

Desafío Sr. Full Stack Developer

Requerimos que desarrolles un proyecto que detecte si una persona tiene diferencias genéticas basándose en su secuencia de ADN. Para eso es necesario crear un programa con un método o función con la siguiente firma:

```
boolean hasMutation(String[] dna);
```

Debe recibir como parámetro un arreglo de cadena caracteres que representan cada fila de una tabla de NxN con la secuencia del ADN. Las letras de los caracteres solo pueden ser: A, T, C, G; las cuales representan cada base nitrogenada del ADN.

Sin mutación:

A	T	G	C	G	A
C	A	G	T	G	C
T	T	A	T	T	T
A	G	A	C	G	G
G	C	G	T	C	A
T	C	A	C	T	G

Con mutación:

A	T	G	C	G	A
C	A	G	T	G	C
T	T	A	T	G	T
A	G	A	A	G	G
C	C	C	C	T	A
T	C	A	C	T	G

Sabrás si existe una mutación si se encuentra más de una secuencia de cuatro letras iguales, de forma oblicua, horizontal o vertical.

Ejemplo de caso con mutación:

```
String[] dna = { "ATGCGA", "CAGTGC", "TTATGT", "AGAAGG", "CCCCTA",  
"TCACTG" };
```

En este caso el llamado a la función `hasMutation(dna)` devuelve `true`. Desarrolla el algoritmo de la manera más eficiente posible.

Desafíos

Nivel 1

Programa (en cualquier lenguaje de programación) que cumpla con lo descrito anteriormente.

Nivel 2

Crear una API REST, hostear esa API en un cloud computing libre (AWS, Azure DevOps, GCP, etc.), crear el endpoint `POST /mutation` en donde se pueda detectar si existe mutación enviando la secuencia de ADN mediante un JSON el cual tenga el siguiente formato:

```
POST /mutation  
{  
  "dna":["ATGCGA", "CAGTGC", "TTATGT", "AGAAGG", "CCCCTA", "TCACTG"]  
}
```

En caso de verificar una mutación, debería devolver un `200`, en caso contrario un `403`.

Nivel 3

Anexar una base de datos, la cual guarde los ADNs verificados con la API. Sólo 1 registro por ADN. Exponer un servicio extra `GET /stats` que devuelva un JSON con las estadísticas de las verificaciones de ADN:

```
{  
  "count_mutations": 40,  
  "count_no_mutation": 100,  
  "ratio": 0.4  
}
```

Tener en cuenta que la API puede recibir fluctuaciones agresivas de tráfico (entre 100 y 1 millón de peticiones por segundo).

No se realizarán pruebas de estrés de nuestra parte.

Entregable:

- Repositorio en GitLab del código
- **README.md** con lo necesario para ejecutarlo local
- URL de la API

¿Qué evaluaremos?

Críterio	Valor
Patrones de diseño	15%
Solución en la nube	10%
Complejidad algorítmica (Big O)	15%
Mejores prácticas	20%
CI/CD	15%
Pruebas (cobertura mínima 40%)	15%
Seguridad	10%