

UNIVERSIDADE SALVADOR – UNIFACS

Ciência da Computação

Equipe:

Manoel Duran - RA: 1272214667,

Hélio Júnior - RA: 12722129211,

Lukas Souza - RA: 1272215050

Data: 30/11/2023

RELATÓRIO DO PROTÓTIPO DE SISTEMAS DISTRIBUIDOS



Salvador, BA

2023

Introdução

Visando resolver o problema de captação de dados nas vendas de sua rede de lojas de uma maneira econômica, uma empresa de departamentos solicitou aos alunos da turma de Sistema Distribuídos da UNIFACS. A solicitação era clara: desenvolver um sistema capaz não apenas de solucionar a problemática em questão, mas de fazê-lo de maneira economicamente viável.

A partir desse ponto, foram definidas algumas das prioridades de desenvolvimento desse programa, como por exemplo: Criação, remoção, edição e leitura das informações dos clientes e estoque. Além disso, a capacidade de receber pedidos de compra e a elaboração de relatórios estatísticos foram identificadas como elementos essenciais. Esses relatórios, compostos por listagens que abrangem desde os produtos mais vendidos até o consumo médio por cliente, foram estrategicamente organizados para oferecer insights valiosos, incluindo análises sobre produtos por cliente e identificação de produtos com baixo estoque.

Com a intenção de maximizar o tempo de entrega, foram escolhidas tecnologias atuais e de alto nível para facilitar o desenvolvimento do software, as tecnologias utilizadas foram Python, JavaScript, SQL, SQLite no banco de dados e API para levar os dados ao usuário.

A presente narrativa não apenas registra o processo de criação do sistema, que tinha como foco principal a resolução do problema apresentado pela rede de lojas, porém também visando explorar e detalhar a metodologia aplicada. Durante o relatório, serão abordados todos os caminhos até a versão final do projeto, oferecendo uma visão geral do trabalho desenvolvido.

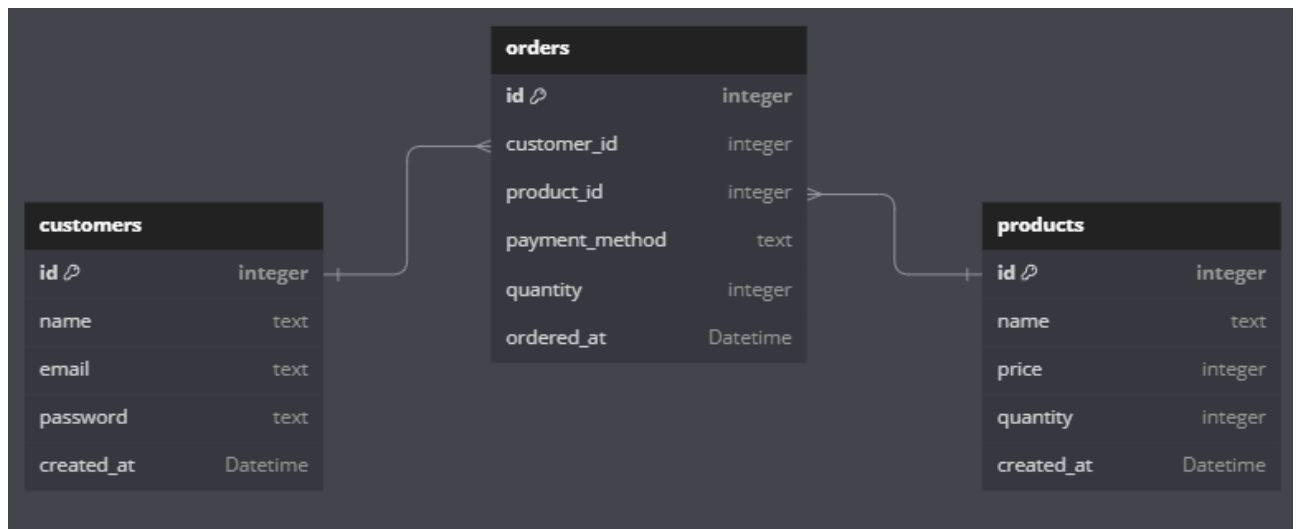
Fundamentação Teórica

Como comentado anteriormente, os membros do projeto optaram pelo uso da linguagem de programação Python para fazer o servidor do sistema (Backend), os principais motivos por trás dessa escolha foram a facilidade de criação, legibilidade e manutenção para futuros desenvolvedores, devido ao fato de ser uma camada fundamental para o funcionamento do programa, por manter toda a regra de negócio do código.

A estrutura de conexão de dados escolhida para utilizar no sistema foi a interface de programação de aplicativos (API). Ela que possibilita a ligação cliente-servidor, tem como principais vantagens a simplificação, fornecendo uma camada de abstração que facilita a interação entre sistemas, a segurança, incorporando diversas práticas de autenticação e autorização, além da facilidade de integração com a linguagem escolhida para fazer o Backend, por meio da biblioteca FastAPI.

Para que o usuário tenha acesso aos relatórios, a equipe desenvolveu uma interface visual web utilizando a biblioteca de código aberto da linguagem de programação Javascript, React. A mesma foi escolhida por se tratar de uma das ferramentas mais práticas e completas da atualidade quando se trata de FrontEnd, combinando com a ideia do sistema de se manter atualizado e ter manutenção simples.

Projeto de implementação



Este diagrama representa o esquema de um banco de dados relacional usado para armazenar informações sobre clientes, pedidos e produtos. Ele é composto por cinco tabelas: clientes, pedidos, produtos e tabelas associativas.

- Tabela clientes: Armazena informações básicas dos clientes, como ID, nome, email e senha. Possui um campo de data e hora (created_at) que registra a data e hora em que o registro foi criado.
- Tabela pedidos: Armazena informações sobre os pedidos realizados pelos clientes, como ID do pedido, ID do cliente, ID do produto, método de pagamento, quantidade e data e hora em que o pedido foi realizado (created_at).
- Tabela produtos: Armazena informações básicas dos produtos, como ID, nome, preço e quantidade disponível. Possui um campo de data e hora (created_at) que registra a data e hora em que o registro foi criado.
- Tabelas associativas: Há duas tabelas associativas neste diagrama. A primeira, entre a tabela clientes e a tabela pedidos, armazena a relação entre os clientes e os pedidos que eles fizeram. A segunda, entre a tabela pedidos e a tabela produtos, armazena a relação entre os pedidos e os produtos que eles incluem.

Além disso permite que as informações sejam facilmente recuperadas e manipuladas.

Em nosso frontend, temos como criar, ler, atualizar e deletar (CRUD) as informações do cliente na página de cliente, para chegar nessa etapa é preciso que na página principal selecione o botão de clientes, na página de clientes temos uma tabela onde está feita a listagem de clientes e a criação de um novo cliente. Para cada cliente da tabela, podemos visualizar, atualizar, deletar os dados do cliente, ver o histórico de pedidos que é referente ao

relatório Produto x Cliente, e além disso, podemos ver o consumo médio do cliente, que também é um relatório.

Para os produtos, temos a tabela de produtos e a opção de criar um novo. Para cada produto podemos visualizar, atualizar e deletar os dados do produto. Nesta tela também temos 2 botões que levam para 2 páginas de relatório, a de produtos mais vendidos em que estão listados todos os produtos em forma de ranking, e o botão que leva para a tela de produtos com baixo estoque, que também é um relatório.

Temos também a tela de pedidos, nesta tela podemos criar um novo pedido ou ver a tabela de pedidos criada. Para cada item da tabela podemos visualizar e editar um pedido, lembrando que na edição de um pedido, a alteração feita na quantidade do pedido, impacta diretamente no estoque do produto selecionado. E também é possível deletar um pedido.

Considerações Finais

Ao longo do relatório buscamos enfatizar sobre uma aplicação projetada para simular a captação de dados de venda em uma extensa rede de lojas. Sabendo que a captação eficiente e análise estratégica de dados de vendas são imperativas para o sucesso de qualquer rede de lojas. Buscamos desenvolver uma aplicação econômica, útil e com funcionalidades objetivas em um curto tempo de entrega a qual nos foi concedido.

Durante o desenvolvimento e implementação desta aplicação, buscamos focar na adaptabilidade e na escalabilidade. Isso não apenas garante que estejamos preparados para atender às demandas dinâmicas do presente, mas também nos posiciona estrategicamente para abraçar as oportunidades futuras, à medida que a rede continua a crescer. Com essas informações decidimos então seguir com as tecnologias citadas durante o relatório.

Tivemos também bons feedbacks de terceiros durante a fase teste da aplicação, onde foi possível considerar cada sugestão e incorporar resoluções para problemas ocorridos durante o mesmo.

Bibliografia

PYTHON SOFTWARE FOUNDATION. Python Language Site: Documentation, 2023. Documentação Oficial. Disponível em: <<https://www.python.org/doc/>>. Acesso em: 06 de nov. de 2023.

RAMÍREZ, Sebastián. FastAPI, 2023. Documentação Oficial. Disponível em: <<https://fastapi.tiangolo.com>>. Acesso em: 08 de nov. de 2023.

React. Documentação do React: Hooks - useState. [S.l.], 2022. Disponível em: <<https://reactjs.org/docs/hooks-state.html>>. Acesso em: 20 de nov. de 2023.