

# LOGITEL

## Tecnologias Utilizadas

- Frontend: React.js com Bootstrap
- Backend (API): Node.js + Express
- Banco de Dados: SQLite3 (utilizando better-sqlite3 – leve, local e sem necessidade de servidor)
- Uploads de arquivos: multer
- Comunicação com a API: via fetch (sem Postman no uso final)
- Hospedagem (em produção): planejada em uma máquina virtual com Ubuntu (via XCP-ng)

## Funcionalidades da Aplicação

### • Movimentações

Tipos de movimentação:

- Entrada
  - Saída
  - Evento
- Campos registrados:
    - Equipamento, Localização, Responsável, Motivo/Evento, Observação
    - Data/Hora automática
    - Nota Fiscal (campo + anexo de imagem ou PDF)

## Devoluções

- Permite marcar movimentações como "devolvidas"
- Possibilidade de adicionar:
  - Imagem da devolução
  - Responsável
  - Observação
  - Data/hora

## **Consultas**

- Listagem geral das movimentações e filtros por tipo
- Visualização de detalhes de cada movimentação
- Download/visualização da nota fiscal (PDF ou imagem)

## **Arquitetura (Resumo técnico)**

- A API foi criada em Node.js com Express, e está organizada na pasta /api
  - Endpoints:
    - GET /movimentacoes
    - POST /movimentacoes
    - PUT /movimentacoes/:id/devolver
    - POST /devolucoes
    - GET /movimentacoes/:id/devolucoes
- O banco de dados local (SQLite) é leve, embutido e simples, com duas tabelas:
  - movimentacoes
  - devolucoes
- Os arquivos (como notas fiscais) são salvos na pasta /uploads local, e servidos pela API
- .env usado para configurar a porta do servidor

## **Execução local**

- Para iniciar o backend:

```
cd api
```

```
npm install
```

```
node server.js
```

- Para iniciar o frontend:

```
npm install
```

```
npm start
```

### **Hospedagem futura**

- Será hospedado em uma VM com Ubuntu usando XCP-ng
- A aplicação será iniciada automaticamente com o servidor
- Acessível via navegador interno da empresa