UNIVERSIDADE DO OESTE DE SANTA CATARINA – UNOESC
CURSO DE CIÊNCIA DE DADOS E INTELIGÊNCIA ARTIFICIAL

PRÁTICAS EXTENSIONISTAS IV
ENTREGA FINAL

Desenvolvedor: Felipe Cabral dos Santos

Chapecó, 03 de dezembro de 2025.

1. Introdução

Nesse trabalho buscarei solucionar o problema de estoque de uma pequena empresa de confecção de roupas e uniformes. Atualmente o controle de estoque é realizado através do papel, o que torna ineficiente e impreciso.

2. Modelagem de Entidade e Relacionamento (Conceitual e Lógico)

2.1 Conceitual

Nessa primeira etapa é necessário o entendimento do negócio, para evitar erros ou omissões no início do projeto e servindo de base para as próximas etapas. Utilizaremos duas entidades, estoque e pedido com seus respectivos atributos.

TabelaEstoque:

idproduto, tem por objetivo identificar qual é o produto.

descproduto, nome ou descrição do produto.

estoqueproduto, quantidade em estoque.

unidadedemedida, identifica como o produto é quantificado (metros, metros quadrados ou unitários).

TabelaPedido:

idpedido, busca distinguir os pedidos.

nomepedido, nome atribuído ao pedido.

descpedido, descrição do pedido e/ou detalhes importantes.

qtadpedido, quantidade do pedido.

statuspedido, situação do pedido (aberto, finalizado, cancelado).

datainicio, data em que o pedido foi iniciado.

datafim, data em que o pedido foi finalizado.

2.2 Lógico

Com a parte conceitual bem definida é possível prosseguir com a modelagem lógica, definindo os tipos de dados e restrições.

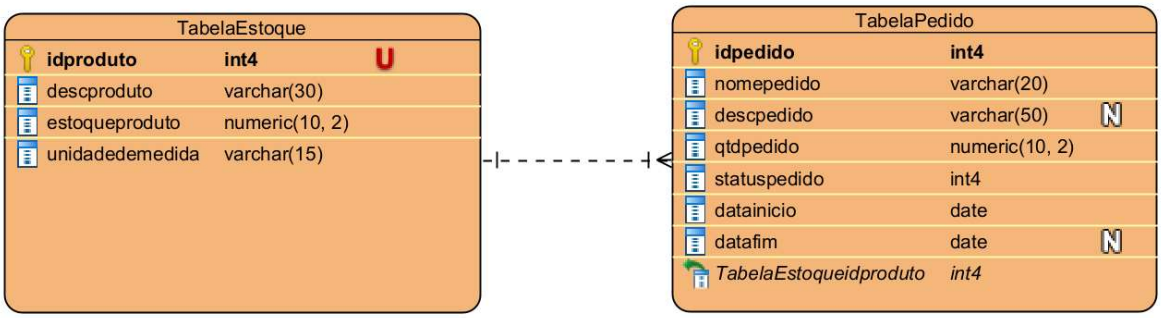
TabelaEstoque:

idproduto: Chave primária da tabela. Identificador único de cada produto, será armazenado como inteiro de até quatro casas decimais (int4).

descproduto: Descrição do produto, será armazenada como texto com até 30 caracteres (varchar(30)).estoqueproduto: Quantidade disponível em estoque, será armazenada como número decimal com até 10 dígitos no total, sendo 2 após a vírgula (numeric(10,2)).
unidadedemedida: Unidade de medida do produto, será armazenada como texto com até 15 caracteres (varchar(15)).

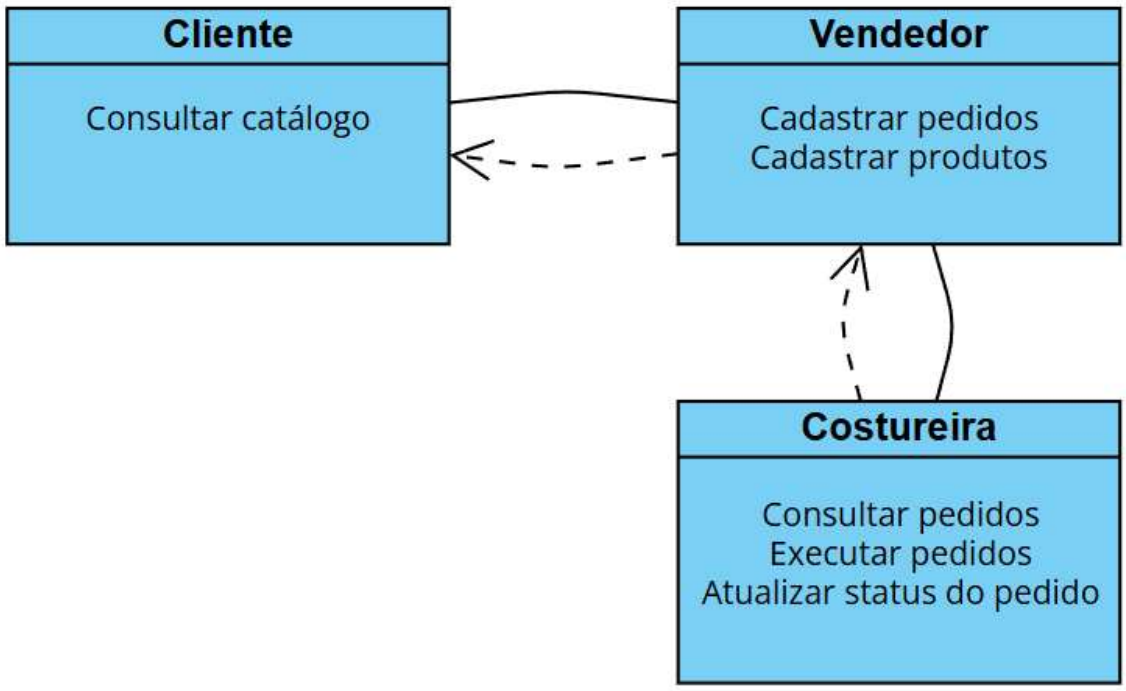
TabelaPedido:

idpedido: Chave primária da tabela. Identificador único de cada pedido, será armazenado como inteiro de até quatro casas decimais (int4).
nomepedido: Nome do pedido ou do solicitante, será armazenado como texto com até 20 caracteres (varchar(20)).
descpedido: Descrição do pedido, será armazenada como texto com até 50 caracteres (varchar(50)).
qtdpedido: Quantidade solicitada no pedido, será armazenada como número decimal com até 10 dígitos no total, sendo 2 após a vírgula (numeric(10,2)).
statuspedido: Indica o status do pedido, será armazenado como inteiro de até quatro casas decimais (int4).
datainicio: Data de início do pedido, será armazenada no formato de data (date).
datafim: Data de finalização do pedido, será armazenada no formato de data (date).
TabelaEstoqueidproduto: Chave estrangeira que referencia o campo idproduto da tabela.



3. Diagrama de Classes

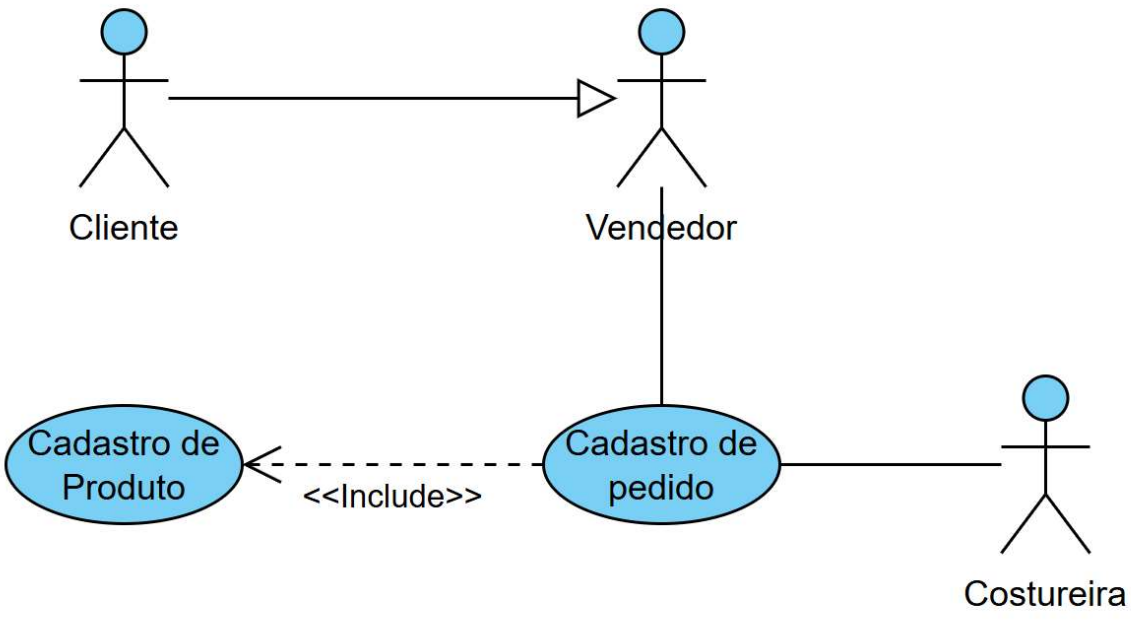
Tem por objetivo representar a estrutura de um sistema, tornando clara a comunicação com a equipe. Muita das vezes é utilizado para análise requisitos ou documentação de sistema para facilitar o entendimento de novos desenvolvedores.



4. Diagrama de Caso de Uso Geral

O objetivo do caso de uso é ajudar entender a regra de negócio, representando os atores e casos de uso.

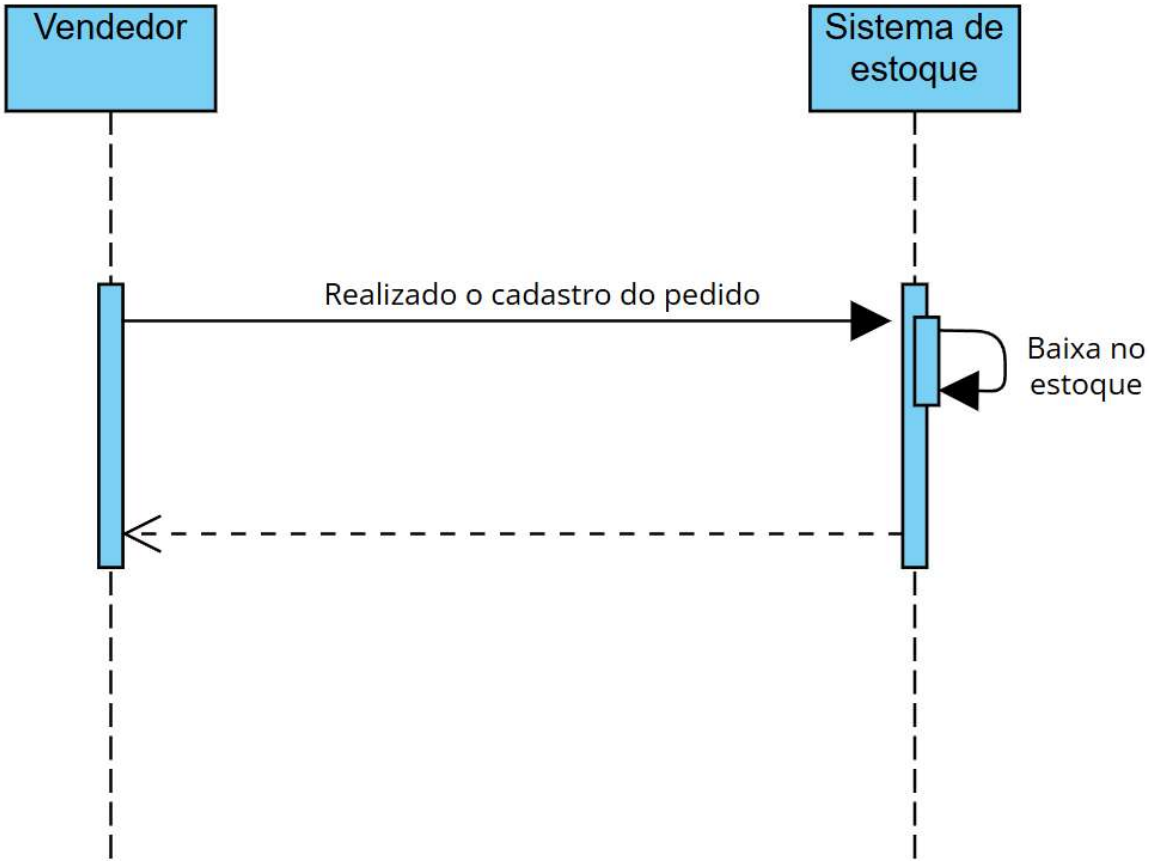
Nessa aplicação os atores são cliente, costureira e vendedor, os casos de uso são cadastro de pedido e de produto



O cliente realiza o pedido com o vendedor, o mesmo realiza o cadastro do pedido e se necessário o cadastro de novos produtos, posteriormente a costureira acessa o pedido.

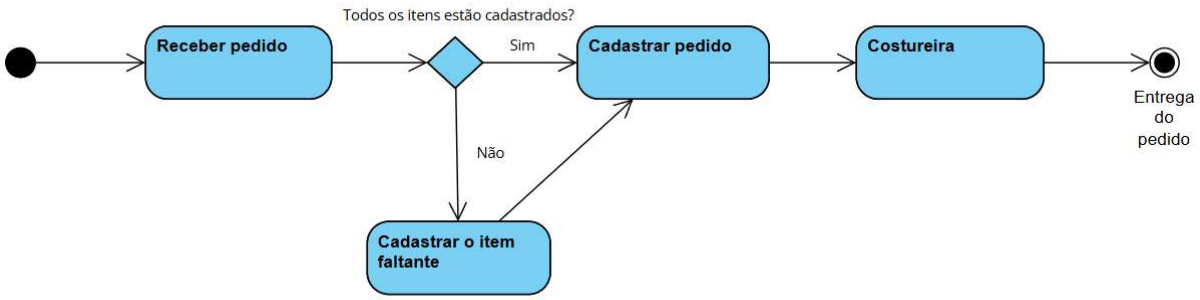
5. Diagrama de Sequência

O diagrama de sequência busca representar a interação entre entidades ou objetos ao longo do tempo



6. Diagramas de Atividades

Tem por objetivo modelar o fluxo de trabalho no processo, representando o controle entre ações. Tem um início e fim bem definidos, atividades realizadas e ramificação de decisões.



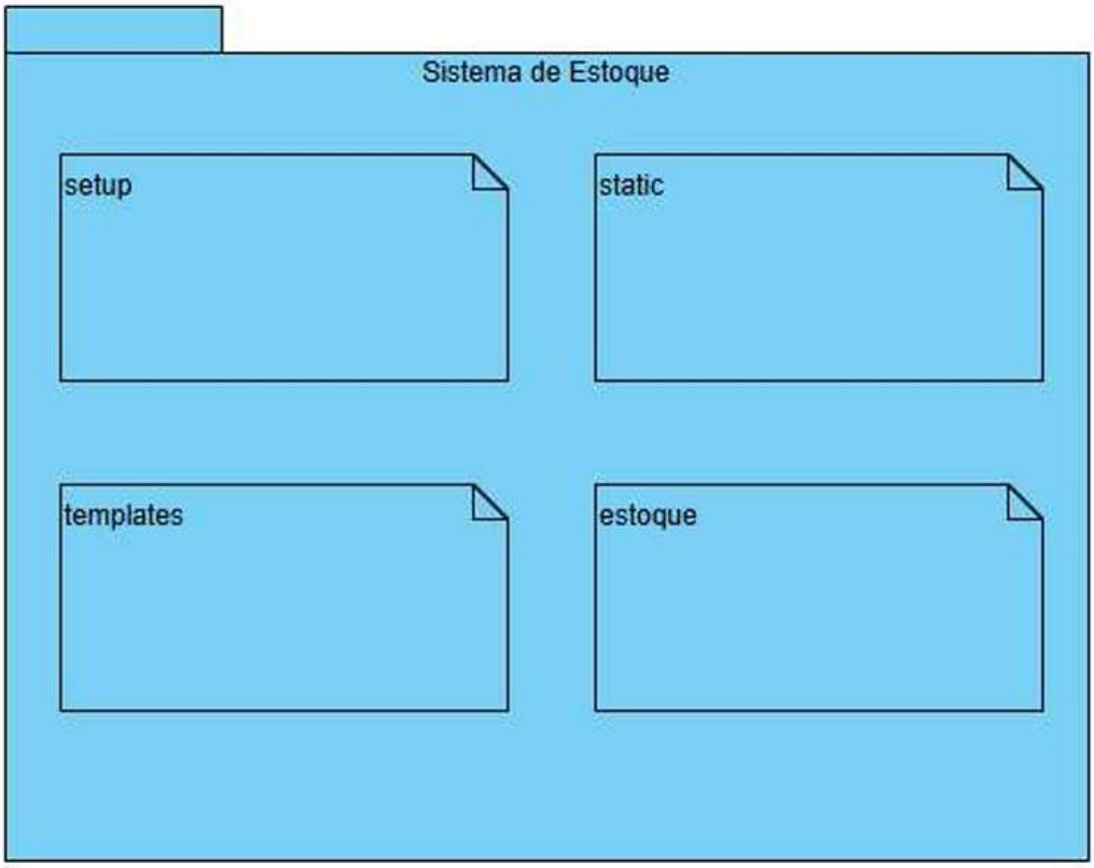
7. Práticas Extensionistas IV.

7.1. Introdução.

Com base na matéria Práticas Extensionistas III buscarei continuar o desenvolvimento do sistema, o mesmo se encontra funcional mas possui muitos pontos de melhorias. 2.2. Diagrama de pacotes. O sistema possui quatro principais pastas que ajudam organizar o projeto.

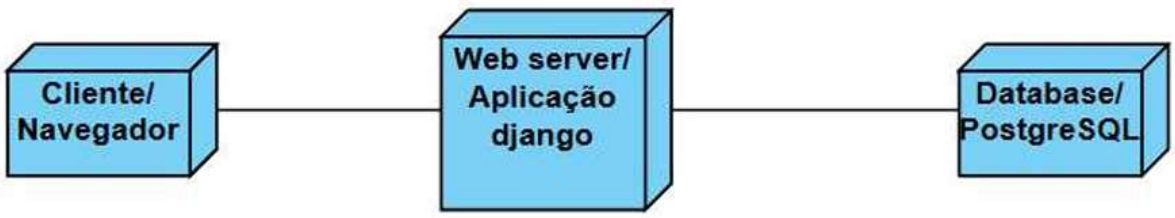
7.2. Diagrama de pacotes.

O sistema possui quatro principais pastas que ajudam organizar o projeto.



7.3.Diagrama de arquitetura de implementação.

O Cliente/Navegador é responsável por enviar as requisições HTTP/HTTPS ao sistema. Essas requisições são recebidas pelo Web Server, onde está hospedada a aplicação Django, responsável por processar as regras e acessar os dados. A comunicação entre o servidor web e o banco de dados PostgreSQL ocorre por meio do ORM do Django, que executa consultas SQL e retorna os resultados para o servidor, que então envia a resposta ao cliente.



7.4. Diagrama de arquitetura de DevOps.

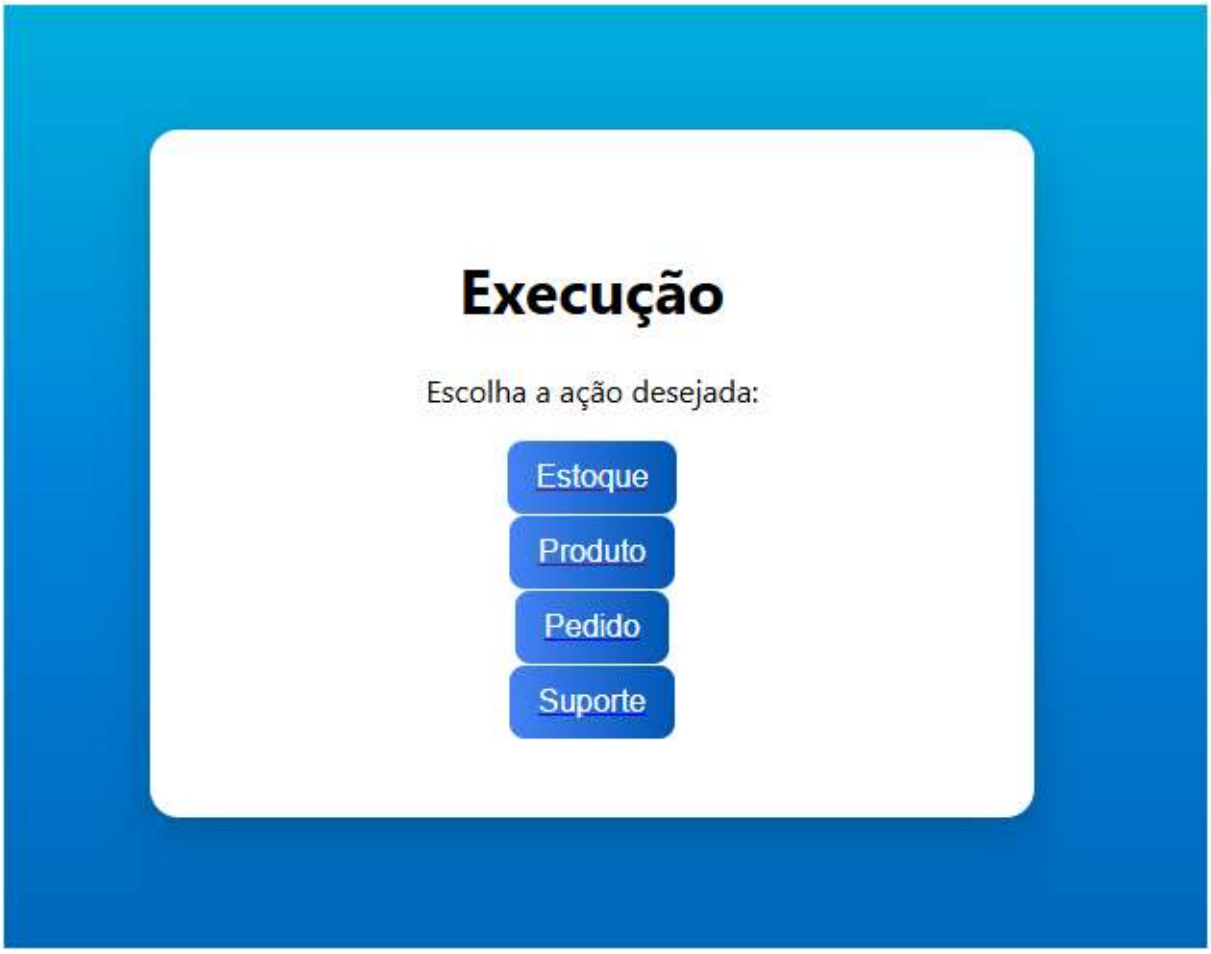
O processo inicia no VSCode onde é desenvolvido e testado, em seguida é enviado para o GitHub que armazena e versiona o código, o Render é integrado ao repositório do Github, sempre que solicitado ele atualiza as mudanças.

7.5 Remodelagem das telas.

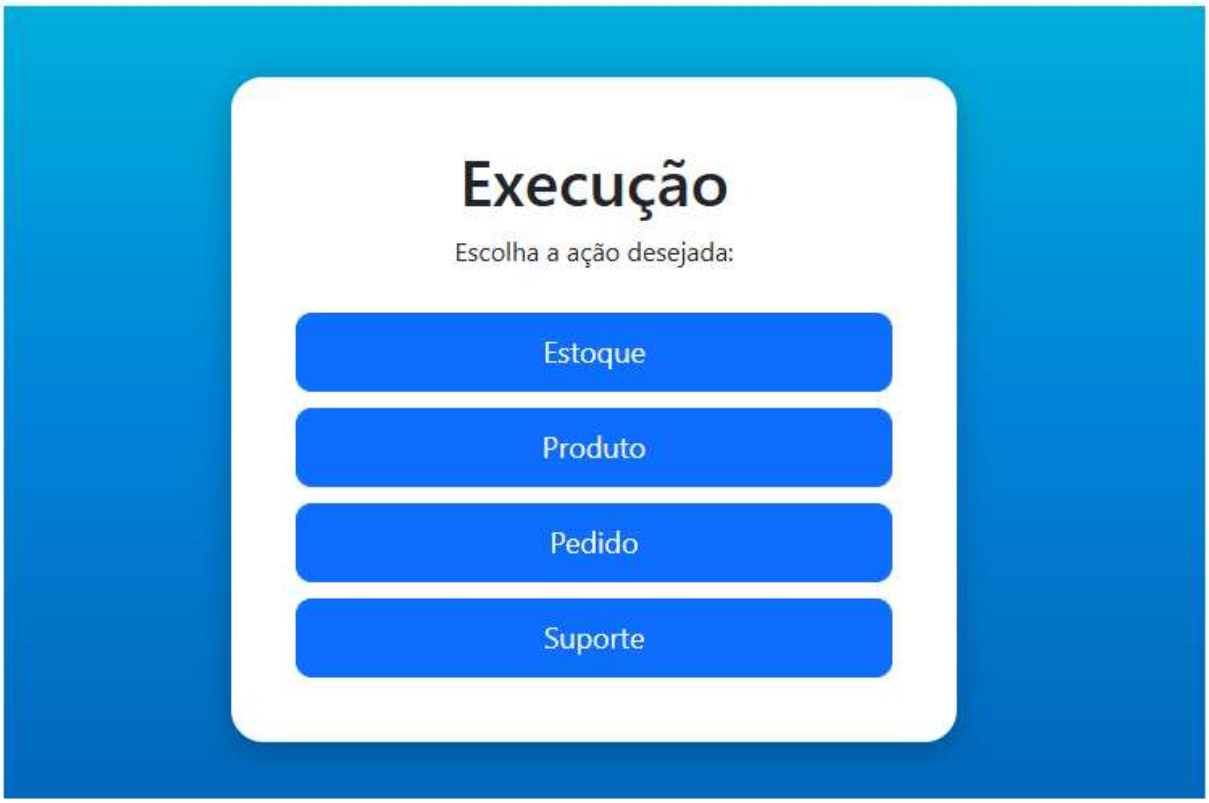
Inicialmente o projeto foi construído com o intuito de ser utilizado via desktop, mas com a demanda mobile o bootstrap foi adicionado ao projeto, porém todas as telas precisaram de algum nível de ajuste para se adaptar ao novo meio. A imagem “background.jpg” que era usada como fundo em todas as telas foi substituída por um código css, que gera um efeito similar e se adéqua melhor a mobile: background: linear-gradient(to bottom, #00f5e1, #0083d8, #003a8c);

Exemplos (mais exemplos na apresentação):

Tela original:



Tela remodelada:



Tela remodelada no mobile:

7.6. API.

O objetivo do desenvolvimento dessa API é manter o usuário informado sobre o estoque, para isso o sistema fará a leitura dos itens e utilizará um bot do telegram para informar o usuário diretamente no celular. A API foi construída usando Django puro. Utilizando rotas que consultam o banco, transformam os dados em JSON e retornam usando JsonResponse. Hospedando a API na Render, permitindo que qualquer sistema possa acessar os dados de estoque da minha aplicação.

7.7. Infraestrutura Cloud.

O projeto consiste em arquitetura SaaS (Software as a Service), visto que é projetado para ser acessado via navegador sem a necessidade de instalação local, rondando na nuvem e podendo ser utilizado por múltiplos usuários ao mesmo tempo. A nuvem Render é uma boa escolha pois tem bom suporte para django e o banco de dados postgresql. Primeiramente é preciso criar um banco de dados na nuvem e utilizá-lo para configurar o software.

7.8. Links importantes

Código fonte: https://github.com/felipecabraldossantos/Projeto-Pr-ticas-Exntensionistas

Projeto na nuvem: https://projeto-pr-ticas-exntensionistas.onrender.com