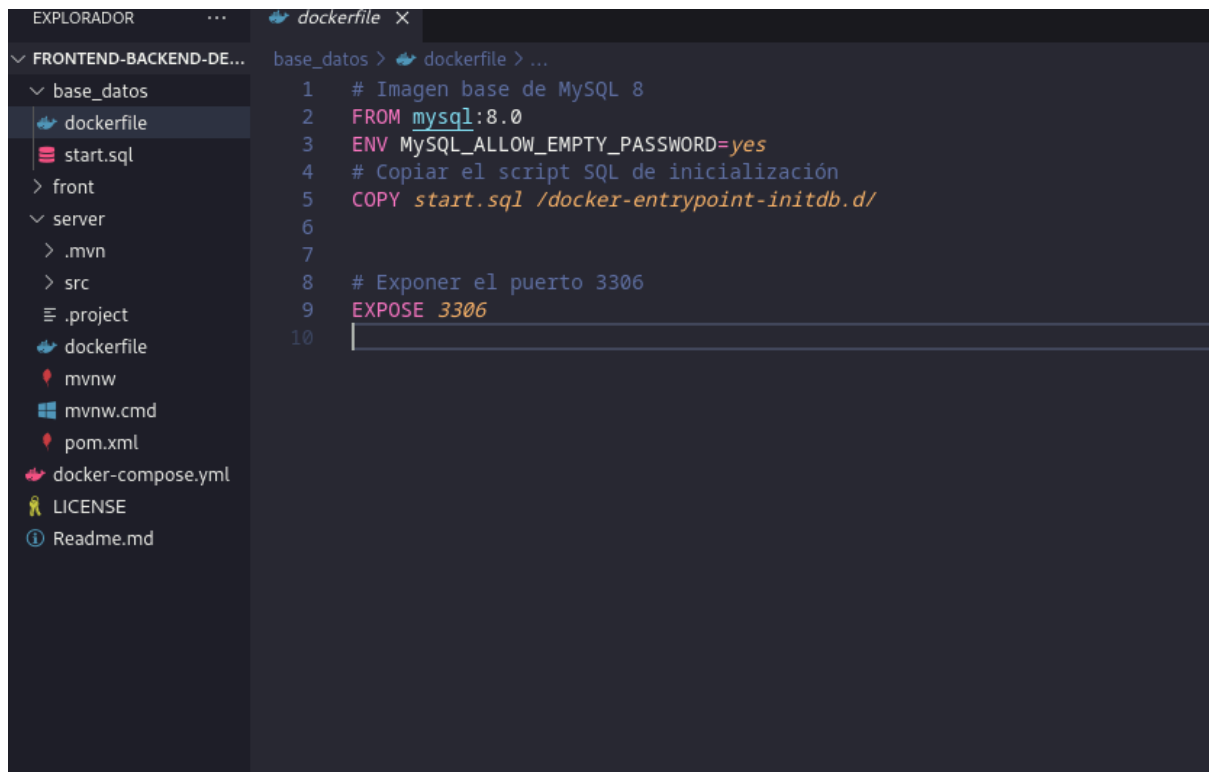


The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'base_datos', 'front', and 'server', and files like 'dockerfile', 'start.sql', 'mvnw', 'pom.xml', 'docker-compose.yml', 'LICENSE', and 'Readme.md'. The code editor is open to 'start.sql' and contains the following SQL script:

```
1
2 -- Crear tabla 'person'
3 CREATE TABLE IF NOT EXISTS person (
4     id INT AUTO_INCREMENT PRIMARY KEY,
5     firstname VARCHAR(50),
6     lastname VARCHAR(50),
7     age INT,
8     version INT
9 );
10
11 -- Insertar datos en la tabla 'person'
12 INSERT INTO person (id, firstname, lastname, age, version) VALUES
13 (1, '2155481-3743', 'lastName1', 22, 0),
14 (2, 'firstName2', 'lastName2', 55, 0),
15 (3, 'firstName3', 'lastName3', 22, 0),
16 (4, 'firstName4', 'lastName4', 44, 0),
17 (5, 'firstName5', 'lastName5', 43, 0),
18 (6, 'firstName6', 'lastName6', 66, 0),
19 (7, 'firstName7', 'lastName7', 77, 0),
20 (8, 'firstName8', 'lastName8', 12, 0),
21 (9, 'firstName9', 'lastName9', 6, 0),
22 (10, 'firstName10', 'lastName10', 44, 0);
23
```



The screenshot shows the same IDE with the 'dockerfile' file selected in the file explorer. The code editor displays the following Dockerfile content:

```
1 # Imagen base de MySQL 8
2 FROM mysql:8.0
3 ENV MySQL_ALLOW_EMPTY_PASSWORD=yes
4 # Copiar el script SQL de inicialización
5 COPY start.sql /docker-entrypoint-initdb.d/
6
7
8 # Exponer el puerto 3306
9 EXPOSE 3306
10
```

EXPLORADOR... docker-compose.yml X

FRONT... docker-compose.yml

base_datos

dockerfile

start.sql

front

server

docker-compose.yml

LICENSE

Readme.md

```
1 version: "3.8"
2 services:
3   backend:
4     build:
5       context: ./server
6       dockerfile: Dockerfile
7   ports:
8     - "8080:8080"
9   depends_on:
10     - db
11   networks:
12     - app-go
13 db:
14   build:
15     context: ./base_datos
16     dockerfile: Dockerfile
17   environment:
18
19     MYSQL_DATABASE: springbootdb
20     MYSQL_ALLOW_EMPTY_PASSWORD: yes
21   ports:
22     - "3306:3306"
23   networks:
24     - app-go
25 networks:
26   app-go:
27     driver: bridge
```

EXPLORADOR... dockerfile X

FRONT...
base_datos
dockerfile
start.sql
front
server
 .mvn
 src
 .project
 dockerfile
 mvnw
 mvnw.cmd
 pom.xml
docker-compose.yml
LICENSE
Readme.md

server > dockerfile > ...
1 # Imagen base de Maven con Java 8 para construir la aplicación
2 FROM maven:3.8.6-eclipse-temurin-8 AS builder
3
4 # Crear y establecer el directorio de trabajo
5 WORKDIR /app
6
7 # Copiar el archivo pom.xml y descargar dependencias
8 COPY pom.xml ./
9 RUN mvn dependency:go-offline -B
10
11 # Copiar el resto del proyecto
12 COPY . .
13
14 # Compilar y construir la aplicación
15 RUN mvn clean install -DskipTests
16
17 # Imagen base de Java 8 para ejecutar la aplicación
18 FROM eclipse-temurin:8-jdk
19
20 # Crear y establecer el directorio de trabajo
21 WORKDIR /app
22
23 # Copiar el archivo JAR desde la fase de construcción
24 COPY --from=builder /app/target/*.jar app.jar
25
26 # Exponer el puerto 8080
27 EXPOSE 8080
28
29 # Comando para ejecutar Spring Boot
30 CMD ["java", "-jar", "app.jar"]
31