# DOCUMENTO GUIA

## 1) Requisitos previos

* Docker (Docker Desktop en Windows o Docker Engine en Linux/WSL2).
* Docker Compose (v2 incluida en Docker Desktop o docker compose integrado).
* Maven y JDK 21 (si vas a compilar localmente). Opcional: compilar dentro del contenedor (multi-stage).
* Estructura del proyecto :

Prueba/  
├─ pom.xml  
├─ src/  
├─ Dockerfile  
├─ docker-compose.yml  
└─ db/  
 └─ init/  
 └─ schema.sql

## 2) Dockerfile (multi-stage, JDK 21)

# ---------- Etapa 1: Build ----------  
FROM maven:3.9.6-eclipse-temurin-21 AS builder  
WORKDIR /app  
COPY pom.xml .  
COPY src ./src  
RUN mvn clean package -DskipTests  
  
# ---------- Etapa 2: Runtime ----------  
FROM eclipse-temurin:21-jdk-alpine  
WORKDIR /app  
COPY --from=builder /app/target/\*.jar app.jar  
EXPOSE 8080  
ENTRYPOINT ["java","-jar","app.jar"]

## 3) docker-compose.yml (app + MySQL BANCO\_DB)

version: "3.9"  
services:  
 app:  
 build: .  
 container\_name: ws-test  
 ports:  
 - "8080:8080"  
 depends\_on:  
 - db  
 environment:  
 SPRING\_DATASOURCE\_URL: jdbc:mysql://db:3306/BANCO\_DB?allowPublicKeyRetrieval=true&useSSL=false&serverTimezone=UTC  
 SPRING\_DATASOURCE\_USERNAME: root  
 SPRING\_DATASOURCE\_PASSWORD: root  
 SPRING\_JPA\_HIBERNATE\_DDL\_AUTO: update  
 networks:  
 - ws-net  
  
 db:  
 image: mysql:8.0  
 container\_name: mysql-db  
 environment:  
 MYSQL\_ROOT\_PASSWORD: root  
 MYSQL\_DATABASE: BANCO\_DB  
 MYSQL\_USER: user  
 MYSQL\_PASSWORD: user123  
 volumes:  
 - db\_data:/var/lib/mysql  
 - ./db/init:/docker-entrypoint-initdb.d  
 ports:  
 - "3307:3306"  
 networks:  
 - ws-net  
  
volumes:  
 db\_data:  
  
networks:  
 ws-net:  
 driver: bridge

## 4) Script de inicialización db/init/schema.sql

CREATE TABLE persons (  
 person\_id BIGINT AUTO\_INCREMENT PRIMARY KEY,  
 name VARCHAR(50) NOT NULL,  
 gender CHAR(1) NOT NULL,  
 age INT NOT NULL,  
 identification VARCHAR(20) NOT NULL UNIQUE,  
 address VARCHAR(100) NOT NULL,  
 phone VARCHAR(20) NOT NULL  
) ENGINE=InnoDB;  
  
CREATE TABLE clients (  
 client\_id BIGINT AUTO\_INCREMENT PRIMARY KEY,  
 person\_id BIGINT,  
 password VARCHAR(100) NOT NULL,  
 status VARCHAR(10) NOT NULL,  
 CONSTRAINT fk\_person  
 FOREIGN KEY (person\_id)  
 REFERENCES persons(person\_id)  
 ON DELETE CASCADE  
 ON UPDATE CASCADE  
) ENGINE=InnoDB;  
  
CREATE TABLE accounts (  
 account\_number BIGINT AUTO\_INCREMENT PRIMARY KEY,  
 account\_type VARCHAR(20) NOT NULL,  
 balance DECIMAL(15,2) NOT NULL,  
 status BOOLEAN NOT NULL,  
 client\_id BIGINT NOT NULL,  
 CONSTRAINT fk\_client  
 FOREIGN KEY (client\_id)  
 REFERENCES clients(client\_id)  
 ON DELETE CASCADE  
 ON UPDATE CASCADE  
) ENGINE=InnoDB AUTO\_INCREMENT=100000;  
  
CREATE TABLE movements (  
 movement\_id BIGINT AUTO\_INCREMENT PRIMARY KEY,  
 date DATE NOT NULL,  
 movement\_type VARCHAR(20) NOT NULL,  
 value DECIMAL(15,2) NOT NULL,  
 initial\_balance DECIMAL(15,2) NOT NULL,  
 balance DECIMAL(15,2) NOT NULL,  
 account\_number BIGINT NOT NULL,  
 CONSTRAINT fk\_account  
 FOREIGN KEY (account\_number)  
 REFERENCES accounts(account\_number)  
 ON DELETE CASCADE  
 ON UPDATE CASCADE  
) ENGINE=InnoDB;

## 5) application.properties

spring.datasource.url=jdbc:mysql://db:3306/BANCO\_DB?allowPublicKeyRetrieval=true&useSSL=false&serverTimezone=UTC  
spring.datasource.username=root  
spring.datasource.password=root  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.show-sql=true

## 6) Comandos principales

# compilar y levantar  
docker compose up -d --build  
  
# ver contenedores  
docker ps  
  
# logs de app o bd  
docker compose logs -f app  
docker compose logs -f db