

Relatório do Trabalho Prático 2 - Sistema de Monitorização Ambiental

Aluno: Ana Valero Redondo

1. Justificação das Escolhas

Tecnologias: Optou-se pelo Spring Boot devido à sua robustez na criação de microserviços e facilidade de integração com diferentes protocolos.

Base de Dados: Utilizou-se PostgreSQL num contentor Docker. A persistência é gerida pelo Hibernate (JPA), o que permitiu mapear automaticamente as entidades Device e Metric sem necessidade de escrever SQL manual (ddl-auto=update).

Protocolos:

- **MQTT:** Escolhido para sensores IoT de baixo consumo (simulado em client-mqtt) pela sua natureza assíncrona e leveza.
- **gRPC:** Escolhido para comunicação de alto desempenho (simulado em client-grpc), utilizando Protobuf para serialização binária eficiente.
- **REST:** Escolhido para integração web padrão e gestão do sistema (Admin CLI), pela facilidade de uso e legibilidade do JSON.

2. Configuração da Base de Dados O sistema requer uma instância PostgreSQL a correr em Docker.

- **Comando de inicialização:** `docker run --name postgres-container -e POSTGRES_PASSWORD=SD2024 -e POSTGRES_DB=sd -p 15432:5432 -d postgres`
 - `docker start postgres-container postgres-container`
- **Estrutura:** O sistema cria automaticamente duas tabelas principais:
- **devices:** Armazena ID, localização e estado.
- **metrics:** Armazena leituras de temperatura/humidade com timestamp.

3. Análise de Performance e Comparação Durante o desenvolvimento, observou-se o seguinte comportamento dos protocolos:

Protocolo	Tipo	Overhead	Velocidade	Observação
gRPC	Síncrono	Muito Baixo (Binário)	Muito Alta	O uso de Protobuf tornou o envio de dados extremamente rápido, ideal para gateways.
MQTT	Assíncrono	Mínimo	Alta	A conexão persistente e o formato leve permitiram envio contínuo sem bloquear o cliente.
REST	Síncrono	Médio (JSON texto)	Média	O overhead do HTTP/1.1 e JSON torna-o mais lento que gRPC, mas suficiente para gestão.

4. Dificuldades e Observações

- A configuração inicial do protoc para gerar as classes gRPC exigiu ajustes no pom.xml para detetar o sistema operativo (Windows).
- A gestão de dependências no Spring Boot 3.2 exigiu atenção às versões compatíveis de grpc-spring-boot-starter.