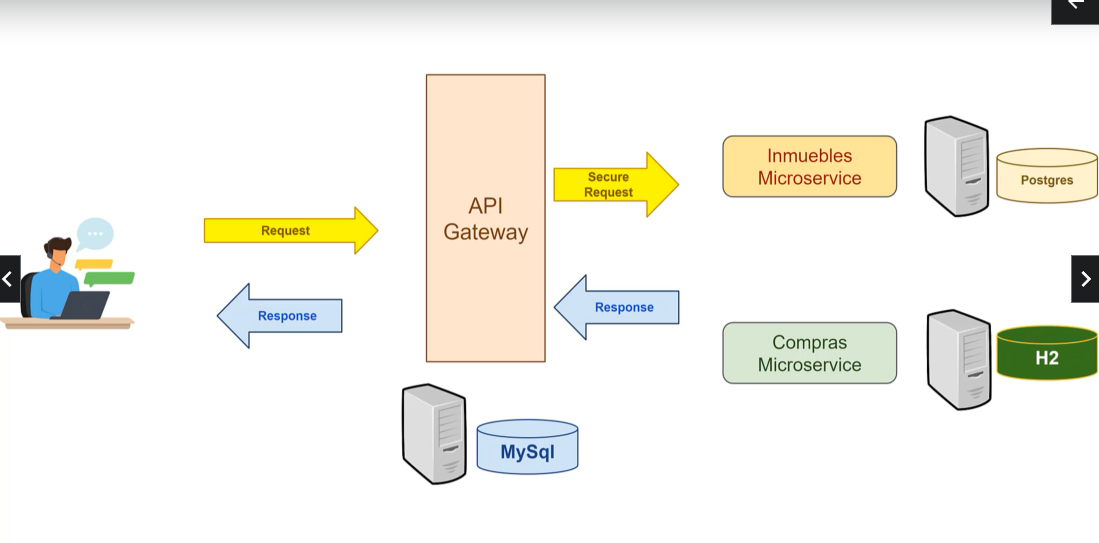
9. API Gateway en Mircorservices

API Gateway

Este sera el interemedario en todas la peticiones que envia el cliente y las microservices del BACKEN, como vemos cada macro service esta todo independiente DB, PUERTO, AUTORIZACION, etc.

Un microservicio se encarga del mantenimiento de inmuebles y el otro encargado de las compras de inmuebles.

Por ende toda la información dispersa procedemos a crear un API GATEWAY(Nuevo Microservicio) para que compacte toda la arquitectura.



Para este proyecto emplearemos un nuevo macroservice llamado API GATEWAY que contendrá todo el tema disperso para conexión de los otros services como DB, SEGURIDAD, PUERTO y también contendrá un DB Mysql dentro de un Container para que almacene y tenga el mantenimiento de todos los usuarios en nuestra app

Crear Servidor de Base de Datos Gateway

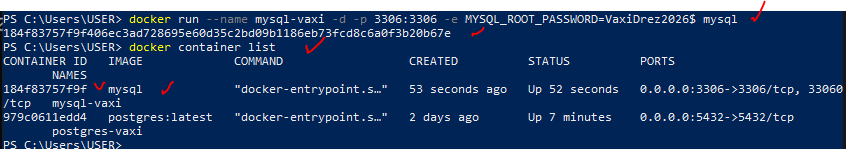
<https://hub.docker.com/>

<https://hub.docker.com/_/mysql>

Y procedo a crear el container desde el PowerSh para la DB luego de tener la imagen descargada de MySql

run --name mysql-vaxi -d -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=VaxiDrez2026$ mysql

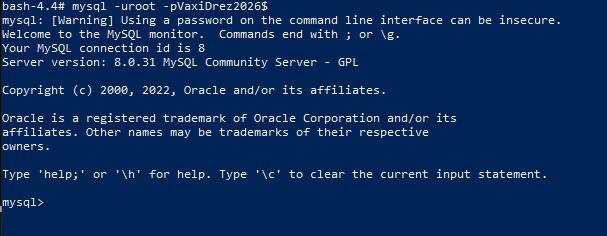
y verifico la creacion del container docker container list



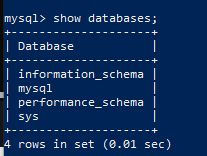
Ingreso al sh del container y luego al sh de la DB



Para ingresar al sh del la DB debo agrega el servidor de DB que instalamos mysql el usuario -u y el password -p



Para listar las tablas de la DB show databases;



Crear una Base de datos create database db\_vaxidrez; y procedo a ver la database y allí esta

Para este proyecto vamos a crear la DB desde spingBoot ojo pero si debemos crear el usuario que va tener permisos para hacer la tarea desde el proyecto.

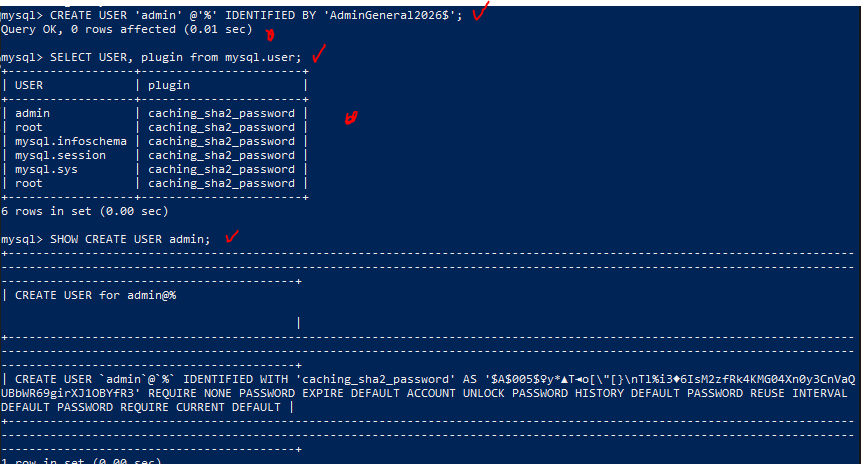
CREATE USER 'admin' @'%' IDENTIFIED BY 'AdminGeneral2026$'; el arroba y el complemento es para indicar desde que tipo se va hacer la conexión si desde cualquier computadora o servidor; el identified es elñ password para yo acceder a este usuario dede mi proyecto.

Verificando el usurio

SELECT USER, plugin from mysql.user;

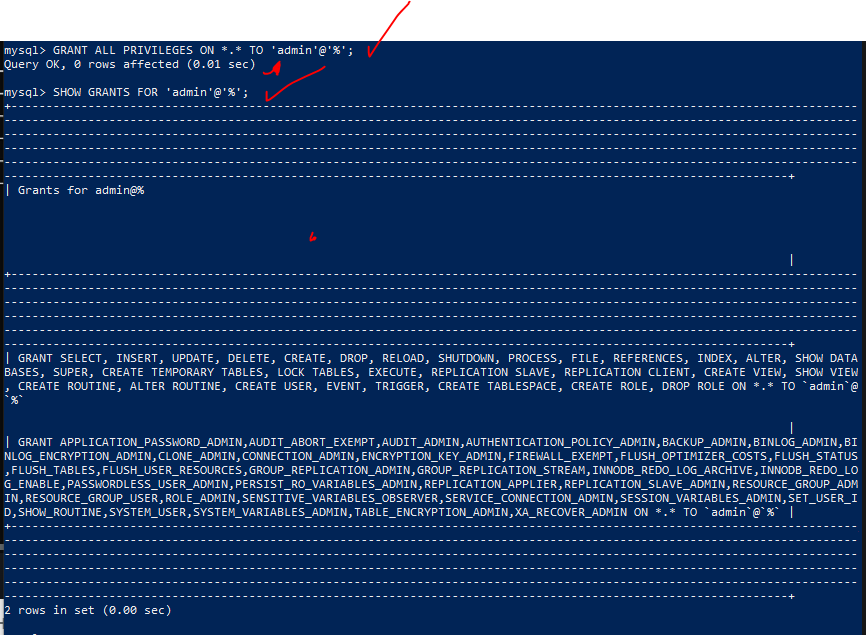
Si quiero ver cuales comando se ejecutaron para hacer efectiva la creación de admin ejecuto

SHOW CREATE USER admin;



Ahora procedo a darle permisos a este usuario

GRANT ALL PRIVILEGES ON \*.\* TO 'admin'@'%'; los asteriscos significan que le estamos dando todos los permisos a este usuario admin y procedo a ver que los permisos se hayan otorgado SHOW GRANTS FOR 'admin'@'%';

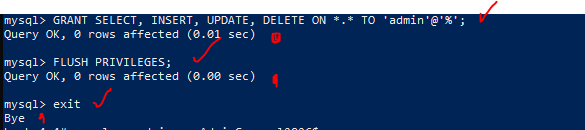


También le dare permiso a los otros usuarios par que puedan hacer CRUD GRANT SELECT, INSERT, UPDATE, DELETE ON \*.\* TO 'admin'@'%';

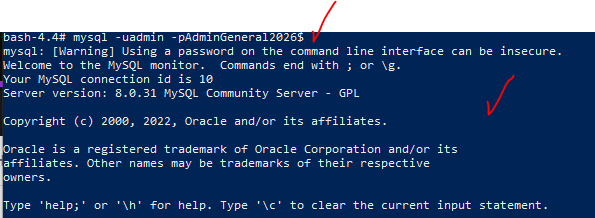
Luego realizo un fluzhz para que todos los cambios se lleven a cambion en el servidor de DB

FLUSH PRIVILEGES;

Procedo a salir del usuario root para entrar al usuario admin y validar si ya tengo todos los permisos otorgados.



Entrando a admin



Listo como tal ya esta OK el el usuario que va hacer la conexión y administración de la DB con el microservice de GATEWAY

Creacion de API Gateway

Importante agregar la dependencia de OpenFeign permite enlazarme a los enpoint de los microservices de compras e inmuebles



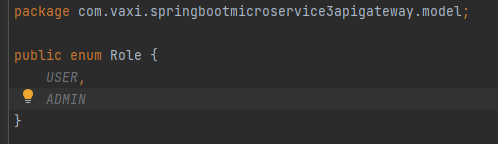
Luego configuro el proyecto como habitual

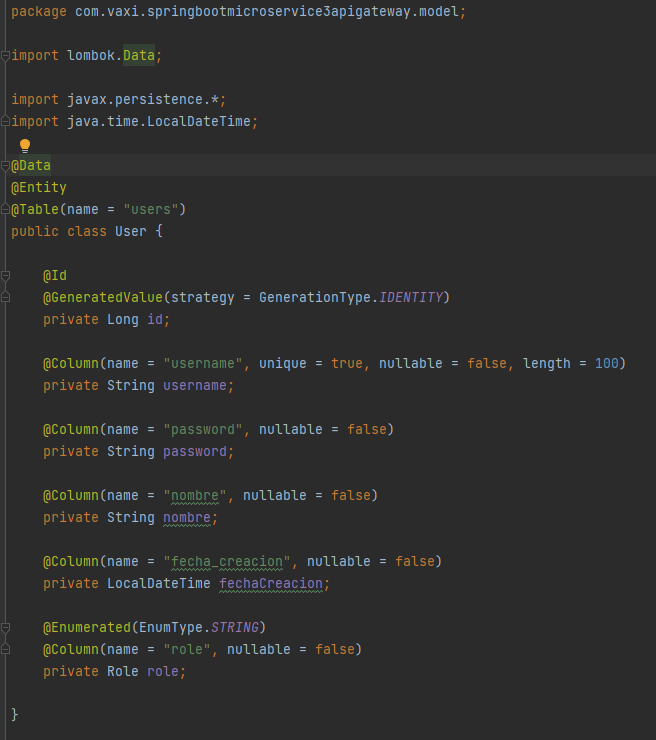
Configuracion de Microservice

Configurando el archivo app.properties

 Creacion de Entity User

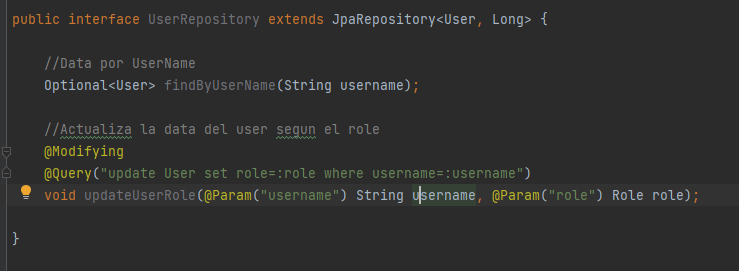
Importante en esta clase implementamos una clase de ENUM extra que contrandra los ROLES string para ser llamado, se ve en el campo role





 Creacion de Repository User

Agrego dos métodos adicionales para hacer búsqueda de data por nombre y modificar data según el usuario y role



Security Service

Recordar que para hacer la isercion de usuarios y roles debo leer la password encriptada para ello creo un método desde el archivo SPRINGBOOTMICROSEVICE principal y como se hace ell llamado del microservice entonces la información vendría por medio del Bean

