**1er Caso de Negocio: Un negocio de alquilar de películas en línea.**

**Microservicio: movie-rental-core-mysql-ms (alquiler)**

SGBD: MySql

BD: movie\_rental, movie\_rental\_replica

Tablas:

|  |  |
| --- | --- |
| rental | |
| id | (PK) Identificador único de alquiler. Es autogenerado. |
| member\_id | Id de socio. No es FK pues las tablas están en distintas BD. |
| movie\_id | (FK) Id de película |
| rental\_date | Fecha de alquiler. |
| return\_date | Fecha de devolución. |

|  |  |
| --- | --- |
| movie | |
| id | (PK) Identificador único de la Película. Es autogenerado. |
| title | Nombre de la película. |
| release\_date | Fecha de estreno. |
| duration\_minutes | Tiempo en minutos. |
| genre\_id | (FK) Id de Genero. |

|  |  |
| --- | --- |
| genre | |
| id | (PK) Identificador único de Genero. Es autogenerado. |
| name | Nombre de género. |

**Microservicio: movie-rental-membership-h2-ms (socios)**

SGBD: H2

BD: movie\_rental

Tablas:

|  |  |
| --- | --- |
| Member | |
| id | (PK) Identificador único del Socio. No es autogenerado. |
| name | Nombre del socio. |
| email | Email del socio. |
| password | Contraseña del socio. |
| active | Estado del socio. |

**Diagrama de Microservicios y Bases de Datos**

**Diagrama

Descripción generada automáticamente**

**Procedimiento para armar el ambiente**

1. BD

1.1. MySql:

Crear la BD: **movie\_rental** y **movie\_rental\_replica**.

1.2. H2: El proyecto creará la base de datos **movie\_rental** de manera automática.

1. Microservicios

2.1. Levantar los 4 proyectos con IntelliJ IDEA:

* + - eureka-server
    - movie-rental-membership-h2-ms
    - movie-rental-core-mysql-ms
    - gateway-server

2.2. Ejecutar el proyecto: eureka-server

2.3. Ejecutar el proyecto: movie-rental-membership-h2-ms

2.4. Ejecutar el proyecto: movie-rental-core-mysql-ms

2.5. Ir a la opción Run –> Edit Configurations.

Volver e ejecutar el proyecto movie-rental-core-mysql-ms, agregando la opción “Allow multiple instances” y agregando en “Add VM options” lo siguiente:

**-Dserver.port=8071 -Dspring.datasource.url=jdbc:mysql://localhost:3306/movie\_rental\_replica?useSSL=false&serverTimezone=UTC**

2.6. Ejecutar el proyecto: gateway-server

**Validación de microservicios registrados en Eureka Server**

<http://localhost:8761/>

**Interfaz de usuario gráfica, Tabla, Sitio web

Descripción generada automáticamente**

**Pruebas en Postman**

<http://localhost:8070/movie-rental/rent>

**Captura de pantalla de un celular

Descripción generada automáticamente**

<http://localhost:8071/movie-rental/rent>

**Texto

Descripción generada automáticamente**

**Usando el Gateway**

http://localhost:8060/api/movie-rental-service/movie-rental/rent

**Captura de pantalla de un celular

Descripción generada automáticamente**

http://localhost:8060/api/movie-rental-membership-service/members/1

**Captura de pantalla de un celular

Descripción generada automáticamente**

**2do Caso de Negocio: Cambiar de estado de socios de activo a inactivo, y, grabar el historial de cambios de un estado a otro.**

**Microservicio: membership-mysql-ms (socios)**

SGBD: MySql

BD: membership, membership\_replica

Tablas:

|  |  |
| --- | --- |
| audit\_log | |
| id | (PK) Identificador único de AuditLog. Es autogenerado. |
| member\_id | (FK) Id de socio. |
| oldStatus | Estado anterior. |
| newStatus | Estado nuevo. |
| changeDate | Fecha de cambio de Estado. |

|  |  |
| --- | --- |
| member | |
| id | (PK) Identificador único del Socio. |
| dni | Documento de identidad del socio. |
| first\_name | Nombre del socio. |
| last\_name | Apellidos del socio. |
| email | Correo del socio. |
| phone\_number | Teléfono del socio. |
| membership\_category\_id | Categoría del socio. |
| status | Estado del socio. |

|  |  |
| --- | --- |
| membership\_category | |
| id | (PK) Identificador único de categoría de socio. Es autogenerado. |
| category\_name | Nombre de categoría de socio. |

**Diagrama de Microservicios y Bases de Datos**

**Diagrama

Descripción generada automáticamente**

**Procedimiento para armar el ambiente**

1. BD

1.1. MySql:

Crear las BD: **membership** y **membership\_replica**.

1. Microservicios

2.1. Levantar los 4 proyectos con IntelliJ IDEA:

* + - eureka-server
    - membership-mysql-ms
    - gateway-server

2.2. Ejecutar el proyecto: eureka-server

2.3. Ejecutar el proyecto: membership-mysql-ms

2.4. Ir a la opción Run –> Edit Configurations.

Volver e ejecutar el proyecto membership-mysql-ms, agregando la opción “Allow multiple instances” y agregando en “Add VM options” lo siguiente:

**-Dserver.port=8096 -Dspring.datasource.url=jdbc:mysql://localhost:3306/membership\_replica?useSSL=false&serverTimezone=UTC**

2.5. Ejecutar el proyecto: gateway-server

**Validación de microservicios registrados en Eureka Server**

<http://localhost:8761/>

**Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente**

**Pruebas en Postman**

<http://localhost:8095/members/change-status>

**Captura de pantalla de un celular

Descripción generada automáticamente**

<http://localhost:8096/members/change-status>

**Captura de pantalla de un celular

Descripción generada automáticamente**

**Usando el Gateway**

<http://localhost:8060/api/membership-service/members/change-status>

**Captura de pantalla de un celular

Descripción generada automáticamente**

**Verificación en tabla audit\_log (histórico de cambios)**

**Imagen que contiene Texto

Descripción generada automáticamente**

**3er Caso de Negocio: Grabar datos de fotografías y la misma foto.**

**Microservicio: fotos\_mongo**

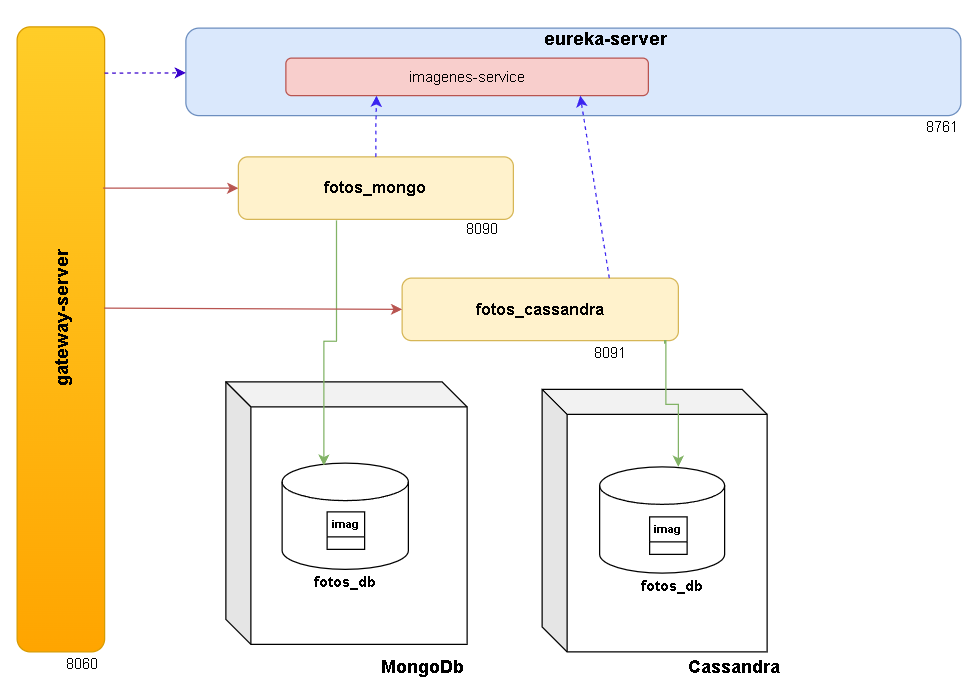
SGBD: MongoDB

BD: fotos\_db

Tablas:

|  |  |
| --- | --- |
| imagenes | |
| id | (PK) Identificador único de imagen. Es autogenerado. |
| nombre | Nombre de la foto |
| contenido | Datos binarios al crear una foto |

**Diagrama de Microservicios y Bases de Datos**



**Procedimiento para armar el ambiente**

1. BD

1.1. MongoDB:

Crear las BD: **fotos\_db**

1. Microservicios

2.1. Levantar los 2 proyectos con IntelliJ IDEA:

* + - eureka-server
    - fotos\_mongo
    - gateway-server

2.2. Ejecutar el proyecto: eureka-server

2.3. Ejecutar el proyecto: fotos\_mongo

2.4. Ejecutar el proyecto: gateway-server

**Validación de microservicios registrados en Eureka Server**

<http://localhost:8761/>

**Interfaz de usuario gráfica

Descripción generada automáticamente**

**Pruebas en Postman**

**Grabar una foto**

http:// localhost:8090/api/imagenes

**Captura de pantalla de computadora

Descripción generada automáticamente**

**Obtener una foto**

[http:// localhost:8090/api/imagenes/65776ce4936b6d5e750fb157](http://localhost:8096/members/change-status)

**Captura de pantalla de computadora

Descripción generada automáticamente**

**Usando el Gateway**

<http://localhost:8060/api/images-service/api/imagenes>

**Captura de pantalla de un celular

Descripción generada automáticamente**

<http://localhost:8060/api/images-service/api/imagenes/6577ab99a1e1811648a89929>

**Interfaz de usuario gráfica

Descripción generada automáticamente**