

**PROJETO INTERDISCIPLINAR**

**Smart Products**

Cadastro de Produtos

**Alunos:**

|  |  |
| --- | --- |
| **RGM** | **Nome** |
| 29445116 | João Victor Conquista |
| 29943183 | Vinícius Barbosa |
| 29409764 | Murilo Vicente |
| 129240743 | William Dias |
| 29401305 | Mayki Pereira |

São Paulo

2023

**UNIVERSIDADE CRUZEIRO DO SUL**

**PROJETO INTERDISCIPLINAR**

**Smart Products**

Cadastro de Produtos

## Trabalho apresentado como parte do requisito para aprovação na Disciplina de Projeto Interdisciplinar do curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Cruzeiro do Sul.

**Orientadores:** Prof. Artur Ubaldo Marques, Prof. Edidio Rubens Dantas Lima e Prof. Paulo Vilela de Melo

São Paulo

2023**Sumário**

[1. Apresentação: 4](#_Toc129875560)

[1.1 Justificativa e Motivação 4](#_Toc129875561)

[2 Requisitos de análise e projeto de sistemas i 4](#_Toc129875562)

[3 Requisitos de engenharia de software 4](#_Toc129875563)

[4 Consideração finais 4](#_Toc129875564)

[5 BIBLIOGRAFIA 4](#_Toc129875565)

[APÊNDICE 1 – Cronograma de entrega de atividades. 5](#_Toc129875566)

# 1. Apresentação:

## 1.1 Justificativa e Motivação

Decidimos criar uma aplicação funcional e simples que estaria nos níveis de conhecimento da equipe.

O software Smart Products foi escolhido para atender problemas de estoque que, geralmente, muitos logistas possuem.

# 2 Requisitos de análise e projeto de sistemas i

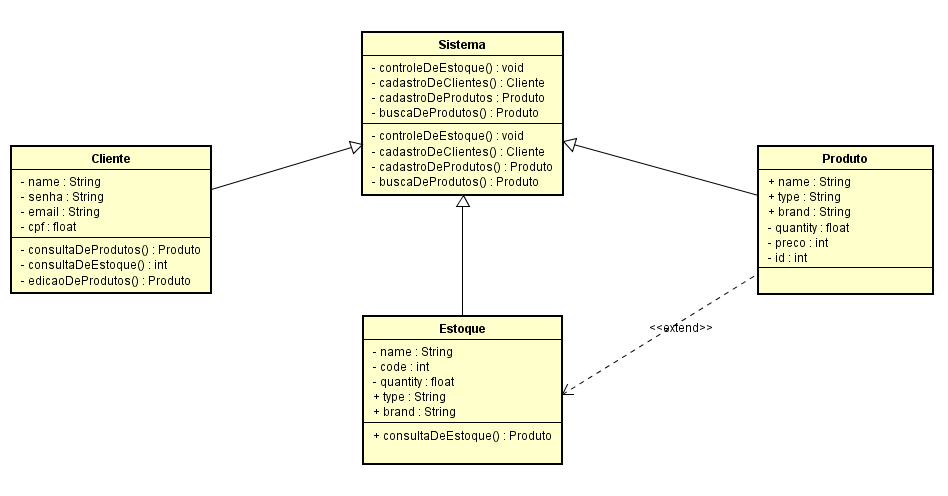
* A proposta deste projeto é criar um sistema de cadastro de produtos simples para um estabelecimento qualquer. Colocando em prática todos os conhecimentos adquiridos no curso de Análise e Desenvolvimento de Sistemas.

**Requisitos Funcionais**

* [RF001] O software deve cadastrar o nome, e-mail, cpf e senha de cada cliente.
* [RF002] O software deve cadastrar a quantidade, tipo e data de validade de cada produto.
* [RF003] O software deve conseguir fazer uma busca por nome de produto na área de busca.
* [RF004] O sistema deve ter o controle de todo o estoque de produtos.

**Requisitos Não Funcionais**

* [RNF001] O sistema deve ser implantado em React, TypeScript, Python, Flask e SQLite.
* [RNF002] O sistema dever ser compatível com os as versões atuais do Google, Firefox e Microsoft EDGE.
* [RNF003] O sistema deve lidar com as função de buscas de produtos sem qualquer tipo de interrupção ou reload da página.
* [RNF004] O sistema deve possuir uma fácil manutenabilidade.

**Diagrama de Classe**

**Caso de Uso**

****

# 

# 3 Requisitos de engenharia de software

* **Histórias de usuário**

**Título**

* **Implementar uma função de cadastro de clientes**

**Descrição**

* **Eu como usuário quero que o sistema armazene em um bando de dados cada cliente com seu nome, cpf , e-mail e senha.**

**Critérios de aceitação**

* **O cadastro do cliente deve acontecer o mais rápido possível e sem o reload da página.**
* **O cadastro deve ocorrer em uma página que tenha como conteúdo um formulário simples e um link que possa redirecionar o usuário para a página de login.**
* **A função deve passar por testes antes da implementação.**

**Título**

* **Implementar uma função de cadastro de produtos**

**Descrição**

* **Eu como usuário quero que o sistema armazene em um bando de dados cada produto com seu nome, marca, preço, tipo e código.**

**Critérios de aceitação**

* **O cadastro do produto deve acontecer o mais rápido possível e sem reload da página.**
* **Os dados de cada produto devem ser armazenados em um banco de dados.**
* **O cadastro do produto deve ter uma interface intuitiva, simples e fácil.**
* **A função deve passar por testes antes da implementação**

**Título**

* **Implementar função de busca de produtos**

**Descrição**

* **Eu como cliente quero que o sistema tenha uma função de busca por nome de produto.**

**Critérios de aceitação**

* **Os resultados da pesquisa devem ser relevantes e precisos para o cliente.**
* **Os produtos devem aparecer em cards com visualização rápida e simples na página de Produtos.**
* **A função de busca não deve depender de um botão para ser executada.**
* **Os produtos devem aparecer de acordo com o que o usuário digita da barra de buscas sem recarregar a página.**
* **Os resultados da busca devem ser exibidos em uma lista clara e simples.**

**Título**

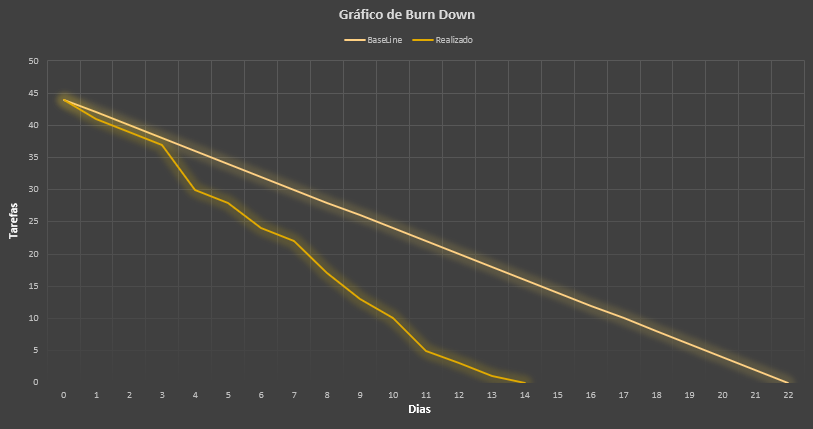
* **Implementar funções de Estoque.**

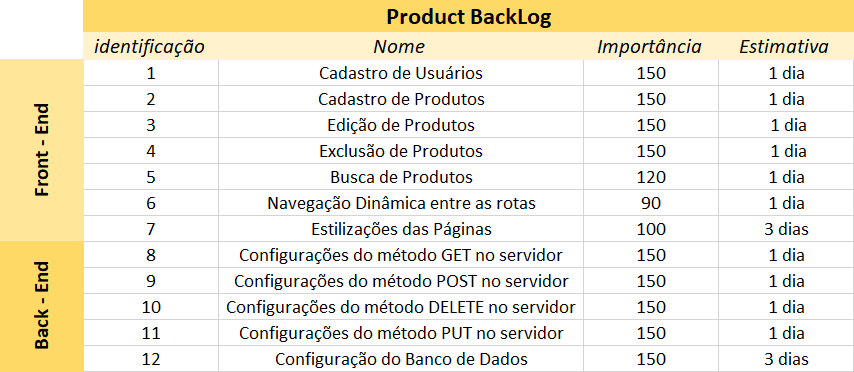
**Descrição**

* **O usuário deve ter acesso a quantidade total de produtos do seu negócio, assim como alertas de baixo estoque quando determinado produto estiver com estocagem baixa.**

**Critérios de aceitação**

* **A quantidade de produtos cadastrado deve ser inserida na página Estoque através de um card.**

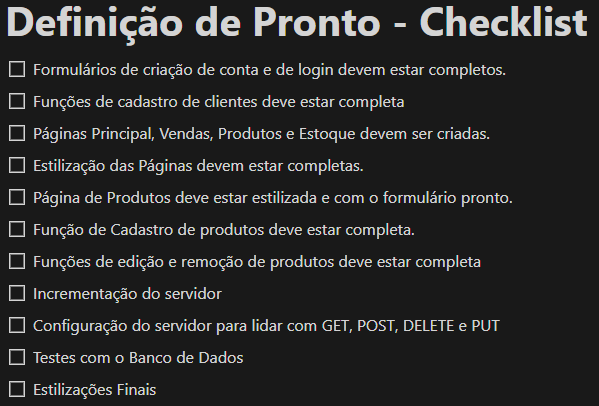
** Gráfico de Burn Down**

** Product BackLog**

**Sprint BackLog**

Uma das Sprints que estávamos realizando durante o desenvolvimento do software

****

** Definition of Ready - Checklist**

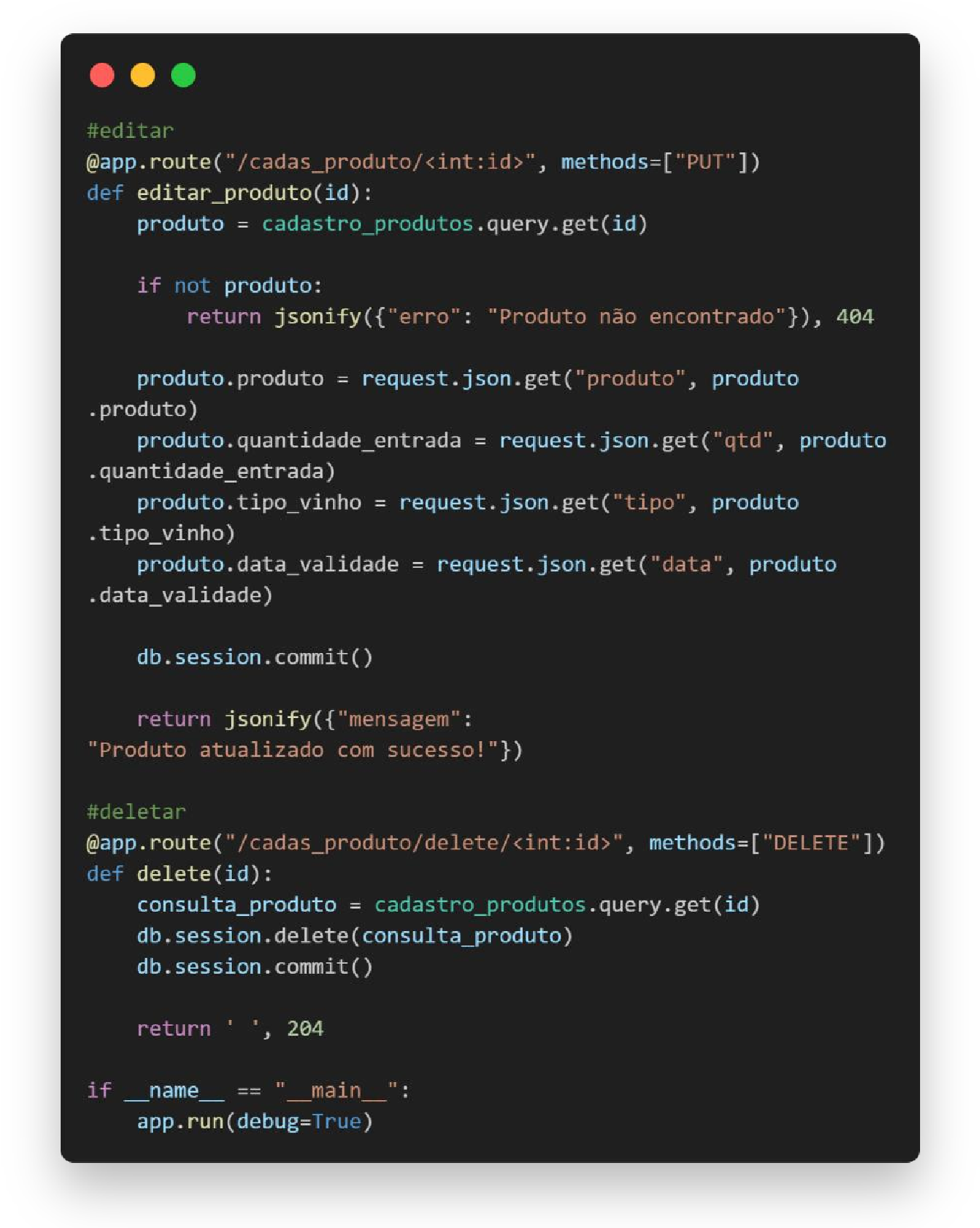
**Reunião de Revisão da Sprint**

****

**Código Fonte do Software**



**Python :**



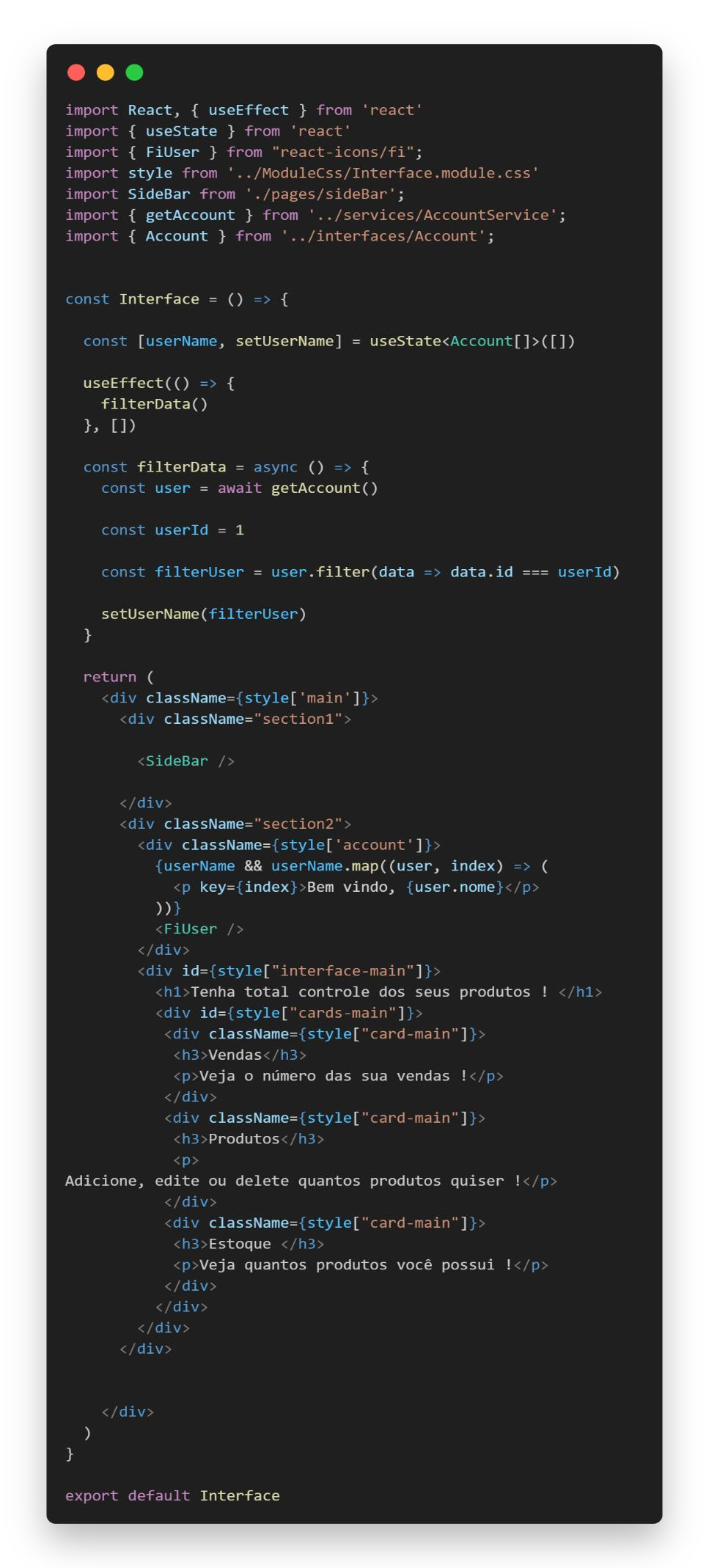




**React :**

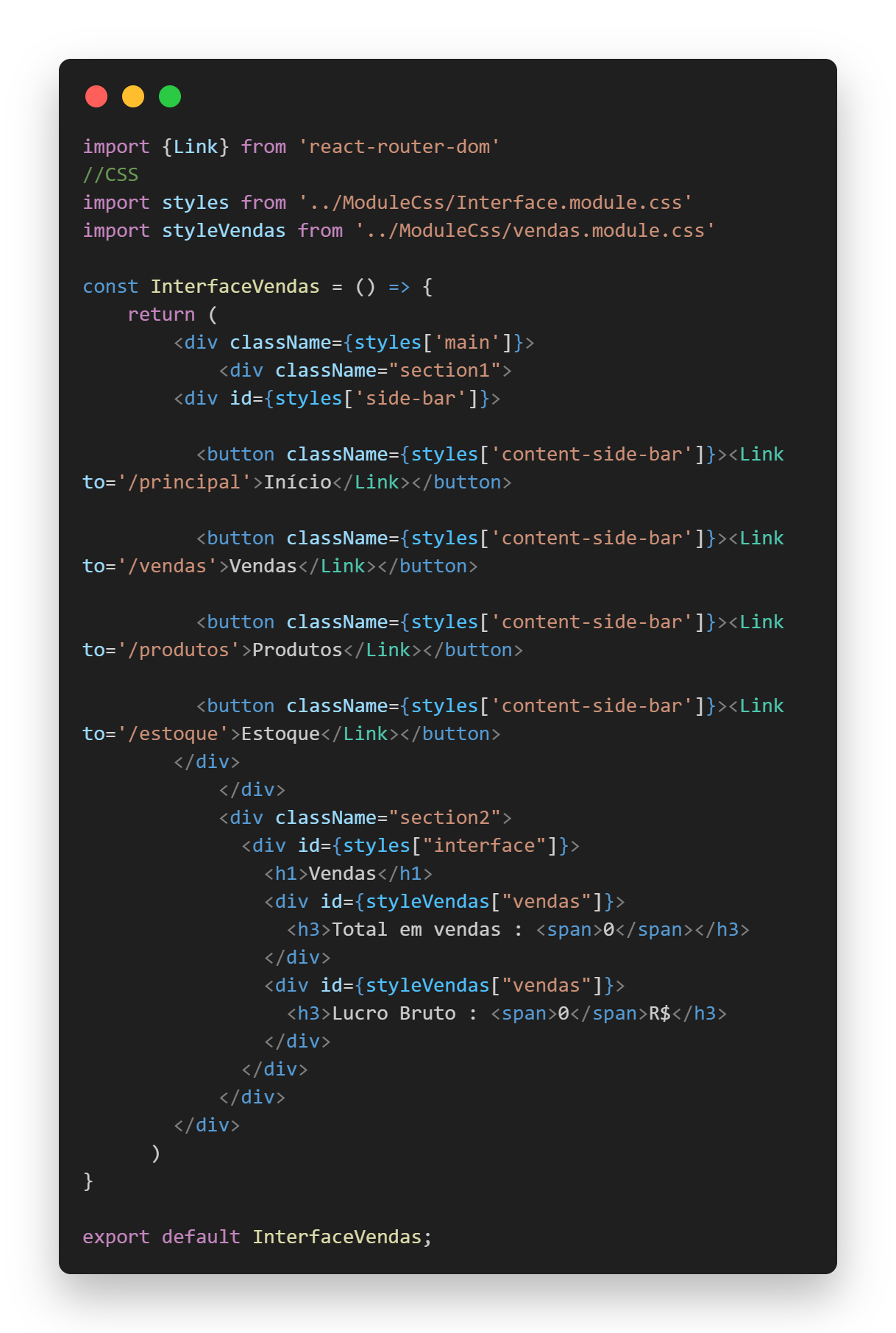






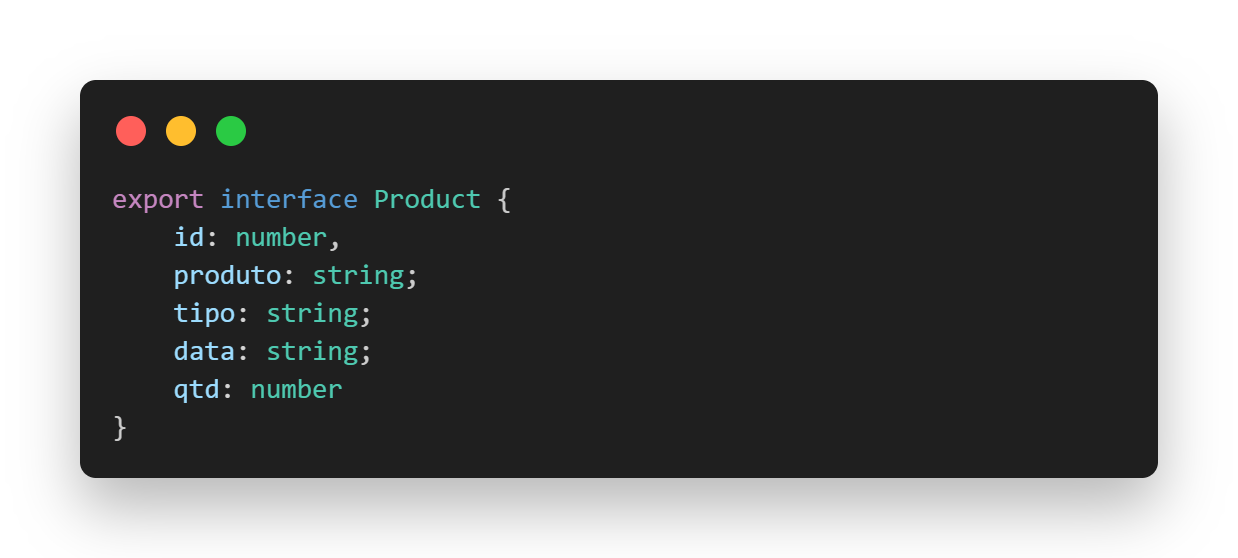
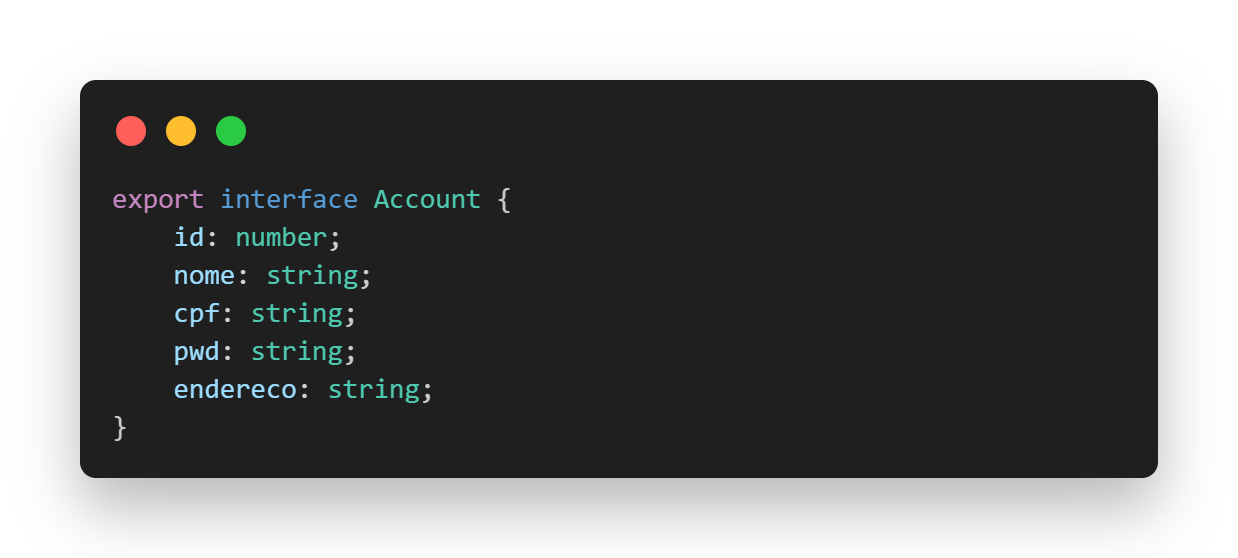








**TypeScript :**







* **Vídeo do funcionamento do software :**

<https://www.youtube.com/watch?v=76HcDIoML-U>

# 4 Consideração finais

Definir os requisitos deste software foi um dos primeiros passos que demos, um dos mais simples também.

De fato, as dificuldades apareceram quando começamos a desenvolver as função de cadastro de clientes e de produtos. A principal dificuldade que tivemos era o bloqueio de Cross Orgin do Browser, e conseguimos resolver esse erro instalando no servidor um código que permite a entrada de dados de um domínio específico.

# 5 BIBLIOGRAFIA

* Análise de Projeto e Sistemas.
* Engenharia de Software.
* Técnicas de Programação.

# APÊNDICE 1 – Cronograma de entrega de atividades.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | Descrição | Data | | Prazo do cronograma em semanas | | | | | | | | | | | | | | | | | |
| Início | Término |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 1 | Definição do tema e planejamento inicial | 01/04 | 05/04 | P | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Elicitação de Requisitos | 06/04 | 17/04 | P | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Análise de Requisitos | 18/04 | 18/04 | P | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Desenvolvimento do Software | 20/04 | 20/05 | P |  |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Testes finais | 21/05 | 22/05 | P | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Entrega do projeto final e apresentação | 18/05 | 18/05 | P | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | P |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**OBS:**

1. **P =** previsto; **R =** realizado