Pruebas técnicas de ingreso Mainsoft

Presentado Por

Jaime Andrés Téllez Gonzalez

c.c. 80657092

para

mainsoft LTDA.

Bogotá D.C. 06 de noviembre de 2020

Tabla de Imágenes Pruebas Técnicas

Docker MySQL 1: instalación imagen MySQL 3

Docker MySQL 2: Imagen lista de imágenes Docker 3

Docker MySQL 3: Creación contenedor Data Base 3

Docker MySQL 4: Imagen de listado de imágenes 3

Docker MySQL 5: imagen de ejecución de MySQL en contenedor Docker 4

Docker MySQL 6: imagen creación de User con todos los permisos 4

Docker MySQL 7: imagen de creación por persistencia en Workbench 5

Docker MySQL 8:Imagen de validación de red 6

Docker MySQL 9: creación de nueva red Docker 6

Docker MySQL 10: Imagen confirmación creación de red 6

Docker MySql 11: Imagen desconectar y conectar a la nueva red 7

Docker MySql 12: Imagen confirmación de contenedor en red 7

Docker MySQL 13: Imagen confirmación config application.properties 7

Docker JAR 1: Imagen creación .JAR 9

Docker JAR 2: Imagen de la generación de la imagen Docker 9

Docker JAR 3: Imagen de validación de las imágenes en Docker 9

Docker JAR 4: Imagen de la creación de contenedor para la imagen .JAR 10

Docker JAR 5: Imagen de validación contenedores en Docker 10

Docker JAR 6: Imagen de la prueba de ejecución correcta del contenedor 10

Postman 1: Imagen prueba de acceso con nuevo puerto 18080 11

Postman 2: Imagen Prueba de lista clientes con puerto 18080 11

Postman 3: Imagen prueba lista productos puerto 18080 12

Postman 4: Imagen prueba listado de ventas puerto 18080 12

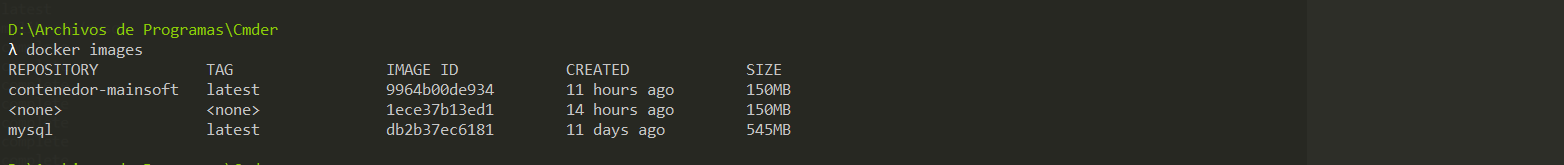
1. Instalación y descarga de imagen de mysql

Comando: docker pull mysql



Docker MySQL 1: instalación imagen MySQL

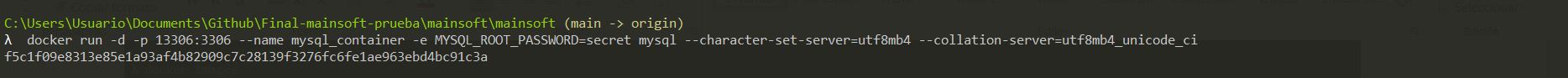
1. Confirmación de la creación de la imagen mysql



Docker MySQL 2: Imagen lista de imágenes Docker

1. Creación del contenedor donde estará alojada la base de datos

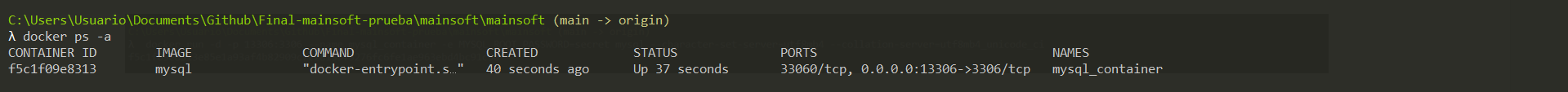
Comando: docker run -d -p 13306:3306 –name mysql\_container -e MYSQL\_ROOT\_PASSWORD=secret mysql --character-set-server=utf8mb4 --collation-server=utf8mb4\_unicode\_ci



Docker MySQL 3: Creación contenedor Data Base

1. Confirmación de la creación del contenedor MySQL

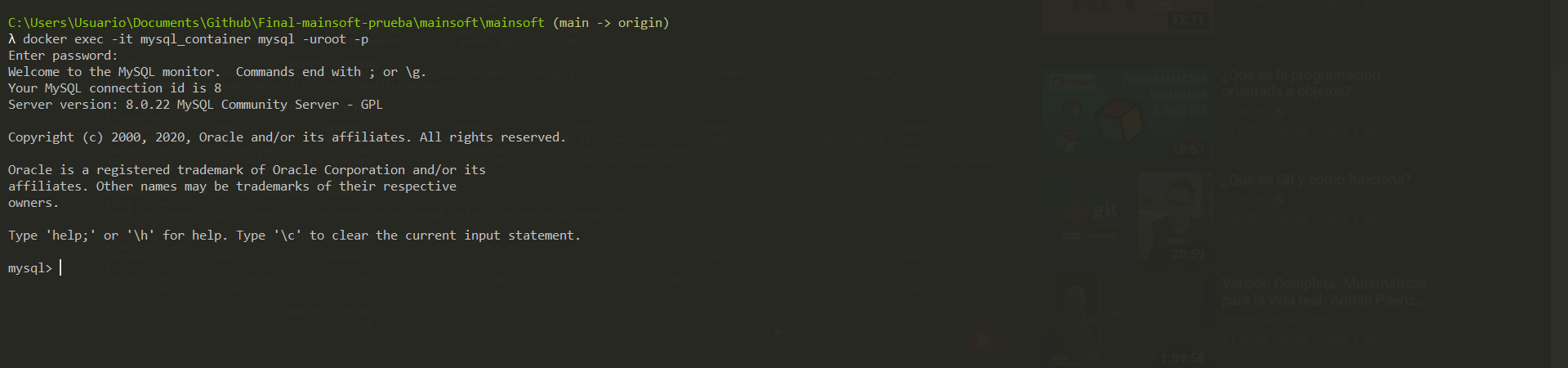
Comando: docker ps -a



Docker MySQL 4: Imagen de listado de imágenes

1. Ejecución de mysql dentro del contenedor

Comando: docker exec -it mysql\_container mysql -uroot -p

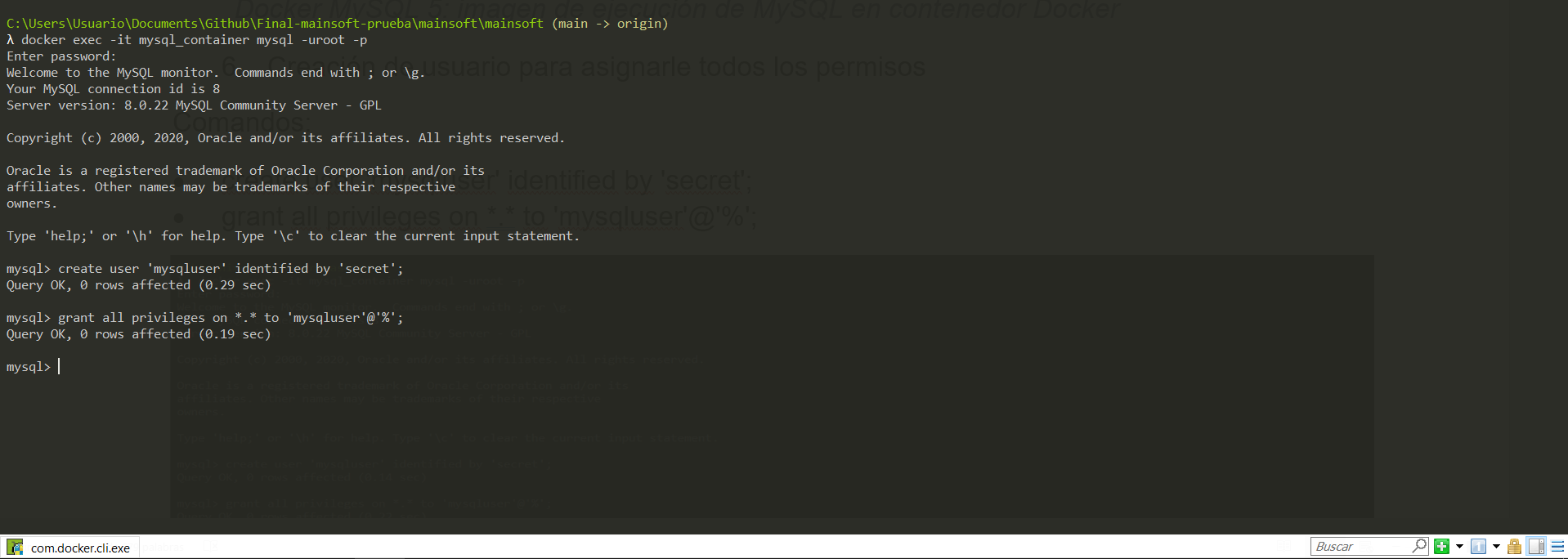


Docker MySQL 5: imagen de ejecución de MySQL en contenedor Docker

1. Creación de usuario para asignarle todos los permisos

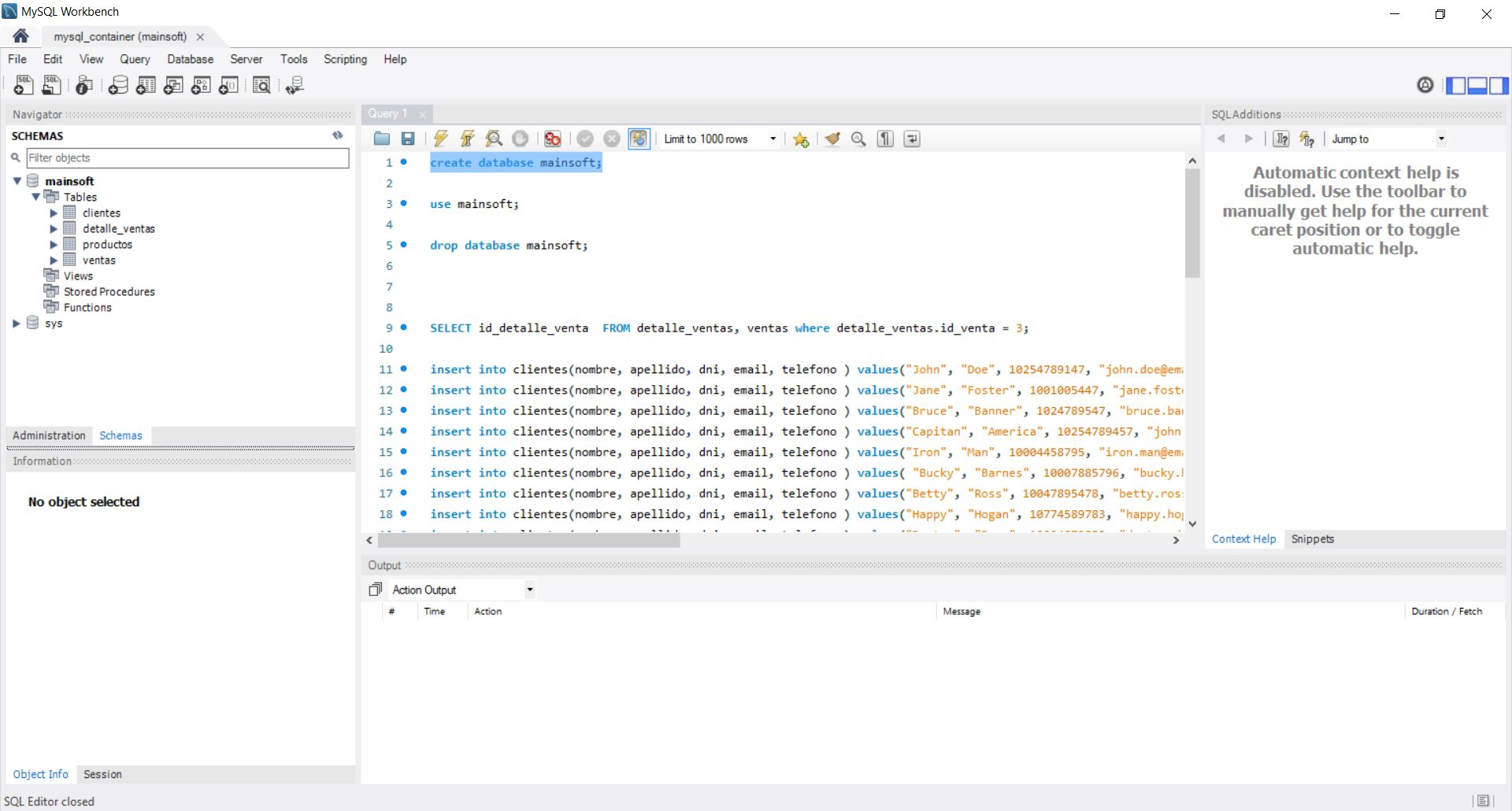
Comandos:

* create user 'mysqluser' identified by 'secret';
* grant all privileges on \*.\* to 'mysqluser'@'%';



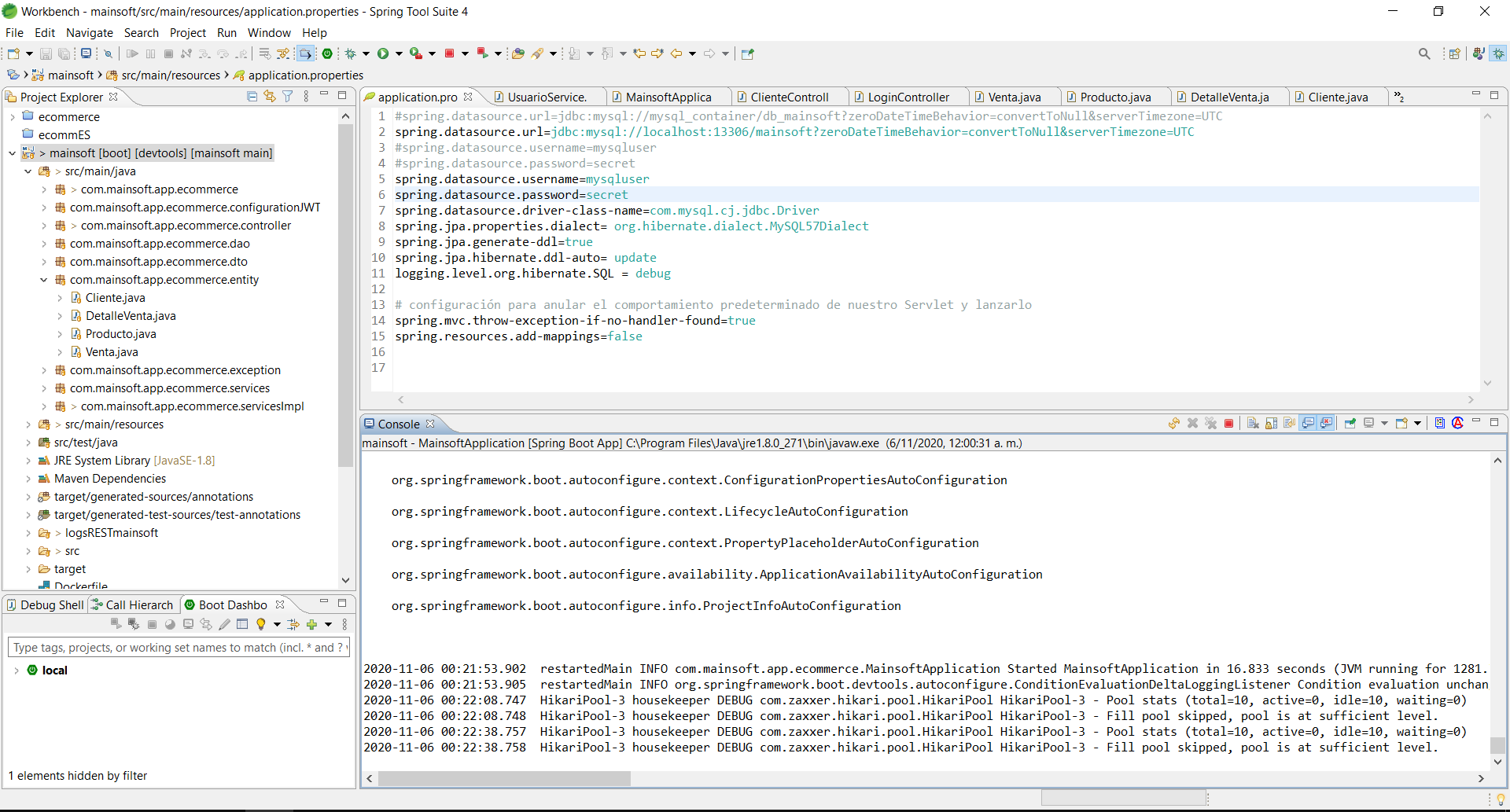
Docker MySQL 6: imagen creación de User con todos los permisos

1. Creación de la base de datos en el contenedor docker por medio de persistencia JPA desde Spring Boot con mysql Wordbrench



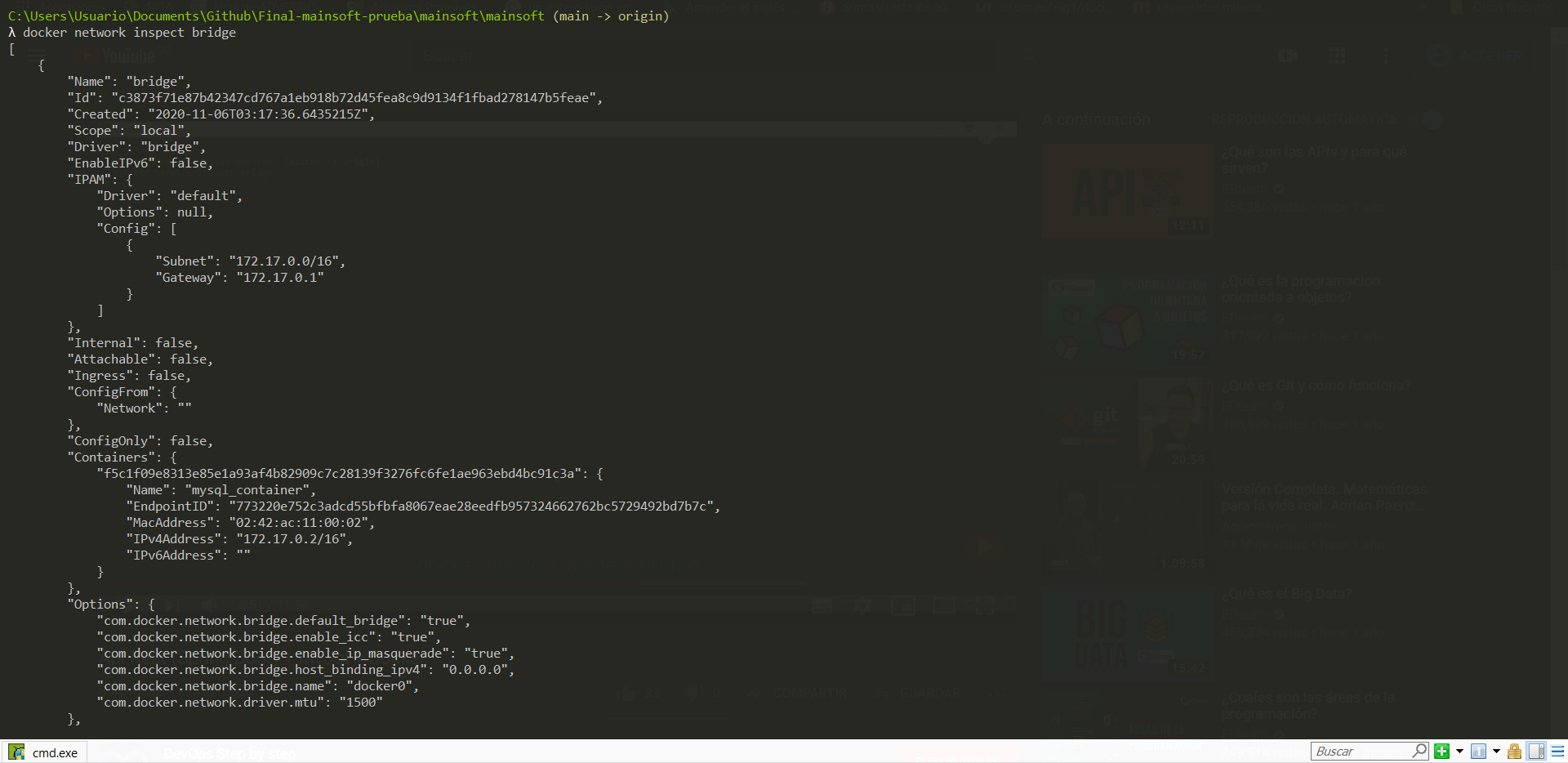
Docker MySQL 7: imagen de creación por persistencia en Workbench

1. Ejecución del aplicativo y direccionamiento del datasource a la base de datos del contenedor



1. Verificación de red en donde se creó el contenedor de la base de datos MySQL

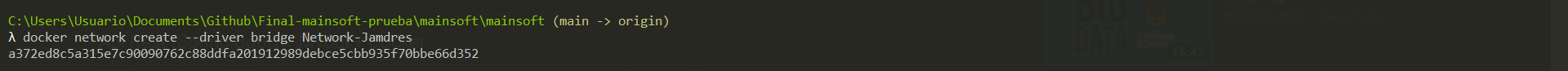
Comando: docker network inspect bridge



Docker MySQL 8:Imagen de validación de red

1. Creación de nueva red donde se alojarán los contenedores Docker de la API y la Data Base

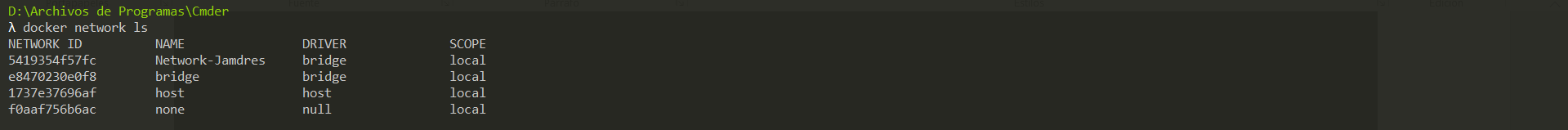
Comando: docker network create --driver bridge Network-Jamdres



Docker MySQL 9: creación de nueva red Docker

1. Confirmación creación nueva red Network-Jamdres

Comando: Docker network list



Docker MySQL 10: Imagen confirmación creación de red

1. Desconectar de la red bridge nuestro contenedor mysql\_container

Comando: docker network disconnect bridge mysql\_container

1. Apuntar nuestra red al contenedor mysql\_container

Comando: docker network connect Network-Jamdres mysql\_container

1. Verificamos que la nueva red “Network-Jamdres” esté apuntando a nuestro contenedor de base de datos mysql: “mysql\_container”.



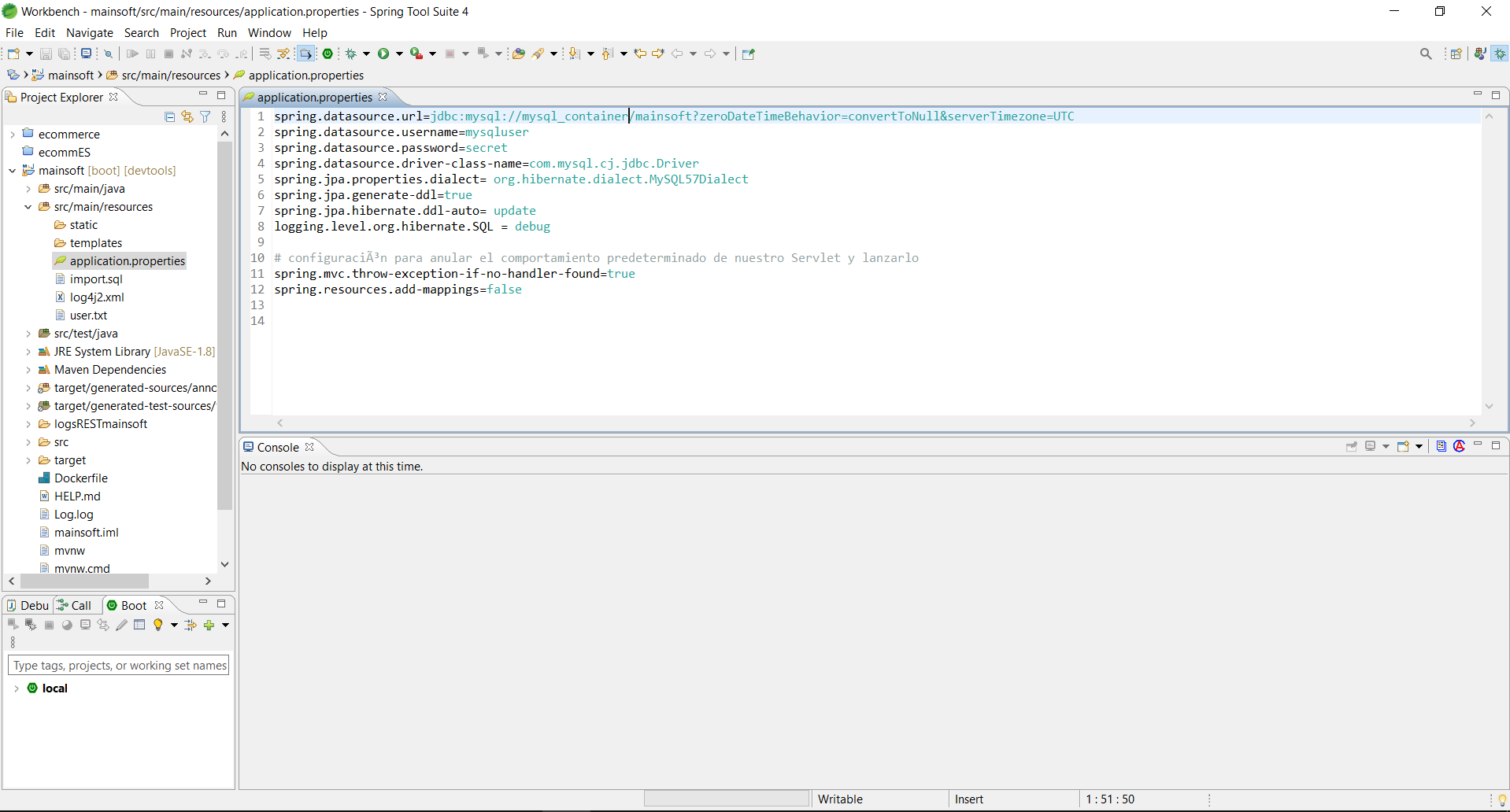
Docker MySql 11: Imagen desconectar y conectar a la nueva red

Comando: docker network inspect Network-Jamdres



Docker MySql 12: Imagen confirmación de contenedor en red

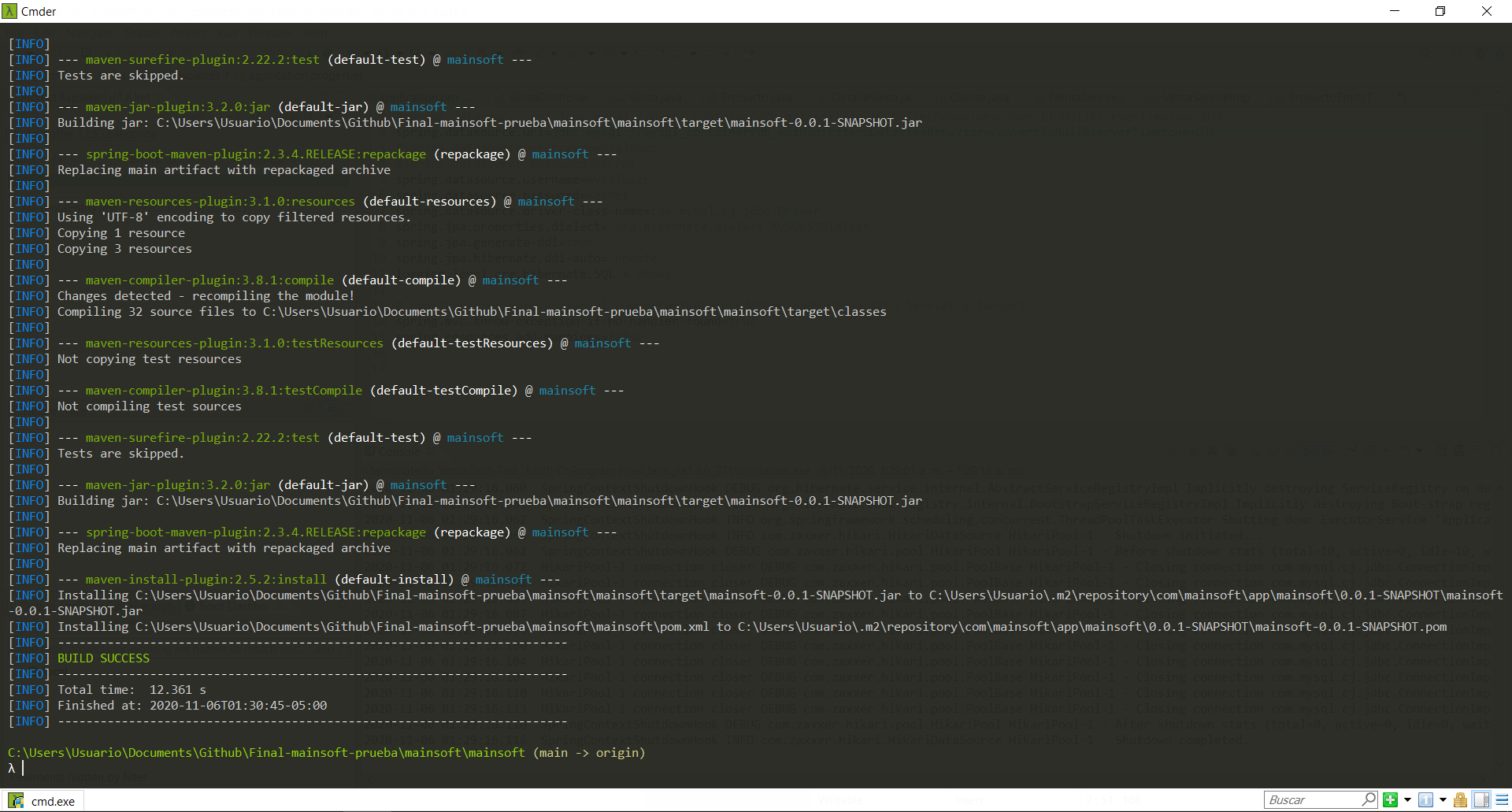
1. Apuntamos en el application.properties nuestro datasource al contenedor de Docker



Docker MySQL 13: Imagen confirmación config application.properties

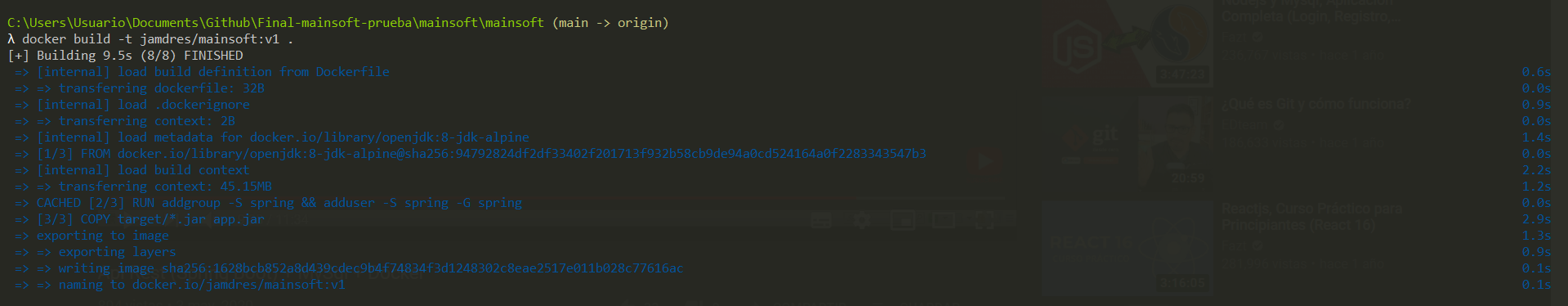
1. Creación del .JAR de las APIs del proyecto.

Comando: .\mvnw.cmd -Dmaven.test.skip=true clean compile package install



Docker JAR 1: Imagen creación .JAR

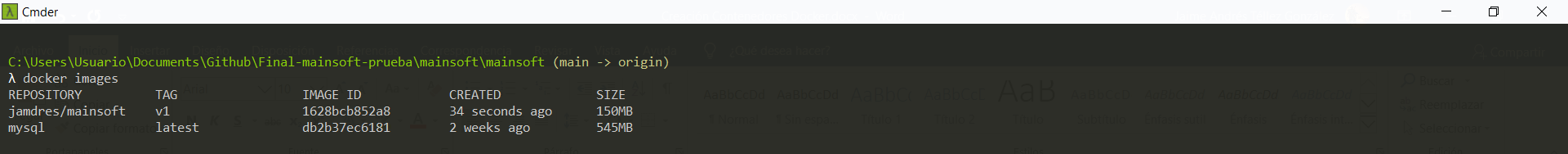
1. Ejecutamos el dockerfile
2. Comando: docker build -t jamdres/mainsoft:v1 .



Docker JAR 2: Imagen de la generación de la imagen Docker

1. Confirmación de la creación de las imágenes en docker

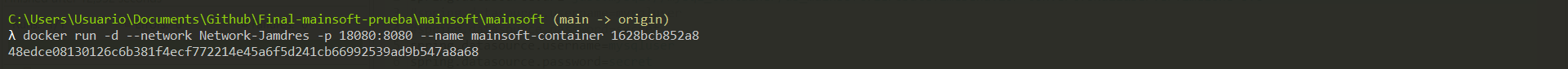
Comando: docker images



Docker JAR 3: Imagen de validación de las imágenes en Docker

1. Creación de un contenedor para la imagen .JAR de las APIs

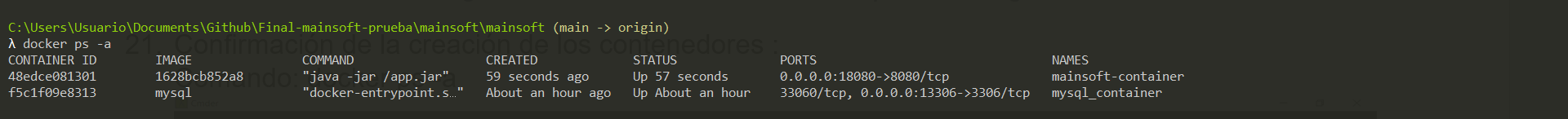
Comando: docker run -d --network Network-Jamdres -p 8080:8080 --name mainsoft-container 1628bcb852a8



Docker JAR 4: Imagen de la creación de contenedor para la imagen .JAR

1. Confirmación de la creación de los contenedores :

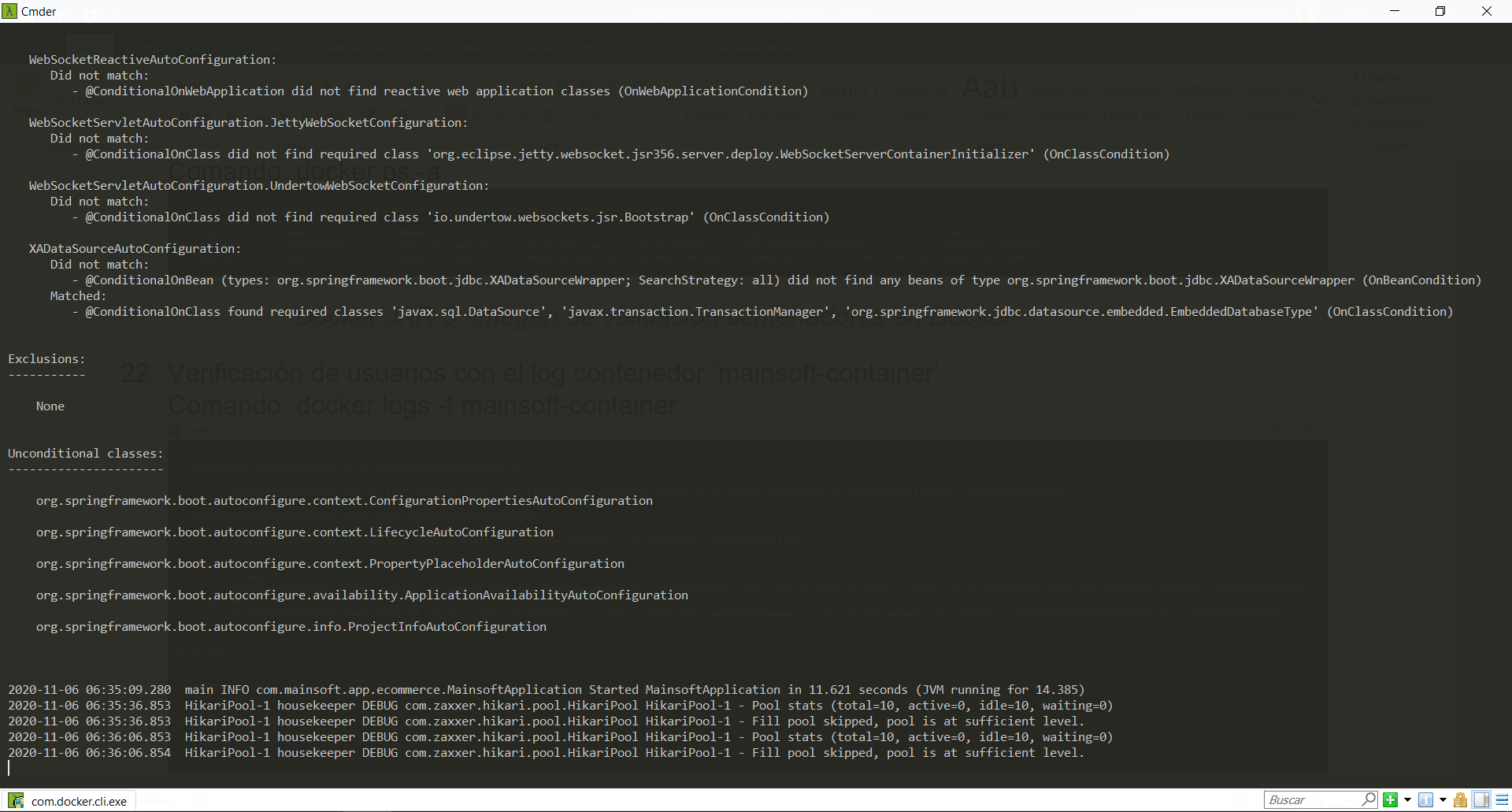
Comando: docker ps -a



Docker JAR 5: Imagen de validación contenedores en Docker

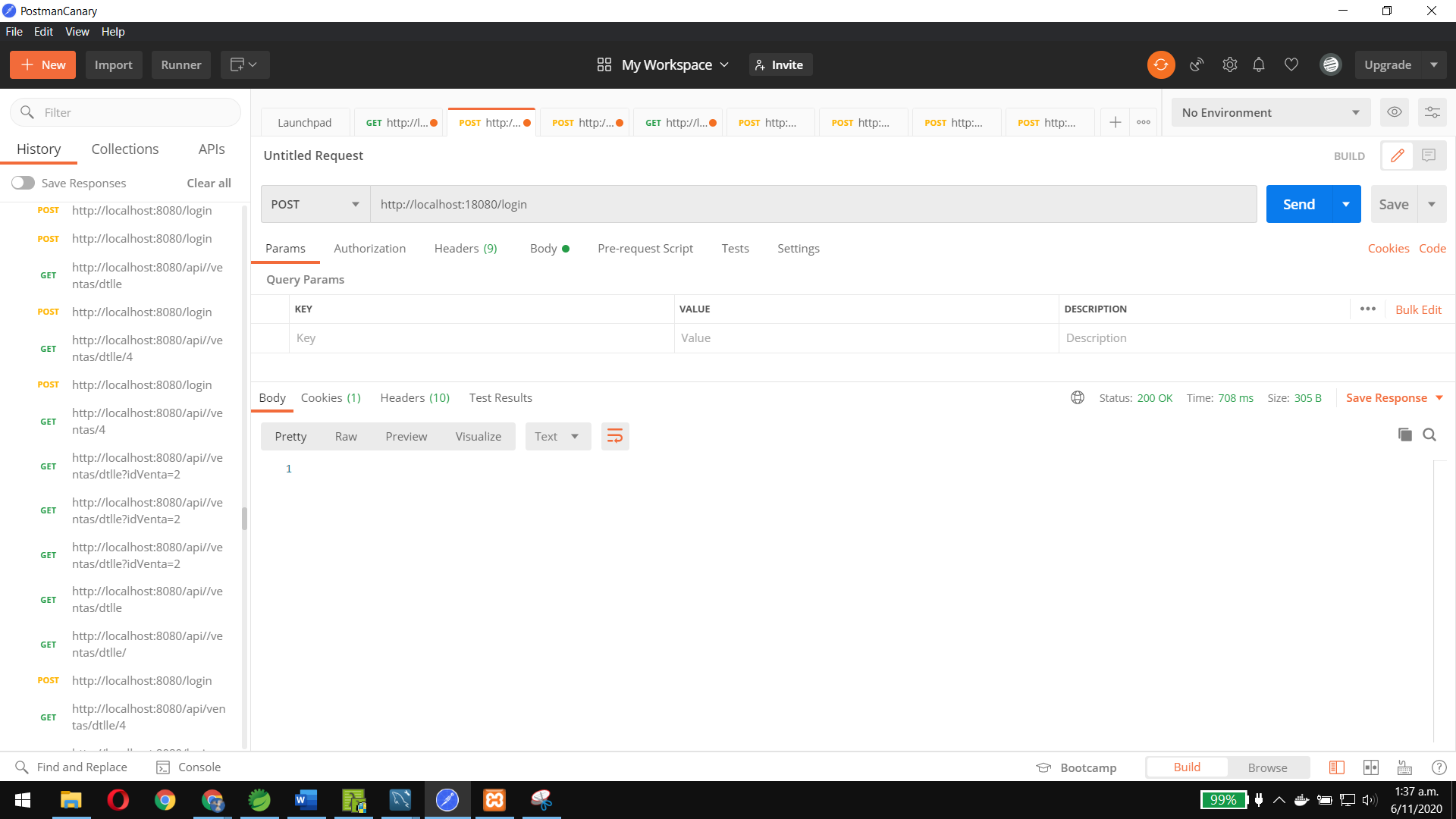
1. Verificación de usuarios con el log contenedor ‘mainsoft-container’

Comando: docker logs -f mainsoft-container



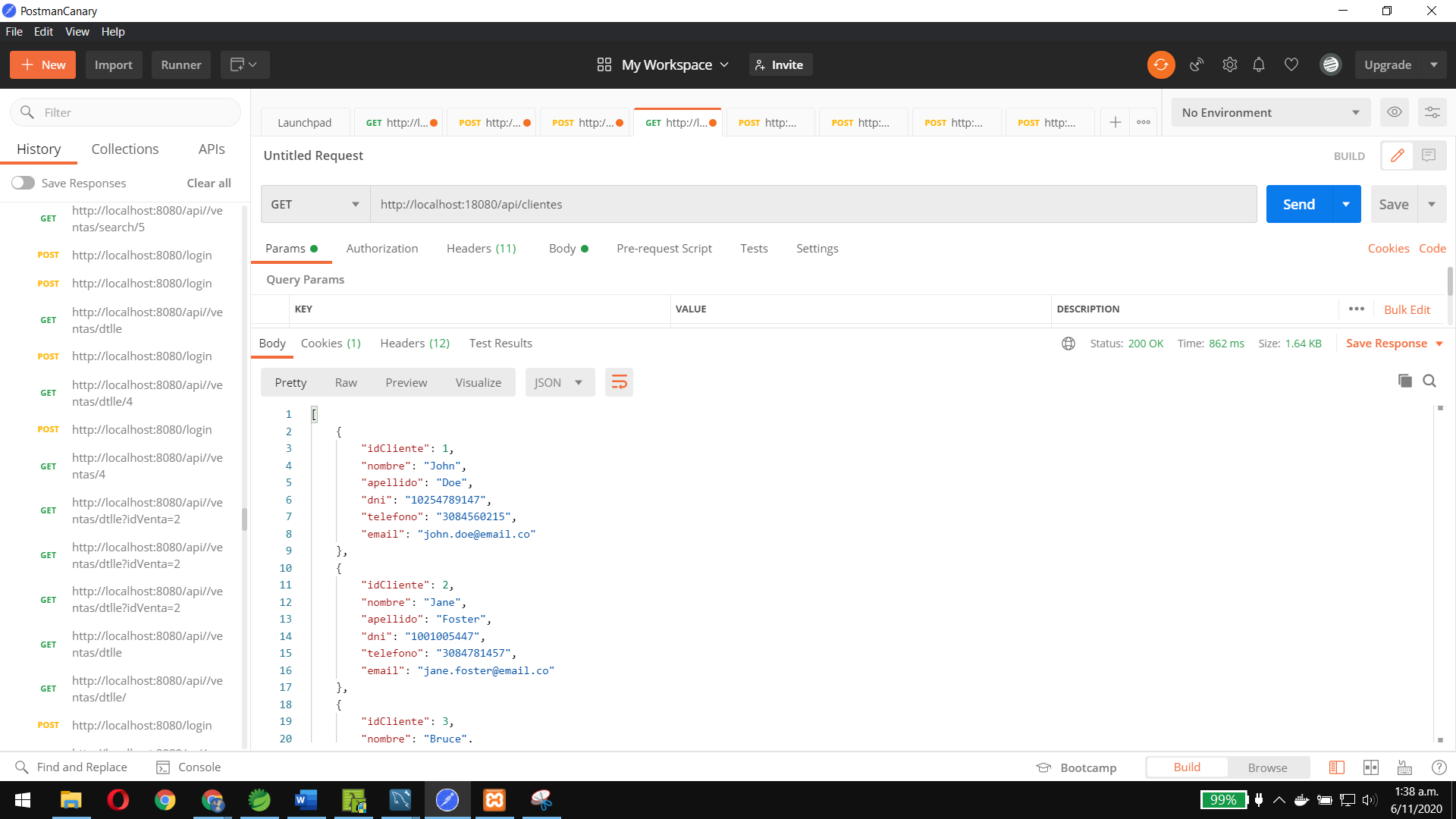
Docker JAR 6: Imagen de la prueba de ejecución correcta del contenedor

1. Prueba de conexión con puerto 18080 en postman



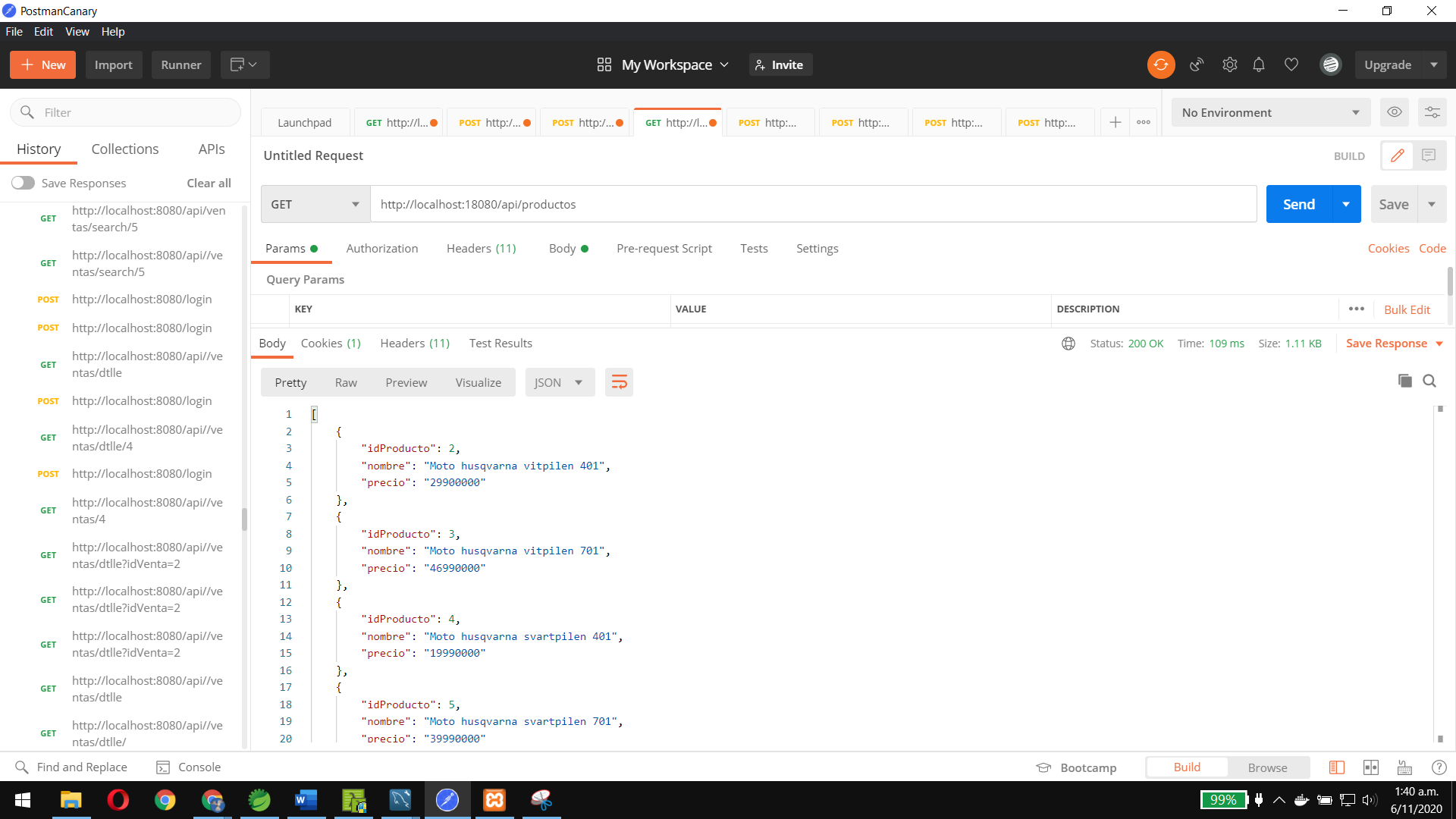
Postman : Imagen prueba de acceso con nuevo puerto 18080

1. Prueba lista de clientes



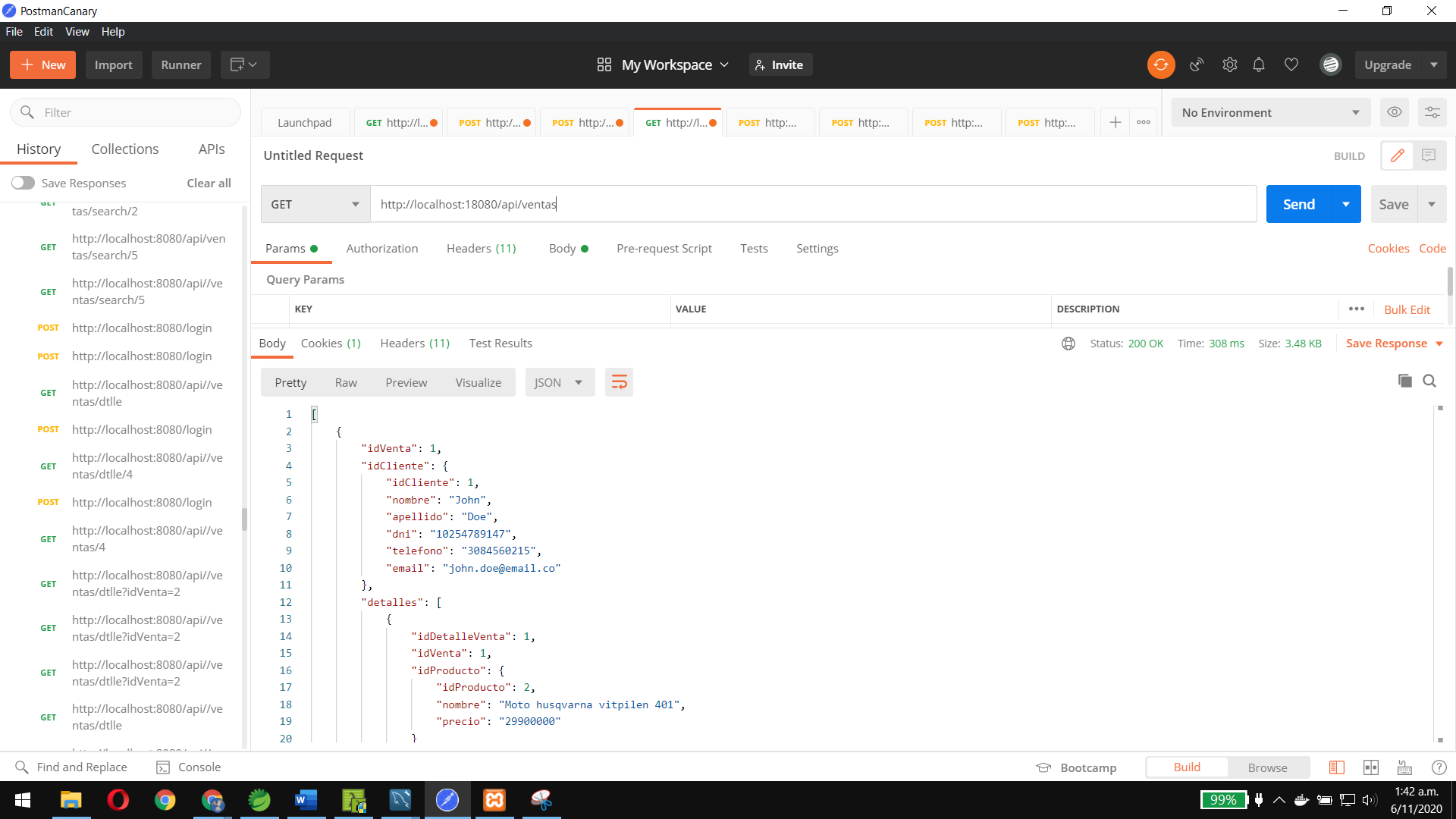
Postman : Imagen Prueba de lista clientes con puerto 18080

1. Prueba postman lista de productos



Postman : Imagen prueba lista productos puerto 18080

1. Prueba postman lista de ventas



Postman : Imagen prueba listado de ventas puerto 18080