

Mutant challenge

Esta aplicación fue creada ya que Magneto quiere reclutar la mayor cantidad de mutantes para poder luchar contra los X-Men.

La aplicación tiene dos endpoints:

- <https://mutants-333112.uc.r.appspot.com/stats> [GET]
- <https://mutants-333112.uc.r.appspot.com/mutant> [POST]

Uso de la aplicación

Se recibirá como parámetro un array de Strings que representan cada fila de una tabla de (NxN) con la secuencia del ADN. Las letras de los Strings solo pueden ser:(A,T,C,G), las cuales representan cada base nitrogenada del ADN.

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

No-Mutante

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

Mutante

Sabrás si un humano es mutante, si encuentras **más de una secuencia de cuatro letras iguales , de forma oblicua, horizontal o vertical.**

Endpoint /stats

Este endpoint retorna la estadística de las verificaciones de ADN <https://mutants-333112.uc.r.appspot.com/stats> [GET]

```
{“count_mutant_dna”:40, “count_human_dna”:100: “ratio”:0.4}
```

Si tratas de acceder a este endpoint **/stats** usando un verbo HTTP diferente de GET, recibirás un status code 405

Endpoint /mutants

Este endpoint valida según el ADN si eres mutante o no <https://mutants-333112.uc.r.appspot.com/mutant> [POST] El body del post es similar al siguiente:

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

- Si la validación de secuencia de ADN da **true** para mutante, retornara HTTP code 200.
- Si la validación de secuencia de ADN da **false** para mutante, retornara HTTP code 403.

Acerca de la arquitectura

La aplicación es desarrollada en GO(Golang), y se conecta a una instancia de base de datos alojada en infraestructura Cloud, mediante el motor de base de datos MongoDB.

Entre las mejoras que se pueden incluir, se puede pensar en una arquitectura donde las estructuras estén en paquetes diferentes, igual que la lógica de mutantes y el acceso a base de datos. De esta forma, abstraemos estos componentes en paquetes independientes.

Entendiendo el flujo de la petición POST:

1. El usuario envia el ADN mediante el body de la petición
2. Se valida si la secuencia de ADN existe en la base de datos
3. Si existe en la base de datos, retorna el valor de si es mutante o no. Si no existe en la base de datos, se valida el ADN y se guarda.

Ahora, entendamos la petición GET;

1. El usuario envia la petición al API.
2. Se consultan los registros encontrados en la base de datos de MongoDB
3. Se realiza el calculo de mutantes, de humanos y el radio.
4. Se envian las estadisticas en el response.

Ejecutando la aplicación

Para ejecutar la aplicación localmente se deben ejecutar los siguientes comandos, en el cmd de Windows:

```
go get github.com/githubnemo/CompileDaemon  
CompileDaemon --command="main.go.exe"
```

