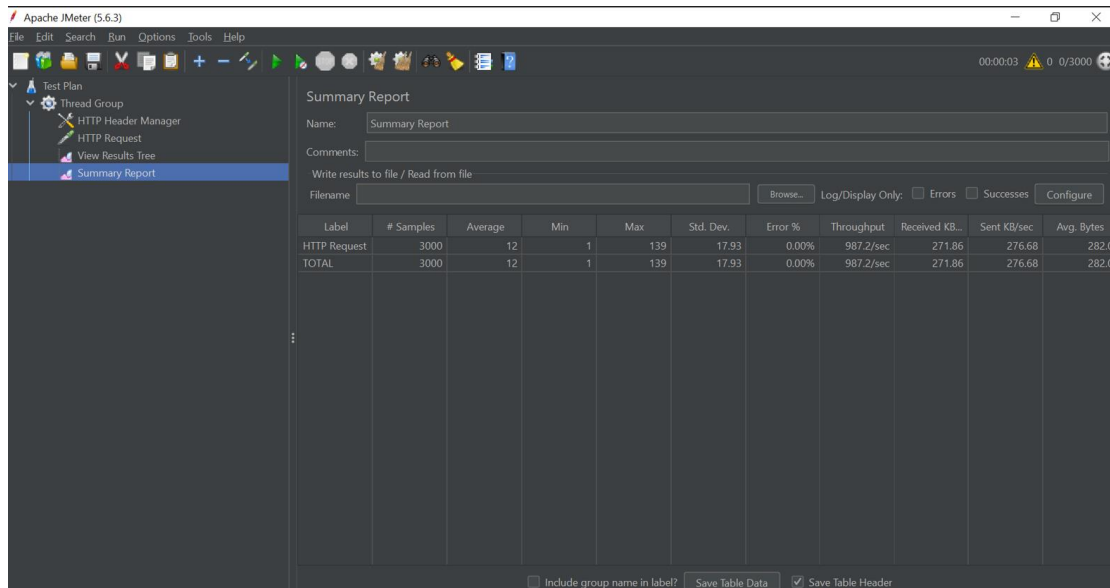
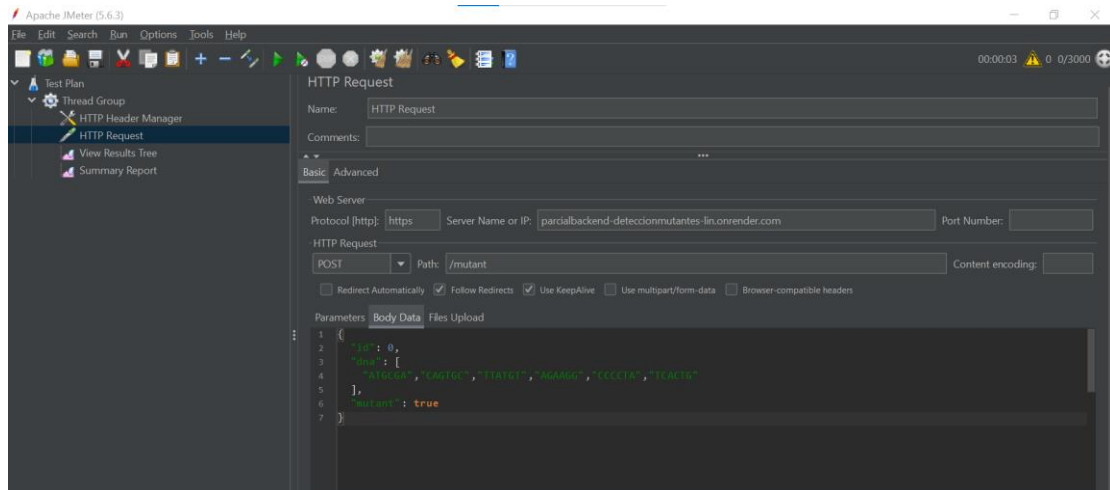
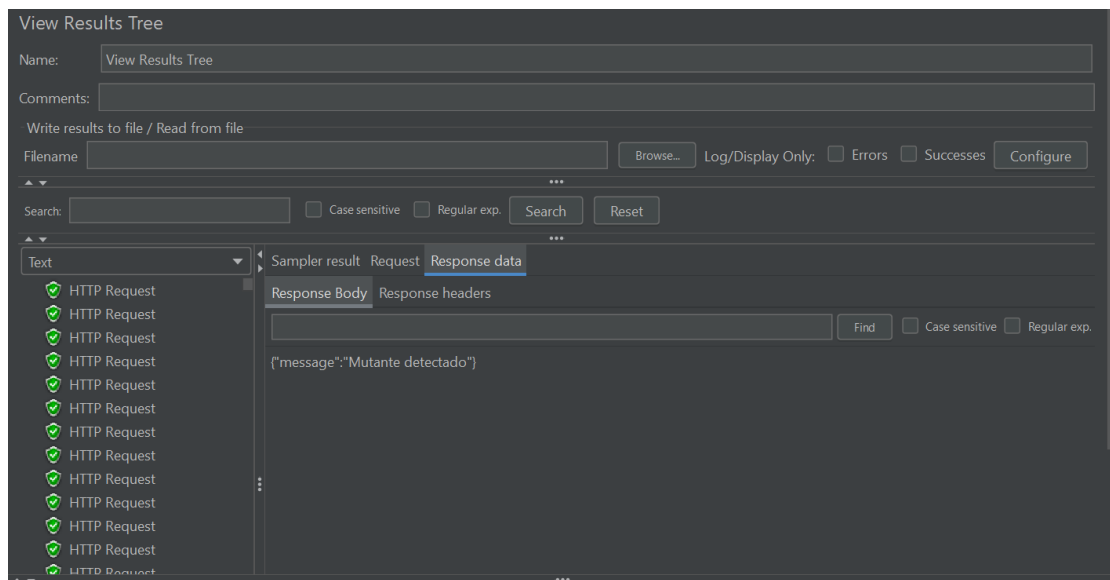


PRUEBAS DE STRESS-Jmeter

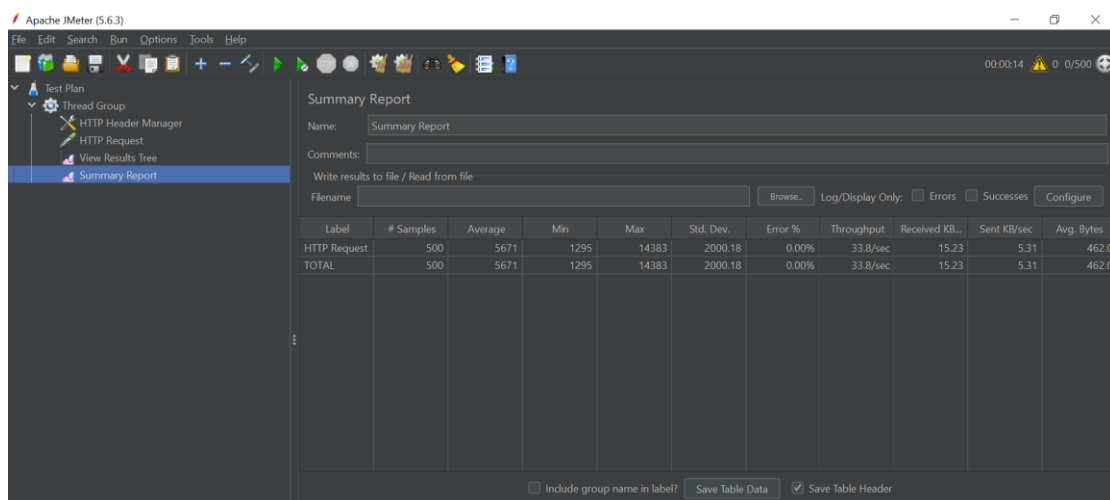
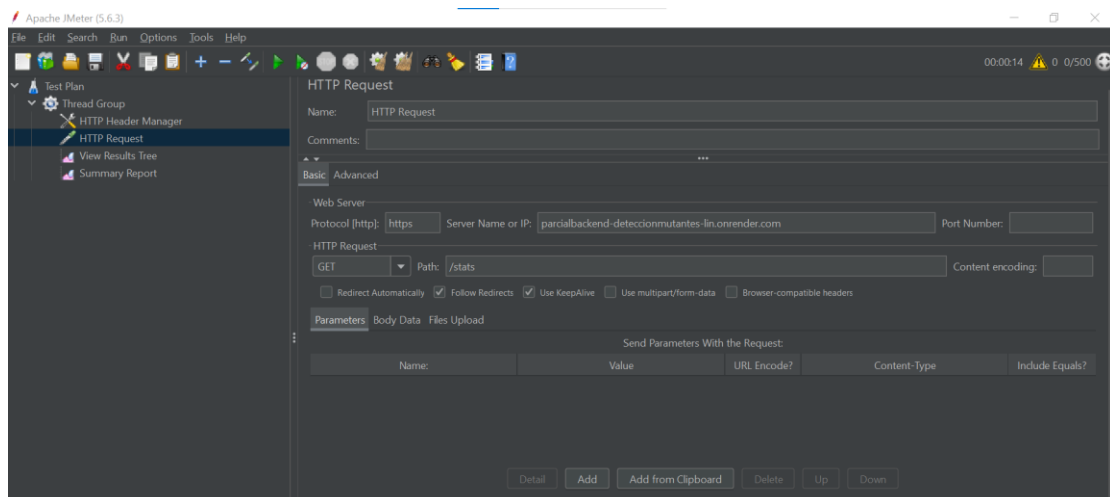
Algoritmo para detectar mutantes:

Se han llevado a cabo pruebas de estrés utilizando Jmeter, sometiendo el sistema a 3000 peticiones en un solo segundo. Los resultados, que se pueden observar a continuación, muestran un error del 0%. Esto indica que el sistema maneja eficientemente cargas de tráfico sin incurrir en errores, asegurando así la robustez y fiabilidad de la API bajo condiciones extremas.





Asimismo, se probó el método de stats enviándole 500 peticiones en 1 segundo, logrando también un 0% de error. Esto demuestra que el sistema mantiene su eficacia y estabilidad, incluso bajo cargas de tráfico significativas, garantizando así la fiabilidad del servicio.



Se han realizado varias investigaciones, y una de las soluciones recomendadas para escalar el sistema en el futuro y soportar una mayor carga de tráfico es implementar el uso de hilos. Esto permitiría

ejecutar tareas en paralelo, optimizando así la eficiencia del programa.

Otra solución posible es incorporar Redis para el almacenamiento de datos en caché, como vimos en clase. Con Redis, la información solicitada se almacena temporalmente, lo que permite que, si otra persona realiza la misma petición, los datos se obtengan directamente de la caché en lugar de consultar nuevamente la base de datos. Esto optimiza el tiempo de respuesta, mejora significativamente el rendimiento del programa y facilita el soporte de fluctuaciones intensas en el tráfico.