

PARCIAL 1 - BACKEND

Desafíos:

Nivel 1: Programa que cumpla con el método pedido por Magneto.

Nivel 2: Crear una API REST, hostear esa API en un cloud computing libre (Google App Engine, Amazon AWS, etc), crear el servicio “/mutant/” en donde se pueda detectar si un humano es mutante enviando la secuencia de ADN mediante un HTTP POST con un Json el cual tenga el siguiente formato:

POST → /mutant/

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


En caso de verificar un mutante, debería devolver un HTTP 200-OK, en caso contrario un 403-Forbidden

Nivel 3: Anexar una base de datos, la cual guarde los ADN's verificados con la API. Solo 1 registro por ADN.


Exponer un servicio extra “/stats” que devuelva un Json con las estadísticas de las verificaciones de ADN: {“count_mutant_dna”:40, “count_human_dna”: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).

secuencias y pruebas de Stress

 Link: <https://github.com/beizae/Parcial1-Backend/Documentacion-Parcial-Beiza>

Proyecto subido a render

 Link: <https://parcial1-backend-9fpj.onrender.com>

Swagger

 Link: <https://parcial1-backend-9fpj.onrender.com/swagger-ui/index.html>

Postman

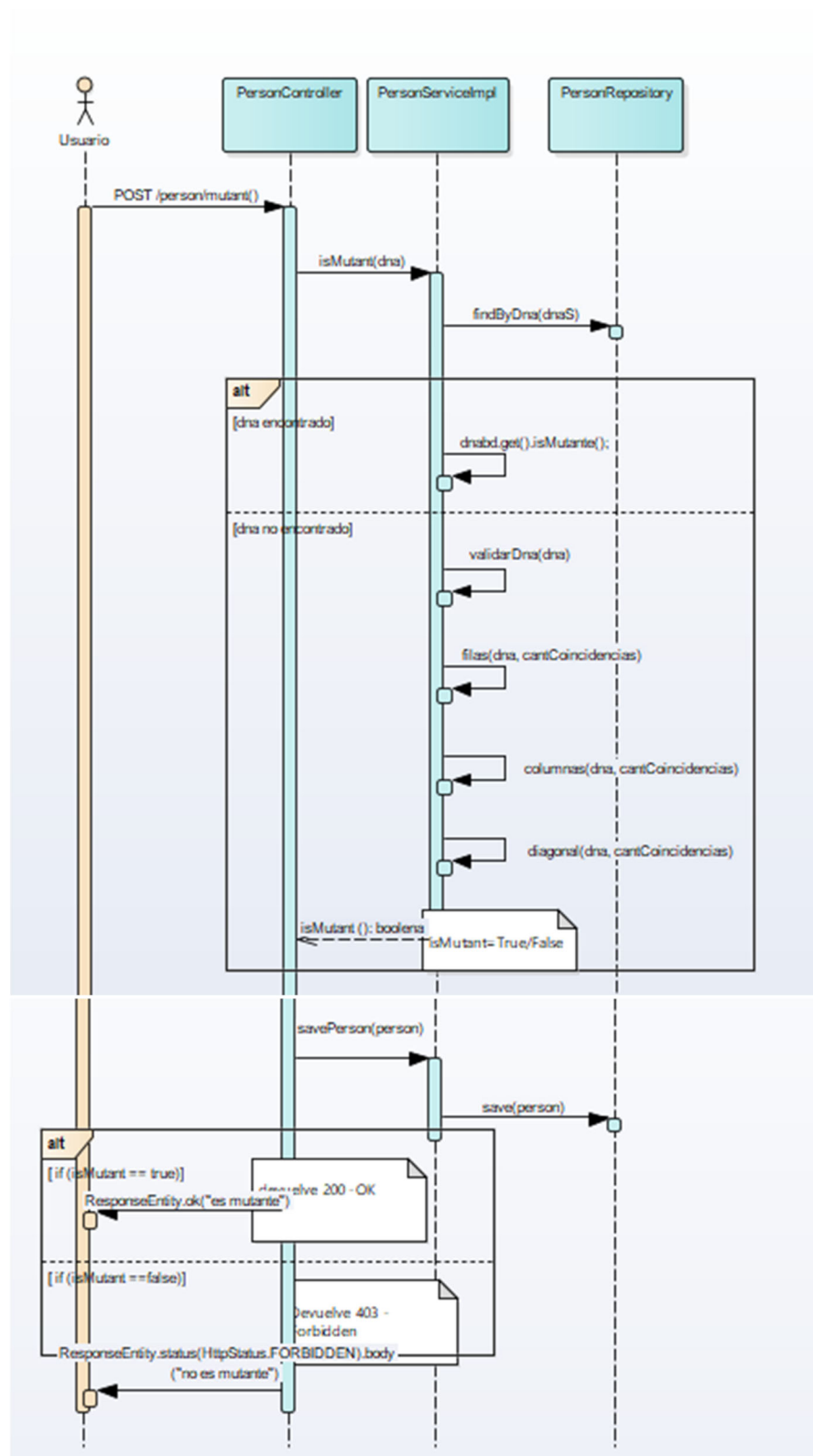
Para realizar las solicitudes utilizando Postman, utilizar los siguientes endpoints:

POST: <https://parcial1-backend-9fpj.onrender.com/enviaradn>

GET: <https://parcial1-backend-9fpj.onrender.com/enviaradn>

El formato de la consulta POST debe ser el siguiente:

```
{  
  "adn":["ATGCGA","CAGTGC","TTCTGT","AAAATG","CTTGGA","TCACTG"]  
}
```

Diagrama de secuencia

Pruebas de Stress JMETER

