UNIFACISA

FASE 1

Locadora de Equipamentos

Brenner Vinicius Pereira Silva
João Lucas Ramalho Barros
Linnik Patricio de Sousa Souto
Matheus Aurelio de Lima

Campina Grande, PB

07 de Novembro de 2025

1 – Introdução

Este projeto tem como objetivo o desenvolvimento de um banco de dados relacional completo para uma Locadora de Equipamentos. O sistema foi projetado para permitir o gerenciamento de clientes, categorias, equipamentos, locações, devoluções, pagamentos e auditorias. Utilizando PostgreSQL em um ambiente Docker, o projeto visa garantir desempenho, integridade e rastreabilidade das operações.

2.1 - REQUISITOS FUNCIONAIS

- O sistema deve permitir o cadastro, edição, exclusão e listagem de clientes.
- O sistema deve permitir o cadastro, edição, exclusão e listagem de categorias de equipamentos.
- O sistema deve permitir o cadastro e gerenciamento de equipamentos, vinculando-os às categorias.
- O sistema deve permitir abrir novas locações, vinculando cliente e equipamentos alugados.
- O sistema deve permitir adicionar e remover itens a uma locação aberta.
- O sistema deve verificar automaticamente a disponibilidade de equipamentos antes da locação.
- O sistema deve atualizar automaticamente a quantidade de equipamentos disponíveis ao serem alugados ou devolvidos.
- O sistema deve registrar devoluções, vinculando os itens devolvidos e seu estado.
- O sistema deve calcular o valor total estimado da locação com base nas diárias negociadas e dias estimados.
- O sistema deve calcular multas por atraso automaticamente, considerando as regras da categoria do equipamento.

- O sistema deve gerar relatórios e views detalhadas, como extrato de clientes e histórico de locações.

- O sistema deve registrar auditoria de operações críticas em uma tabela própria.

2.2 - REQUISITOS NÃO FUNCIONAIS

- O sistema deve utilizar PostgreSQL como banco de dados relacional principal.

- O ambiente de execução deve ser containerizado via Docker, garantindo portabilidade e isolamento.

- O sistema deve garantir integridade referencial entre as tabelas utilizando chaves estrangeiras e constraints.

- O sistema deve responder a consultas e operações comuns em tempo inferior a 3 segundos.

- O sistema deve possuir logs e auditoria para rastrear todas as operações críticas.

- Os scripts SQL devem ser documentados e separados em DDL, DML e Views, para facilitar manutenção e reuso.

2.3 - REGRAS DE NEGÓCIO

- Um equipamento só pode ser alugado se seu status for 'disponivel'.

- A data prevista de devolução deve ser posterior à data da locação.

- O valor da diária padrão e da multa são definidos pela categoria do equipamento, mas podem ser negociados na locação.
- A multa por atraso é calculada com base na quantidade de dias de atraso e na multa diária da categoria.

- O status da locação deve ser atualizado automaticamente conforme o progresso (aberta, em curso, concluída, etc).

- Ao alugar um equipamento, sua quantidade disponível é reduzida em 1 unidade.

- Ao registrar uma devolução, a quantidade disponível do equipamento é incrementada em 1 unidade.

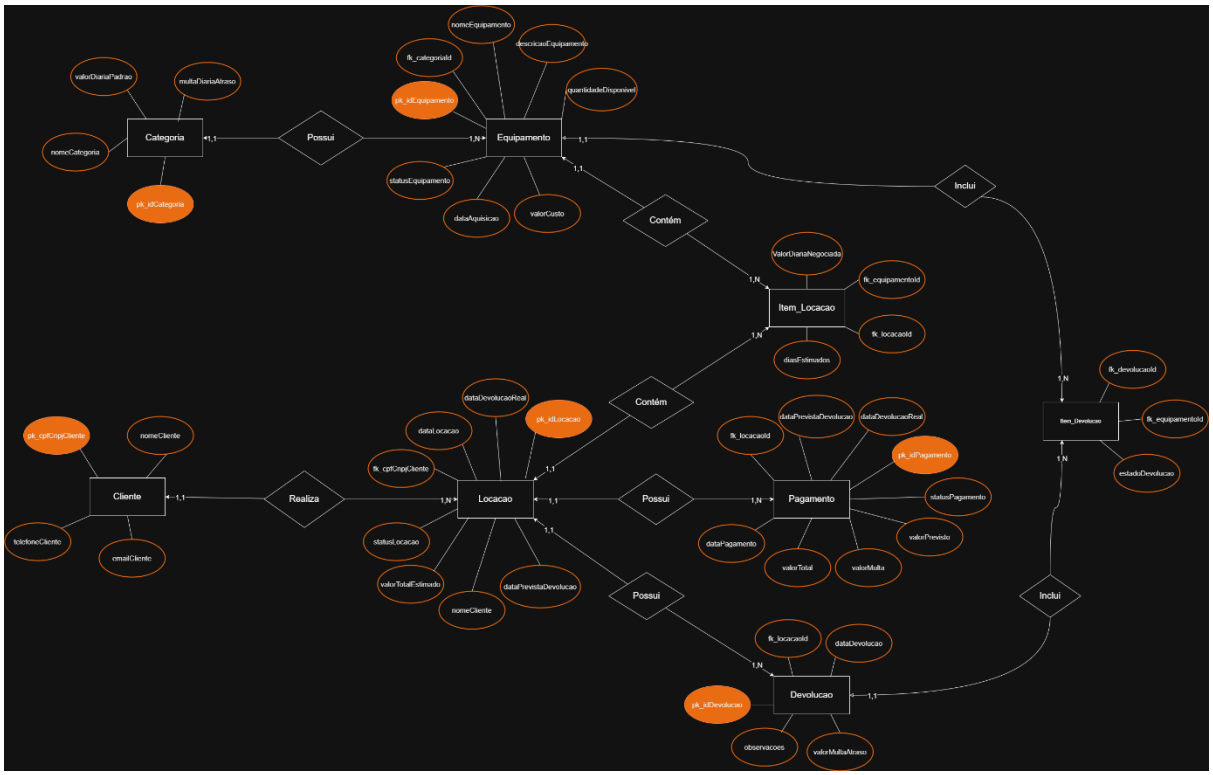
- A devolução deve registrar o estado dos equipamentos devolvidos.

- Cada pagamento deve estar vinculado a uma locação e ser atualizado automaticamente após a devolução.

- A auditoria deve registrar todas as inserções, atualizações e exclusões em tabelas principais do sistema.

3 – Modelo Diagrama

- No nosso modelo, optamos por utilizar a ferramenta “draw.io” por ser uma ferramenta já utilizada anteriormente em outras competências e consequentemente de maior familiaridade com a utilização. Desta forma, acreditamos que foi mais eficiente.



- Arquivo do Modelo: ..\Desktop\Locadora de Equipamentos.drawio.pdf

4 – Popular Dados (DML)

Para popular o banco de dados com registros de teste, executamos o script de DML que insere categorias, equipamentos, clientes, locações, itens de locação, devoluções e pagamentos.

Passos realizados:

- Garantir que o container PostgreSQL esteja em execução (docker compose up -d).
- Copiar o script populate_locadora.sql para o diretório onde se encontra o container ou torná-lo acessível.
- Executar o script dentro do container com o comando:
docker exec -i postgres_locadora psql -U admin -d locadora < populate_locadora.sql

O script populate_locadora.sql adiciona registros de exemplo que permitem testar os gatilhos e regras de negócio, tais como redução/repôs de estoque, cálculo automático de valor estimado da locação e cálculo de multas em pagamentos.

5 – Views (Relatórios e Consultas

Foram criadas views que consolidam informações das locações e do extrato dos clientes, facilitando consultas e relatórios.

Views implementadas:

1. vw_locacoes_detalhadas — objetivo: fornecer uma visão completa de cada locação, incluindo:

• Identificador da locação (id_locacao).

• Nome e CPF/CNPJ do cliente.

• Data da locação, data prevista de devolução e data de devolução real.

• Status da locação (aberta, emcurso, concluida, concluida_com_atraso, cancelada).

• Valor previsto/estimado da locação, valor da multa e valor total pago.

• Total de itens alugados, lista de equipamentos alugados e categorias envolvidas.

Essa view facilita consultas para análises como: faturamento por período, valores de multas aplicadas, equipamentos mais alugados e composição das categorias por locação.

2. vw_extrato_cliente — objetivo: fornecer o histórico financeiro do cliente, incluindo:

• Nome e CPF/CNPJ do cliente.

• Identificador da locação e datas relevantes (locação, prevista e real).

• Status da locação.

• Valor total estimado da locação, valor da multa e valor pago.

Essa view facilita a geração de extratos e relatórios de pagamentos, permitindo identificar clientes inadimplentes, histórico de locações e receita por cliente.

Adicionalmente, é possível criar views derivadas que filtrem apenas locações ativas ou apenas pagamentos com status 'concluido' para facilitar consultas específicas (ex.: locações em curso ou faturamento do mês).

Exemplo de uso das views em consultas:
- Listar locações em curso: SELECT * FROM vw_locacoes_detalhadas WHERE status_locacao = 'emcurso';
- Extrato de um cliente específico: SELECT * FROM vw_extrato_cliente WHERE cpf_cnpj_cliente = '222.222.222-22' ORDER BY data_locacao DESC;

6 – Execução Final e Observações

Após criar o esquema (DDL) e popular os dados (DML), as views e triggers passam a operar automaticamente. Recomenda-se verificar os logs do container e executar queries de validação para assegurar que as triggers (por exemplo, redução de estoque e cálculo de multas) estão funcionando como esperado.

Comandos úteis:

- Subir o ambiente: docker compose up -d
- Executar DDL: docker exec -i postgres_locadora psql -U admin -d locadora < ddl_locadora_postgres.sql
- Executar DML: docker exec -i postgres_locadora psql -U admin -d locadora < populate_locadora.sql
- Acessar o container: docker exec -it postgres_locadora bash

GITHUB DO PROJETO

- Link: Github Locadora de Equipamentos