

FAPEN FACULDADE PENTÁGONO.

CURSO POLITÉCNICO.

DAVID LEME GONÇALVES

DOUGLAS CASTANHA

EMERSON DE MELO MARTINS

JAQUELINE LEME GONÇALVES VILAR

NATHAN SIMÃO DOS SANTOS

SEUPHONE

SISTEMA DE E-COMMERCE

SANTO ANDRÉ-SP

2021

DAVID LEME GONÇALVES

DOUGLAS CASTANHA

EMERSON DE MELO MARTINS

JAQUELINE LEME GONÇALVES VILAR

NATHAN SIMÃO DOS SANTOS

SEUPHONE

SISTEMA DE E-COMMERCE

Trabalho de Sistema de E-commerce para Leasing de Iphone e conclusão do quarto semestre do Curso de Análise de Desenvolvimento de Sistemas.

Orientador: Professor Antônio Tessari.

SANTO ANDRÉ-SP

2021

DAVID LEME GONÇALVES

DOUGLAS CASTANHA

EMERSON DE MELO MARTINS

JAQUELINE LEME GONÇALVES VILAR

NATHAN SIMÃO DOS SANTOS

SEUPHONE

SISTEMA DE E-COMMERCE

Aprovado em: \_\_\_ /\_\_\_ /\_\_\_

BANCA EXAMINADORA

Prof.

FAPEN Faculdade Pentágono.

Prof.

FAPEN Faculdade Pentágono.

Prof.

FAPEN Faculdade Pentágono.

**RESUMO**

Nesse trabalho estamos descrevendo as tecnologias e métodos usados na criação do Sistema E-commerce para o negócio de leasing de iPhones, nesse quarto semestre iremos desenvolver um módulo de compras.

Nesse trabalho abordamos várias áreas do conhecimento em tecnologia da informação. Primeiramente falaremos da matéria Empreendedorismo e Marketing Digital.

Vamos navegar em estruturas de dados como API REST e JSON. Entrar na área de programação e suas tecnologias HTML, CSS, Javascript, Bootstrap, React, JSON, C#, API REST, ASP.NET, Entity Framework, SQL e mencionando suas divisões entre Model, View e Controller.

**Palavras-chaves:** Locação, Praticidade, Comercio Eletrônico.

LISTA DE FIGURAS

[**FIGURA 1-** (Tabela MER) 28](#_Toc451974196)

[**FIGURA 2-** (Tabela DER) 30](#_Toc451974197)

[**FIGURA 3-** (Diagrama de Classes) 32](#_Toc451974196)

[**FIGURA 4-** (Visão Lógica da Arquitetura) 36](#_Toc451974197)

[**FIGURA 5-** (Diagrama de Classes) 37](#_Toc451974196)

[**FIGURA 6-** (Configuraçõ de Rede Física) 43](#_Toc451974197)

[**FIGURA 7-** (Página do Servidor) 44](#_Toc451974196)

[**FIGURA 8-** (Página do Servidor) 45](#_Toc451974197)

[**FIGURA 9-** (GITHU) 46](#_Toc451974196)

[**FIGURA 10-** Tela inicial. 57](#_Toc451974197)

[**FIGURA 11-** Tela Sobre 58](#_Toc451974196)

[**FIGURA 12-** Tela Carrinho 59](#_Toc451974197)

[**FIGURA 13-** Tela Detalhe Produto 60](#_Toc451974196)

[**FIGURA 14-** Tela Perfil 61](#_Toc451974197)

SUMÁRIO

[INTRODUÇÃO 6](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974193)

[1. EMPREENDEDORISMO E MARKETING DIGITAL 7](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974194)

[1.1 Propósta de Valor 7](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974196)

1.2 Segmento de Clientes 8

[1.3 Canais de Distribuição 9](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974196)

[1.4 Relacionamento com os Clientes 9](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974196)

[1.5 Atividade Chave 10](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974196)

[1.6 Recursos Chave 10](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974196)

[1.7 Pareceiros Chave 12](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974196)

[2. DESENVOLVIMENTO FRONT-END 13](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974198)

[2.1 HTML 13](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[2.2 CSS 14](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[2.3 Javascript 15](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[2.4 Bootstrap 15](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[2.5 React 16](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[2.6 JASON 17](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974202)

[3. DESENVOLVIMENTO BACK-END 19](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974211)

[3.1 C-Sharp 19](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

3[.2 Arquitetura MVC 20](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[3.3 API REST 21](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[3.4 ASP.NET 23](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[3.5 Entity Framework 23](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[4. DESENVOLVIMENTO BANCO DE DADOS 25](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974212)

[4.1 SQL 25](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

4[.2 Tabela MER 27](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[4.3 tabela DER 29](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[4.4 Diagrama de Classes 31](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[5. DOCUMENTO DE ARQUITETURA DE SOFTWARE 33](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974212)

[5.1 Finalidade 33](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

5[.2 Escopo 33](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[5.3 Definições, Acrônimos e Abreviações 34](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[5.4 Referências 34](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[5.5 Visão Geral 34](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[6. METAS E RESTRIÇÕES DE ARQUITETURA 34](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974212)

[7.vIsão lógica 35](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974212)

[7.1 Visão Geral 38](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

7.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura

[30](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[8. vISÃO DE PROCESSOS 42](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974212)

[9.vIsão DE IMPLANTAÇÃO 43](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974212)

[10. vISÃO DE IMPLEMENTAÇÃO 44](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974212)

[10.1 Visão Geral 44](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[11.vIsão GERENCIAMENTO DE CÓDIGO 45](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974212)

11[.1 Visão Geral 44](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974200)

[12. CONCLUSÃO 47](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974212)

[13. BIBLIOGRAFIA 48](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974212)

[ANEXO A - SCRIPT DE CRIAÇÃO BANCO DE DADOS 50](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974213)

[ANEXO b - SCRIPT DE INSERÇÃO DE DADOS NO BANCO DE DADOS......](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974213)

[53](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974213)

[ANEXO C – TABELA DICIONÁRIO DE DADOS 55](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974213)

[ANEXO d – Telas do sistema 57](file:///C:\Users\jaque\Downloads\TRABALHO%20%2012-06-2020.docx#_Toc451974213)

**INTRODUÇÃO**

Em sites e-commerce podemos comprar ou contratar os mais diversos serviços de onde estivermos tendo apenas o acesso à Internet. Ter um sites e-commerce é muito importante atualmente pois esse canal com o público não é mais um diferencial e sim uma exigência do mercado.

Esse semestre o foco de nosso projeto é um site para vendas. Em nosso caso será um site para locação de celulares Iphones obedecendo as regras de negócio.

As principais tecnologias utilizadas para o desenvolvimento das telas desse site foram o React e a Api Rest em conjunto com o C# para criações como telas de apresentação, login, cadastro, carrinho e finalização de locação.

1. **EMPREENDEDORISMO E MARKETING DIGITAL**

Há algum tempo o celular vem se tornando objeto de desejo seja por status ou pelas funcionalidades que facilitam tarefas no dia-a-dia. Ter um aparelho de ponta para muitas pessoas representa uma ligação com tecnologia, promovendo status social.

Além de ser o principal meio de comunicação, os celulares podem resolver grande parte dos problemas do dia-a-dia, ser nosso lazer com jogos eletrônicos, redes sociais, aplicativos de mídia e também ferramenta de trabalho, pois cada vez mais as empresas investem em aplicativos para essa plataforma.

Com cada vez mais tecnologia embarcada e o câmbio do Dólar desfavorável em relação ao Real, os preços dos celulares aumentam exponencialmente e os usuários no Brasil acabam por manter seus celulares, notadamente os Iphones e aparelhos da Samsung, por períodos mais longos do que usualmente ocorre em outros países, não podendo comprar lançamentos. Essa prática acaba por intimidar também novos usuários.

A SeuPhone é uma empresa que entrou no mercado para que esse acesso mais rápido as novas tecnologias de celulares se tornem realidade para os brasileiros.

Com o serviço de *Leasing* (locação em inglês) que é um aluguel com um contrato de dois anos, o cliente escolhe o modelo de celular e, ao término desse contrato, ele tem a opção de devolvê-lo, adquirir o aparelho pelo preço preestabelecido no início do aluguel, ou fazer outro contrato de um novo aparelho.

* 1. **Proposta de Valor**

O termo “*leasing”* e a cultura que esse termo carrega não é usual em nosso país. A ideia de pagar por alguma coisa somente pelo seu tempo de uso é limitada apenas a alguns produtos em nosso país.

Os smartphones no Brasil, notadamente das marcas Apple e Samsung, carregam um preço consideravelmente elevado, principalmente quando falamos de lançamentos. Esses aparelhos, sonho de consumo de muitos brasileiros, a cada lançamento tem se tornado um sonho mais distante.

Foi pensando nessas duas situações que a empresa SeuPhone criou sua proposta de valor: mostrar que o aluguel de um smartphone pode beneficiar tanto o consumidor quanto ao fabricante do produto.

E como isso funciona?

O serviço de *Leasing* (locação em inglês) é um aluguel com um contrato de dois anos. O cliente escolhe o modelo de smartphone e ao término desse contrato ele tem a opção de devolvê-lo, adquirir o aparelho pelo preço preestabelecido no início do aluguel, ou fazer outro contrato de um novo aparelho.

* 1. **Segmento de Clientes**

O segmento de clientes da SeuPhone está, inicialmente, dividido em Pessoas Físicas e Jurídicas.

Quanto as Pessoas Físicas, são aquelas que precisam de smartphones com as últimas tecnologias, por questões profissionais ou simplesmente de lazer e pessoas que gostam de aparentar o *status* que um aparelho desses oferece.

São preponderantemente jovens e adultos, com boa cultura tecnológica, entre 14 e 45 anos, ambos os sexos, notadamente moradores das grandes metrópoles brasileiras e com uma renda *per capita* superior a cinco salários-mínimos. Esses smartphones, por seus altos custos, já selecionam seus consumidores geralmente de classe média alta a alta. A SeuPhone busca abranger um público maior, incluindo a classe média á média baixa, devido as facilidades financeiras para adquirir o aparelho.

Quanto as Pessoas Jurídicas, são empresas que oferecem celulares coorporativos aos seus colaboradores e desejam mantê-los atualizados, ao mesmo tempo, sem dispender grandes valores para a aquisição. Podem ser empresas de quaisquer ramos de atividades e, geograficamente, da mesma forma que as Pessoas Físicas, podem estar localizadas em qualquer ponto geográfico do território brasileiro.

**1.3** **Canais de Distribuição**

Como um canal de varejo intermediário de produtos com valores consideráveis a SeuPhone não necessita de um canal de distribuição físico. O Canal virtual supre todas as necessidades do negócio, até mesmo pela ideia de que buscamos passar de praticidade, que o nosso público não precisa de muito esforço para contratar um aluguel de celulares.

Inclusive, a SeuPhone vai eliminar a necessidade da presença física do cliente para assinaturas de contrato. Através de assinaturas digitais, será possível finalizar todo o processo de contratação do serviço de leasing, sem deslocamentos ou burocracia.

* 1. **Relacionamento com os Clientes**

A estratégia de Relacionamento com os Clientes da SeuPhone leva em consideração as seguintes ferramentas:

**ASSISTÊNCIA PESSOAL:** Assistência Pessoal nada mais é do que os pontos de contato que a SeuPhone vai disponibilizar ao prospect que entrou no site da empresa. A Seuphone disponibilizará formas de contato através de seu site, como Chat e um Fale Conosco para preenchimento de dúvidas para posterior resposta da empresa. Também disponibilizará um canal de SAC (Serviço de atendimento ao Cliente), e-mail e número de telefone e WhattsApp para contato. Estes três últimos, funcionarão em horário comercial, das 9:00h às 17:00h, de segunda à sexta-feira.

**ASSISTÊNCIA DEDICADA:** Por ser um aluguel, existe a possibilidade que durante a vigência do contrato ocorrer dúvidas com relação as cláusulas e imprevistos posteriores no produto oferecido, como defeito de fábrica, furto, entre outros problemas que possam afetar o cumprimento do contrato de ambas as partes. Por isso quando o cliente entrar em contato por nosso canal de SAC (Serviço de atendimento ao Cliente), e-mail, telefone ou via WhatsApp, poderá solicitar uma consulta direta com o setor jurídico ou técnico, que auxiliará a resolver quaisquer tipos de problemas, tanto na venda como no pós-venda.

**FAÇA VOCÊ MESMO:**  A Seuphone tem como premissa que a navegação em seu site seja intuitiva e que o cliente possa realizar os processos da escolha a finalização do aluguel de maneira independente, porém, é esperado que dúvidas surjam para alguns clientes durante o processo. Dessa maneira, para aqueles que desejarem concluir a negociação sem nenhum contato pessoal, a SeuPhone vai disponibilizar o FAQ (perguntas frequentes). Dessa forma, será possível que grande parte das dúvidas dos clientes sejam sanadas sem necessidade de contato direto com a empresa.

* 1. **Atividade Chave**

A atividade chave da empresa Seuphone é oferecer o serviço de *Leasing,* que seria basicamente o aluguel de um smartphone.

O conceito de *Leasing* que a empresa tenta passar para o mercado é o de que é possível pagar por um smartphone somente pelo seu tempo de uso.

Como atividade adicional, a Seuphone oferece a opção de compra do aparelho ao final do contrato.

Ou seja, resumidamente, as atividades chave da Seuphone estão atreladas aos serviços de compra e venda de aparelhos celulares de alto valor agregado.

**1.6 Recursos Chave**

Os Recursos Chave podem ser considerados os equipamentos, ferramentas, matérias-primas, imóveis, pessoas, finanças e criatividade que faz com que a empresa consiga produzir e/ou comercializar um produto ou serviço em seu mercado de atuação.

Os principais Recursos Chave da Seuphone são:

**Físicos:** Um escritório para concentrar a infraestrutura necessária para o negócio como:

-Equipamento (Computadores, impressoras e telefones);  
- Rede de computadores interna;

- Base de dados (contratar um serviço de SaaS e/ou IaaS);

- Local para armazenamento de estoque e logística dos produtos;

- Pequeno Estoque de Smartphones para locação;

- Contratação de serviços externos para a logística de envios e recebimentos;

**Intelectuais:** precisaremos que esses contratos de aluguéis sejam bem elaborados para não haver problemas de relacionamento com nossos clientes, de um site bem elaborado e intuitivo que suporte todas as funções necessárias e uma gestão de toda a logística que nossa empresa oferece.

**Humanos:**

-Contador;

-Advogado:

- Equipe de atendimento;

-Equipe de desenvolvimento;

-Equipe de divulgação e marketing;

-Equipe de RH.

**Financeiros:**

**-**Empréstimos;

- Investidores;

-Parcerias;

-Capital de Giro para manter estoque;

- Reserva para possíveis acontecimentos imprevistos;

**1.7 Parceiros Chave**

A SeuPhone tem como foco de negócio a distribuição de aparelhos celulares de última geração através de contratos de Leasing.

Com isso, nossos principais parceiros chaves serão os fornecedores de smartfones, como a Samsung, Apple, Xiaomi e Motorola.

Será fundamental que esses parceiros ofereçam bons preços, condições de pagamento e garantia de entrega, principalmente, dos lançamentos.

**2.DESENVOLVIMENTO FRONT-END**

Front-End ou Client-Side (lado do cliente) é onde o usuário consegue ver, interagir, visualizar todos os conteúdos que o chamam a atenção e onde desenvolvimento web ganha vida. Informações podem ser inseridas e muitas empresas se apresentam com mostras de seus produtos ou serviços prestados.

Para o desenvolvimento do Front-End para o sistema e-commerce, com base nas regras de negócios para o Leasing de iPhones, utilizamos algumas ferramentas como HTMl, CSS, Javascript, Bootstrap4 e React.

**2.1 HTML**

HTML (Hypertext Markup Language ou Linguagem de Marcação de Hipertexto) foi criado por Tim Berners-Lee, com o objetivo de comunicar e disseminar pesquisas entre ele e seu grupo de colegas e ficou mais conhecido quando começou a ser utilizado para formar na época a rede pública o que mais tarde se tornou a internet que conhecemos hoje.

O HTML é uma linguagem utilizada para desenvolver websites. Ela é a linguagem base da internet. Baseada no conceito de hipertexto, conjuntos de elementos ligados por conexões, como palavras, imagens, vídeos, áudios e documentos foi criada para ser de fácil entendimento por seres humanos e por máquinas.

Podemos dizer que basicamente o HTML é a linguagem padrão para a estruturação de toda página web. Por ela ser padronizada é aceita por vários dispositivos, desde computadores, celulares e até mesmo televisores (EIS, 2011).

Todas as nossas telas utilizam TAGS, como textos informativos utilizando a tag <p>, imagens ilustrativas com <img>, menus separados pelas tags <nav>, seções separando cada tópico com <section>, entre outros muitos elementos, tornando-se o esqueleto das telas.

**2.2 CSS**

O CSS (Cascading Style Sheets ou Folhas de Estilo em Cascata) é uma folha de estilo composta por camadas e utilizada para definir a apresentação em páginas da internet que adotam para o seu desenvolvimento linguagens de marcação como no HTML. O CSS define como serão exibidos os elementos contidos no código de uma página da internet e sua maior vantagem é efetuar a separação entre o formato e o conteúdo de um documento. (Tecmundo).

São as folhas de estilos de CSS que definem o layout da página, é a partir dela que podemos definir cores, fontes, tamanhos e apresentações diferentes dos elementos da página, seria um complemento ao HTML, onde um seria a estrutura (HTML) e o outro a estilização (CSS).

Conforme dito acima, as folhas de estilos para código CSS definem todo o comportamento do documento, em nosso projeto utilizamos praticamente em todos os elementos HTML.

Foi alterado tamanhos para os títulos, cores diferentes, definimos que nosso menu principal é fixo no topo da página, facilitando o acesso de nosso cliente.

Além dessas estilizações de nossa página, o CSS possui também o benefício de evitar repetição de código.

O mesmo código CSS pode ser reaproveitado em outros elementos HTML quando definimos classes e/ou ids e aplicados a estilização diretamente neles.

O CSS trabalha juntamente com o HTML, fazendo com que a maneira que o cliente, pessoa que acessa nosso sistema, tenha uma visão e experiência mais agradável.

* 1. **Javascript**

O Javascript é a linguagem que torna os sites dinâmicos, permitindo ao usuário maneiras de interagir com a página e, ao programador, permite utilizar recursos que não são possíveis apenas com HTML ou CSS como, por exemplo, interação com o sistema do computador para apresentar data e hora.

Por fim em Front-End, podemos conhecer um pouco de Javascript, tecnologia responsável por toda dinâmica e interação do nosso sistema com a pessoa que está acessando.

O Javascript pode ser utilizado em diversos lugares de formas muito complexas, porém, em nosso sistema foi utilizado somente para dinâmica e interação do sistema, como por exemplo animar a abertura do menu, realizar um carrossel de imagens de nossos produtos, permitindo avançar ou retroceder produtos.

Ele também fez com que em nosso sistema, durante o cadastro de um novo usuário, ao informar o CEP consiga retornar todos os dados em relação ao endereço da pessoa, facilitando muito o cadastro.

Podemos concluir que além de podermos utilizar a estilização com o CSS para melhorar o visual, o Javascript possui um papel importante para experiência do usuário.

**2.4 Bootstrap**

O Bootstrap foi desenvolvido para o site Twitter por volta de 2010 por Mark Otto e Jacó Thornton. Antes de ser uma estrutura de código aberto foi conhecido como Twitter Blueprint. Após alguns meses de desenvolvimento, o Twitter realizou sua [primeira Hack Week](https://blog.twitter.com/2010/hack-week) ou Semana do Hack, um evento que reúne vários profissionais da área com intuito de desenvolver ou aperfeiçoar um software ou hardware até o final do evento. Com isso projeto explodiu à medida que os desenvolvedores de todos os níveis evoluíram sem nenhuma orientação externa. Por mais de um ano serviu exclusivamente como guia de estilo para o desenvolvimento de ferramentas internas no Twitter, a empresa ainda utiliza o Bootstrap, mas agora ele também é disponível para o público. (GETBOOTSTRAP)

O Bootstrap é um framework Front- End de código aberto criado para facilitar a rotina de um desenvolvedor, pois ele possibilita a criação de sites mais rapidamente. Ele possui todos os tipos de templates baseados em HTLM (Linguagem de Marcação de Hipertexto) e CSS (Folhas de Estilo em Cascata) para várias funções e componentes, como por exemplo, botões, navegação, sistema de grades e carrosséis de imagens. Ele também possui uma infinita lista de bibliotecas disponíveis. (LOGEN,2020)

Em nosso sistema o Bootstrap foi utilizado para toda a base de estilização do projeto. Ele serviu como estrutura de todo o sistema de grid, manter a estilização de formulários, fontes, TAGS com cores e botões disponíveis na biblioteca do framework como menus de navegação, cards e efeito carrossel.

Podemos dizer que o framework agilizou muito o desenvolvimento do layout do nosso site, pelo fato de ter muitos recursos pré-construídos através de bibliotecas ele facilitou todo o processo.

* 1. **React**

O React foi criado em 2011 pelo engenheiro de softwares Jordan Walk dentro do Facebook, onde por algum tempo foi preparado para atingir um estado suficientemente estável e em 2013 foi aberto ao público e hoje em dia o Instagram também e movido por ele. (PODILA,2013)

O React é basicamente uma biblioteca com uma coleção de funcionalidades relacionadas que podem ser usadas por um desenvolvedor para solução de problemas específicos. No caso do React, sua única função é a criação de interface de usuários, que organiza o que será mostrado para o usuário final.

Com a lógica de componente escrita em JavaScrip e não em templates, ele permite facilmente passar diversos tipos de dados ao longo de sua aplicação. (UDACITY BRASIL, 2018)

A biblioteca React possibilitou a construção de toda interface da aplicação de forma rápida por sua facilidade em manipular elementos dos dados exibidos. Fazendo com que ocorra a apresentação dos dados que estão no servidor de forma dinâmica. Como por exemplo mostrando produtos disponíveis cadastrados anteriormente para o usuário que está acessando o e-commerce.

Além do que foi citado, o conceito de componentização utilizado nessa biblioteca faz com que o código seja de fácil manutenção pois cada parte da aplicação está separada com suas próprias regras e apenas o necessário para o seu funcionamento.

**2.6 JSON**

O JSON (JavaScritp Object Notation ou JavaScritp Notação de Objeto) foi difundido pelo programador e empresário Douglas Crockford. Por ser grande conhecedor de JavaScritp popularizou o formato de dados JSON e desenvolveu outras ferramentas relacionadas.

O JSON é uma forma leve e compacta de troca de dados entre sistemas com linguagem fácil de ler e escrever aos humanos e de fácil intepretação a máquinas. Apesar do nome JavaScritp Object Notation ele pode ser usado por muitas outras linguagens além do JavaScritp e vem se tornando muito popular em aplicações Web.

Muito utilizado na comunicação entre o navegador e o servidor, faz com que as páginas consumam pequenos conjuntos de dados em vez de ter que pegar todos os dados necessários de uma vez para o carregamento de uma página. (FERNANDES,2020)

Em nosso projeto utilizamos o JSON pelas inúmeras vantagens que ele oferece para o desenvolvimento de um aplicativo, como mais rapidez na execução de transporte de dados e o arquivo com tamanho reduzido.

**3.DESENVOLVIMENTO BACK-END**

Como o nome sugere, é a parte por "trás" ou server-side (parte do servidor) do sistema, é onde fica todo o raciocínio da página web e só o SERVIDOR consegue ler, não é possível vê-lo ou interagir igual o Front-End.

É responsável por toda a lógica do sistema, desde uma simples conta de matemática, consultar dados de um banco, ou até uma autenticação do sistema.

**3.1 C-Sharp**

O C# escrito também C-Sharp, é uma linguagem de multiparadigma desenvolvida pela Microsoft. Ela é a linguagem principal do NET. Framework, o framework para desenvolvimento da Microsoft. Apesar de desenvolvida do zero foi baseada na linguagem de C ++ e possui muitos elementos da linguagem Pascal e Java. ( PACIEVITCH)

Por ser multiparadigma o C-Sharp suporta orientação a objetos e conceitos comuns como encapsulamento, herança e polimorfismo. É também fortemente tipada e com diferenciação entre letras maiúsculas e minúsculas, ou seja, case sensitive.

O C# foi fundamental para o desenvolvimento do backend da aplicação em conjunto com a arquitetura MVC (Model, View, Controller) de forma segura e poderosa. Por ser uma linguagem orientada a objetos facilitou o entendimento da utilização de sua sintaxe e o uso dela para a criação da estrutura da aplicação como por exemplo os modelos com seus atributos e métodos.

Essas facilidades se dão por conta de a linguagem ser da Microsoft e possuir licença opensource e é multiplataforma.

**3.2 Arquitetura MVC**

Em meados da década de 70 o engenheiro civil Christopher Alexander criou o que é considerado um dos primeiros padrões de arquitetura documentada, seu trabalho foi reconhecido por profissionais da área de desenvolvimento de software, que começaram a utilizar os conceitos para iniciar as primeiras documentações de padrões de projetos.

Foi então que o funcionário da corporação Xerox Trygve Reenskaug criou em 1979 o que nós temos hoje como padrão MVC(MEDEIROS,2013).

O MVC (Model-View-Controller ou Modelo-Visão-Controle) é um padrão de arquitetura que se divide em três tipos de camadas:

Model: é a parte onde fica toda a lógica de programação, onde contém as classes do sistema, o acesso ao banco de dados e toda a regra de negócio.

View: é a parte onde o usuário interage com o sistema, onde por exemplo fica botões, a parte gráfica do negócio, campos para preencher cadastros etc.

Controller:o nome já define sua função é a parte responsável pelo controle do view e do model, ele faz o processamento de dados informados pelo usuário pela camada do view e passa para camada do model (RAMOS,2011).

Nosso projeto é todo arquitetado com o padrão MVC com os Models que contém todos os métodos e atributos em classes como as de usuário. O View contendo as telas do aplicativo e o Controller onde irá possuir todos os métodos responsáveis pelas funcionalidades do aplicativo.

Devido a divisão do padrão MVC em três camadas fica muito fácil a organizar e arquitetar um projeto para programação, a própria sintaxe do programa fica menos confusa por esse motivo e por ser um padrão que pode ser utilizado por qualquer linguagem orientada a objeto o programador que conhecer esse conceito tem mais facilidade na compreensão de uma linguagem da qual ele não tem familiaridade.

**3.3** **API REST**

API (Application Programming Interface ou Interface de Programação de Aplicações) REST (Representational State Transfer ou Transferência Representacional de Estado) é um estilo de arquitetura de software proposto por Roy Thomas Fielding cientista da computação e um dos principais autores da especificação HTTP (Hypertext Transfer Protocol ou Protocolo de Transferência de Hipertexto) no ano de 2000. (JUNIOR, 2019)

API REST é um conjunto de boas práticas utilizadas nas requisições HTTP por uma API em uma aplicação web, para entendermos isso melhor precisamos primeiro explicar o que é um API, que é um conjunto de requisições que possibilita a comunicação de dados entre as aplicações, utilizando para isso requisições HTTP que são responsáveis pelas operações necessárias para a manipulação de dados. Essas requisições são:

* POST: Esse método cria dados;
* GET: Esse método faz a leitura de dados;
* DELETE: Esse método exclui informações;
* PUT: Esse método atualiza registros;
* PATCH: Esse método também atualiza registros, porém de forma mais

específica.

E REST é um conjunto de restrições usadas para as requisições HTTP compreendam as diretrizes definidas na arquitetura. As restrições da arquitetura REST são:

* Cliente-servidor: as aplicações que existem no servidor e no cliente devem ser separadas;
* Sem estado: as requisições são feitas de forma independente, cada uma executa apenas uma determinada ação;
* Cache: A API deve usar o cache para evitar chamadas repetidas ao servidor;
* Interface uniforme: reuni outros quatro conceitos em que determina que os recursos devem ser identificados, a manipulação dos recursos deve ser por meio de representação, com mensagens auto descritivas e utilizar links para navegar pelo aplicativo.

Então API REST é utilizar uma API para acessar aplicações na parte do servidor, de forma que essa comunicação seja feita com os padrões definidos pela arquitetura REST.

Existem resumidamente três tipos de APIs:

* Privadas: São usadas entre as aplicações internas de uma empresa;
* Parceiros: São usadas para permitir integração entre diferentes

softwares entre parceiros de negócios;

* Públicas: Como o nome mesmo sugere podem ser usadas livremente geralmente são usadas por empresas que disponibilizam para seus desenvolvedores poderem fazer integração com outras aplicações. (SOUZA, 2020)

Em nosso projeto, a API REST foi desenvolvida possibilitando utilizar as requisições HTTP para que possamos manipular dados, como por exemplo a utilização do método GET na rota de produtos da aplicação que será responsável por buscar todos os dados previamente cadastrados para que essas informações sejam exibidas em diferentes plataformas.

Então concluímos que API REST é uma forma de se consumir uma fonte de dados de uma forma global, onde diversos tipos de aplicações podem acessá-las, por essa facilidade de comunicação com outras aplicações as interfaces permitem adicionar funcionalidades ou informações ao aplicativo de maneira rápida e simples.

**3.4** **ASP.NET**

O ASP.NET foi o primeiro framework de desenvolvimento web da Microsoft, lançado oficialmente no ano de 2002 junto com a plataforma NET 1.0. Ele foi derivado do ASP 3.0 e inicialmente se chamava ASP+.

ASP.NET é uma plataforma da [Microsoft](https://pt.wikipedia.org/wiki/Microsoft) para o desenvolvimento de aplicações [Web](https://pt.wikipedia.org/wiki/World_Wide_Web). Permite, através de uma linguagem de programação integrada no [.NET Framework](https://pt.wikipedia.org/wiki/.NET_Framework), criar páginas dinâmicas.

O ASP.NET por ser baseado no [Framework](https://pt.wikipedia.org/wiki/Framework) .NET herdou todas as suas características e como qualquer aplicação .NET, pode ser escrita em várias linguagens como por exemplo [C#](https://pt.wikipedia.org/wiki/C_Sharp).

Uma aplicação para web desenvolvida em ASP.NET pode reutilizar código de qualquer outro projeto escrito para a plataforma.NET

Em conjunto com a linguagem C#, a adição do ASP.NET nos permitiu construir de forma fácil as APIS, aonde as rotas retornam dados no formato JSON. Um tipo de dado que pode ser utilizado de forma global em diversas aplicações diferentes, mas que nesse caso foi usado na aplicação web sendo consumida pelo React, biblioteca utilizada para o Front-end.

Com as rotas construídas no Controller com a facilidade do ASP.NET, podemos consumir serviços como por exemplo de produtos, pedidos, entre outros

**3.5** **Entity Framework**

O Entity Framework e uma ferramenta de persistência com uma longa história de presença na plataforma .Net com parte integrante do pacote de tecnologias ADO.NET.

Entity Framework disponibiliza soluções para diminuir o problema de impedância, abstraindo do desenvolvedor detalhes dos bancos de dados relacionais. Também oferece diversos recursos que melhoram a produtividade no desenvolvimento de aplicações persistentes. Em nosso projeto ele foi utilizado para trabalhar com dados na forma de objetos e propriedades específicos de domínio, como clientes e endereços de clientes, dessa maneira não foi preciso nos preocuparmos com as tabelas e colunas de banco de dados subjacentes

O Entity Framework é uma ORM (Object Relational Mapper) que nos auxiliou no mapeamento das entidades da base de dados para código no formato de objeto. Seu uso nos permitiu tirar o máximo proveito da produtividade da aplicação durante o desenvolvimento, diminuindo toda complexidade. Diferentemente caso fosse utilizado por exemplo o ADO.NET, mecanismo de acesso aos dados também do C#, porém, de forma mais manual para suas consultas.

**4.****DESENVOLVIMENTO BANCO DE DADOS**

Banco de dados resumidamente se trata de uma coleção de informações que se relacionam de forma que faça algum sentido, isto é, uma estrutura organizada de dados.

**4.1 SQL**

SQL foi uma linguagem criada para banco de dados na década de 70. E seu criador foi Edgar Frank Codd, ele era um dos membros do laboratório da IBM

de San Jose na Califórnia. O SQL (Structured Query Language ou Linguagem de Consulta Estruturada) foi um completo sucesso no seu lançamento, tanto que na época a ANSI (American National Standarts Institute ou Instituto Americano de Normas Nacionais) padronizou a linguagem, sendo assim, a maioria das linguagens de Bando de Dados usam os padrões do SQL até hoje podendo ter pequenas variações na sua sintaxe.

A linguagem SQL é usada para comandos em bancos de dados relacionais (Baseado em tabelas). Existem diversas linguagens hoje em dia, porém é a mais usada no mercado atualmente. Seu objetivo é ajudar os programadores acessar, consultar e modificar dados de uma empresa ou organização, também ajuda com a comunicação entre DBAs (Data Base Administrator ou Administrador de Banco de Dados) e desenvolvedores.

Com o SQL é possível fazer Query's (Consultas) ou manipulações de dados mais complexas, porém existem formas mais simples para fazer consultas em tabelas. (SANTOS,2008)

Como:

* Insert (Inserir);
* Delete (Excluir);
* Update (Atualizar);
* Search (Pesquisar);
* Select (Selecionar).

SQL pode ser uma ferramenta até simples de ser usada, mas com alto potencial para fazer manipulações diretas dos dados,algo essencial para criação dos mais diversos softwares e sistemas digitais. No SQL existem alguns subgrupos para seu

desenvolvimento.

Tais como:

* **DML –** Data Manipulation Language ou Linguagem de manipulação de dados: comandos para alterar as informações nas tabelas, para inserir ou excluir dados (exemplo: select, delete e insert);
* **DDL –** Data Definition Language ou Linguagem de definição de dados: comandos para modificar o banco de dados (exemplo: drop para apagar algum objeto e create que permite a criação de novos objetos);
* **DCL –** Data Control Language ou Linguagem de controle de dados: grupo responsável pelas permissões, restrições ou bloqueios (exemplo: grant permite o acesso e/ou modificações no banco de dados);
* **DTL –** Linguagem de Transição de Dados: é responsável por salvar as alterações feitas pelos usuários (exemplo: commit autoriza que as alterações sejam salvas).

SQL foi usado como nosso banco de dados relacional, guardando todos os dados de usuários, pedidos, produtos etc. Sendo possível fazer Query's para buscar

uma lista de produtos para o painel de administrador e com métodos em nossa API usamos para inserir, deletar e atualizar informações em nossas tabelas.

O SQL por ser umas das linguagens mais usadas por desenvolvedores, existem diversas aulas e tutoriais pela internet ensinando vários detalhes da linguagem e ajudando em seu entendimento. E por ser a linguagem aonde se criou um padrão, a partir dela é possível migrar para outra linguagem sabendo pelo menos o seu básico.

**4.2 Tabela MER**

O MER (Model Relationships Mode ou Modelo Entidade Relacionamento)como o nome sugere, é um modelo conceitual utilizado na [Engenharia de Software](https://www.devmedia.com.br/principios-da-engenharia-de-software/29630) para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos).

Em geral, este modelo representa de forma abstrata a estrutura que possuirá o [banco de dados](https://www.devmedia.com.br/cursos/banco-de-dados) da aplicação. Obviamente, o banco de dados poderá conter várias outras entidades, tais como chaves e tabelas intermediárias, que podem só fazer sentido no contexto de [bases de dados relacionais](https://www.devmedia.com.br/curso/curso-modelagem-de-bancos-de-dados-relacionais/409).(Devmedia ,2014).

Em um MER (Modelo Entidade Relacionamento) podemos definir em geral como o banco de dados funcionará, desde definindo entidades que serão as tabelas, atributos dessas entidades onde definimos a forma de como será salvo o dado e pôr fim a relação entre as entidades.

Conforme havíamos dito, em nosso modelo utilizamos essa estrutura, podemos citar por exemplo nossa entidade (tabela) “tb\_user” onde serão armazenados dados necessários para efetuar a autenticação em nosso sistema, essa entidade possui atributos como “e-mail” e “senha”, esses que estarão armazenados no banco de dados como tipo “VARCHAR”, ou seja, tipo texto

Concluímos que ter o MER (Figura 1) é muito importante no desenvolvimento do sistema, pois é nele que podemos definir como funcionará como um todo o armazenamento de dados no qual definimos entidades, atributos e as relações entre eles.

Diagrama

Descrição gerada automaticamenteFigura 1 (Tabela MER)

**4.3 Tabela DER**

Enquanto a tabela MER é um modelo conceitual, o DER (Entity Relationships Mode ou Diagrama Entidade Relacionamento)é a sua representação gráfica e principal ferramenta. Em situações práticas, o diagrama é tido muitas vezes como sinônimo de modelo, uma vez que sem uma forma de visualizar as informações, o modelo pode ficar abstrato demais para auxiliar no desenvolvimento do sistema. Dessa forma, quando se está modelando um domínio, o mais comum é já criar sua representação gráfica, seguindo algumas regras.

As entidades devem ser representadas por retângulos, seus atributos ligados diretamente na própria entidade, os relacionamentos por losangos, ligados ás entidades por linhas, contendo também sua cardinalidade (1..1, 1..n ou n..n). (Devmedia ,2014).

É muito importante ter uma representação gráfica através do DER (Diagrama Entidade Relacionamento) pois são dele que teremos facilidade em visualizar como o banco de dados irá funcionar entendendo melhor a relação entre as entidades e suas cardinalidades.

Em nosso projeto, o DER foi utilizado justamente para essa tal facilidade em visualizar a cardinalidade entre as entidades (tabelas), podemos citar por exemplo que a tabela “tb\_ product” possui cardinalidade de 1 para 1 com a entidade “tb\_ provider”, sendo assim para cada um registro de login podemos relacionar um registro de pessoa.

Concluímos que o DER (Figura 2) nada mais é do que a descrição do banco de dados de forma gráfica para facilitar o entendimento das relações entre as entidades.

**Diagrama

Descrição gerada automaticamente**

Figura 2 (Tabela DER)

**4.4 Diagrama De Classes**

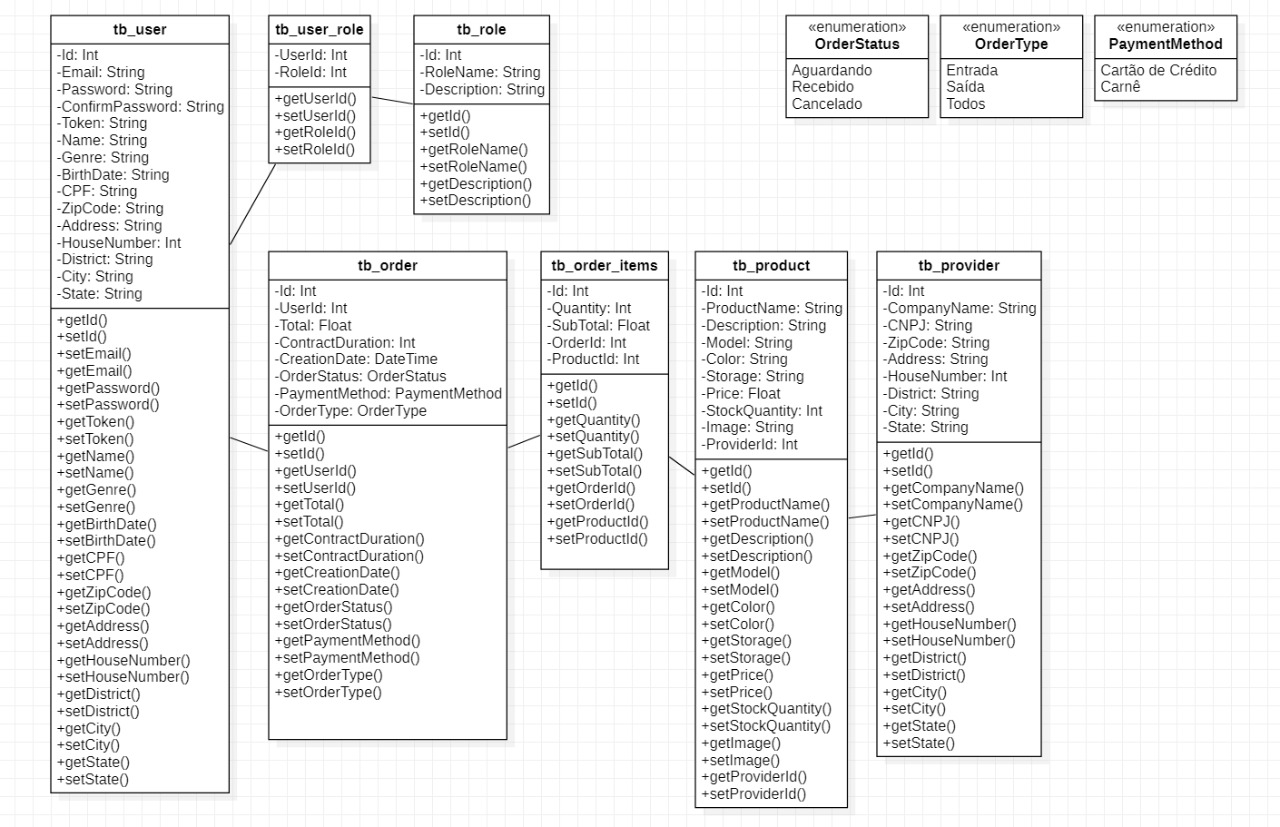
O diagrama de classes é bastante útil no desenvolvimento de sistemas e de softwares de computação, pois é uma representação estática utilizada na programação para descrever a estrutura de um sistema, definindo todas as classes que um sistema necessita, atributos, operações, relações entre os objetos e serve também para a base da construção de outros diagramas.

Os itens que compõem um diagrama de classes representam as classes que serão programadas, assim como os principais objetos ou a interação entre classes e objetos do sistema (Significados, 2018).

Em nosso projeto a modelagem Diagrama de Classe era necessário pelo fato de representar a estrutura e a relações das classes que servem de modelo para objetos em nosso sistema.

Toda a estrutura de nosso projeto foi baseada no diagrama de classes, é dele que podemos nos guiar para a criação do código, das funcionalidades através dos métodos e também como o banco de dados está estruturado através dos atributos e das relações entre as tabelas, tudo isso para entendermos melhor o sistema.

Além de servir como um guia para o desenvolvimento do projeto, o diagrama de classes (Figura 3) também ajuda quem não tem relação alguma no desenvolvimento a entender melhor como o sistema funciona.

Figura 3 (Diagrama de Classes)

**5.DOCUMENTO DE ARQUITETURA DE SOFTWARE**

O documento de arquitetura de software, tem o seu propósito no contexto do processo, o seu escopo de representação, vinculado ao software em desenvolvimento, bem como as definições, os acrônimos, as abreviações, e as referências utilizadas. Para finalizar a seção, será apresentada uma visão geral do documento em relação ao projeto.

A partir da seção dois, ter-se-á a representação detalhada da arquitetura do sistema, bem como as suas visões arquiteturais mais expressivas.

##### **5.1 Finalidade**

A finalidade deste artefato é fornecer uma visão geral do projeto para o desenvolvimento do software.

##### **5.2 Escopo**

O mercado de e-commerce no Brasil tem mudado o comportamento do consumidor enquanto cresce nos últimos anos, com a sua expansão chegando várias vezes aos dois dígitos percentuais.

A pandemia do novo coronavírus está provocando impactos em todas as áreas, e por conta disso as empresas devem se adaptar a modalidade atual, se por um lado algumas atividades estão sendo paralisadas ou reduzidas, por outro, alguns serviços estão com uma demanda maior, é o caso das vendas de produtos por deliverys e e-commerces.

Algumas lojas virtuais registraram um grande crescimento das vendas através do e-commerce.

Construir um site de comercio eletrônico, ecommerce, para comercialização dos produtos da empresa Seuphone. O projeto em si deverá conter as funcionalidades de realizar login, cadastro de cliente, listagem de produtos, carrinho de compras e realização do pagamento.

##### **5.3 Definições, Acrônimos e Abreviações**

Não Há.

##### **5.4 Referências**

Não Há.

##### **5.5 Visão Geral**

Nas próximas seções deste documento serão representados os detalhes significativos de construção arquitetural do Seuphone, através da ordem a seguir:

* Metas e Restrições de Arquitetura
* Visão Lógica
* Visão de Processos
* Visão de Implantação
* Visão de Implementação
* Visão de Gerenciamento de código

#### **6. METAS E RESTRIÇÕES DE ARQUITETURA**

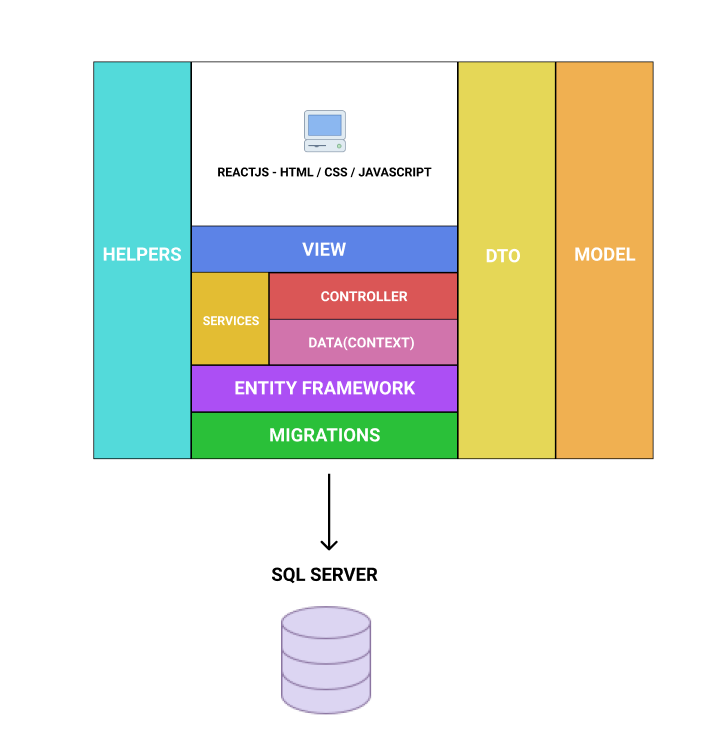
Para a proposta da arquitetura, foram considerados fatores como a finalidade do sistema, tipo de usuários e ambiente de execução. Sendo assim, a arquitetura a ser adotada precisa atender às seguintes características:

* Modularidade: o sistema será desenvolvido em camadas, utilizando a arquitetura MVC no backend com a utilização de suas APIS para alimentar o frontend.
* Manutenibilidade: o sistema oferecerá disponibilidade 24x7 (24 horas nos 7 dias da semana), o que exige mecanismos de monitoração do funcionamento do sistema e recuperação em caso de falha. O sistema deve adotar padrões de documentação e codificação bem definidos.
* Reusabilidade: o sistema permitirá a utilização de classes e componentes em outros projetos, favorecendo o tempo de produção e a qualidade do produto gerado, no caso de nosso projeto há a possibilidade da reusabilidade, porém, não foi utilizada no momento.
* Portabilidade: a fim de garantir portabilidade e independência de plataforma, o sistema será desenvolvido na plataforma .NET como backend e React como frontend.
* Estrutura da equipe: o time de desenvolvimento será o grupo Seuphone com os seguintes integrantes: David, Douglas, Emerson, Jaqueline e Nathan.

#### **VISÃO LÓGICA**

Essa seção descreve as partes significantes da arquitetura a partir do modelo de projeto, E para cada pacote, sua composição em classes. São introduzidas classes significantes para a arquitetura e suas responsabilidades, bem como os relacionamentos mais importantes, operações e atributos.

A figura a seguir fornece uma visão lógica da arquitetura do sistema.

Figura 4 (Visão Lógica da Arquitetura)

##### **Visão Geral**

Essa seção descreve a decomposição do modelo de projeto através da hierarquia de pacotes e camadas.

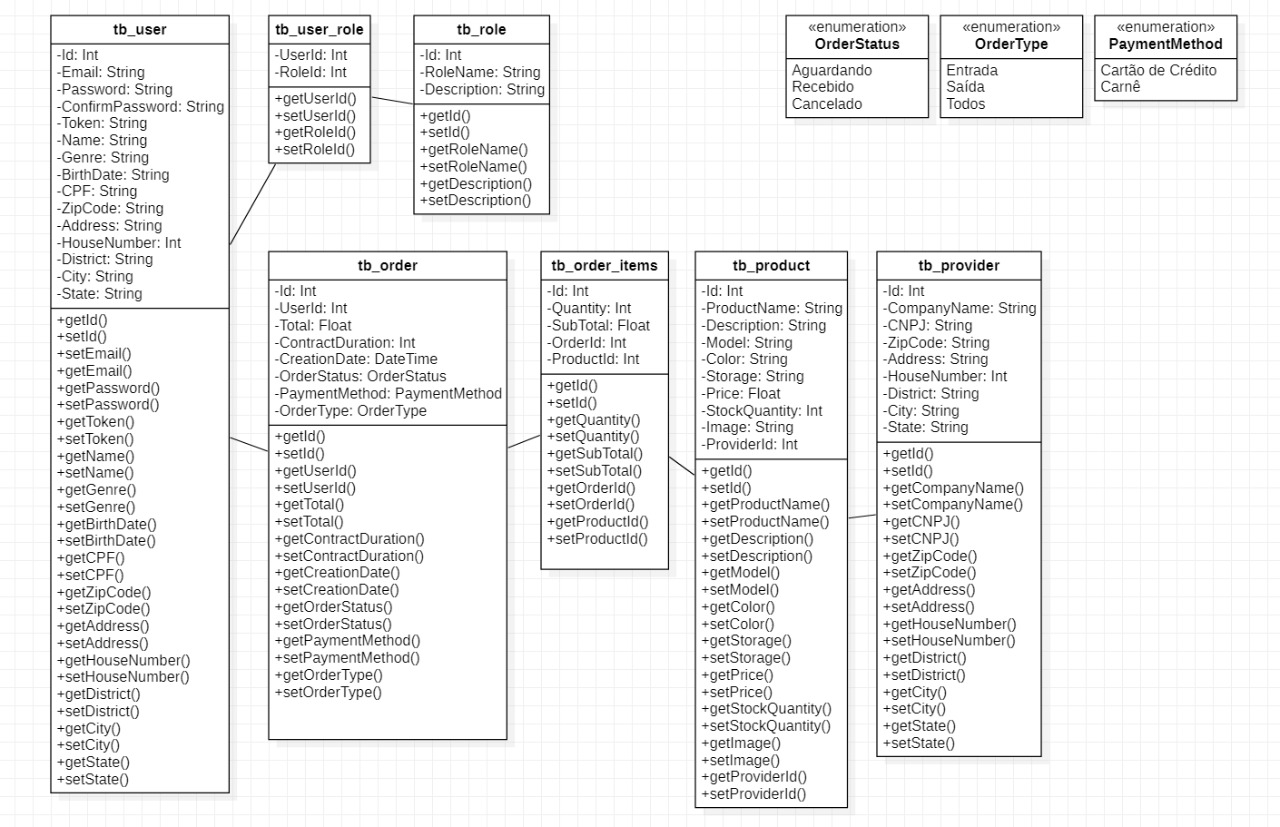


Figura 5 (Diagrama de Classes)

##### **Pacotes de Design Significativos do Ponto de Vista da Arquitetura**

Nessa seção será apresentado a transformação dos dados retornados no formato JSON pelas APIS do backend para que no frontend, o ReactJS possa ler os mesmos e transformar na página que o cliente final possa visualizar.

**WEB - ReactJS**

|  |  |
| --- | --- |
| **Páginas** | **Descrição** |
| Home | Tela inicial com pequenas seções de apresentação da empresa, produtos e banner no formato carrossel. |
| About | Tela com informações sobre a empresa e como funciona o serviço de leasing, locação de iphones. |
| Login | Tela de autenticação do cliente. |
| Forgot-password | Tela de redefinição de senha. |
| Register | Tela de cadastro do cliente. |
| Products | Tela de listagem de produtos com filtro de busca. |
| Products/1 | Tela com a especificação de um produto através da identificação dele. |
| Cart | Tela onde estará todos os produtos escolhidos pelo cliente. |
| Profile | Tela com todas as informações do usuário realizadas durante. |
| User-order-detail/1 | Tela com detalhamento do pedido do usuário. |
| Contract | Tela informativa sobre o contrato de leasing. |
| Cart-payment | Tela de pagamento com os produtos selecionados no carrinho com seleção de duração de contrato e método de pagamento. |

* **Controller**

Esse pacote contém classes relacionados a conexão com o Serviço.

|  |  |
| --- | --- |
| **Classes- C#** | **Descrição** |
| OrdersController | Classe responsável pelas rotas da API relacionadas a Pedido, as chamadas estão relacionadas aos métodos de criação, inserção e visualização de pedidos. |
| ProductsController | Classe responsável pelas rotas da API relacionadas a Produto, as chamadas estão relacionadas aos métodos de criação, inserção, visualização e atualização de pedidos. |
| ProvidersController | Classe responsável pelas rotas da API relacionadas a Fornecedor, as chamadas estão relacionadas aos métodos de criação, inserção, visualização e atualização de pedidos. |
| UsersController | Classe responsável pelas rotas da API relacionadas a Usuário, as chamadas estão relacionadas aos métodos de criação, inserção, visualização e atualização de pedidos. |

* **Model**

Esse pacote contém classes relacionados ao Serviço, onde vai realizar a conexão com o SQL.

|  |  |
| --- | --- |
| **Classes- C#** | **Descrição** |
| Authentication | Classe Model onde irá conter atributos e métodos necessários para realizar a autenticação do usuário. |
| Order | Classe Model onde irá conter atributos e métodos necessários guardar e utilizar informações do pedido para efetuar as devidas ações da ORM EntityFramework. |
| OrderItems | Classe Model onde irá conter atributos e métodos necessários guardar e utilizar informações dos itens do pedido para efetuar as devidas ações da ORM EntityFramework. |
| PasswordReset | Classe Model onde irá conter atributos e métodos necessários para realizar a alteração/atualização da senha do usuário. |
| Product | Classe Model onde irá conter atributos e métodos necessários guardar e utilizar informações do produto para efetuar as devidas ações da ORM EntityFramework. |
| Provider | Classe Model onde irá conter atributos e métodos necessários guardar e utilizar informações do fornecedor para efetuar as devidas ações da ORM EntityFramework. |
| Role | Classe Model onde irá conter atributos e métodos necessários guardar e utilizar informações do perfil de acesso para efetuar as devidas ações da ORM EntityFramework. |
| User | Classe Model onde irá conter atributos e métodos necessários guardar e utilizar informações do usuário para efetuar as devidas ações da ORM EntityFramework. |
| UserRole | Classe Model onde irá conter atributos e métodos necessários guardar e utilizar informações da lista de perfis do usuário para efetuar as devidas ações da ORM EntityFramework. |

* **DTO**

Esse pacote contém classes de auxílio ao serviço, onde vai facilitar a utilização de atributos específicos.

|  |  |
| --- | --- |
| **Classes- C#** | **Descrição** |
| AddressDTO | Classe DTO para facilitar e auxiliar a alteração de endereço do usuário. |
| PasswordDTO | Classe DTO para facilitar e auxiliar a alteração de senha do usuário. |
| OrderStatusDTO | Classe DTO para facilitar e auxiliar a alteração de situação de pedido. |
| ProductDTO | Classe DTO para facilitar e auxiliar a alteração de produto. |
| UserEditDTO | Classe DTO para facilitar e auxiliar a alteração de usuário. |

* **Services**

Esse pacote contém classes com métodos com ações específicas e auxiliar nos controllers da aplicação.

|  |  |
| --- | --- |
| **Classes- C#** | **Descrição** |
| MailService | Classe contendo métodos relacionados a envio de e-mail. |
| OrderService | Classe contendo métodos para gerar um arquivo do tipo PDF com o contrato escolhido e itens do pedido do usuário. |
| UserService | Classe contendo métodos para a autenticação do usuário e gerar um token de acesso. |

* **Helpers**

Esse pacote contém classes comuns que podem ser utilizadas em todos outros pacotes.

|  |  |
| --- | --- |
| **Classes- C#** | **Descrição** |
| CommonMethods | Classe que conterá métodos de criptografia que será utilizada por exemplo antes da gravação de uma senha no banco de dados. |
| AppSettings | Classe responsável pelo “Secret”, um código utilizado para a autenticação do usuário. |

* **Context**

Esse pacote contém classe com métodos e propriedades gerados pelo EntityFramework para ter uma espécie de conexão com o banco de dados

|  |  |
| --- | --- |
| **Classes- C#** | **Descrição** |
| SeuphoneApiContext | Objeto relacionado ao banco de dados, gerado pelo pacote do ntityFramework |

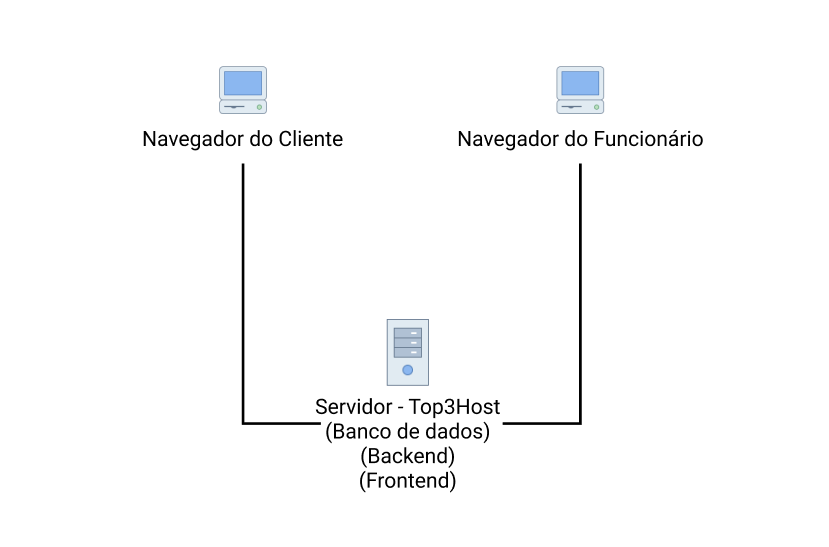
#### **8.VISÃO DE PROCESSOS**

Esta seção descreve a decomposição dos processos da aplicação:

* Servidor Top3Host – Servidor geral para a hospedagem do frontend com as telas para home, dados do usuário, produtos, carrinho e pagamento, juntamente com o backend onde terá todas as regras de negócio da aplicação e o banco de dados que armazenará todas as informações.

#### **9. VISÃO DE IMPLANTAÇÃO**

Esta seção descreve uma ou mais configurações da rede física (hardware) na qual o software é implantado e executado.

Figura 6 (Configuração de Rede Física)

* **Serviço Banco de Dados** – Indica que o servidor possui um serviço de banco de dados SQL Server.
* **Serviço Backend** – Indica que o mesmo servidor, além de possuir o banco de dados hospedado, também conterá o serviço de backend escrito em C# ASP.NET.
* **Serviço Frontend** – Por fim terá hospedado o frontend escrito com ReactJS.
* **Clientes e Funcionários** – Representam os vários clientes do sistema, que podem estar geograficamente distribuídos e adotar plataformas distintas. A interconexão entre o servidor e os terminais será realizada através da internet.

#### **10. VISÃO DE IMPLEMENTAÇÃO**

Esta seção descreve a estrutura geral do modelo de implementação, apontando a configuração do servidor da aplicação

##### **10.1 Visão Geral**

O sistema Seuphone foi publicado no servidor Top3Host que utiliza o IIS, pois a hospedagem é fácil e eficiente, já que ela foi desenvolvida pela Microsoft, uma grande empresa no ramo sendo assim podendo ser gerenciado da seguinte forma:

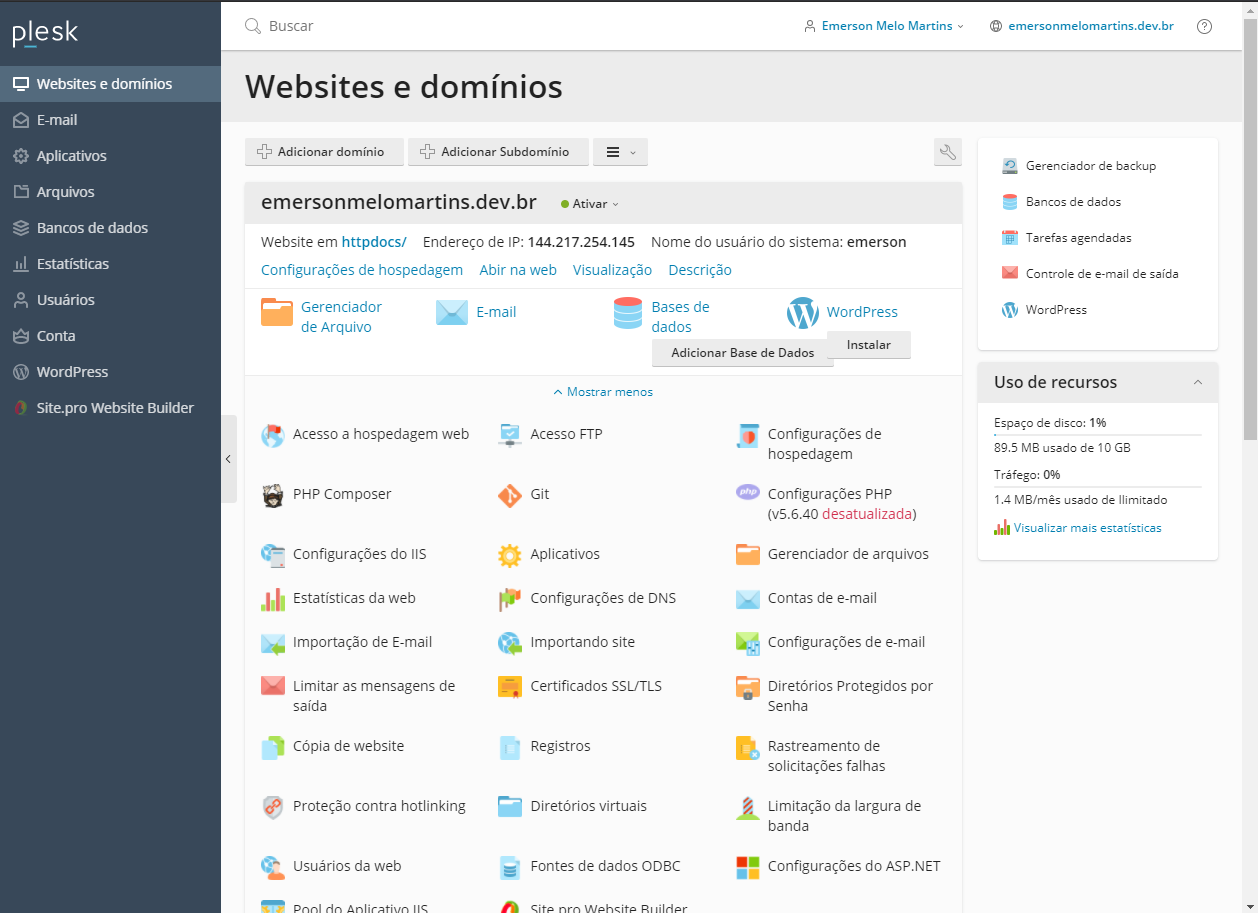


Figura 7 (Página do Servidor)

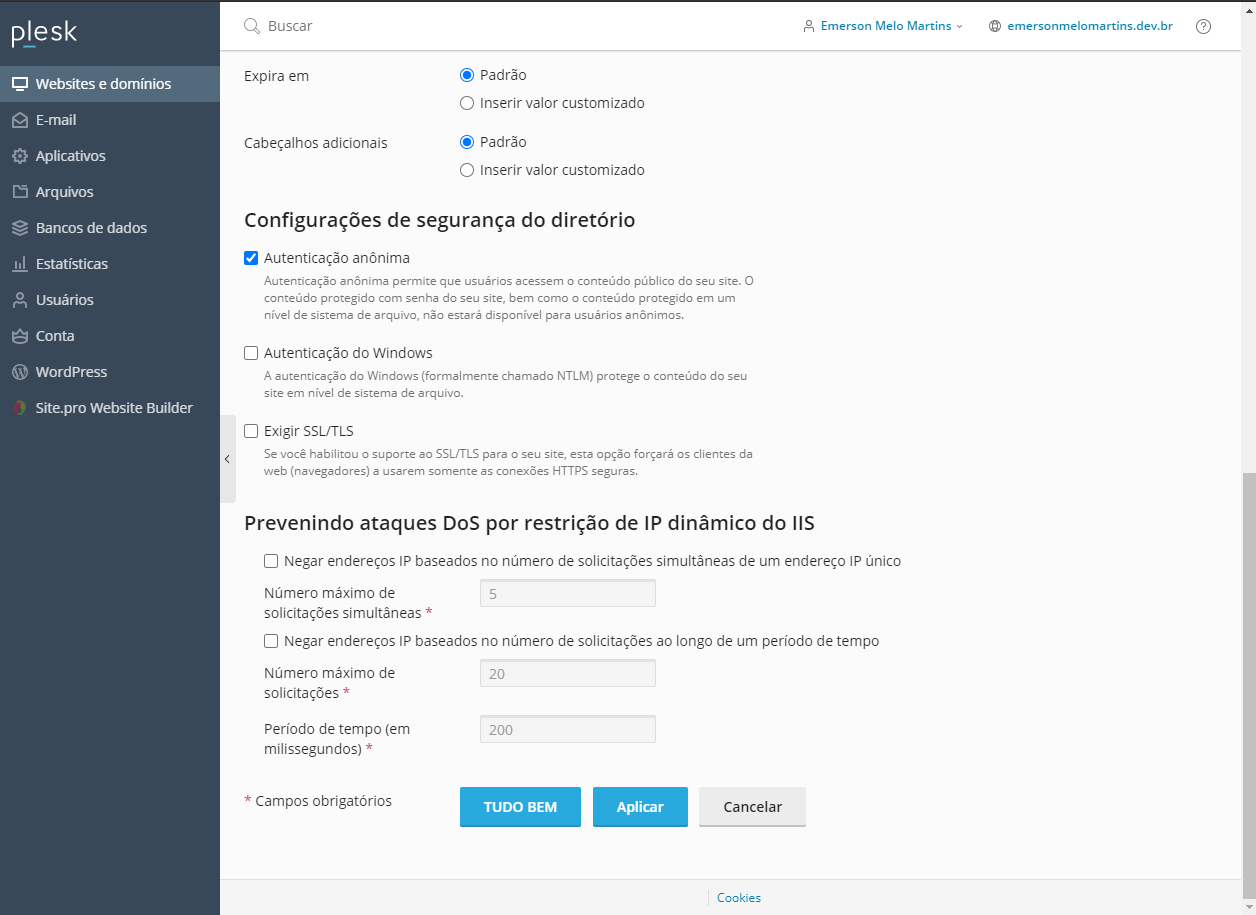
**

Figura 8 (Página do Servidor)

#### **11. VISÃO DE GERENCIAMENTO DE CÓDIGO**

Esta seção descreve a estrutura geral da manutenção e armazenamento do código fonte do sistema

##### **11.1 Visão Geral**

O código fonte do sistema estará armazenado em nuvem através da plataforma GITHUB, acessado pelo link:

* Backend: <https://github.com/emersonmelomartins/seuphone-web-backend-api-csharp>
* Frontend: <https://github.com/emersonmelomartins/seuphone-web-frontend-react>

Todo o código estará disponível na branch principal chamada “Master”.

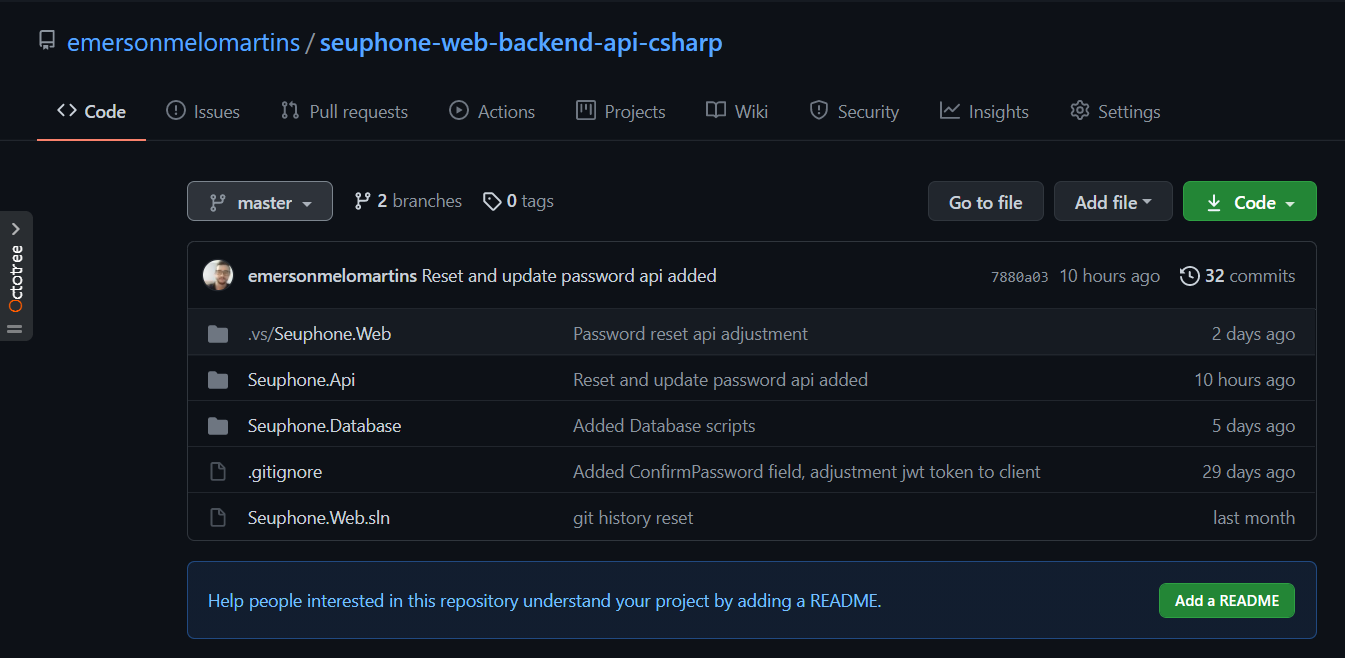


Figura 9 (GITHU)

**12. CONCLUSÃO**

Nesse projeto foi desenvolvido um sistema foco em e-commerce, onde podemos efetuar atividades como filtros de pesquisa, recuperação de senha e visualizar dados de produtos entre outros.

Para desenvolvimento web existem ferramentas que podem ser utilizadas. Muitas delas nasceram para facilitar e agilizar a rotina de um programador.

Utilizamos a ferramenta React para o desenvolvimento das telas do site e o C# para disponibilizar a API RESTde acordo com as regras de negócio. Com isso o projeto foi baseado na arquitetura MVC no qual nos permite ter uma melhor organização de responsabilidades de cada uma das partes.

O sistema cumpriu todos os objetivos propostos pelo projeto que é um site e-commerce.

**13. BIBLIOGRAFIA**

EIS, D.O Básico: O que é HTML? **Tableless**, 2011.Disponível em: < <https://tableless.com.br/o-que-html-basico/>>Acesso em: 03 de Outubro de 2020 ás 20:10 h.

EIS, D.O Básico: O que é HTML? **Tableless**, 2011.Disponível em: < <https://tableless.com.br/o-que-html-basico/>>Acesso em: 03 de Outubro de 2020 ás 20:10 h.

JUNIOR, M. Origem e significados de REST e RESTFUL. **Eximiaco. tech**,2019. Disponível em: < https://www.eximiaco.tech/pt/2019/09/13/origem-e-significado-de-rest-e-restful/ >Acesso em: 08 de Outubro de 2020 as 16:00 h.

MEDEIROS, H.Introdução ao Padrão MVC. **Devmedia**,2013.Disponível em:< https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308> Acesso em: 03 de Outubro de 2020 ás 15:44 h.

O que é um Diagrama de Classes. **Significados**, 2018. Disponível em :<<https://www.significados.com.br/diagrama-de-classes/>*>*Acesso em: 05 de Outubro de 2020 ás 15:13 h.

Os 7 benefícios do HTML5. **Celke**, Disponível em: <<https://celke.com.br/artigo/os-7-beneficios-do-html5> >Acesso em: 13 de novembro de 2019 às 12:59h.

Pena Nieradka, Itamar. Javascript+CSS+DOM: Desenvolvimento Para Web.1. ed. Rio de Janeiro: NovaTerra, 2014. Contracapa.

PEREIRA. A. P. O que é CSS? **Tecmundo** , 2009 .Disponível em: < https://www.tecmundo.com.br/programacao/2705-o-que-e-css.htm >Acesso em: 11 de novembro de 2019 as 14:06h.

Por que usar CSS? **MDN web docs moz://a** ,2021. Disponível em: < https://developer.mozilla.org/pt-BR/docs/conflicting/Learn/CSS/First\_steps/How\_CSS\_works\_61240ddc35a4ddd533ea33cc8ccf1c87 >Acesso em: 03 de Junho de 2021 ás 19:32 h.

RODRIGUES, J. Modelo Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER). **DevMedia**, 2014. Disponível em < https://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>Acesso em: 30 de Abril de 2020 às 18:40 h.

RAMOS, D. O Que Model-View-Controller(MVC)?.**Oficinadanet**,2011.Disponível em: <https://www.oficinadanet.com.br/artigo/desenvolvimento/o\_que\_e\_model-view-controller\_mvc > Acesso em: 09 de junho de 2020 ás 16:01 h.

SANTOS, C. J. Entendendo a Linguagem SQL . **Devmedia**, 2008 Disponível em:<

https://www.devmedia.com.br/entedendo-a-linguagem-sql/7775 > Acesso em: 02 de Junho de 2021 ás 19:10 h.

SOUZA, S. Entenda o que é REST API e a importância dele para o site da sua empresa. **Stage** ,2020. Disponível em: <<https://rockcontent.com/br/blog/rest-api/>>Acesso em: 08 de Outubro 2020 às 16:45 h.

SQL - o que é e como funciona na prática? **Pravaler** ,2020 Disponível em: https://www.pravaler.com.br/sql-o-que-e-e-como-funciona-na-pratica/> Acesso em: 02 de Junho de 2021 ás 19:32 h.

**ANEXO A – SCRIPT DE CRIAÇÃO BANCO DE DADOS.**

-- Script criação de banco de dados e tabelas

CREATE DATABASE Seuphone

ON

( NAME = SeuphoneDB,

FILENAME = 'C:\vs-workspace\seuphonedb.mdf',

SIZE = 10MB,

MAXSIZE = 500MB,

FILEGROWTH = 1MB )

LOG ON

( NAME = SeuphoneLOG,

FILENAME = 'C:\vs-workspace\seuphonelog.ldf',

SIZE = 10MB,

MAXSIZE = 200MB,

FILEGROWTH = 1MB

);

USE Seuphone;

CREATE TABLE tb\_provider (

Id int IDENTITY(1,1) NOT NULL,

CompanyName VARCHAR(60) NOT NULL,

CNPJ VARCHAR(25) NOT NULL UNIQUE,

ZipCode VARCHAR(10) NOT NULL,

Address VARCHAR(60) NOT NULL,

HouseNumber int NOT NULL,

District VARCHAR(60) NOT NULL,

City VARCHAR(40) NOT NULL,

State VARCHAR(20) NOT NULL,

CONSTRAINT PK\_tb\_provider PRIMARY KEY (Id)

)

CREATE TABLE tb\_role (

Id int IDENTITY(1,1) NOT NULL,

RoleName VARCHAR(20) NOT NULL UNIQUE,

Description VARCHAR(60) NULL,

CONSTRAINT PK\_tb\_role PRIMARY KEY (Id)

)

CREATE TABLE tb\_user (

Id int IDENTITY(1,1) NOT NULL,

Email VARCHAR(40) NOT NULL UNIQUE,

Password VARCHAR(60) NOT NULL,

ConfirmPassword VARCHAR(60) NOT NULL,

Token VARCHAR(60) NULL,

Name VARCHAR(40) NOT NULL,

Genre CHAR NOT NULL,

BirthDate DateTime NOT NULL,

CPF VARCHAR(14) NOT NULL UNIQUE,

ZipCode VARCHAR(10) NOT NULL,

Address VARCHAR(60) NOT NULL,

HouseNumber int NOT NULL,

District VARCHAR(40) NOT NULL,

City VARCHAR(40) NOT NULL,

State VARCHAR(20) NOT NULL,

CONSTRAINT PK\_tb\_user PRIMARY KEY (Id)

)

CREATE TABLE tb\_order (

Id int IDENTITY(1,1) NOT NULL,

UserId int NOT NULL,

Total float NOT NULL,

ContractDuration int NOT NULL,

CreationDate DateTime NOT NULL,

OrderStatus int NOT NULL,

PaymentMethod int NOT NULL,

OrderType int NOT NULL,

CONSTRAINT PK\_tb\_order PRIMARY KEY (Id),

CONSTRAINT FK\_tb\_order\_tb\_user\_UserId FOREIGN KEY (UserId) REFERENCES tb\_user(Id)

)

CREATE TABLE tb\_product (

Id int IDENTITY(1,1) NOT NULL,

ProductName VARCHAR(60) NOT NULL,

Description VARCHAR(60) NOT NULL,

Model VARCHAR(60) NOT NULL,

Color VARCHAR(60) NOT NULL,

Storage VARCHAR(60) NOT NULL,

Price float NOT NULL,

StockQuantity int NOT NULL,

[Image] varchar(MAX) NOT NULL,

ProviderId int NOT NULL,

CONSTRAINT PK\_tb\_product PRIMARY KEY (Id),

CONSTRAINT FK\_tb\_product\_tb\_provider\_ProviderId FOREIGN KEY (ProviderId) REFERENCES tb\_provider(Id)

)

CREATE TABLE tb\_user\_role (

UserId int NOT NULL,

RoleId int NOT NULL,

CONSTRAINT PK\_tb\_user\_role PRIMARY KEY (UserId,RoleId),

CONSTRAINT FK\_tb\_user\_role\_tb\_role\_RoleId FOREIGN KEY (RoleId) REFERENCES tb\_role(Id),

CONSTRAINT FK\_tb\_user\_role\_tb\_user\_UserId FOREIGN KEY (UserId) REFERENCES tb\_user(Id)

)

CREATE TABLE tb\_order\_items (

Id int IDENTITY(1,1) NOT NULL,

Quantity int NOT NULL,

SubTotal float NOT NULL,

OrderId int NOT NULL,

ProductId int NOT NULL,

CONSTRAINT PK\_tb\_order\_items PRIMARY KEY (Id),

CONSTRAINT FK\_tb\_order\_items\_tb\_order\_OrderId FOREIGN KEY (OrderId) REFERENCES tb\_order(Id),

CONSTRAINT FK\_tb\_order\_items\_tb\_product\_ProductId FOREIGN KEY (ProductId) REFERENCES tb\_product(Id)

)

**ANEXO B – SCRIPT DE INSERÇÃO DE DADOS NO BANCO DE DADOS.**

-- Script inserção de dados no banco de dados

USE Seuphone;

-- TB\_USER

INSERT INTO tb\_user VALUES ('emerson@email.com', '123456', '123456',

'698dc19d489c4e4db73e28a713eab07b', 'Emerson Melo', 'M', '05/06/1995',

'316.678.480-20', '09360-490', 'Av. Caetano Scila', '285', 'Vila Assis Brasil', 'Mauá', 'SP' );

INSERT INTO tb\_user VALUES ('thais@email.com', '654321', '654321',

'e8a88bb6f4d420a8517965d25cd54a14', 'Thais Martins', 'F', '01/01/1990',

'011.615.830-17', '09112-000', 'Rua Silla Nalon Gonzaga', '70', 'Parque Marajoara', 'Santo André', 'SP' );

INSERT INTO tb\_user VALUES ('everton@email.com', '123654', '123654',

'70ecdf02ed3885202139357cc595622d', 'Everton Chabariberi', 'M', '02/02/1989',

'620.305.390-23', '09112-000', 'Rua Silla Nalon Gonzaga', '70', 'Parque Marajoara', 'Santo André', 'SP' );

-- TB\_ROLE

INSERT INTO tb\_role VALUES ('ROLE\_ADMIN', 'Administrador do Sistema');

INSERT INTO tb\_role VALUES ('ROLE\_CLIENTE', 'Cliente do Sistema');

INSERT INTO tb\_role VALUES ('ROLE\_DEV', 'Desenvolvedor do Sistema');

-- TB\_USER\_ROLE

INSERT INTO tb\_user\_role VALUES (1, 1);

INSERT INTO tb\_user\_role VALUES (2, 1);

INSERT INTO tb\_user\_role VALUES (3, 2);

-- TB\_PROVIDER

INSERT INTO tb\_provider VALUES ('Apple Computer', '47.071.378/0001-40',

'04542-000', 'Av Leopoldo C de Magalhaes Jr', 700, 'Bota Fogo', 'São Paulo', 'SP');

INSERT INTO tb\_provider VALUES ('Rei das Capinhas', '92.655.064/0001-49',

'09932-012', 'Rua Lazar Segal', 120, 'Brasil', 'São Paulo', 'SP');

INSERT INTO tb\_provider VALUES ('Carregadores Apple', '94.336.917/0001-05',

'09360-888', 'Rua Ruerato', 458, 'Vila América', 'Santo André', 'SP');

-- TB\_PRODUCT

INSERT INTO tb\_product VALUES('iPhone 12', 'iPhone 12 - 64GB - Azul',

'iPhone 12', 'Azul', '64GB', 6999.99, 0, 'codigo\_base64', 1);

INSERT INTO tb\_product VALUES('iPhone 12 Pro', 'iPhone 12 Pro - 128GB - Vermelho',

'iPhone 12 Pro', 'Vermelho', '128GB', 8999.99, 0, 'codigo\_base64', 1);

INSERT INTO tb\_product VALUES('iPhone 12 Pro Max', 'iPhone 12 Pro Max - 512GB - Grafite',

'iPhone 12 Pro Max', 'Grafite', '512GB', 12999.99, 0, 'codigo\_base64', 1);

-- TB\_ORDER

INSERT INTO tb\_order VALUES (1, 15000.99, 2, '27/05/2021', 1, 2, 1);

INSERT INTO tb\_order VALUES (2, 9999.99, 1, '26/05/2021', 2, 1, 1);

INSERT INTO tb\_order VALUES (1, 6999.00, 2, '25/05/2021', 1, 1, 2);

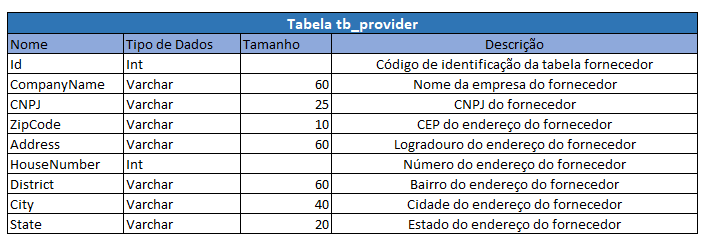
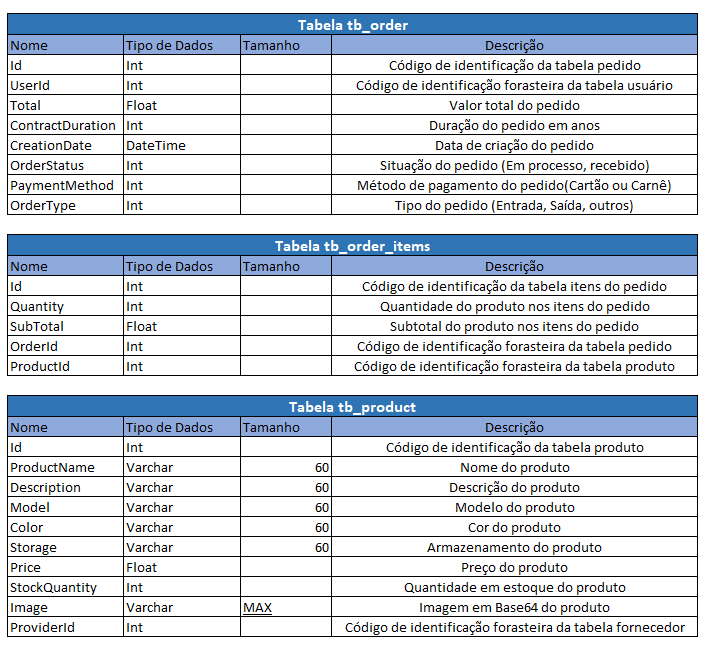
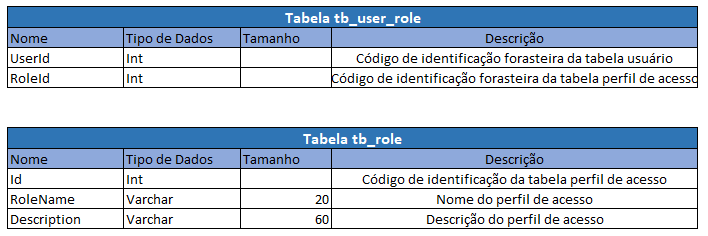
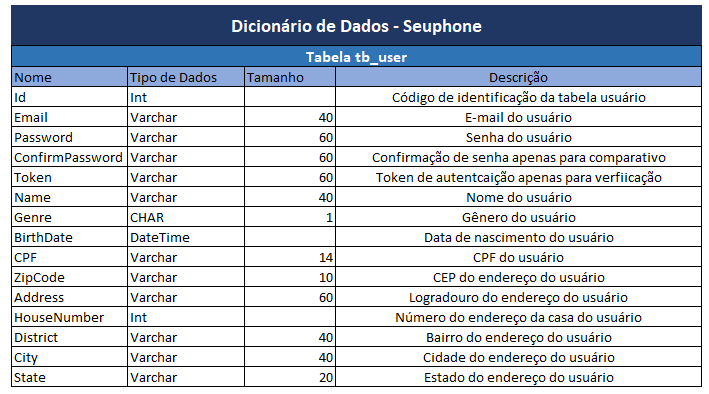
-- TB\_ORDER\_ITEMS

INSERT INTO tb\_order\_items VALUES(2, 9999.99, 1, 1);

INSERT INTO tb\_order\_items VALUES(5, 29999.99, 1, 3);

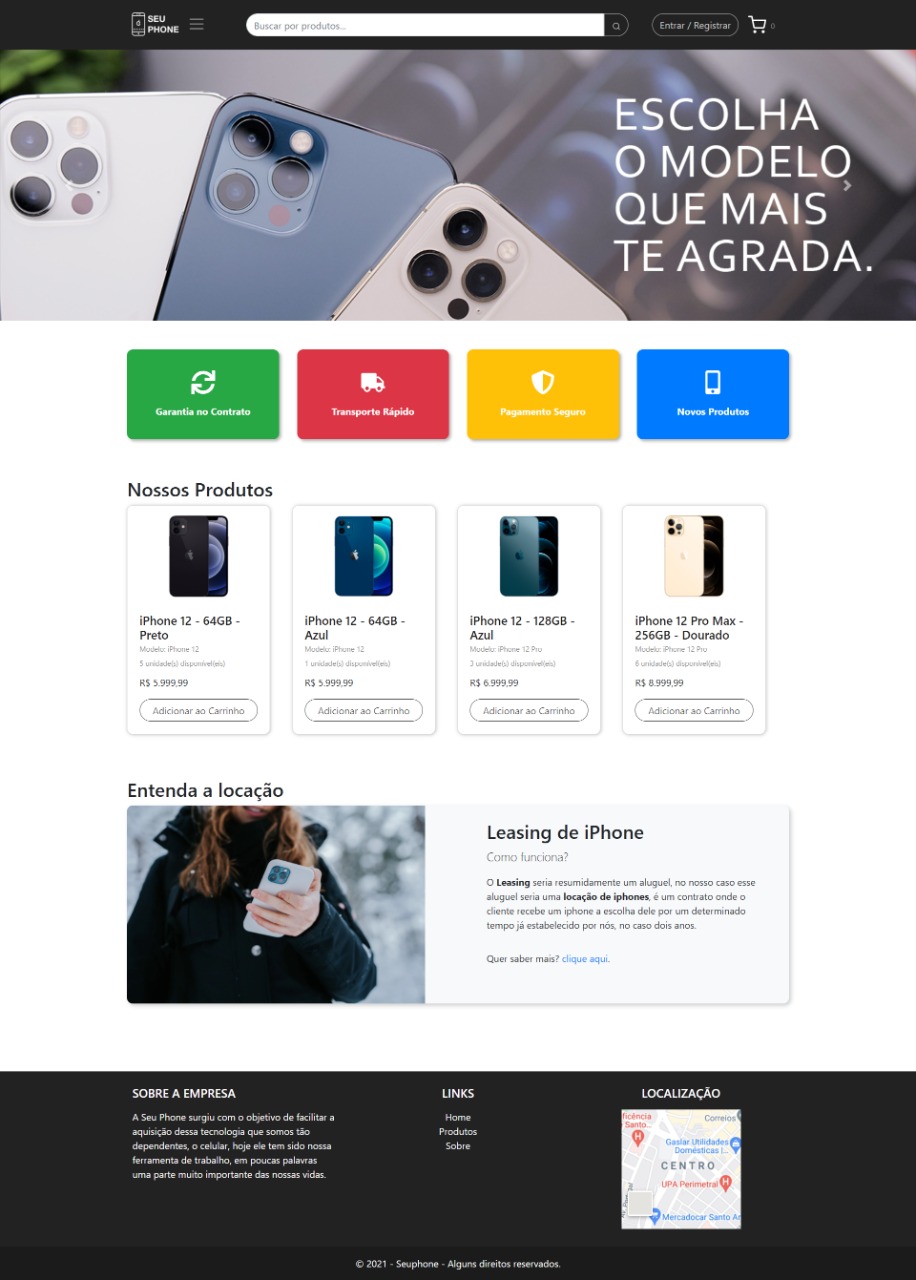
INSERT INTO tb\_order\_items VALUES(1, 6999.99, 2, 2);

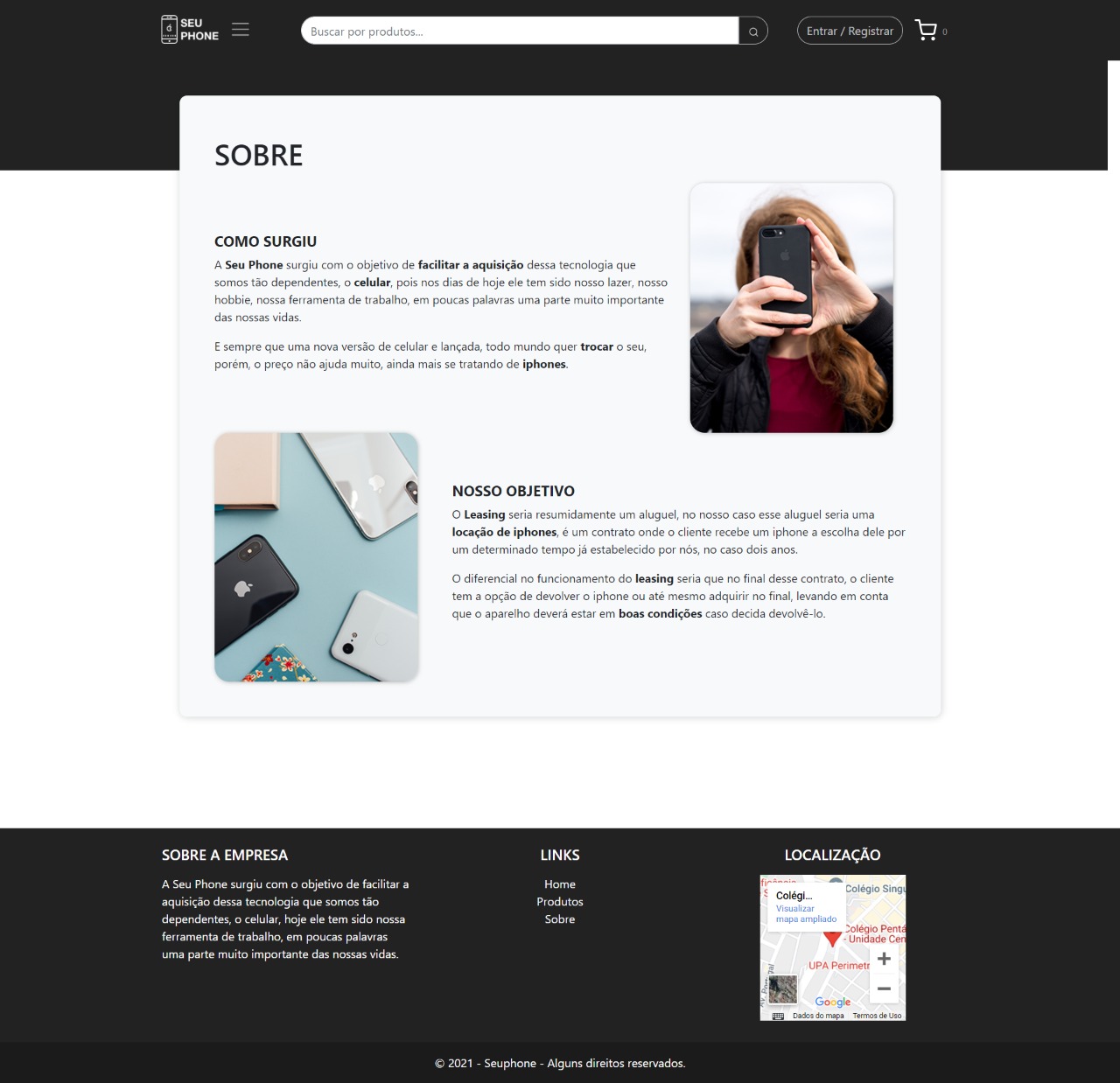
**ANEXO C – TABELA DICIONÁRIO DE DADOS.**

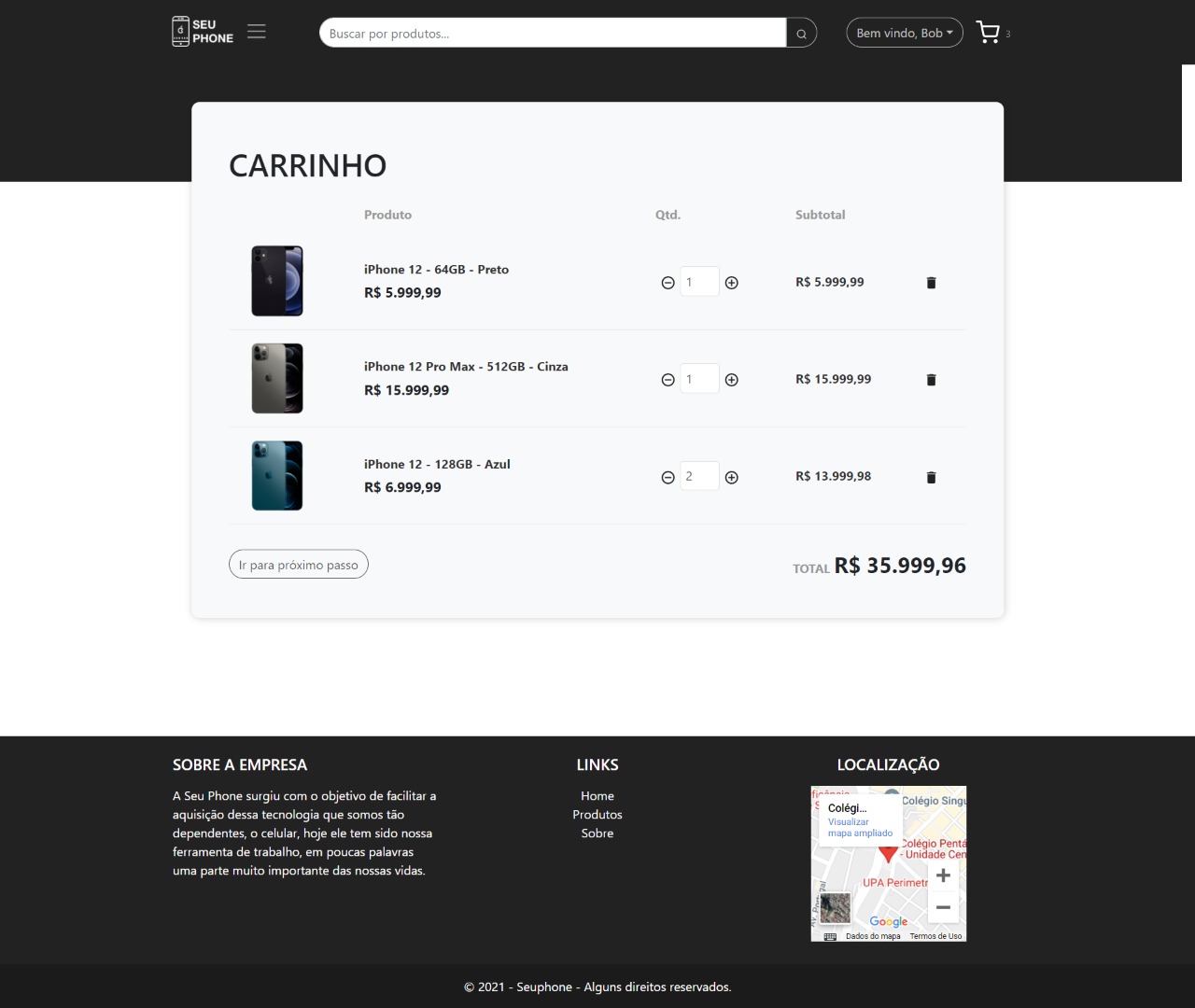


**ANEXO D – TELAS DO SISTEMA**

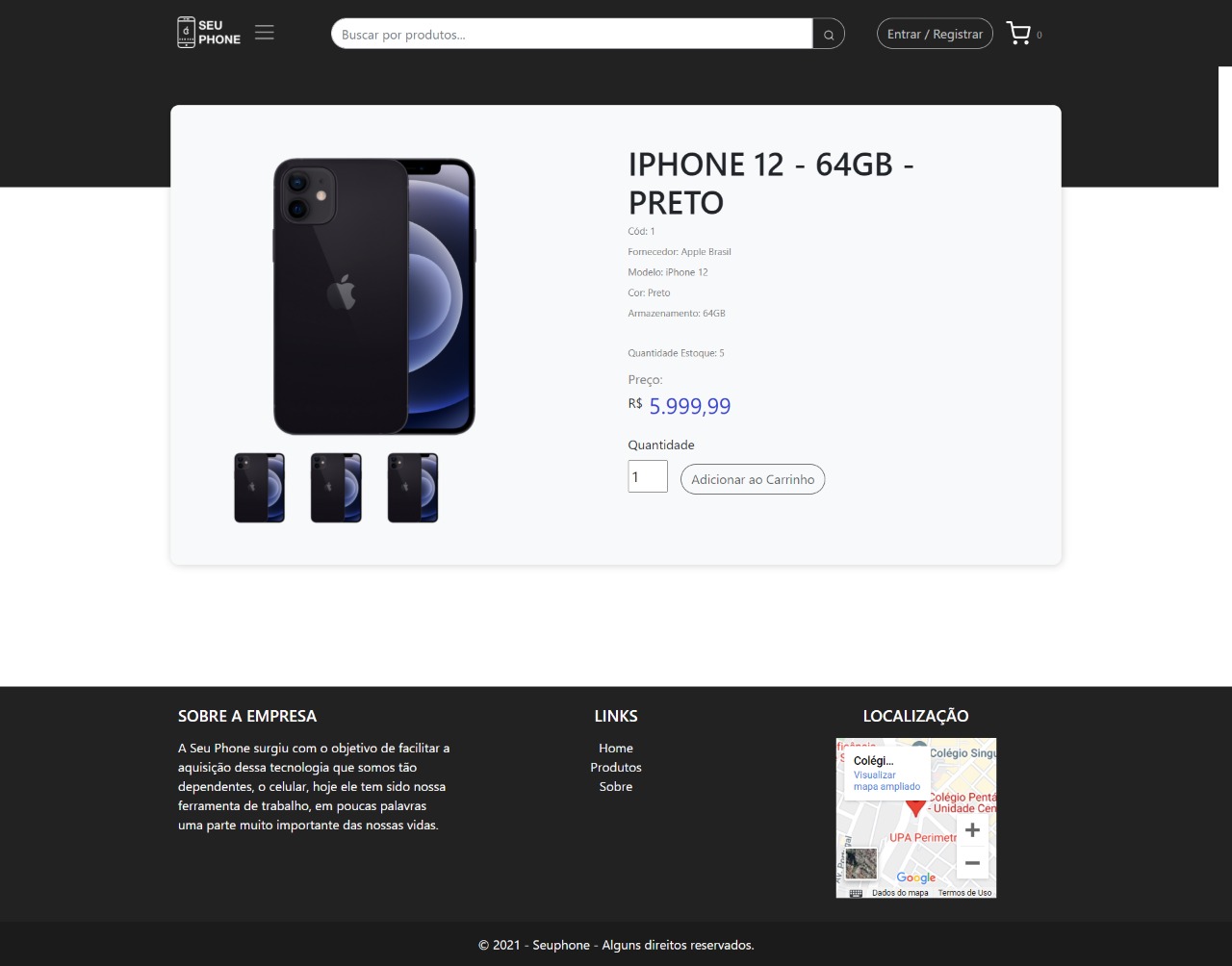
**Figura 10:** Tela inicial.



**Figura 11:** Tela Sobre.

**Figura 12:** Tela Carrinho.

**Figura 13:** Tela Detalhe Produto.



**Figura 14:** Tela Perfil.

