Detección de Mutantes:** Implementar la función isMutant(String[] dna) que reciba una matriz de ADN y retorne true si la persona es mutante y false si no lo es.
2. **API REST:** Crear una API REST para exponer el servicio de detección de mutantes, que permita realizar la verificación a través de un método POST.
3. **Almacenamiento y Estadísticas:** Almacenar los ADN verificados en una base de datos H2 y proporcionar un endpoint adicional /stats que exponga estadísticas sobre la cantidad de mutantes y humanos verificados.

## Herramientas y Tecnologías a Utilizar

1. **Lenguaje:** Java (versión 17)
2. **IDE:** IntelliJ IDEA Ultimate
3. **Framework:** Spring Boot (para el desarrollo de la API REST)
4. **Base de Datos:** H2 (base de datos en memoria)
5. **Herramienta de construcción:** Gradle (para la gestión de dependencias y construcción del proyecto)
6. **Control de versiones:** Git y GitHub (para el manejo del código fuente)
7. **Hosting:** Render (para desplegar la API de forma gratuita)
8. **Herramientas de prueba:**
   * **JUnit:** para pruebas automáticas.
   * **Postman:** para pruebas manuales de la API.

## Pasos a Seguir

### 1. Preparación del Entorno

* **Instalación del JDK:**
  + Asegúrate de tener instalado JDK 17 o superior.
* **Instalación de IntelliJ IDEA:**
  + Descarga e instala IntelliJ IDEA Ultimate desde [el sitio oficial](https://www.jetbrains.com/idea/download/).
* **Configuración de Git:**
  + Instala Git y configura tu cuenta de GitHub. Crea un nuevo repositorio para el proyecto.
* **Creación del Proyecto:**
  + Abre IntelliJ IDEA y crea un nuevo proyecto.
  + Selecciona "Spring Initializr" y configura las siguientes opciones:
    - **Project Name:** MutantesProg3
    - **Type:** Gradle
    - **Language:** Java
    - **Spring Boot:** 3.3.4
    - **Java Version:** 17
  + Selecciona las dependencias necesarias:
    - Spring Web
    - H2 Database
    - Spring Boot DevTools

### 2. Implementar la Lógica del Proyecto

* **Función isMutant:**
  + Implementar la lógica para detectar secuencias de ADN mutante en la función isMutant(String[] dna).
* **Controlador REST:**
  + Crear un controlador en Spring Boot que maneje el endpoint POST /mutant/ para recibir secuencias de ADN en formato JSON.
* **Endpoint de Estadísticas:**
  + Implementar otro endpoint GET /stats que devuelva estadísticas sobre la cantidad de mutantes y humanos verificados.

### 3. Pruebas Locales

* **Pruebas de API:**
  + Utiliza Postman o Insomnia para probar los endpoints localmente, enviando peticiones POST y validando las respuestas de la API.
* **Pruebas Automáticas:**
  + Implementa pruebas automáticas usando JUnit para garantizar que tanto el algoritmo como la API funcionen correctamente.

### 4. Despliegue en Render

* **Subida a GitHub:**
  + Sube el proyecto a tu repositorio en GitHub.
* **Creación de Servicio en Render:**
  + Crea un servicio en Render para desplegar la API, conectando tu cuenta de GitHub.
* **Configuración de Variables de Entorno:**
  + Configura las variables de entorno necesarias y verifica que la aplicación se despliegue correctamente.

### 5. Optimización y Documentación

* **Optimización del Rendimiento:**
  + Añadir optimizaciones para manejar grandes cantidades de tráfico y asegurar una respuesta rápida de la API.
* **Documentación:**
  + Escribir un README en GitHub que contenga instrucciones de instalación, configuración y uso de la API.
* **Pruebas Automáticas Adicionales:**
  + Asegúrate de tener pruebas automáticas que cubran al menos el 80% del código.

## Conclusión

El proyecto "MutantesProg3" está diseñado para ser un sistema escalable y eficiente que cumpla con los requisitos establecidos por Magneto. A través de la implementación de la lógica de detección de mutantes y la creación de una API REST utilizando herramientas modernas como Spring Boot y Gradle, se busca ofrecer un servicio robusto que pueda ser fácilmente desplegado y mantenido.