**0) Prerrequisitos**

* **Java 17** (JDK 17)
* **Maven 3.8+**
* **Python 3.10+**
* **MySQL 8.x**
* **Git** (opcional, para clonar)

Verifica versiones:

java -version

mvn -v

python --version

mysql --version

**1) Estructura del repo (carpetas relevantes)**

EmpujeComunitario/

├─ ClienteSpring/ # Proyecto Spring Boot (REST + gRPC client)

├─ ServerPython/ # Servidor gRPC (Python)

│ ├─ server.py

│ ├─ .env # Variables de conexión MySQL / puerto gRPC

│ └─ app/

│ ├─ db.py # Conexión MySQL (PyMySQL)

│ ├─ models.py # DAOs (usuarios/roles)

│ ├─ security.py # bcrypt utilidades

│ └─ services/

│ └─ user\_service.py # Implementación gRPC

└─ Proto/

└─ ong.proto # Proto unificado

**Rutas relativas**: el pom.xml de Spring ya compila el ../Proto/ong.proto.

**3) Servidor Python (gRPC → MySQL)**

**3.1 Crear y activar venv (recomendado)**

cd ServerPython

python -m venv .venv

.\.venv\Scripts\Activate.ps1

**3.2 Instalar dependencias**

pip install grpcio grpcio-tools pymysql bcrypt cryptography python-dotenv cryptography

**3.3 Variables del servidor (ServerPython/.env)**

Crea un archivo **.env** (texto plano):

DB\_HOST=localhost

DB\_PORT=3306

DB\_USER=empuje (CAMBIAR POR TU USUARIO)

DB\_PASS=empuje123 (CAMBIAR POR TU CLAVE)

DB\_NAME=empujecomunitario

GRPC\_PORT=50051

KAFKA\_BOOTSTRAP\_SERVERS=localhost:9092

ORG\_ID=42

**3.4 Generar stubs (solo si cambiaste ong.proto)**

cd ServerPython

python -m grpc\_tools.protoc -I../Proto --python\_out=./ --grpc\_python\_out=./ ../Proto/ong.proto

**3.5 Levantar el servidor**

**cd ServerPython**

**.\.venv\Scripts\Activate.ps1**

**python server.py**

Salida esperada en terminal :

[DB] Conectado a MySQL localhost:3306/empujecomunitario

[gRPC][Python] Escuchando en 0.0.0.0:50051 ...

**4) Cliente Spring Boot**

**4.1 Construir / arrancar**

cd ClienteSpring

mvn clean install -DskipTests

mvn spring-boot:run

**6) Solución de errores comunes**

* **No plugin found for prefix 'spring-boot'**
  + Ejecutaste fuera de ClienteSpring o falta el plugin en <build><plugins>.
  + Corre:
  + cd ClienteSpring
  + mvn spring-boot:run
* **invalid salt al loguear usuarios viejos**
  + Significa que tu BD tenía contraseñas en texto plano.
  + Solución: borra usuarios viejos y crea nuevos (server ya guarda **bcrypt**), o implementa migración transparente (opcional).
* **VS Code ve errores “fantasma”**
  + Haz **Developer: Reload Window** o borra .idea/.project caches.

**7) Comandos rápidos (para guardar)**

**Server (Python):**

cd ServerPython

.\.venv\Scripts\Activate.ps1

python server.py

**Cliente (Spring):**

cd ClienteSpring

mvn spring-boot:run

**Build sin tests:**

cd ClienteSpring

mvn clean install -DskipTests

**8) Usuarios de prueba (solo para demo)**

**Clave presidente**

Usuario: Matias

Pass : U1fMcV2NyHwa

**Voluntarios**

username: jose\_Vol

pass : pLmfCCRnuzYz

**Vocal**

username: maria\_Voc

email : malo@mail.com

pass : URsIbu9hRHWx

**Coordinador**

username: Rami\_Coor

email : ragu@mail.com

pass : N43JYA0D81wi

# Preparacion para docker

**Windows 10/11 64 bits**

* Debe soportar **virtualización (VT-x / SVM)** en BIOS.
  + SVM Mode → Enabled (AMD)
  + o Intel VT-x (Intel).

**Docker Desktop**

* Descargar desde: https://www.docker.com/products/docker-desktop/
* Instalar con la opción de habilitar **WSL2 backend**.
* Tras instalar, abrir Docker Desktop y verificar que aparece “Engine running”.

**WSL2 (si no está instalado)**  
Ejecutar en PowerShell (como admin):

wsl --install

Reiniciar la PC y luego abrir Docker Desktop

**acceder a carpeta de Kafka y Levantar contenedores**

cd EmpujeComunitario\infra\kafka

docker compose up -d

Verificá que estén corriendo:

docker ps

**Crear topics del TP**

**Si PowerShell bloquea scripts, usar:  
powershell -ExecutionPolicy Bypass -File .\create-topics.ps1**

Ejecutar el script incluido:

.\create-topics.ps1

docker exec kafka bash -lc "kafka-topics --bootstrap-server localhost:9092 --list"

\*deberia devolver algo vacio

**Probar Kafka manualmente (opcional)**

**Terminal 1 Consumidor (escucha mensajes):**

docker exec -it kafka bash -lc "kafka-console-consumer --bootstrap-server localhost:9092 --topic solicitud-donaciones --from-beginning"

**Terminal 2 Productor (envía mensajes):**

docker exec -it kafka bash -lc "kafka-console-producer --bootstrap-server localhost:9092 --topic solicitud-do

deberias ver en terminal 1

{"org\_id":42,"solicitud\_id":"SOL-2025-0001","items":[{"categoria":"ALIMENTOS","descripcion":"Puré de tomates"}],"fecha\_hora":"2025-09-15T03:10:00-03:00","idempotency\_key":"42:SOL-2025-0001"}

**Detener contenedores**

docker compose down

**Si necesitás resetear todo (incluyendo datos):**

docker compose down -v

if (Test-Path .\data) { Remove-Item -Recurse -Force .\data }

y volver a docker compose up -d.