



Desafio Técnico Mobile

Simulador de Ciclo Automatizado de Transporte na Mineração



Contexto do Problema

Na mineração de grande escala, equipamentos como caminhões, escavadeiras, perfuratrizes e pás-carregadeiras trabalham de forma coordenada para extrair, transportar e descarregar minério. Este desafio foca no processo de **automação do ciclo completo de um caminhão de transporte de minério**, sem intervenção manual do operador.

O aplicativo que você irá construir tem como objetivo **interpretar automaticamente os dados de sensores embarcados**, identificar em qual etapa do ciclo o caminhão se encontra e exibir essas informações ao operador. Cada ciclo será registrado e, ao final, deverá ser armazenado localmente e posteriormente sincronizado com o "servidor" simulado — via exportação para um arquivo de log.



Objetivo

Criar um aplicativo mobile que:

- Carregue leituras simuladas de sensores linha a linha de um arquivo `.jsonl`;
- Interprete as leituras para determinar a etapa atual do ciclo de transporte;
- Atualize a interface com os dados processados;
- Armazene ciclos completos localmente;
- Simule envio de dados via rede escrevendo em um arquivo de saída `.jsonl`;
- Evite duplicação de dados já sincronizados;
- Seja capaz de funcionar completamente offline;



Regras de Negócio – Etapas do Ciclo

1. EM FILA CARREGAMENTO

- Velocidade = 0 por mais de 5 segundos
- E (detecta caminhão em fila ou em carregamento para a mesma escavadeira)

2. EM CARREGAMENTO

- Velocidade = 0 por mais de 5 segundos
- Detecta escavadeira a menos de 2m

- E **nenhum outro caminhão** em carregamento para a mesma escavadeira

3. TRÂNSITO CHEIO

- Velocidade > 0
- Escavadeira a mais de 2m
- **Estado atual é EM CARREGAMENTO**

4. EM FILA BASCULAMENTO

- Velocidade = 0 por mais de 5 segundos
- GPS igual ou próximo ao ponto de basculamento
- Sensor de báscula **DESATIVADO**
- E (detecta caminhão em fila ou **estado atual é TRÂNSITO CHEIO**)

5. EM BASCULAMENTO

- Velocidade = 0
- GPS igual ao ponto de basculamento
- Sensor de báscula **ATIVADO**

6. TRÂNSITO VAZIO

- Velocidade > 0
- GPS distante 5m ou mais do ponto de basculamento
- **Estado atual é EM BASCULAMENTO**



Sensores Utilizados

Os dados dos sensores virão na simulação como:

- **Beacons:** array com equipamentos próximos (tipo: escavadeira, caminhão ou sensor_bascula)
- **GPS:** localização atual e velocidade (em m/s)

O **sensor de báscula** estará no array de beacons. Abaixo um exemplo de como ele estaria representado se estivesse instalado num caminhão de id CAM-001:

```
{
  "id": "CAM-001",
  "tipo": "sensor_bascula",
  "distancia": 0,
  "status": "DESATIVADO"
}
```

Estrutura do Arquivo de Entrada (**simulacao.jsonl**)

Formato: **.jsonl** (JSON Lines). Cada linha representa a leitura dos sensores em um momento no tempo.

Exemplo:

```
{
  "data_hora": "2025-06-27T15:00:05",
  "equipamento_carga": "EXCAV-001",
  "ponto_basculamento": {
    "x": 100,
    "y": 200,
    "z": 0
  },
  "beacons": [
    {
      "id": "EXCAV-001",
      "tipo": "escavadeira",
      "distancia": 5,
      "status": "OPERANDO"
    },
    {
      "id": "CAM-001",
      "tipo": "sensor_bascula",
      "distancia": 0,
      "status": "DESATIVADO"
    }
  ],
  "gps": {
    "velocidade": 0.0,
    "localizacao": {
      "x": 95,
      "y": 198,
      "z": 0
    }
  }
}
```



Interface Esperada

A cada leitura simulada (via botão **SIMULAR**), o app deve exibir:

- **ETAPA ATUAL**: status determinado pela lógica
 - **EQUIPAMENTO DE CARGA**: conforme leitura
 - **PONTO DE BASCULAMENTO**: conforme leitura
 - **VELOCIDADE ATUAL**: convertida de m/s para km/h
 - **DADOS SINCRONIZADOS**: indica se há pendências ou se tudo foi enviado
-



Sincronização Offline → Online

- O app deve registrar **ciclos completos** (até **TRÂNSITO VAZIO**) localmente
- Quando o app detectar “rede disponível” (simulada), deve exportar os ciclos não sincronizados para:

`sync_servidor.jsonl` (1 linha por ciclo completo):

```
{
  "ciclo_id": "ciclo-20250627150200",
  "data_inicio": "2025-06-27T15:02:00",
  "data_fim": "2025-06-27T15:17:33",
  "etapas": [
    { "etapa": "EM FILA CARREGAMENTO", "timestamp": "2025-06-27T15:02:00" },
    ...
    { "etapa": "TRÂNSITO VAZIO", "timestamp": "2025-06-27T15:17:33" }
  ],
  "equipamento_id": "CAM-001",
  "equipamento_carga": "EXCAV-001",
  "ponto_basculamento": { "x": -19.923, "y": -43.937, "z": 0 },
  "status_sincronizacao": "SINCRONIZADO"
}
```



Entrega Esperada

O candidato deve entregar:

1. ☒ **Link para repositório GitHub**
2. ☒ Código-fonte completo do app
3. ☒ **APK** para instalação (link no README)
4. ☒ Arquivo **README.md** com:
 - Instruções de instalação e uso
 - Explicação da arquitetura e decisões técnicas (opcional, mas um diferencial)
 - Local onde será gerado o arquivo sync_servidor.jsonl
 - Dívidas técnicas caso não tenha conseguido implementar alguma coisa da forma que entende ser a mais adequada;