



# Estructuración de la Iniciativa

*Examen  
Mercadolibre*


## Magneto

**Fecha de Creación:**

2021-08-17


**Versión:**

1.0

	<b>Formato:</b> Documento Estructuración de la Iniciativa		<b>Código:</b> 001
	<b>PROCESO:</b> Gestión de Requerimientos		
	<b>Subproceso:</b> Gestión de la Demanda	<b>Emitido:</b> 17/08/2021	<b>Actualizado:</b>


### Objetivo del Documento

La iniciativa que se presenta en este documento la compone esencialmente evidenciar el proceso de la solicitud de Magneto y la propuesta que realizamos para cumplir dicho requerimiento.

	<b>Formato:</b> Documento Estructuración de la Iniciativa		<b>Código:</b> 001
	<b>PROCESO:</b> Gestión de Requerimientos		
	<b>Subproceso:</b> Gestión de la Demanda	<b>Emitido:</b> 17/08/2021	<b>Actualizado:</b>

## Contenido

1.	CLASIFICACIÓN DE LA INICIATIVA.....	4
2.	DESCRIPCIÓN DEL PROBLEMA.....	4
3.	DESCRIPCIÓN DEL PROCESO ACTUAL.....	4
4.	ALCANCE DE LA INICIATIVA .....	4
5.	REQUERIMIENTOS FUNCIONALES.....	4
6.	CRITERIOS DE ACEPTACION .....	5
7.	REGLAS DE NEGOCIO .....	5
8.	BENEFICIOS ESPERADOS .....	6
9.	IMPLEMENTACION.....	6
9.1.	DIAGRAMAS .....	6
9.2.	ILUSTRACIONES: Para cumplir con el requerimiento funcional (RF_001) y entender los criterios de aceptación y las reglas del negocio, se debe interpretar la siguiente ilustración: .....	7
9.3.	RAZONAMIENTO: Para cumplir con el objetivo el requerimiento, se plantea la siguiente implementación: .....	7
10.	REQUERIMIENTOS NO FUNCIONALES .....	9
11.	ENTREGABLES .....	9

	<b>Formato:</b> Documento Estructuración de la Iniciativa		<b>Código:</b> 001
	<b>PROCESO:</b> Gestión de Requerimientos		
	<b>Subproceso:</b> Gestión de la Demanda	<b>Emitido:</b> 17/08/2021	<b>Actualizado:</b>

## 1. CLASIFICACIÓN DE LA INICIATIVA

Requerimiento del negocio.  
Nueva funcionalidad.

## 2. DESCRIPCIÓN DEL PROBLEMA

Actualmente Magneto quiere reclutar la mayor cantidad de mutantes para poder luchar contra los X-Men.

## 3. DESCRIPCIÓN DEL PROCESO ACTUAL


N/A.

## 4. ALCANCE DE LA INICIATIVA

Tener un proceso funcional el cual le permita a Magneto mediante un conjunto de pasos sencillos identificar si un humano es mutante basándose en su secuencia de ADN.

## 5. REQUERIMIENTOS FUNCIONALES

- **RF\_001:** Crear una API REST, hostear esa API en un cloud computing libre.
- **RF\_002:** Crear un servicio “/mutant” en donde se pueda detectar si un humano es mutante enviando la secuencia de ADN.
- **RF\_003:** Crear un servicio “/stats” que devuelva las estadísticas de las verificaciones de ADN.
- **RF\_004:** Anexar una base de datos.
- **RF\_005:** Tener en cuenta que la API puede recibir fluctuaciones agresivas de tráfico (Entre 100 y 1 millón de peticiones por segundo).

	<b>Formato:</b> Documento Estructuración de la Iniciativa		<b>Código:</b> 001
	<b>PROCESO:</b> Gestión de Requerimientos		
	<b>Subproceso:</b> Gestión de la Demanda	<b>Emitido:</b> 17/08/2021	<b>Actualizado:</b>

## 6. CRITERIOS DE ACEPTACION


- **RF\_001:** Crear un programa con alguno de los siguientes lenguajes: Java / Golang / C-C++ / Javascript (node) / Python / Ruby.
- **RF\_002:**
  - Crear servicio mediante un HTTP POST con un Json el cual tenga el siguiente formato:  
**POST → /mutant**  

```
{
  "dna":["ATGCGA","CAGTGC","TTATGT","AGAAGG","CCCCTA","TCACTG"]
}
```
  - En caso de verificar un mutante, debería devolver un HTTP 200-OK, en caso contrario un 403-Forbidden.
- **RF\_003:** Crear servicio mediante un HTTP GET que devuelva un Json el cual tenga el siguiente formato:  
**GET → /stats**  

```
{ "count_mutant_dna":40, "count_human_dna":100: "ratio":0.4 }
```
- **RF\_004:** Se deben guardar los ADN's verificados con la API.
- **RF\_005:** Test-Automáticos, Code coverage > 80%.

## 7. REGLAS DE NEGOCIO

- **RF\_002:**
  - Recibir 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 representa cada base nitrogenada del ADN.

	<b>Formato:</b> Documento Estructuración de la Iniciativa		<b>Código:</b> 001
	<b>PROCESO:</b> Gestión de Requerimientos		
	<b>Subproceso:</b> Gestión de la Demanda	<b>Emitido:</b> 17/08/2021	<b>Actualizado:</b>

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

- **RF\_003:** Solo 1 registro por ADN.

## 8. BENEFICIOS ESPERADOS

Desarrolla el algoritmo de la manera más eficiente posible.

## 9. IMPLEMENTACION

### 9.1. DIAGRAMAS

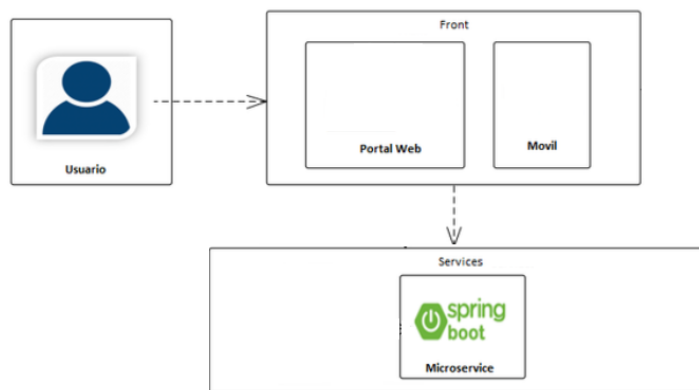


Diagrama de componente.

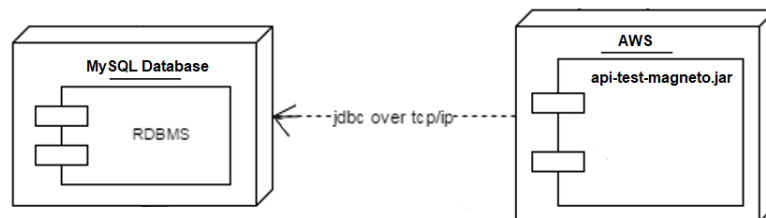



Diagrama de despliegue.

	<b>Formato:</b> Documento Estructuración de la Iniciativa		<b>Código:</b> 001
	<b>PROCESO:</b> Gestión de Requerimientos		
	<b>Subproceso:</b> Gestión de la Demanda	<b>Emitido:</b> 17/08/2021	<b>Actualizado:</b>

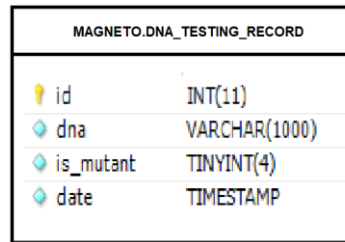


Diagrama de base de datos.

## 9.2. ILUSTRACIONES:

Para cumplir con el requerimiento funcional (RF\_001) y entender los criterios de aceptación y las reglas del negocio, se debe interpretar la siguiente ilustración:

A	T	G	C	G	A	A	T	G	C	G	A
C	A	G	T	G	C	C	A	G	T	G	C
T	T	A	T	T	T	T	T	A	T	G	T
A	G	A	C	G	G	A	G	A	A	G	G
G	C	G	T	C	A	C	C	C	C	T	A
T	C	A	C	T	G	T	C	A	C	T	G


No-Mutante

Mutante

## 9.3. RAZONAMIENTO:


Para cumplir con el objetivo el requerimiento, se plantea la siguiente implementación:

- Crear un nuevo repositorio en github.
- Crear un microservicio en el lenguaje JAVA utilizando como Framework Spring Boot.
- Crear un RDS en Amazon Web Services con motor de base de datos MySQL que debe ser configurado como lo muestra el Diagrama de base de datos y además crear los siguiente SP:
  - sp\_insert\_dna\_testing\_record (IN dna VARCHAR(1000), IN isMutant TINYINT).

	<b>Formato:</b> Documento Estructuración de la Iniciativa		<b>Código:</b> 001
	<b>PROCESO:</b> Gestión de Requerimientos		
	<b>Subproceso:</b> Gestión de la Demanda	<b>Emitido:</b> 17/08/2021	<b>Actualizado:</b>

- sp\_select\_dna\_testing\_record().
- Crear el Api y el Controller con dos métodos que contengan las siguientes declaraciones:
  - **Mutant:**
    - RequestMapping = "/mutant".
    - Method = POST.
    - RequestBody.
    - ResponseEntity.
  - **Stats:**
    - RequestMapping = "/stats".
    - method = .GET.
    - ResponseEntity.
- Crear el Business de los métodos que consume el controller:
  - **Mutant:** En este métodos se debe cumplir con los criterios de aceptación y reglas de negocios del **RF\_002** consumiendo de los siguientes métodos:
    - boolean validate(String[] dna,String dnaLetters, String dnaSize).
    - char[][] convertArray(String[] str).
    - boolean isMutant(char[][]).
    - void insert(String dna, boolean isMutan).
  - **Stats:** En este métodos se debe cumplir con los criterios de aceptación y reglas de negocios del **RF\_003** consumiendo de los siguientes métodos:
    - object get().
- Crear los DTO request y response necesarios.
- Crear el DAO para insertar y consultar la información en la base de datos:
  - void insert(String dna, boolean isMutan): Debe ejecutar el SP sp\_insert\_dna\_testing\_record.



	<b>Formato:</b> Documento Estructuración de la Iniciativa		<b>Código:</b> 001
	<b>PROCESO:</b> Gestión de Requerimientos		
	<b>Subproceso:</b> Gestión de la Demanda	<b>Emitido:</b> 17/08/2021	<b>Actualizado:</b>

- object get(): Debe ejecutar el SP sp\_select\_dna\_testing\_record(?,?).
- Crear una clase ProcessChain donde se realice la implementación de los métodos:
  - boolean validate(String[] dna,String dnaLetters, String dnaSize): Devolverá verdadero siempre y cuando el arreglo de String no sea NULL, no este vacío, que su tamaño no supere el valor parametrizado y que solo permita las base nitrogenada también parametrizadas en el sistema.
  - char[][] convertArray(String[] str): Devolverá una matriz de (NxN) donde las filas son el numero de datos en el arreglo de String y las columna la longitud del String.
  - boolean isMutant(char[][]): Devolverá verdadero si al recorrer la matriz de forma oblicua, horizontal o vertical cumple la condición de una secuencia de cuatro letras iguales.
- Crear un Elastic Beanstalk en Amazon Web Services donde se debe subir el proyecto .jar

## 10. REQUERIMIENTOS NO FUNCIONALES

- Latencia
- Escalabilidad
- Alta disponibilidad
- Confiabilidad

## 11. ENTREGABLES

- Código fuente en el repositorio github.
- Instrucciones de cómo ejecutar el programa o la API en README de github.
- URL de la API
- DDL o DML necesarios.