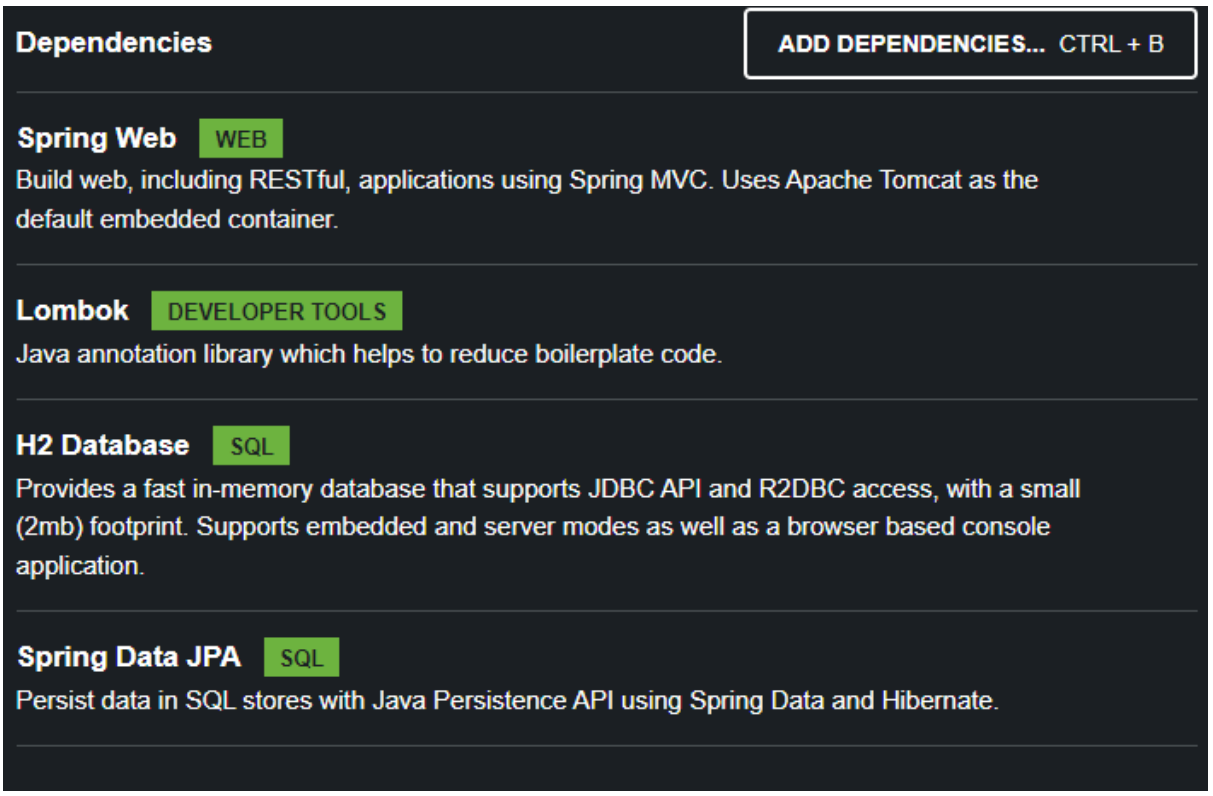
Prueba Técnica Innoqa

Aspirante : Steven Piedra Garcia

1. Creación del proyecto

Se crea mediante “Spring initializr” el proyecto Spring bajo la arquitectura maven, y se añaden las siguientes dependencias:



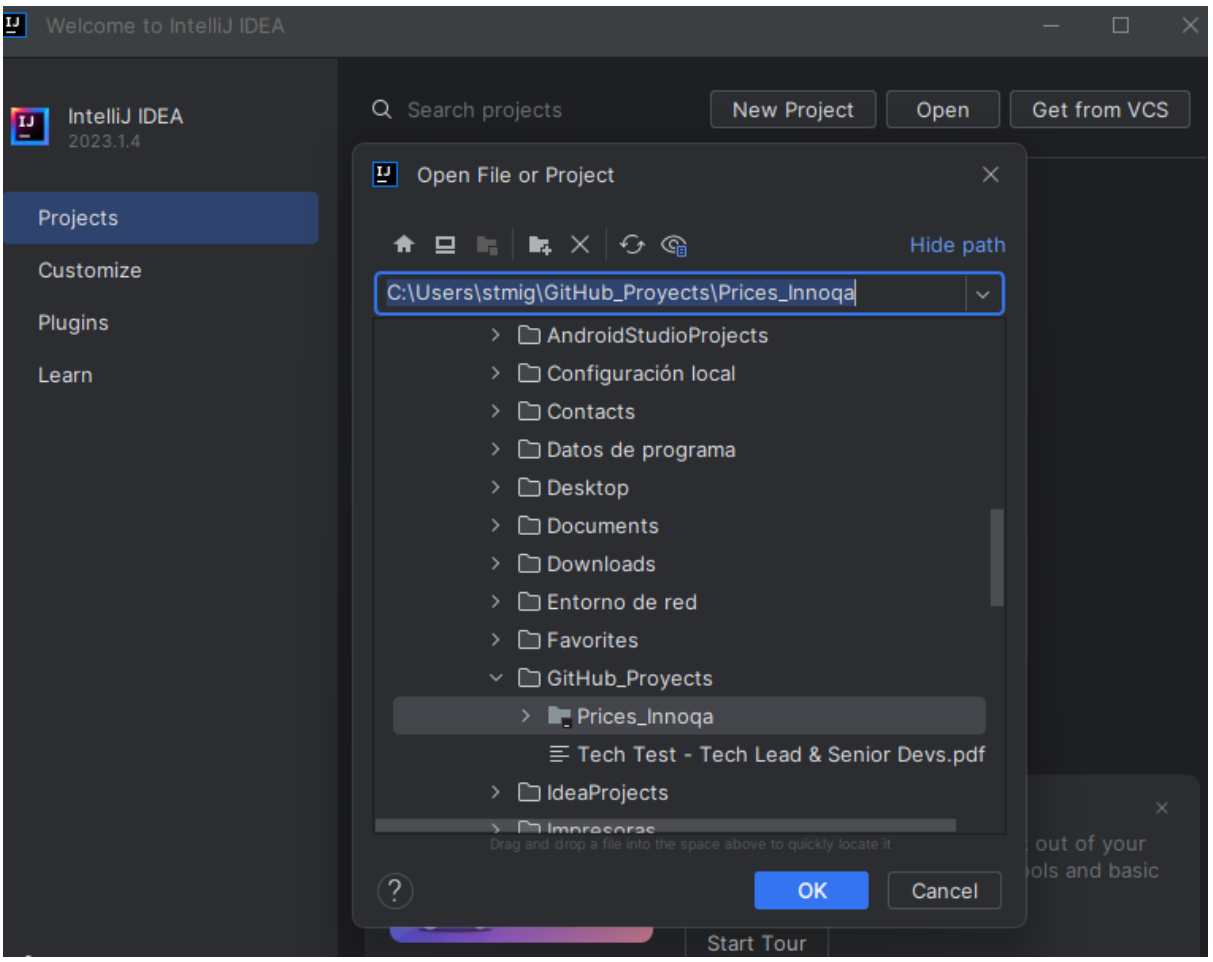
Además hemos añadido la dependencia de Junit para poder realizar el testing del servicio .

2.Importación del proyecto y Ejecución del programa

En este caso vamos a utilizar como entorno de desarrollo INTELLIJ IDEA.

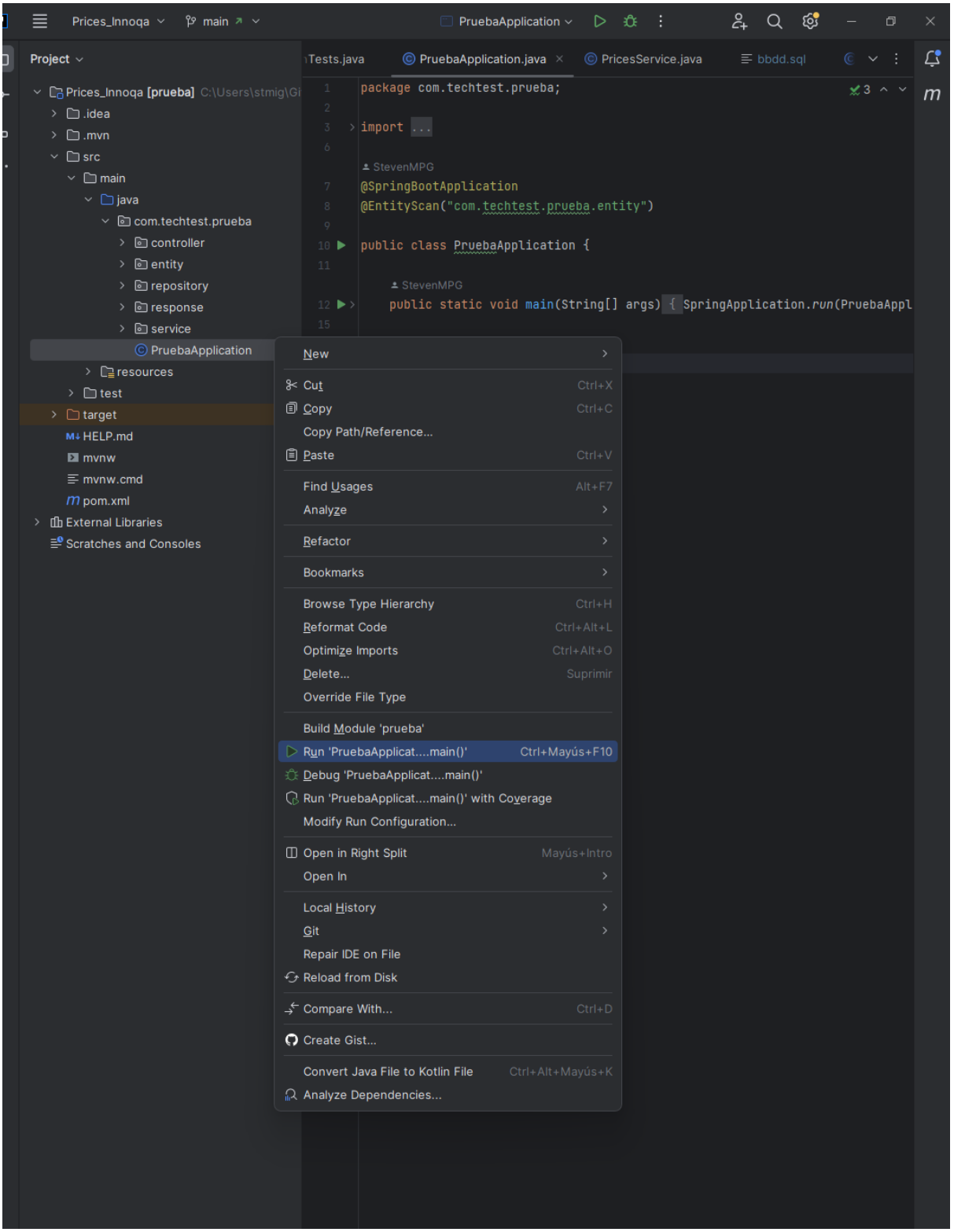
Para poder importar el proyecto lo primero que haremos será descargar o clonar el repositorio de github.

Tras haber descargado el proyecto lo importamos mediante el gestor de nuestro entorno de desarrollo INTELLIJ



Tras haber importado el proyecto dejamos que se carguen las dependencias correctamente y procedemos a ejecutarlo.

Para ejecutarlo clicamos en el botón de RUN o realizando el comando **Mayus+F10**, otra opción es la de , mientras nos encontramos en el archivo **“ PruebaApplication”**, desplegamos el menú de opciones y seleccionamos la opción de RUN


Prices_Innoqa
main
PruebaApplication
Project
Tests.java
PruebaApplication.java
PricesService.java
bbdd.sql
Prices_Innoqa [prueba] C:\Users\stmig\Gi
.idea
.mvn
src
main
java
com.techtest.prueba
controller
entity
repository
response
service
PruebaApplication
resources
test
target
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
package com.techtest.prueba;
import ...
StevenMPG
@SpringBootApplication
@EntityScan("com.techtest.prueba.entity")
public class PruebaApplication {
StevenMPG
public static void main(String[] args) { SpringApplication.run(PruebaAppl
New
Cut Ctrl+X
Copy Ctrl+C
Copy Path/Reference...
Paste Ctrl+V
Find Usages Alt+F7
Analyze
Refactor
Bookmarks
Browse Type Hierarchy Ctrl+H
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Suprimir
Override File Type
Build Module 'prueba'
Run 'PruebaApplicat....main()' Ctrl+Mayús+F10
Debug 'PruebaApplicat....main()'
Run 'PruebaApplicat....main()' with Coverage
Modify Run Configuration...
Open in Right Split Mayús+Intro
Open In
Local History
Git
Repair IDE on File
Reload from Disk
Compare With... Ctrl+D
Create Gist...
Convert Java File to Kotlin File Ctrl+Alt+Mayús+K
Analyze Dependencies...

3. Gestion y configuracion de base de datos H2

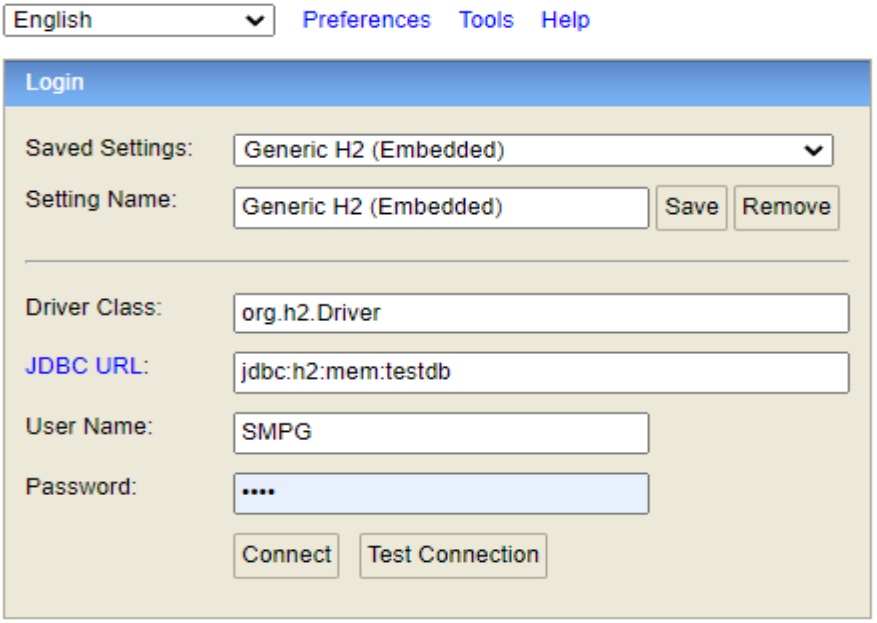
Se ha usado H2 como la base de datos en memoria y consola.
Tras ejecutar el proyecto nos dirigimos a nuestra url de nuestra base de datos H2 en mi caso se encuentra en el en LOCALHOST, en los puertos 8080.

-> http://localhost:8080/h2-console

JDBC URL = jdbc:h2:mem:testdb

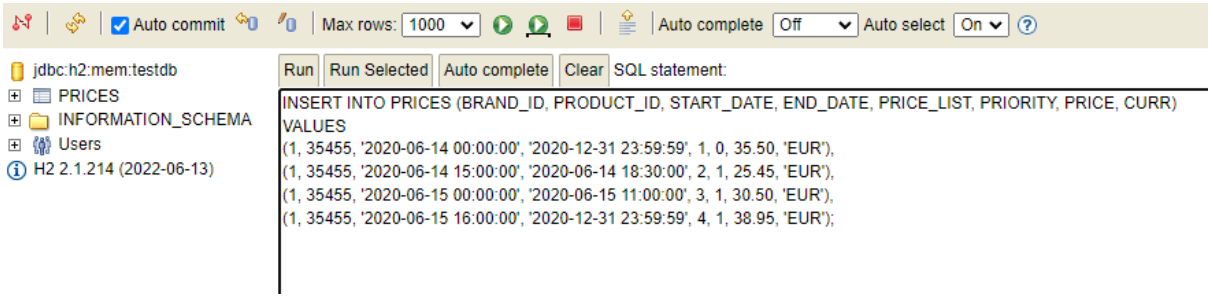
username=SMPG

password=asas



Conectamos con la base de datos y realizamos la siguiente consulta para poder introducir los datos en memoria y así poder realizar las diferentes peticiones

INSERT INTO PRICES (BRAND_ID, PRODUCT_ID, START_DATE, END_DATE, PRICE_LIST, PRIORITY, PRICE, CURR)
VALUES
(1, 35455, '2020-06-14 00:00:00', '2020-12-31 23:59:59', 1, 0, 35.50, 'EUR'),
(1, 35455, '2020-06-14 15:00:00', '2020-06-14 18:30:00', 2, 1, 25.45, 'EUR'),
(1, 35455, '2020-06-15 00:00:00', '2020-06-15 11:00:00', 3, 1, 30.50, 'EUR'),
(1, 35455, '2020-06-15 16:00:00', '2020-12-31 23:59:59', 4, 1, 38.95, 'EUR');



3.Comprobación.

Para este paso , vamos a utilizar el cliente de POSTMAN para poder realizar peticiones de prueba para ello utilizamos como endpoint la siguiente dirección.

http://localhost:8080/price?brandId=1&productId=35455&applyDate=2023-09-08%2021:00:00

Como se puede ver se estan utilizando como parámetros de entrada:

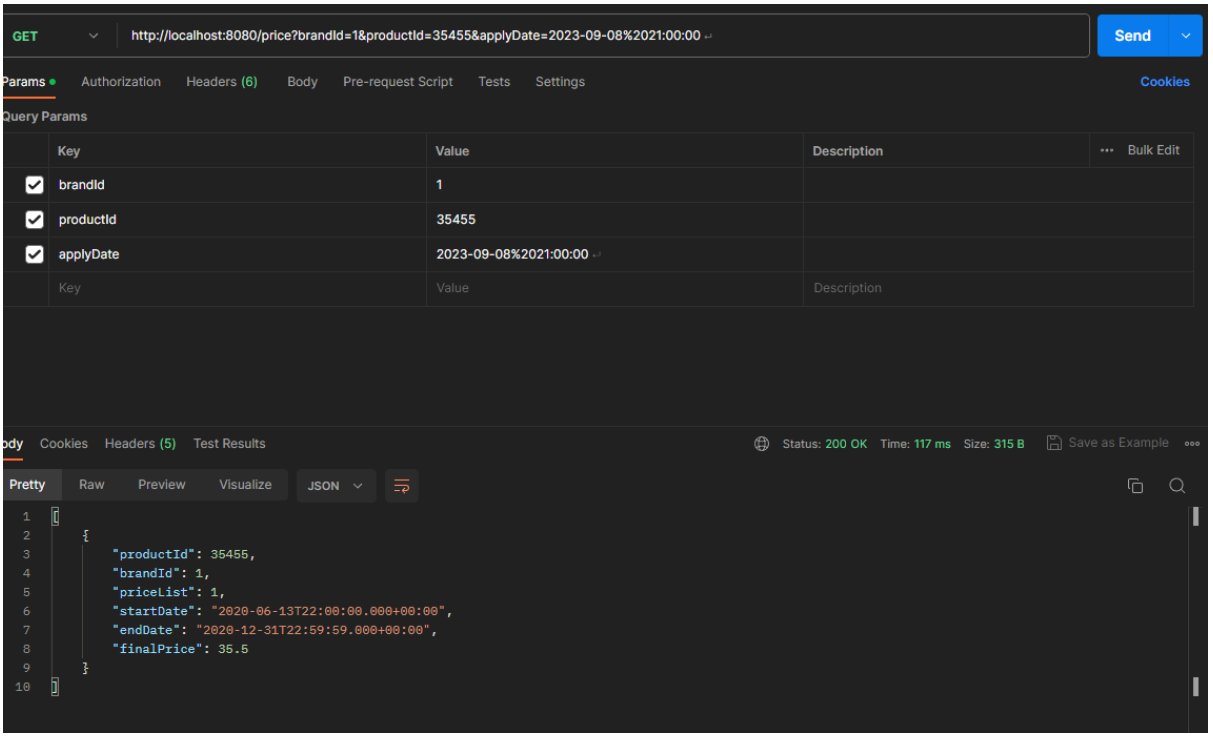
- **brandId:** El cual corresponde con el identificador de la marca.
- **productId** : El cual corresponde con el identificador del producto.
- **applyDate**: Es la fecha de consulta, en este caso la estamos tomado de los siguientes datos proporcionados para este caso de uso

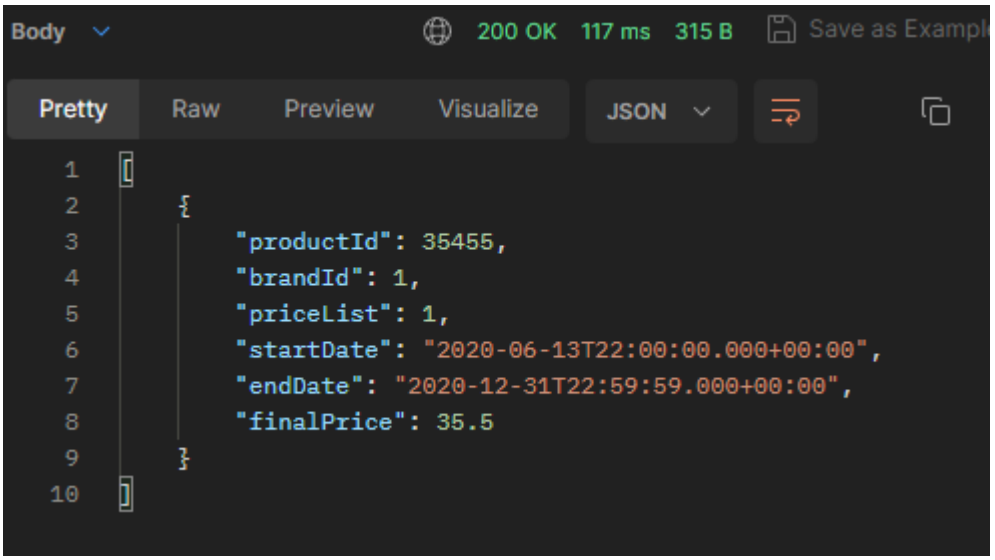
PRICES

BRAND_ID	START_DATE	END_DATE	PRICE_LIST	PRODUCT_ID	PRIORITY	PRICE	CURR
1	2020-06-14-00.00.00	2020-12-31-23.59.59	1	35455	0	35.50	EUR
1	2020-06-14-15.00.00	2020-06-14-18.30.00	2	35455	1	25.45	EUR
1	2020-06-15-00.00.00	2020-06-15-11.00.00	3	35455	1	30.50	EUR
1	2020-06-15-16.00.00	2020-12-31-23.59.59	4	35455	1	38.95	EUR

Ejecutamos la consulta haciendo click en el botón de **SEND** lo cual nos dará como resultado los parámetros de salida expuestos en la prueba que son :

- Identificador de producto,
- Identificador de cadena,
- Tarifa a aplicar
- Fechas de aplicación
- Precio final a aplicar.

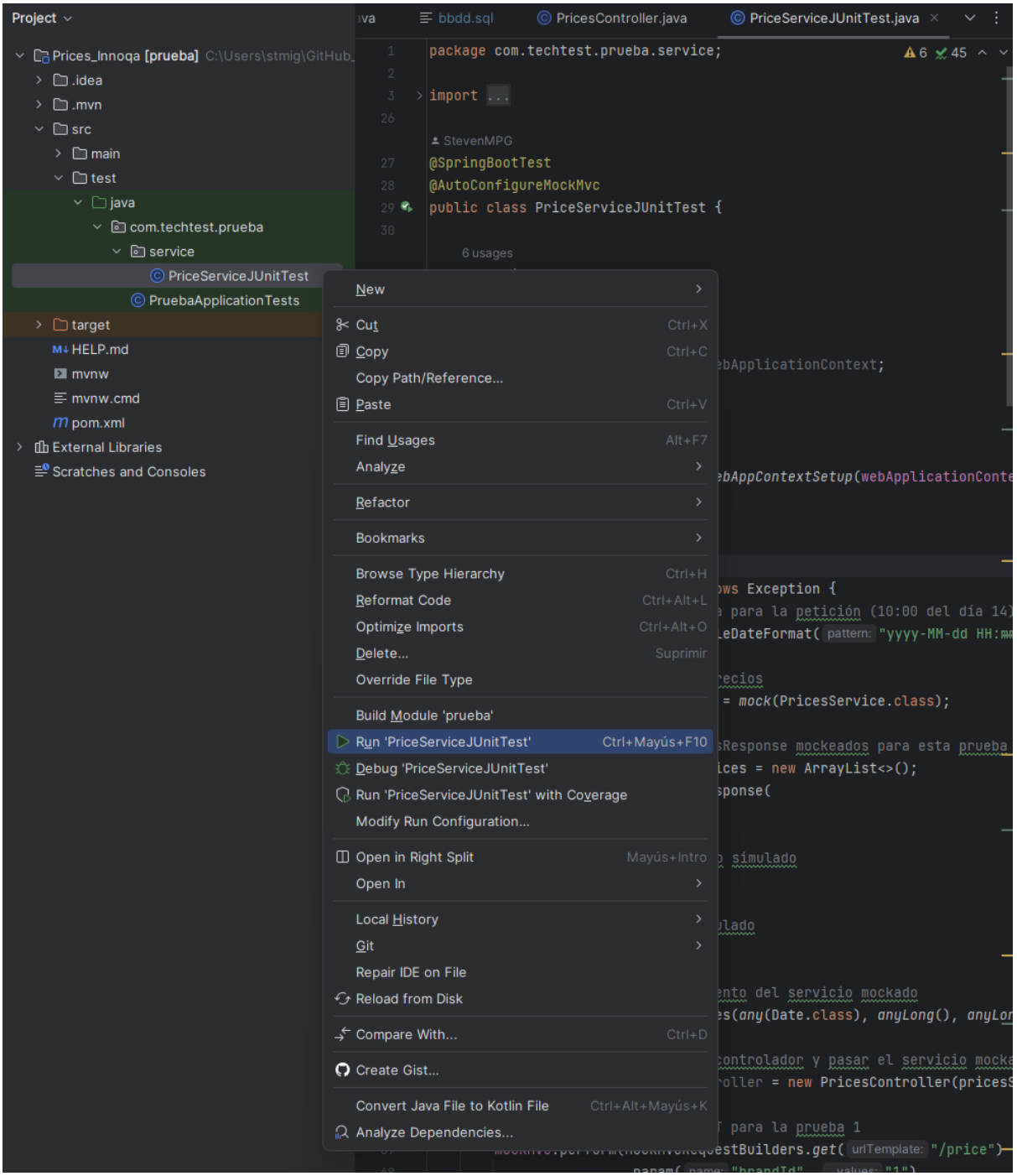




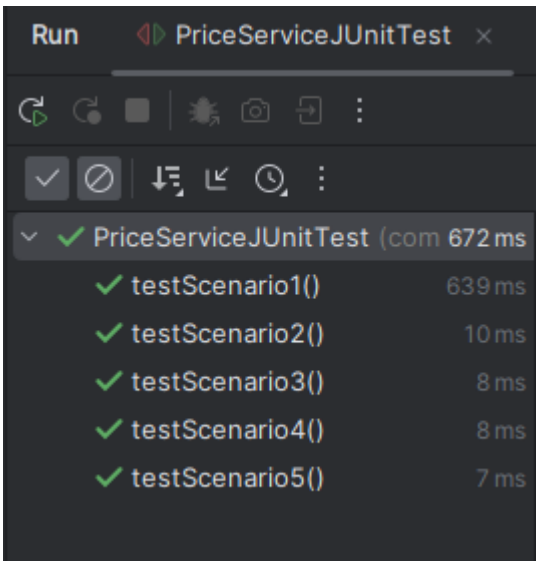
Con esto concluimos la primera parte de la prueba que consiste en devolver los datos que se piden mediante la consulta Get.

4.Test

Para la segunda parte de la prueba se pide realizar distintos escenarios de test para el servicio que hemos creado para poder ejecutar el test no dirigimos a la carpeta SRC del proyecto y desplegamos la carpeta TEST y nos dirigimos a la carpeta del servicio de los test y clicamos derecho encima de el archivo y ejecutamos el RUN



Como resultado nos aparecerá en consola si son correctos



Como se puede apreciar los distintos test que hemos creado para los distintos escenarios son correctos. con esto damos finalizado la prueba técnica de Innoqa.