

PRUEBAS UNITARIAS – TEST COVERAGE

Detección de Mutantes-Mutante Detectado

HTTP Mutante API REQUEST / Mutante Save Share

POST ▼ https://parcialbackend-deteccionmutantes-lin.onrender.com/mutant Send ▼

Params Authorization Headers (9) **Body** ● Scripts Settings Cookie

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Beautify

```
1  {"dna":["ATGCGA","CAGTGC","TTATGT","AGAAGG","CCCCTA","TCACTG"]}
2
```

Body Cookies Headers (12) Test Results 200 OK • 2.26 s • 458 B • 🌐 📄 🔍

Pretty Raw Preview Visualize Text ▼ 🔗

```
1  {"message":"Mutante detectado"}
```

Detección de Mutantes-No es Mutante

HTTP Mutante API REQUEST / Mutante Save Share

POST ▼ https://parcialbackend-deteccionmutantes-lin.onrender.com/mutant Send ▼

Params Authorization Headers (9) **Body** ● Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Beautify

```
1  {"dna":["TTGCGA","CAGTGC","TTTTGT","GAGGAC","GAGGAC","GAGGAC"]}
2
```

Body Cookies Headers (12) Test Results 403 Forbidden • 1973 ms • 461 B • 🌐 📄 🔍

Pretty Raw Preview Visualize Text ▼ 🔗

```
1  {"message":"No es mutante"}
```

Algunos manejos de errores:

a. Recibir un array vacío

HTTP Mutante API REQUEST / Mutante Save Share

POST ▼ Send ▼

Params Authorization Headers (9) **Body** ● Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Beautify

```
1  {"dna": []}
2
```

Body Cookies Headers (12) Test Results **400 Bad Request** • 44.08 s • 486 B •

Pretty Raw Preview Visualize **Text** ▼

```
1  {"error": "El array de ADN no puede estar vacío."}
```

b. Recibir un array de NxM en vez de un NxN

HTTP Mutante API REQUEST / Mutante Save Share

POST ▼ Send ▼

Params Authorization Headers (9) **Body** ● Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Beautify

```
1  {"dna": ["ATG", "CAGT", "TTAG"]}
```

Body Cookies Headers (7) Test Results **400 Bad Request** • 13 ms • 278 B •

Pretty Raw Preview Visualize **Text** ▼

```
1  {"error": "El array debe ser de tamaño NxN."}
```

Postbot Runner Start Proxy Cookies Vault Trash

c. Recibir un array de números (entre comillas para que sean string)

HTTP Mutante API REQUEST / Mutante Save Share

POST ▼ Send ▼

Params Authorization Headers (9) **Body** ● Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ▼ Beautify

```
1 {"dna":["1111", "3456", "3421", "4111"]}
```

Body Cookies Headers (7) Test Results 400 Bad Request • 9 ms • 278 B • 🌐 🔍 ⋮

Pretty Raw Preview Visualize Text ▼ 🔗

```
1 {"error":"El ADN no puede contener números"}
```

🔗 Postbot ▶ Runner 🔗 Start Proxy 🔗 Cookies 🏠 Vault 🗑️ Trash

d. Recibir null

POST ▼ Send ▼

Params Authorization Headers (9) **Body** ● Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ▼ Beautify

```
1 {"dna":null}
```

Body Cookies Headers (7) Test Results 400 Bad Request • 9 ms • 279 B • 🌐 🔍 ⋮

Pretty Raw Preview Visualize Text ▼ 🔗

```
1 {"error":"El array de ADN no puede ser null."}
```

🔗 Postbot ▶ Runner 🔗 Start Proxy 🔗 Cookies 🏠 Vault 🗑️ Trash

e. Recibir un array de NxN de nulls

HTTP Mutante API REQUEST / Mutante Save Share

POST ▼ Send ▼

Params Authorization Headers (9) **Body** ● Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Beautify

```
1 {"dna":[null, null, null, null]}
```

Body Cookies Headers (7) Test Results **400 Bad Request** • 10 ms • 281 B • 🌐 🔍 ⋮

Pretty Raw Preview Visualize **Text** ▼ 🔍

```
1 {"error":"Cada fila del ADN no puede ser null."}
```

Postbot Runner Start Proxy Cookies Vault Trash

f. Recibir un array de NxN con letras distintas a las propuestas {"B","H"}

HTTP Mutante API REQUEST / Mutante Save Share

POST ▼ Send ▼

Params Authorization Headers (9) **Body** ● Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Beautify

```
1 {"dna":["ATGCGA", "CAGTGC", "TTATGT", "AGAAGG", "CCCCTA", "TCACTB"]}
```

Body Cookies Headers (7) Test Results **400 Bad Request** • 9 ms • 294 B • 🌐 🔍 ⋮

Pretty Raw Preview Visualize **Text** ▼ 🔍

```
1 {"error":"El ADN solo puede contener las letras A, T, C, G."}
```

Postbot Runner Start Proxy Cookies Vault Trash

Estadísticas-Stats

HTTP

Mutante API REQUEST / Stats

Save

Share

GET

http://localhost:8080/stats

Send

Params

Authorization

Headers (6)

Body

Scripts

Settings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

200 OK

62 ms

303 B

Pretty

Raw

Preview

Visualize

JSON

1

{

2

"countMutantDna": 3,

3

"countHumanDna": 2,

4

"ratio": 1.5

5

}

Postbot

Runner

Start Proxy

Cookies

Vault

Trash

A continuación se muestra la información que había en la base de datos, lo cual verificamos que está funcionando correctamente el método stats.

No seguro

192.168.56.1:8082/login.do?sessionId=f46c677915050e2cd8bf2f2a4eca6ccb

A

Auto commit

Número máximo de filas: 1000

Auto completado

Desactivado

Auto select

On

jdbc:h2:tcp://localhost/~mutante

Ejecutar

Run Selected

Auto completado

Eliminar

Instrucción SQL:

DNA

DNA_AUD

REVINFO

INFORMATION_SCHEMA

Usuarios

H2 2.3.232 (2024-08-11)

SELECT * FROM DNA

SELECT * FROM DNA;

ID	DNA	IS_MUTANT
1	[ATGCGA, CAGTGC, TTATGT, AGAAGG, CCCCTA, TCACTG]	TRUE
2	[TTGCGA, CAGTGC, TTTTGT, GAGGAC, GAGGAC, GAGGAC]	FALSE
3	[TGAC, AGCC, TGAC, GGTC]	TRUE
4	[TTGCGA, GCTGTA, GCTGTA, ATGCAA, GAGGAC, GAGGAC]	FALSE
5	[TTGCGA, CAGTGC, TTTTGT, AGGACG, CCCCTA, GTTTTT]	TRUE

(5 filas, 3 ms)

Editar

HTTP

Mutante API REQUEST / Stats

Save

Share

GET

https://parcialbackend-deteccionmutantes-lin.onrender.com/stats

Send

Params

Authorization

Headers (6)

Body

Scripts

Settings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (12)

Test Results

200 OK

12.13 s

477 B

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "countMutantDna": 0,
3   "countHumanDna": 0,
4   "ratio": 0.0
5 }
```

Test Coverage

El enunciado solicita una cobertura de código superior al 80%.

```
}main
└─ java 100% classes, 94% lines covered
   └─ org.example.mutant 100% classes, 94% lines covered
      └─ controllers 100% classes, 100% lines covered
         ├── DnaMutanteController 100% methods, 100% lines covered
         └── StatsController 100% methods, 100% lines covered
      └─ dtos
         └── DTOStats
      └─ entities 100% classes, 100% lines covered
         └── Dna 100% methods, 100% lines covered
      └─ repositories
         └── DnaRepository
      └─ services 100% classes, 95% lines covered
         ├── DnaMutanteService 100% methods, 95% lines covered
         ├── StatsService 100% methods, 100% lines covered
         └── Main 50% methods, 33% lines covered
```

Coverage Tests in 'mutant.test' x				
Element	Class, %	Method, %	Line, %	Branch, %
org.example.mutant	100% (6/6)	94% (16/17)	94% (86/91)	86% (71/82)
controllers	100% (2/2)	100% (4/4)	100% (13/13)	100% (2/2)
DnaMutanteController	100% (1/1)	100% (2/2)	100% (8/8)	100% (2/2)
StatsController	100% (1/1)	100% (2/2)	100% (5/5)	100% (0/0)
entities	100% (1/1)	100% (1/1)	100% (2/2)	100% (0/0)
Dna	100% (1/1)	100% (1/1)	100% (2/2)	100% (0/0)
repositories	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
DnaRepository	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
dtos	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
DTOStats	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
services	100% (2/2)	100% (10/10)	95% (70/73)	86% (69/80)
StatsService	100% (1/1)	100% (2/2)	100% (8/8)	100% (2/2)
DnaMutanteService	100% (1/1)	100% (8/8)	95% (62/65)	85% (67/78)
Main	100% (1/1)	50% (1/2)	33% (1/3)	100% (0/0)

Como podemos observar, según el reporte de cobertura de ejecución en IntelliJ, casi todos los tests de las distintas clases proporcionadas cubren cerca del 100% de las líneas de código. Esto indica un alto nivel de cobertura de pruebas unitarias.

Estas pruebas permiten verificar que cada componente y función del sistema opere según lo esperado, detectando posibles errores o fallos. Además, aseguran la calidad del código y el correcto funcionamiento del proyecto.