# Entrega Final Projeto BD – 2025/2

Davi Bragança e Silva - 242001473

Caio Fernando Rocha de Albuquerque – 242034518

**GITHUB:** https://github.com/davibraganca10/Projeto_BD-2025-2

**TEMA**: Sistema de Registro de Ocorrências Urbanas

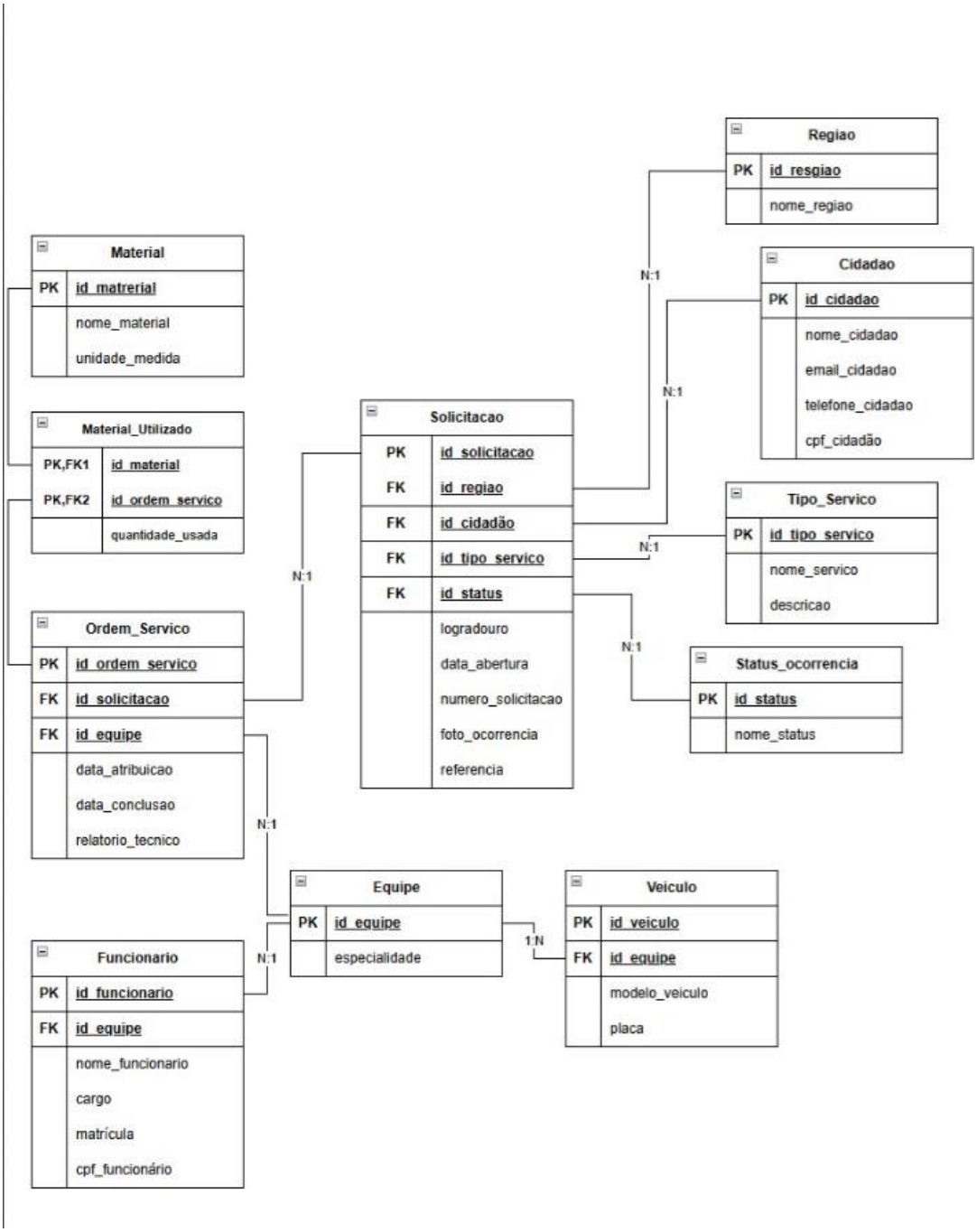
**SGBD**: PostgreSQL

**LINGUAGEM**: Java com Maven

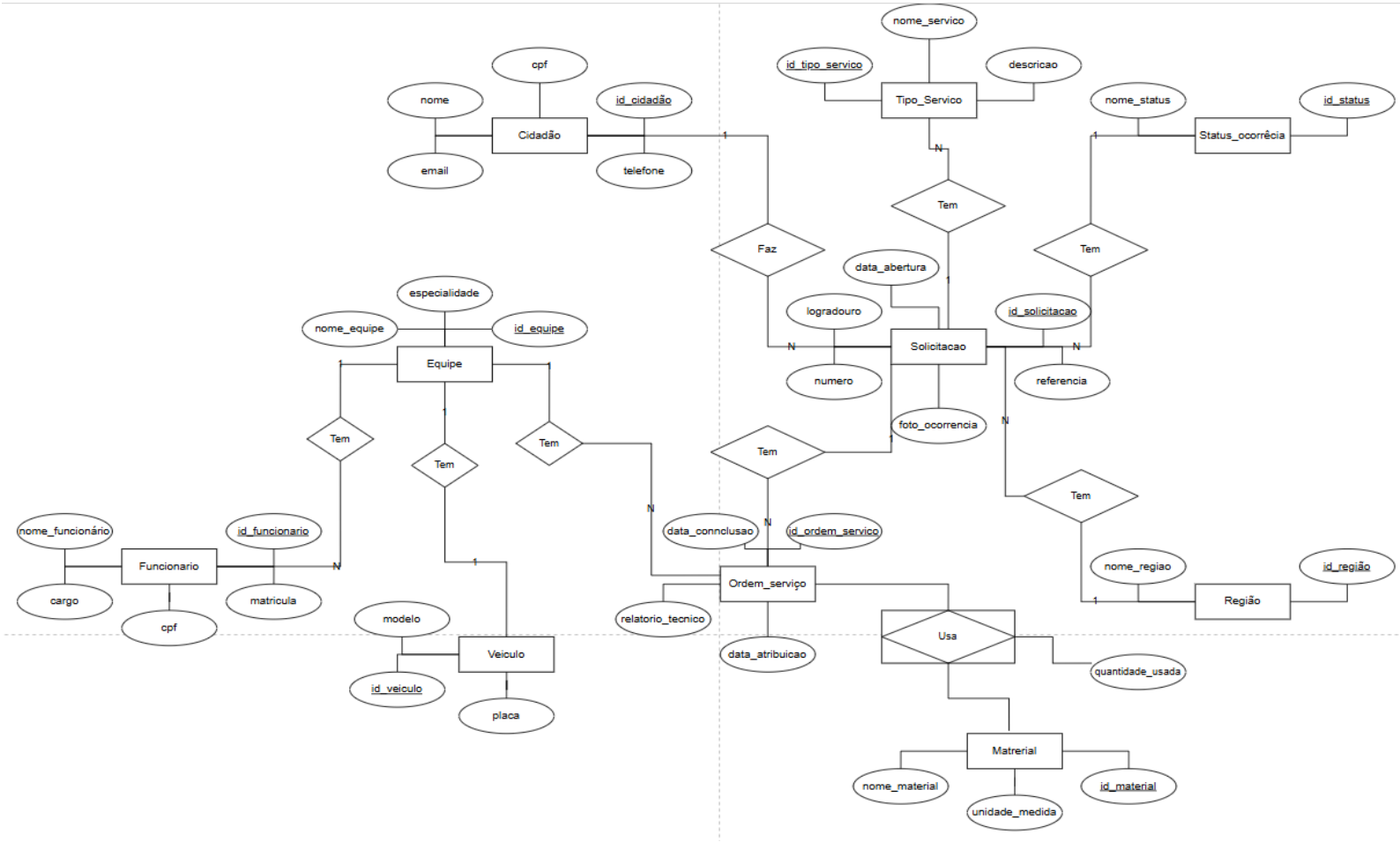
## Introdução

O trabalho consiste em um sistema de zeladoria, em que é permitido notificar as autoridades sobre problemas urbanos (Poste queimado, buracos na pista, placas desgastadas etc). A ideia é o cidadão poder fazer solicitações, informando a região do problema e o tipo de serviço necessitado, e a Administração irá definir a ordem de serviço, o veículo que será enviado, a equipe combatente (com seus funcionários), a ordem de serviço e os materiais comprados e utilizados. Assim, este projeto pode ser usado para otimizar e aumentar a eficácia da gestão de uma cidade.

# MR

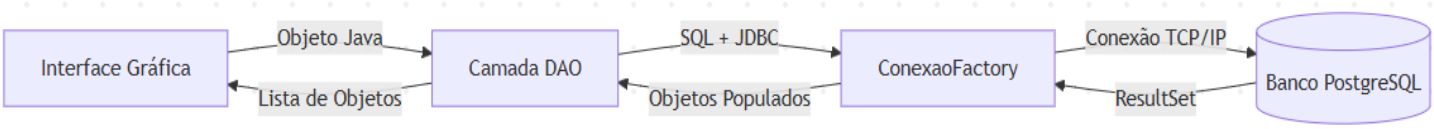

Regiao
PK id_resgiao
nome_regiao
Material
PK id_matrerial
nome_material
unidade_medida
Cidadao
PK id_cidadao
nome_cidadao
email_cidadao
telefone_cidadao
cpf_cidadão
Material_Utilizado
PK,FK1 id_material
PK,FK2 id_ordem_servico
quantidade_usada
Solicitacao
PK id_solicitacao
FK id_regiao
FK id_cidadão
FK id_tipo_servico
FK id_status
logradouro
data_abertura
numero_solicitacao
foto_ocorrencia
referencia
Tipo_Servico
PK id_tipo_servico
nome_servico
descricao
Ordem_Servico
PK id_ordem_servico
FK id_solicitacao
FK id_equipe
data_atribuicao
data_conclusao
relatorio_tecnico
Status_ocorrencia
PK id_status
nome_status
Equipe
PK id_equipe
especialidade
Veiculo
PK id_veiculo
FK id_equipe
modelo_veiculo
placa
Funcionario
PK id_funcionario
FK id_equipe
nome_funcionario
cargo
matrícula
cpf_funcionário
N:1
N:1
N:1
N:1
N:1
N:1
N:1
1:N

# MER




---

# Camada de persistência e mapeamento



Implementação:

- **Model:** Classes Java (Cidadao, Solicitacao) que espelham as tabelas do banco.
- **DAO:** Classes (CidadaoDAO, SolicitacaoDAO) que encapsulam os comandos SQL da CRUD (INSERT, SELECT, UPDATE, DELETE).
- **Factory:** A classe ConexaoFactory que realiza efetivamente a conexão com o PostgreSQL

## Atendimento aos requisitos técnicos do trabalho

### ->Gerador de Chave Primária Automático

Todas as tabelas do sistema utilizam o tipo SERIAL do PostgreSQL para autoincremento dos identificadores.

**Exemplo:** id_solicitacao SERIAL PRIMARY KEY na tabela Solicitacao.

### ->Inserção de Dado Binário (Foto)

O sistema permite o armazenamento de fotos das ocorrências. Foi utilizada a coluna foto_ocorrencia com o tipo BYTEA na tabela Solicitacao.

### ->Trigger

Foi implementada uma Trigger para automatizar o fluxo de trabalho. Quando uma nova Ordem de Serviço é criada, o sistema entende que o trabalho começou e atualiza automaticamente o status da Solicitação original para "Em Andamento" (ID 3).

## ->Procedure

Foi criada a procedure Finalizar_Servico para facilitar o encerramente de uma Ordem de Serviço. Ela realiza duas operações no SGBD: atualiza o relatório técnico e registra o timestamp da conclusão.

## ->View

A view View_Contatos_Por_Regiao foi desenvolvida para facilitar a visualização dos dados, cruzando dados de três tabelas (Cidadao, Solicitacao, Regiao) para listar os contatos (email e telefone) dos Cidadãos solicitantes junto de sua localização.

## ->O CRUD para um conjunto de no mínimo 3 tabelas com relacionamento entre elas

O CRUD está funcionando para todas as 10 tabelas do banco de dados. A tabela Solicitacao, por exemplo, possui 4 FKs: id_regiao, da tabela Regiao; id_cidadao, da tabela Cidadao; id_tipo_servico, da tabela Tipo_Servico; id_status, da tabela Status_Ocorrencia. É possível criar uma Solicitacao pela interface gráfica do projeto.

## ->Interface gráfica para usuário leigo

O sistema possui diversas telas que permitem criar, ler, atualizar e deletar dados do banco de forma intuitiva.

## ->Uso de IA

Foi usado Inteligência Artificial no projeto principalmente para o arquivo ConexaoFactory, responsável por estabelecer a conexão do projeto com o Banco de Dados; no desenvolvimento das telas; e nas

primeiras CRUDs feitas, com o objetivo de aprender como funciona e como fazer para replicar nas demais. Também foi usado nos scripts de Trigger, Procedure e View para corrigir erros e bugs.

## Scripts SQL

No Github do projeto, a view, procedure e trigger estão no arquivo scripts_avançados.sql e o ddl e dml de criação e população do banco estão nos arquivos ddl.sql e dml.sql.