Universidade Federal de Mato Grosso do Sul - UFMS
Sistemas de Informação
Laboratório de Banco de Dados
Docente Prof. Vanessa Borges

Trabalho Prático
Aplicação WEB com SGDB Relacional

Discente Guilherme Carvalho
RGA 2018.1907.071-9

20/11/2020

Índice

Descrição do Trabalho

Implementar uma aplicação web integrada com um SGBD relacional.

Requisitos do Trabalho

Modelo Relacional

- 1 relacionamento **n:n**
- 1 relacionamento **1:n**
- 1 relacionamento **1:1**
- **5** relações distintas
- Trigger para registro de **log** de todas as alterações

Aplicação

- CRUD
- Frontend
- Backend

Especificação do Problema

Para desenvolvimento deste projeto o discente optou por modelar uma aplicação para realizar a digitalização do negócio de uma lavanderia fantasia.

O Problema

Para aprimorar a eficiência, agilidade, atendimento e faturamento da referida lavanderia, é preciso modernizar o negócio. Para tal, deve-se integrar um sistema informativo.

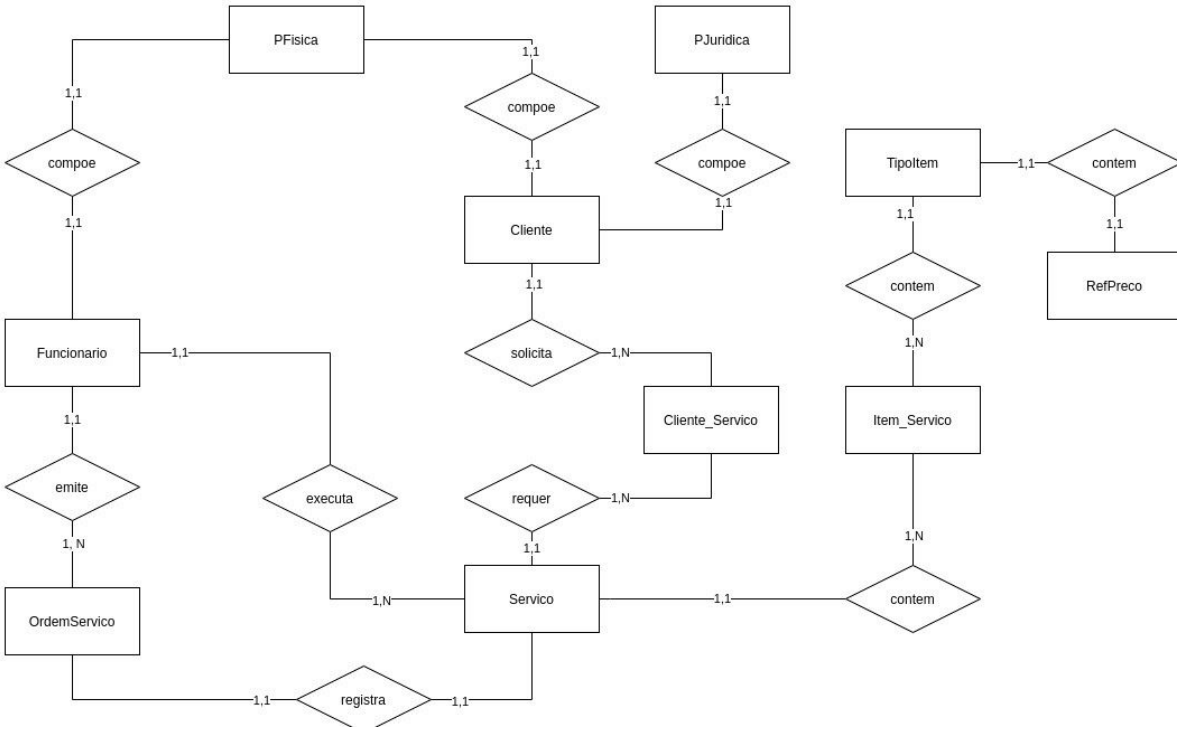
Atualmente os processos são gerenciados por meio de cadernos de anotações e planilhas. Em dias de alta demanda este sistema de gerenciamento se mostra ineficiente pois não garante a confiabilidade do processo. Ocasionando em falhas graves como a não entrega de peças de vestuário, roupas de cama, de mesa etc.. Ou até mesmo a entrega equivocada, ou seja, um cliente recebe as peças de outros clientes. Por conta desses eventos indesejáveis os clientes estão optando por outras lavanderias, impactando diretamente na empresa, podendo levá-la à falência.

Visando uma solução para tal problema, o dono optou por realizar a informatização da empresa e de seus processos.

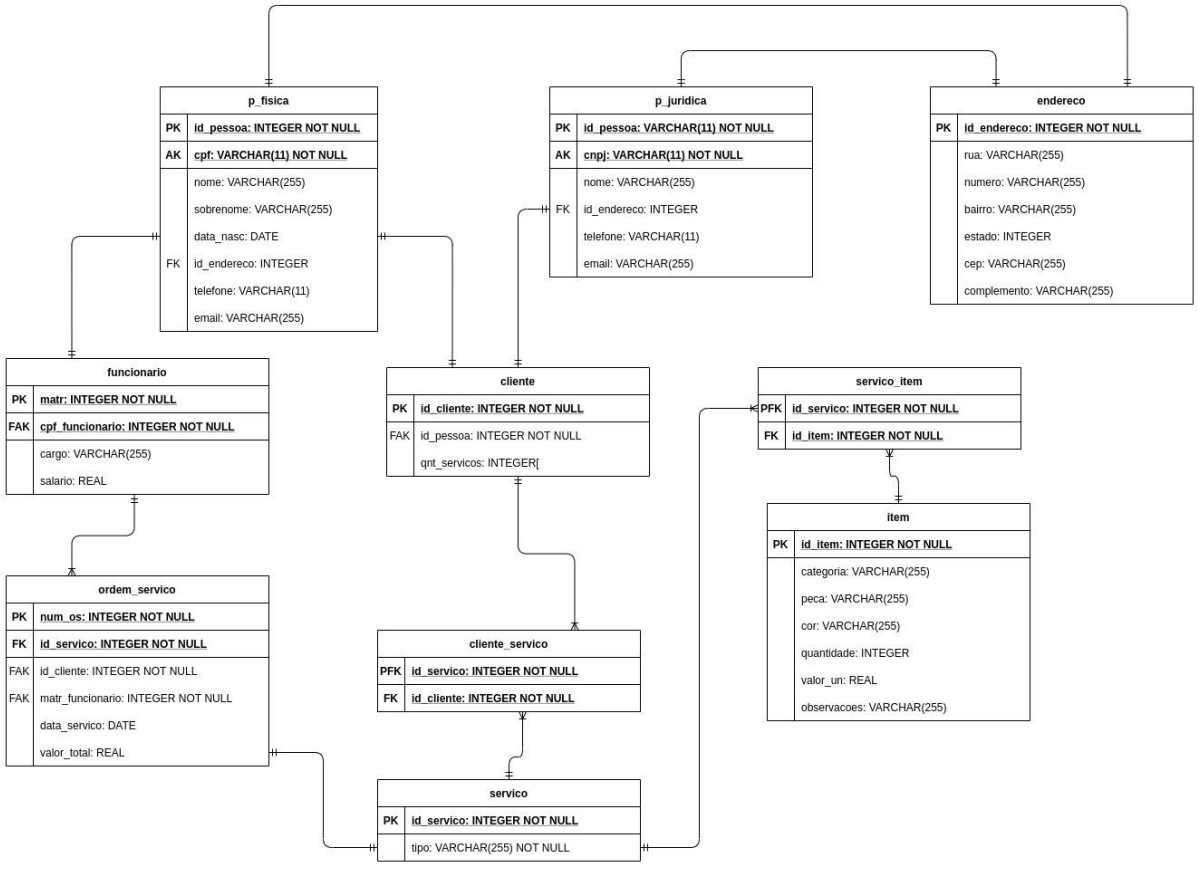
Esquema Relacional e Trigger

Para criação do banco foram desenvolvidos três modelos:

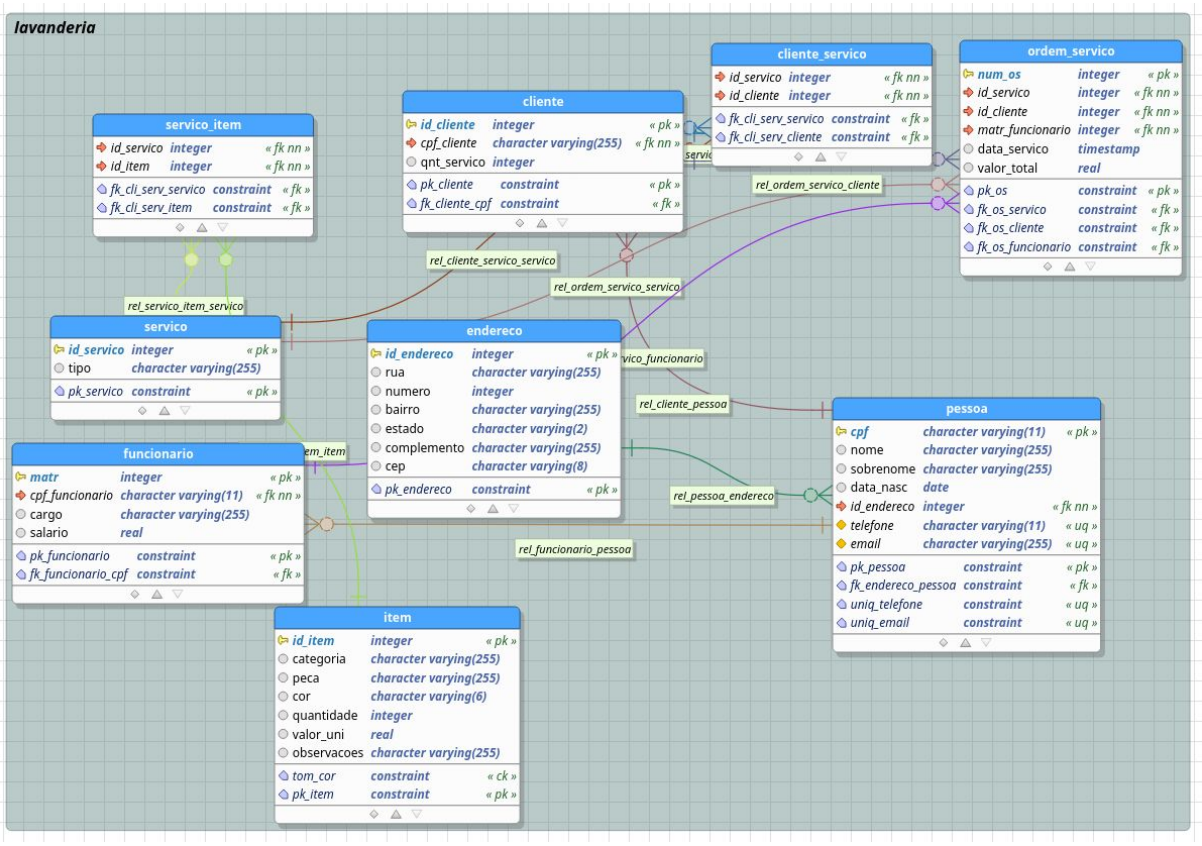
Modelo Relacional - Inicial



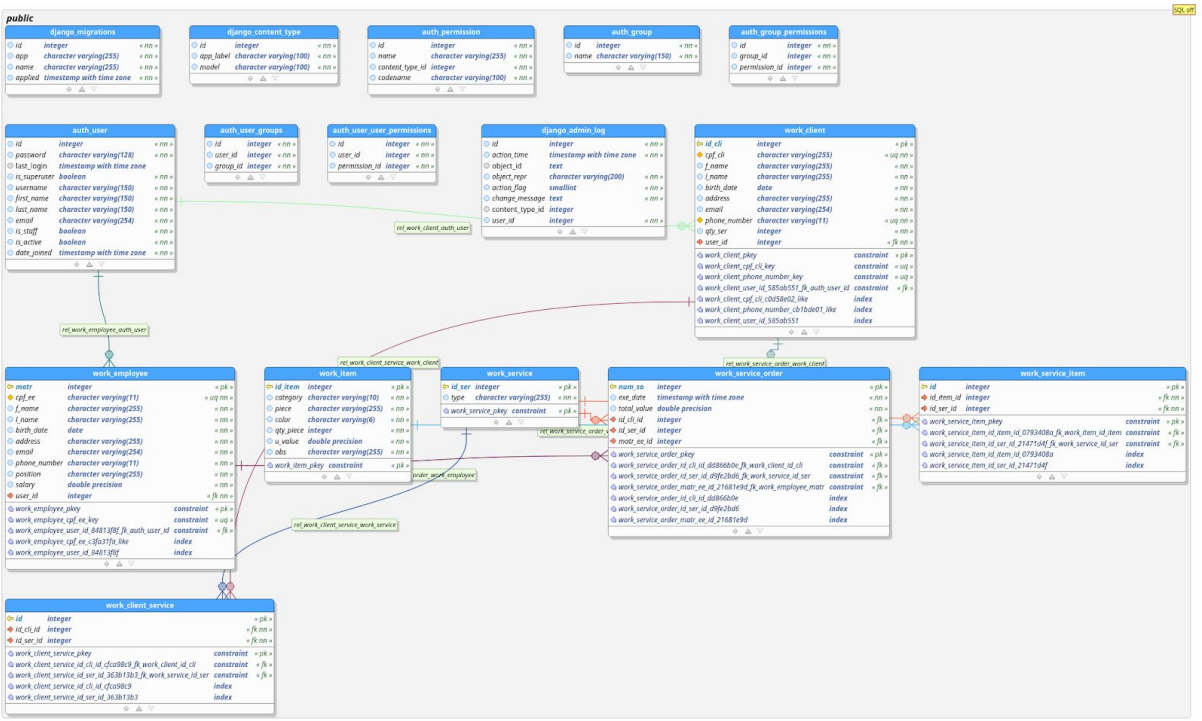
Modelo Relacional - Normalizado



Modelo gerado no pgModeler através do sql executado no banco de dados postgres



Modelo gerado pelo ORM do Framework Django



Descrição do Modelo

A lógica do projeto foi de que, a lavanderia conteria Funcionários e Clientes, sendo os funcionários são Pessoas Físicas com certos atributos e os Clientes poderiam ser tanto Pessoas Físicas quanto Pessoas Jurídicas, ambos tendo um Endereço de localização. A lavanderia executaria Serviços os quais são compostos por Itens e geram uma Ordem de Serviço.

Com o correr do desenvolvimento do projeto enfrentei dificuldades com os conceitos do framework e por isso tive dificuldade na implementação. Por conta disso, tive de alterar as tabelas para que elas pudessem funcionar minimamente. A remoção da tabela de Endereços é um exemplo.

A trigger para geração de logs não foi implementada.

Tecnologias Utilizadas

Para este projeto dentre as ferramentas obrigatórias permitidas as escolhidas foram:

- Linguagem: Java, Python ou PHP
  - Escolha: Python
- Banco de Dados: PostgreSQL ou MySQL
  - Escolha: PostgreSQL

Definidas as ferramentas obrigatórias solicitadas, segue as ferramentas adicionais optadas pelo discente:

- SandBox
  - Docker Containers
  - Docker-Compose
  - Devcontainer
  - Virtualenv
- Editor
  - VS Code
- Framework
  - Django
- Modelagem
  - Diagrams.net
  - pgModeler
- Ferramenta de Gerenciamento do Banco de Dados
  - pgAdmin

Tutorial de Instalação

O projeto também se encontra no repositório do Github.

Para iniciar o projeto deve-se instalar as ferramentas **docker** e **docker-compose** e após a instalação é necessário executar o seguinte comando na raíz do projeto:

- **Subir serviço**
    - docker-compose up -d --build
- **Criar *migrations***
    - docker exec -i -t app python manage.py makemigrations work
    - docker exec -i -t app python manage.py migrate
- **Criar *superuser* responsável por executar as operações**
    - docker exec -i -t app python manage.py createsuperuser
- **Parar serviço**
    - docker-compose down

Isso fará com que os serviços da aplicação, postgresql e pgAdmin subam. Para acessar os respectivos serviços basta acessar as seguintes URLs:

- Aplicação
    - 127.0.0.1:8000
- PgAdmin
    - 127.0.0.1:80
    - **Para identificar o IP do banco de dados postgresql execute:**
        - docker inspect app-db | grep IPAddress

Conclusão

O trabalho foi implementado de maneira insatisfatória e contém funcionalidades mínimas. Para um desenvolvimento satisfatório deveria ser desenvolvido a integração que permite a inserção de dados dos Funcionários e Clientes corretamente, sendo possível ambos adicionarem e editarem seu endereço, porém tive dificuldades com lógica e exibição dos formulários. Para criação de Serviços deveria adicionar um formulário de item interativo, com um botão capaz de criar mais formulários para adição de mais itens. A criação da ordem de serviço deveria ser automática, sendo criada logo após a criação de um serviço.

Em suma, a falta de comprometimento com o projeto, aliada à inexperiência com framework escolhido resultou em um trabalho não tão bom quanto poderia ter sido.