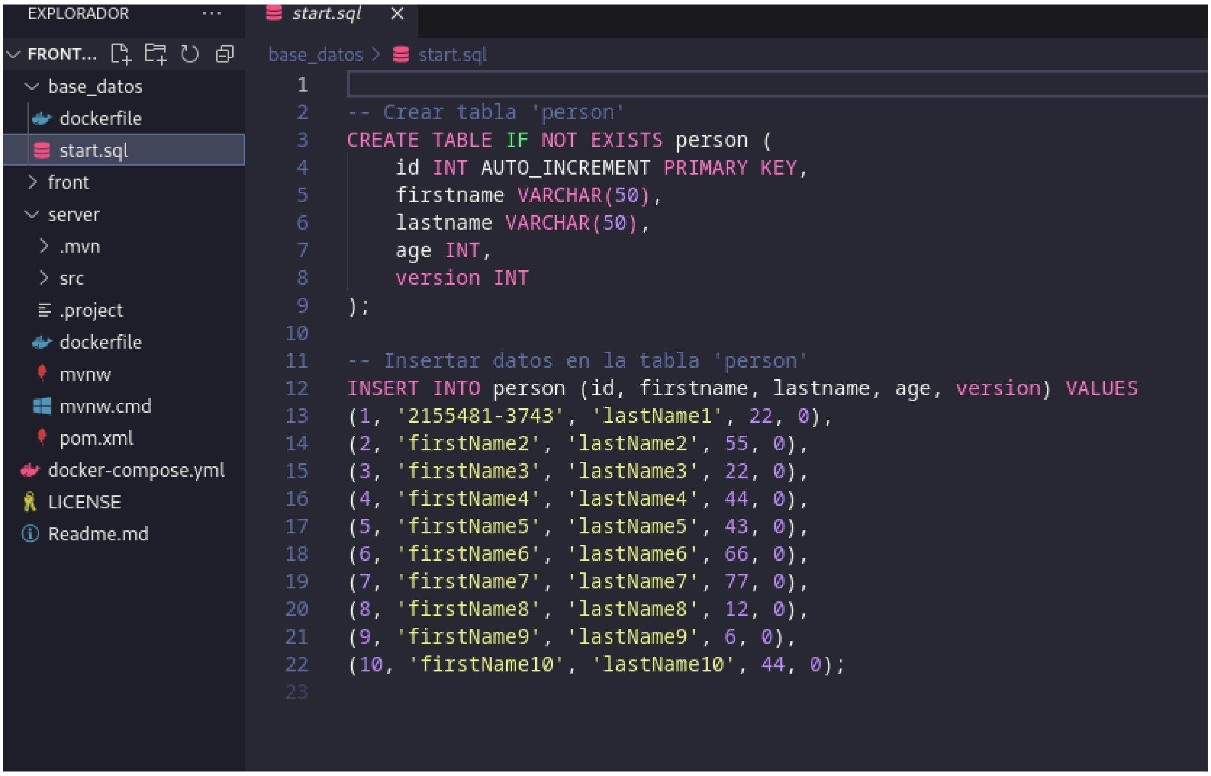
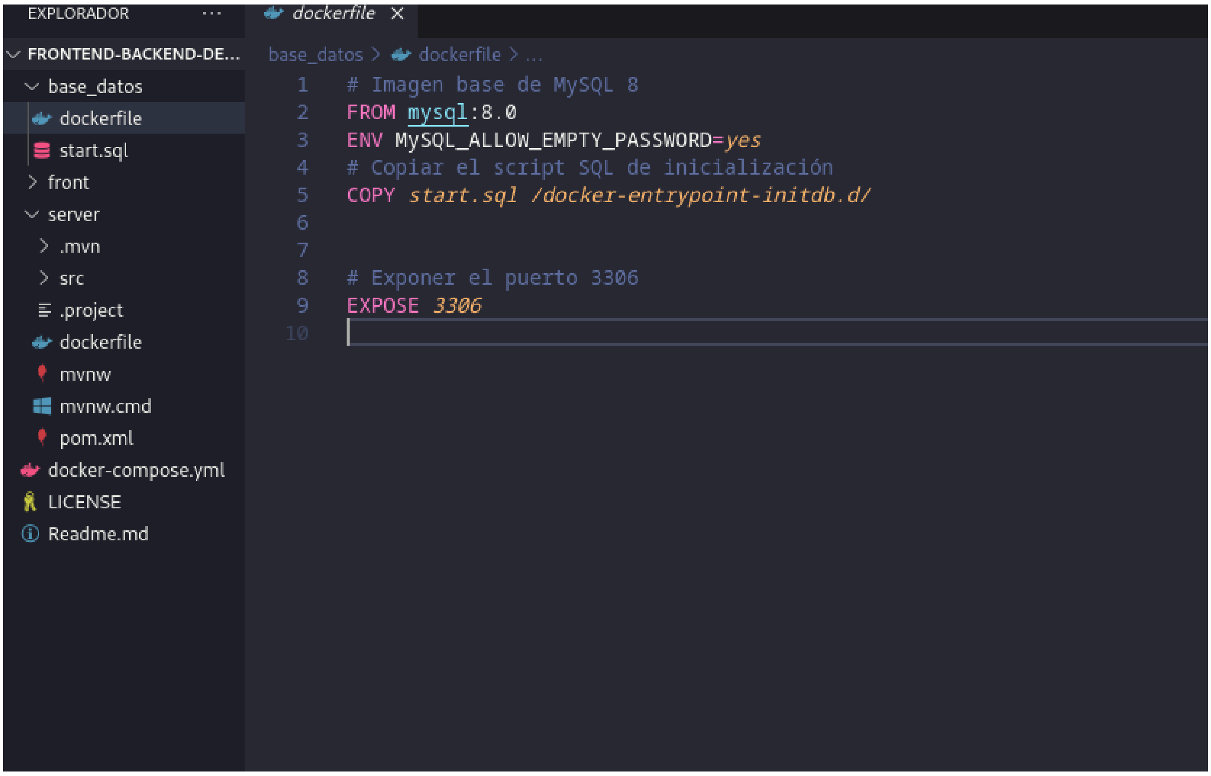
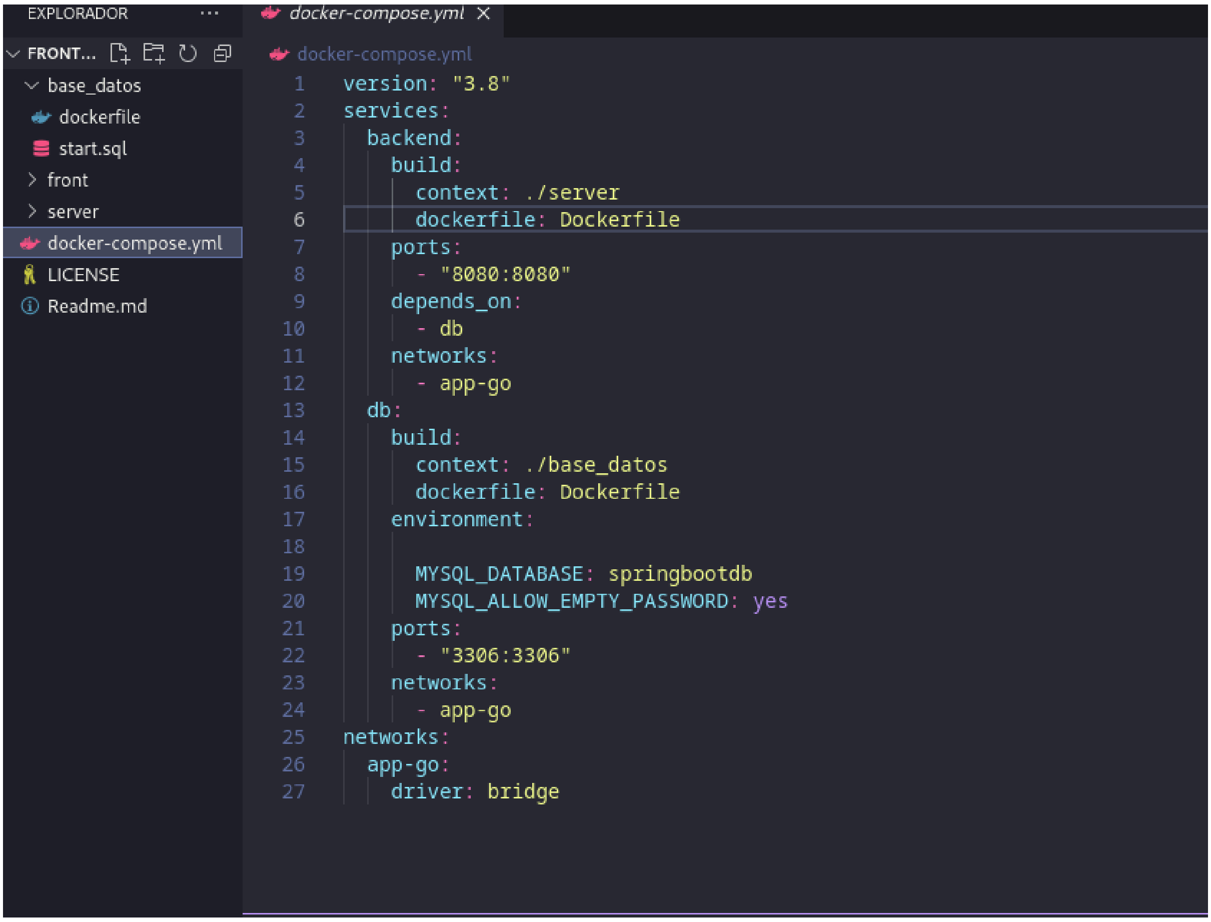


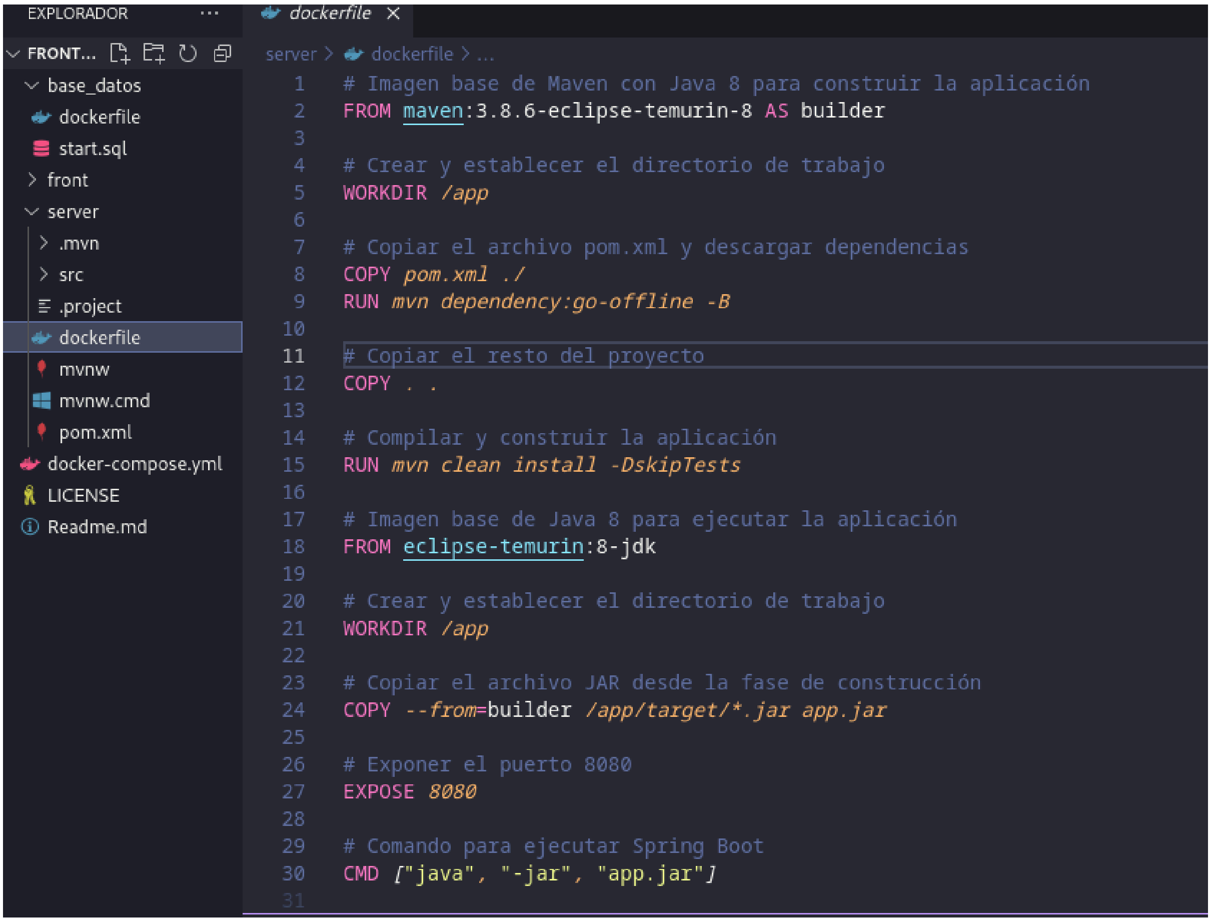
```sql

-- Crear tabla 'person'
CREATE TABLE IF NOT EXISTS person (
    id INT AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(50),
    lastname VARCHAR(50),
    age INT,
    version INT
);

-- Insertar datos en la tabla 'person'
INSERT INTO person (id, firstname, lastname, age, version) VALUES
(1, '2155481-3743', 'lastName1', 22, 0),
(2, 'firstName2', 'lastName2', 55, 0),
(3, 'firstName3', 'lastName3', 22, 0),
(4, 'firstName4', 'lastName4', 44, 0),
(5, 'firstName5', 'lastName5', 43, 0),
(6, 'firstName6', 'lastName6', 66, 0),
(7, 'firstName7', 'lastName7', 77, 0),
(8, 'firstName8', 'lastName8', 12, 0),
(9, 'firstName9', 'lastName9', 6, 0),
(10, 'firstName10', 'lastName10', 44, 0);
```



```dockerfile
# Imagen base de MySQL 8
FROM mysql:8.0
ENV MySQL_ALLOW_EMPTY_PASSWORD=yes
# Copiar el script SQL de inicialización
COPY start.sql /docker-entrypoint-initdb.d/


# Exponer el puerto 3306
EXPOSE 3306

```



```yaml
version: "3.8"
services:
  backend:
    build:
      context: ./server
      dockerfile: Dockerfile
    ports:
      - "8080:8080"
    depends_on:
      - db
    networks:
      - app-go
  db:
    build:
      context: ./base_datos
      dockerfile: Dockerfile
    environment:

      MYSQL_DATABASE: springbootdb
      MYSQL_ALLOW_EMPTY_PASSWORD: yes
    ports:
      - "3306:3306"
    networks:
      - app-go
networks:
  app-go:
    driver: bridge
```



```
# Imagen base de Maven con Java 8 para construir la aplicación
FROM maven:3.8.6-eclipse-temurin-8 AS builder

# Crear y establecer el directorio de trabajo
WORKDIR /app

# Copiar el archivo pom.xml y descargar dependencias
COPY pom.xml ./
RUN mvn dependency:go-offline -B

# Copiar el resto del proyecto
COPY . .

# Compilar y construir la aplicación
RUN mvn clean install -DskipTests

# Imagen base de Java 8 para ejecutar la aplicación
FROM eclipse-temurin:8-jdk

# Crear y establecer el directorio de trabajo
WORKDIR /app

# Copiar el archivo JAR desde la fase de construcción
COPY --from=builder /app/target/*.jar app.jar

# Exponer el puerto 8080
EXPOSE 8080

# Comando para ejecutar Spring Boot
CMD ["java", "-jar", "app.jar"]
```