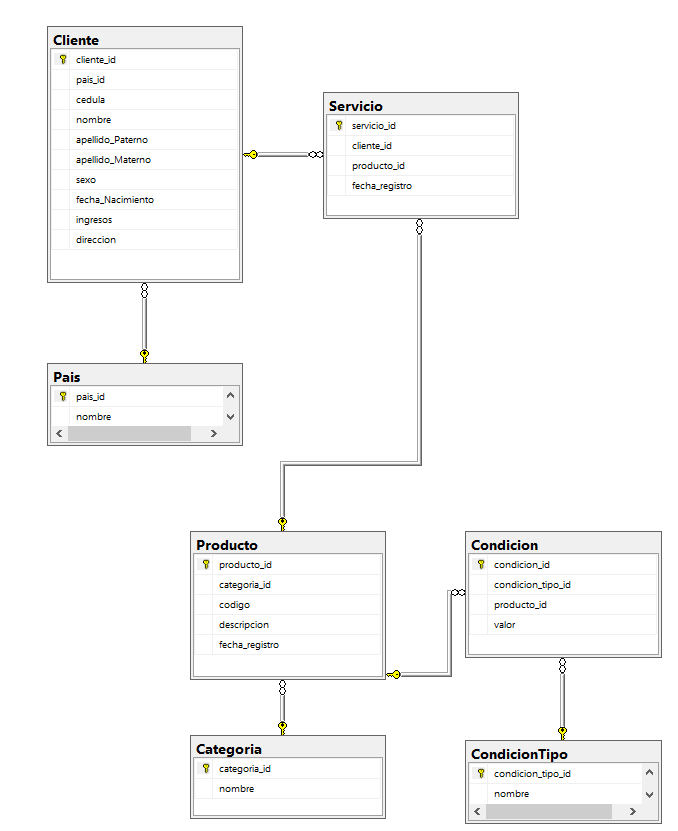
Proyecto:

<https://github.com/cesarpb80/testproject.git>

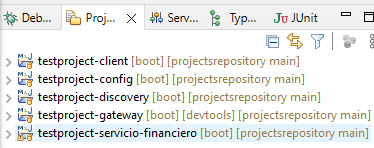
Análisis del proyecto (Esto solo como referencia, ya que se implementó bajo arquitectura orientada a micro servicios, dentro de cada servicio esta implementada las tablas para cada uno y se utiliza como BD H2, en el repositorio se encuentra el script para generar todo la Base de datos (SqlServer)):



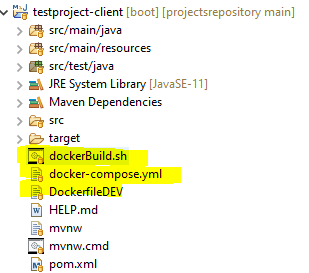
Servicios:

SpringBoot, SpringCloud, Eureka, Gateway y otras librerías de ayuda.

Java 11



Archivos Docker para la creación y despliegue de los servicios:



Comandos (dentro del directorio del proyecto):

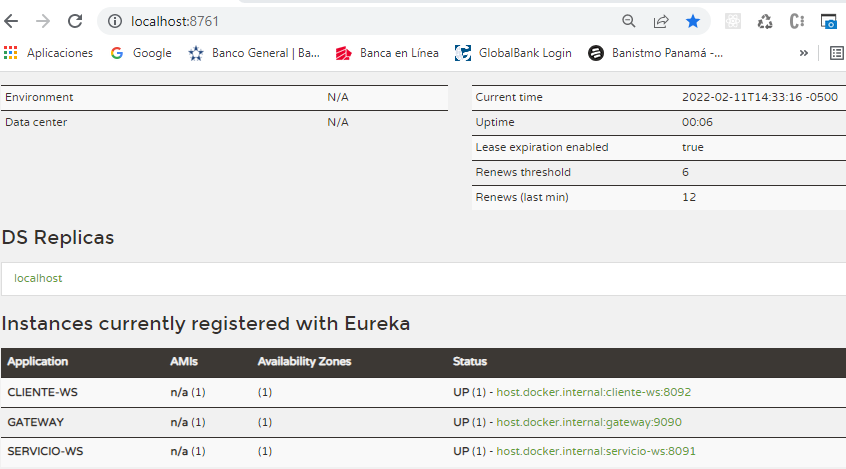
$ ./dockerBuild.sh -> Crea una imagen con los últimos cambios en docker

$ docker compose up –d -> Levantar el contenedor

$ docker compose down -> parar el contenedor

Servicio de Eureka para el registro de los servicios y la implementación del Gateway para la arquitectura basado en micro servicios:

Eureka levanta en el puerto: 8761 y el cual registra los servicios de Cliente, Servicio y Gateway



App:

Trabajado con React con témplate de NextJs.

Despliegue:

Visual Code (Dentro del proyecto en el terminal):

$ npm run dev

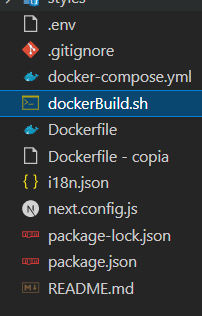
Contenedor:

Comandos (dentro del directorio del proyecto):

$ ./dockerBuild.sh -> Crea una imagen con los últimos cambios en docker

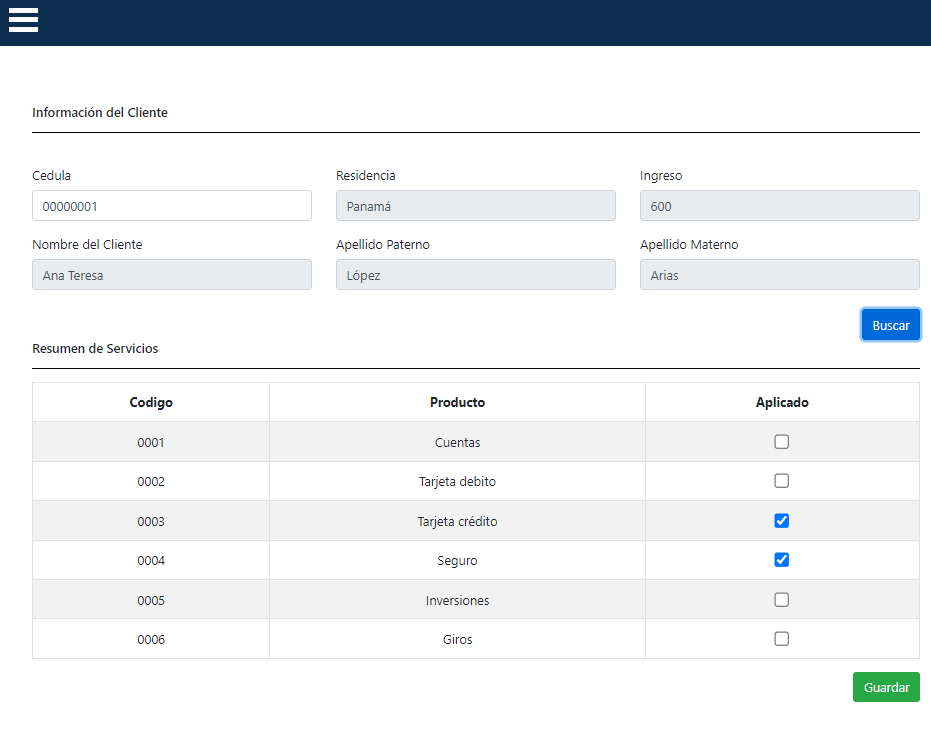
$ docker run -p 3000:3000 testproject-web -> ejecutar la app

$ docker compose down -> parar el contenedor



La aplicación tiene las operaciones básicas para el registro de clientes, productos y registro de servicios para los clientes:

Servicios (en el script de la Base de Datos se incluye el filtrado de productos por cliente, ya que por falta de tiempo no logre implementarlo):



Clientes:



Productos:

