

Prueba Técnica BackEnd Node.js

Instrucciones:

- Lea detenidamente los aspectos que se solicitan en cada parte o sección.
- Se dará un tiempo para la culminación de la prueba de 72 Horas, pasado este tiempo debe responder el correo con el enlace al repositorio y el enlace al despliegue.

Introducción

El profesor Bruce Banner está haciendo un estudio de los astronautas que regresan a la tierra después de una misión para determinar si la radiación les causó algún tipo de anomalía. Para determinarlo se les practica una prueba de ADN.

Requerimientos:

- Se debe crear una API REST con node.js que contenga los siguientes End Point
 - **POST: /validate-anomaly**

```
// POST validate-anomaly

{
  "dna": [
    ["A", "C", "A", "D"],
    ["A", "B", "C", "D"],
    ["A", "C", "C", "A"],
    ["C", "B", "C", "D"]
  ]
}
```

Debe recibir un JSON con el parámetro **DNA**, el cual es una matriz bidireccional con **N** cantidad de elementos (desde 3X3 hasta 2000X2000).

La misma debe validar si existe anomalía (esta se presenta cuando se encuentran 3 o más veces la misma letra de manera consecutiva en horizontal, vertical o diagonal de la matriz) como los siguientes ejemplos.

1

A	B	C	D	E
A	A	A	D	D
B	A	D	A	E
A	B	C	D	A
B	A	C	D	E

2

A	B	C	D	E
A	B	C	D	D
B	A	D	D	E
A	B	C	D	A
B	A	C	D	E

3

A	B	C	D	E
A	B	C	D	C
B	A	D	C	E
A	B	C	D	A
B	A	C	D	E

4

A	B	C	D	E
A	B	C	D	B
B	A	D	A	E
A	B	A	D	A
B	A	C	D	E

En caso de que exista una anomalía, debería devolver un código 200, en caso contrario un 403.

Se debe guardar en alguna base de datos cada una de las validaciones realizadas al EndPoint con su respectivo resultado

- **GET: /stats**

El cual debe responder un **JSON** con las estadísticas de uso del endpoint anterior.

```
// GET stats  
  
{  
  "count_anomalies": 30,  
  "count_no_anomalies": 70,  
  "ratio": 0.3  
}
```

- **count_anomalies:** cuantas peticiones se encontraron con anomalía.
- **count_no_anomalies:** Cuantas peticiones no se encontraron anomalías.
- **ratio:** factor que indica del 100 % de las peticiones hechas cuánto porcentaje se encontró anomalías.

Puntos a Evaluar:

- Análisis y desarrollo de los requerimientos.
- Conocimiento en desarrollo en Node.js / TypeScript
- Patrones de diseño
- Solución en la nube
- Conocimiento en creación y uso de API REST
- Complejidad algorítmica, Big O, **Plus**
- Pruebas unitarias, **Plus**

Consideraciones:

- Se debe desarrollar en Node.js (JavaScript o TypeScript).
- Debe enviarse el enlace al repositorio en GitHub público.
- Debe desplegarse en algún servicio que permita su uso y validación, preferiblemente AWS.