

# Mercado Libre

## Test

## X-Men

**Nombre:** Felipe Esteban Riaño Pineda

**Email:** [felipepineda1997@gmail.com](mailto:felipepineda1997@gmail.com)

**Celular:** +57 3053063835

**WhatsApp:** +57 3107907321

En el presente documento se busca detallar algunos detalles del desarrollo del proyecto.

En primer lugar, el desarrollo del proyecto se implemento con una arquitectura hexagonal, con el fin de brindar atributos de calidad como escalabilidad y fácil mantenimiento.

Adjunto a este documento se detalla un modelo de como se implemento la arquitectura.

El proyecto se desplego en una instancia de EC2 de AWS, la cual se puede acceder con el siguiente link:

<http://ec2-34-204-47-229.compute-1.amazonaws.com:8080>

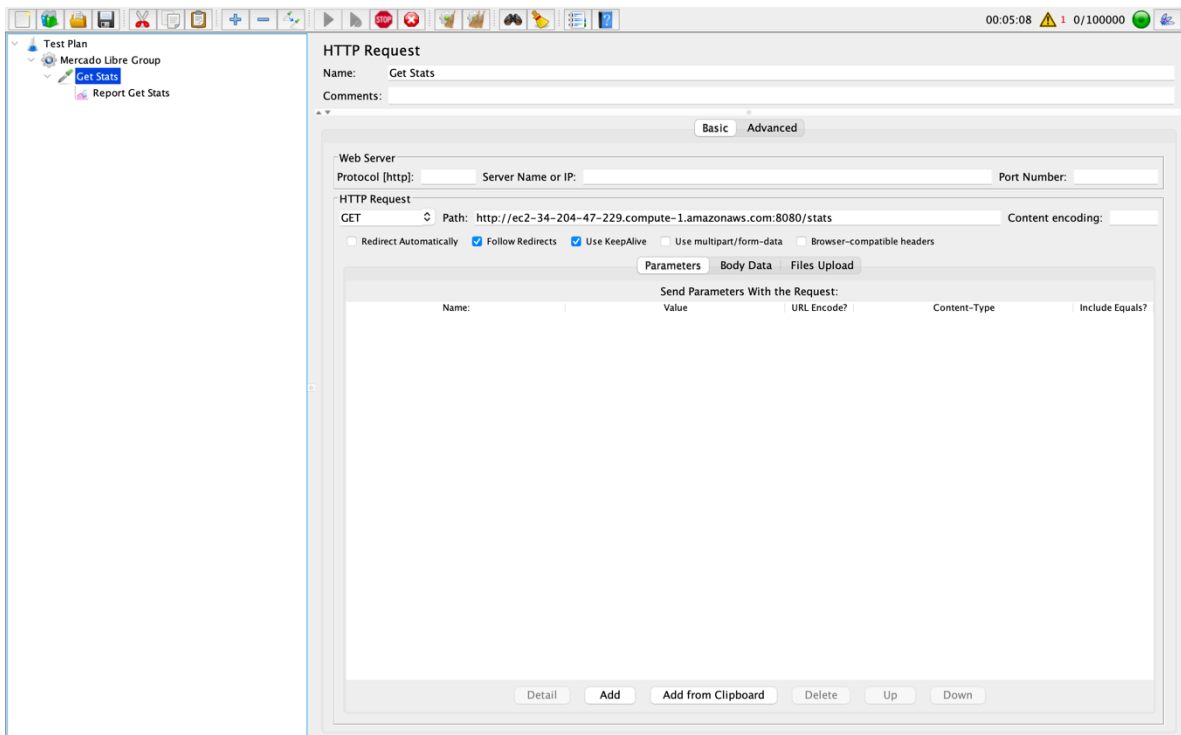
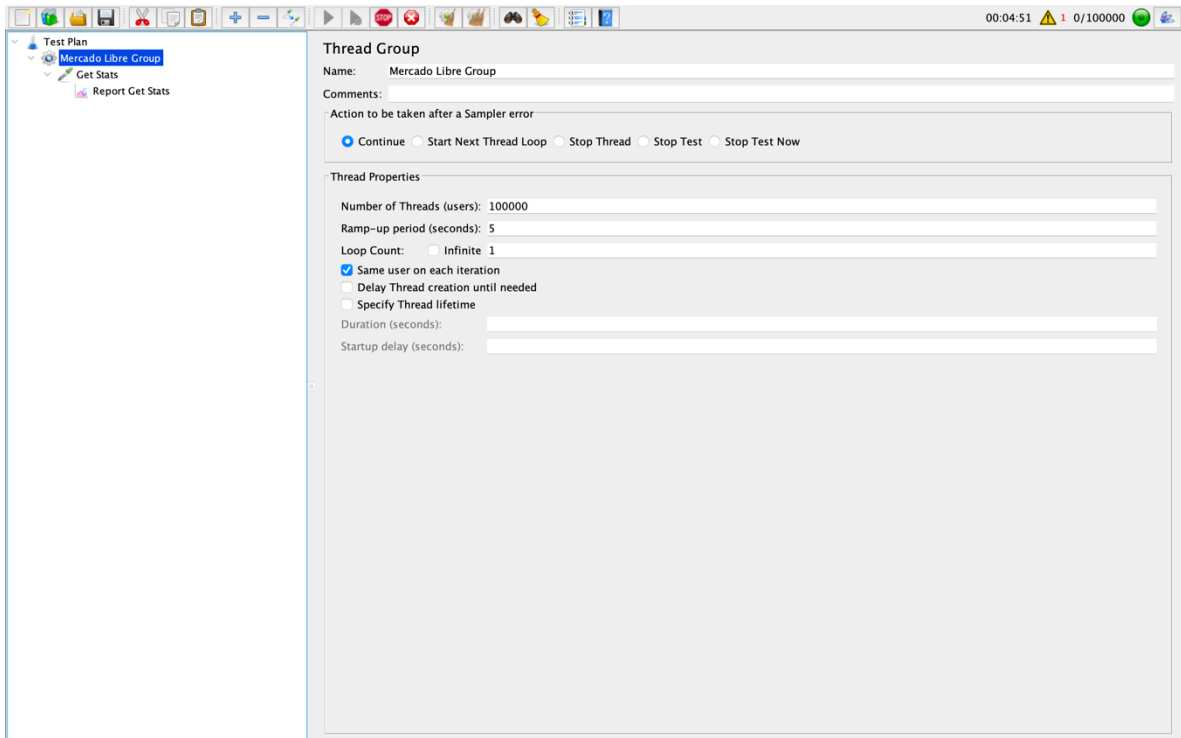
El servicio expuesto en el link enviado posee 2 endpoint, tal cual lo plantea el ejercicio uno para almacenar el ADN de un humano y verificar si es humano o mutante y el otro en el cual se pueden consultar las estadísticas de los ADN's procesados, a continuación se escriben los links en los cuales se puede acceder a esta información:

- <http://ec2-34-204-47-229.compute-1.amazonaws.com:8080/mutant>
- <http://ec2-34-204-47-229.compute-1.amazonaws.com:8080/stats>

El servicio también cuenta con un endpoint, en el cual se expone una documentación en Swagger, el link al cual se debe acceder es:

- <http://ec2-34-204-47-229.compute-1.amazonaws.com:8080/swagger-ui.html>

Adicionalmente, como se planteo en el ejercicio este servicio podría someterse a cargas de estrés muy altas. Por esta razón se realizo una prueba de rendimiento con ayuda de la herramienta de JMeter (Se selecciono el método HTTP Get debido a que la persistencia de la información se esta almacenando en una instancia MySQL del servicio RDS de Amazon, y el guardar una gran cantidad de datos podría exceder la capa gratuita), la prueba de estrés se realizo con 100.000 peticiones y estos son los resultados que arrojo:



**Summary Report**

Name: Report Get Stats

Comments:

Write results to file / Read from file

Filename   Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Get Stats	2087	15573	349	31262	8943.75	0.00%	65.0/sec	16.88	9.71	266.0
TOTAL	2087	15573	349	31262	8943.75	0.00%	65.0/sec	16.88	9.71	266.0

☐ Include group name in label?  ☒ Save Table Header

En este caso tuvo un porcentaje de error del 0% para 100.000 en 5 segundos. Igualmente cabe resaltar que estos rendimientos de por si podrían mejorar un escalamiento horizontal en infraestructura.

En cuanto a la cobertura de código, se realizó por medio del IDE IntelliJ y arrojó los siguientes resultados:

The screenshot shows the IntelliJ IDEA interface with the following components:

- Project View:** Shows the project structure with folders like 'main', 'test', 'application', 'config', 'generator', and 'infrastructure'.
- Code Editor:** Displays the 'build.gradle' file for the 'xmen\_mutant\_ms' project. The code includes SonarQube properties, group, version, sourceCompatibility, repositories, and dependencies.
- Coverage Window:** Shows the coverage results for 'Tests in 'xmen\_mutant\_ms.test''. The table indicates 91% classes and 76% lines covered in 'all classes in scope'. The 'persistence' package is highlighted with 100% coverage.
- Run Window:** Shows the test results for 'Tests in 'xmen\_mutant\_ms.test''. The output includes task names like 'compileJava', 'processResources', 'classes', 'compileTestJava', 'processTestResources', 'testClasses', and 'test'. It also shows the IntelliJ IDEA coverage runner output.

Element	Class, %	Method, %	Line, %
com	91% (22/24)	82% (53/64)	76% (171/223)
persistence	100% (0/0)	100% (0/0)	100% (0/0)

Para esto se creó una organización con un repositorio para almacenar el proyecto desarrollado y otro donde se almacenara dicho documento y algunos otros archivos de importancia.

Link de la organización:

<https://github.com/MercadoLibreTestFelipeRiano>

Link repositorio del proyecto:

[https://github.com/MercadoLibreTestFelipeRiano/xmen\\_mutant\\_ms](https://github.com/MercadoLibreTestFelipeRiano/xmen_mutant_ms)