

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA**

**Faculdade de Tecnologia Baixada Santista
Rubens Lara**

**Curso Superior de Tecnologia em
Sistemas para Internet**

**BRUNO DA SILVA PEREIRA
GUSTAVO LUIGI BRANCACIO ALVES BUENO
RAUL CALDAS TESTA**

SOFTWARE DE GESTÃO DE LOJAS DE ROUPAS

**Santos, SP
2023**

**BRUNO DA SILVA PEREIRA
GUSTAVO LUIGI BRANCACIO ALVES BUENO
RAUL CALDAS TESTA**

SOFTWARE DE GESTÃO DE LOJAS DE ROUPAS

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia Rubens Lara, como exigência para a obtenção do Título de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Felipe Cannarozzo Lourenço

**Santos, SP
2023**

RESUMO

Este trabalho apresenta a análise e implantação de um sistema para ajudar empresários a gerenciar seus estoques e vendas de lojas de roupas. O sistema foi criado com base nas necessidades de informatização de uma loja, e tem como objetivo facilitar o atendimento aos clientes, proporcionar maior controle de dados e trazer informações claras e objetivas para a tomada de decisões. Para entender melhor as necessidades de um possível, foi feita uma análise de uma rotina e dos principais possíveis problemas que enfrentaria na gestão da loja. Com base nessas informações, foram levantadas as principais funcionalidades do sistema e, em seguida, ele foi desenvolvido. Os resultados obtidos demonstram que o software atenderia os principais requisitos para o gerenciamento de uma loja e pode ser útil para outros estabelecimentos de ramos similares, ajudando empreendedores a otimizar seus negócios, crescer e atrair mais clientes.

Palavras-chaves: Gerenciamento. Estoque. Otimização.

ABSTRACT

This work presents the analysis and implementation of a system to help entrepreneurs manage their inventory and sales in clothing stores. The system was created based on the computerization needs of a store, and aims to facilitate customer service, provide greater data control, and provide clear and objective information for decision-making. To better understand the needs of a customer, an analysis was made of a routine and the main potential problems it would face in managing the store. Based on this information, the main functionalities of the system were raised and then it was developed. The results show that the software would meet the main requirements for managing a store and can be useful for other establishments in similar fields, helping entrepreneurs to optimize their businesses, grow and attract more customers.

Keywords: Management. Stock. Optimization.

LISTA DE ABREVIATURAS E SIGLAS

IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA.....	15
UML - LINGUAGEM DE MODELAGEM UNIFICADA	18
CPF - CADASTRO DE PESSOA FÍSICA	20
SGBD - SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS	21
MER - MODELO ENTIDADE RELACIONAMENTO	21
BAAS - BACKEND AS A SERVICE	21
API - APPLICATION PROGRAMMING INTERFACE	21
REST - REPRESENTATIONAL STATE TRANSFER	21
CRUD - CREATE, READ, UPDATE, DELETE	22
SPA - SINGLEPAGE APPLICATION.....	26

LISTA DE ILUSTRAÇÕES

ILUSTRAÇÃO 01 – ARQUIVOS CRIADOS PARA A ORGANIZAÇÃO DOS SERVIÇOS	23
ILUSTRAÇÃO 02 – LISTA DE MÉTODOS DISPONÍVEIS DO SERVIÇO DE PRODUTOS	23
ILUSTRAÇÃO 03 – MÉTODO UTILIZADO PARA LISTAR OS PRODUTOS REGISTRADOS	24
ILUSTRAÇÃO 04 – CAMADA DE VISUALIZAÇÃO ONDE CONTÉM A CHAMADA AO MÉTODO DE LISTAGEM DE PRODUTOS	25

SUMÁRIO

1 INTRODUÇÃO.....	15
1.1 OBJETIVO	15
1.1.1 OBJETIVO GERAL.....	15
1.1.2 OBJETIVOS ESPECÍFICOS	16
1.2 ESTADO DA ARTE.....	16
2 DESENVOLVIMENTO.....	18
2.1 ANÁLISE DO SISTEMA	18
2.1.1 ANÁLISE DE REQUISITOS	19
2.1.2 DIAGRAMA DE CASO DE USO.....	19
2.1.3 FLUXO DE EVENTOS.....	20
2.2 BANCO DE DADOS	20
2.2.1 TECNOLOGIAS COMPLEMENTARES.....	21
2.3 CAMADA DE NEGÓCIO	22
2.4 CAMADA DE APRESENTAÇÃO	255
3 RESULTADO	27
REFERÊNCIAS BIBLIOGRÁFICAS	29
APÊNDICE A	291
APÊNDICE B	32
APÊNDICE C	34
APÊNDICE D	37
APÊNDICE E	38

1 INTRODUÇÃO

De acordo com uma pesquisa realizada pelo IBGE (Instituto Brasileiro de Geografia e Estatística) em 2020, 80% das micro e pequenas empresas fecham antes de completar um ano, e cerca de 60% fecham antes dos cinco anos, devido principalmente a problemas de gestão financeira e de estoque. Essa realidade se aplica também às lojas de roupas, que enfrentam desafios específicos em relação à gestão de estoque, como controlar a entrada e saída de produtos, gerenciar os tamanhos e cores disponíveis, e evitar prejuízos com a falta de giro de estoque.

Para evitar que sua loja faça parte dessa estatística, é fundamental contar com um controle eficiente das operações e do estoque. Pensando nisso, foi desenvolvido um sistema que facilita o gerenciamento de vendas, estoque e controle dos materiais utilizados. Esse sistema permite uma gestão mais ágil e precisa dos materiais disponíveis, com informações sobre as vendas realizadas e os produtos em estoque em tempo real.

A pesquisa do IBGE deixa claro que um dos maiores desafios das pequenas empresas é manter uma boa gestão de estoque e controle financeiro. O uso de um sistema para controle de estoque e vendas pode ser a solução para reduzir custos, analisar os resultados da empresa, melhorar preços de venda e buscar por mais oportunidades de melhoria. Além disso, o sistema apresenta os dados de maneira clara e objetiva, permitindo uma tomada de decisão mais assertiva e eficiente.

Dessa forma, investir em um sistema para gestão de estoque e vendas é uma estratégia essencial para garantir a longevidade e sucesso de sua loja de roupas, proporcionando uma vantagem competitiva no mercado.

1.1 OBJETIVO

O objetivo deste projeto é aprimorar o fluxo de trabalho em lojas de roupas, minimizando erros humanos e possibilitando uma visualização clara de eventos diários, consultas a registros anteriores e um gerenciamento eficiente de estoque.

1.1.1 OBJETIVO GERAL

Desenvolver um sistema de gestão para uma loja de roupas. Onde será possível cadastrar produtos e vendas, como também gerar relatórios das vendas.

1.1.2 OBJETIVOS ESPECÍFICOS

O sistema de gerenciamento de loja de roupas irá proporcionar aos proprietários e colaboradores um controle mais preciso do estoque, possibilitando a realização de um inventário mais assertivo, que permitirá identificar a necessidade de reposição de produtos, prevenindo a falta de itens mais vendidos e evitando desperdício de espaço com produtos pouco vendidos.

Além disso, a funcionalidade de gerenciamento de vendas permitirá que os colaboradores registrem as vendas de forma mais rápida e eficiente, evitando erros de registro e possibilitando a identificação dos produtos mais vendidos e os que estão em estoque por meio de relatórios gerenciais.

Outra vantagem do sistema de gerenciamento de loja de roupas é a possibilidade de consulta a dados de clientes, como histórico de compras, preferências e informações de contato, permitindo que os proprietários e colaboradores possam oferecer um atendimento personalizado e mais eficiente.

Com essas funcionalidades, o sistema de gerenciamento de loja de roupas se torna uma ferramenta essencial para a gestão eficiente de uma loja de roupas, possibilitando uma melhor organização, controle financeiro e um atendimento mais satisfatório aos clientes.

1.2 ESTADO DA ARTE

Existem várias opções de sistemas para gerenciamento de lojas de roupas, desde *softwares* mais complexos até aplicativos mais simples. Entre as opções mais populares no mercado, prede-se citar o Hiper, o *Bling* e o Conta Azul.

O Hiper é um aplicativo desenvolvido pelo grupo *Stone*, que permite o gerenciamento de estoque e vendas de forma integrada. Ele possui uma interface intuitiva, fácil de usar, e pode ser utilizado em *smartphones*, *tablets* ou computadores. Além disso, o Hiper oferece recursos como cadastro de clientes, controle de contas a pagar e a receber, emissão de notas fiscais, entre outros. As principais vantagens do Hiper são a sua facilidade de uso e o fato de ser um aplicativo gratuito. No entanto,

ele pode apresentar limitações em termos de funcionalidades para lojas de maior porte.

O *Bling* é um *software* de gestão empresarial que oferece recursos para gerenciamento de estoque, vendas, financeiro, emissão de notas fiscais e outras funcionalidades. Ele é indicado para micro e pequenas empresas, e pode ser integrado a diversas plataformas de e-commerce. O *Bling* possui uma interface amigável, fácil de usar, e oferece recursos como gestão de pedidos de venda, controle de comissões, entre outros. As principais vantagens do *Bling* são a sua capacidade de integração com outras plataformas, a variedade de recursos oferecidos e o suporte técnico de qualidade. No entanto, ele pode ser um pouco mais caro que outras opções de sistemas.

O Conta Azul é um *software* de gestão empresarial que permite o gerenciamento de estoque, vendas, financeiro, emissão de notas fiscais e outras funcionalidades. Ele é indicado para micro e pequenas empresas, e oferece uma interface fácil de usar e recursos como gestão de fluxo de caixa, controle de contas a pagar e a receber, entre outros. As principais vantagens do Conta Azul são a sua facilidade de uso e a integração com outras plataformas, como bancos e sistemas de pagamento online. No entanto, ele pode apresentar limitações em termos de personalização de relatórios e funcionalidades mais avançadas.

É importante ressaltar que, antes de escolher um sistema para gerenciamento de lojas de roupas, é preciso levar em conta as necessidades específicas do negócio e avaliar as funcionalidades e preços de cada opção disponível no mercado.

2 DESENVOLVIMENTO

Neste capítulo, serão apresentadas as etapas do desenvolvimento do sistema proposto, incluindo a análise do sistema, a estruturação do banco de dados, as ferramentas utilizadas para o desenvolvimento do *front-end* e as principais rotinas implementadas no *back-end*.

Inicialmente, foi realizada uma análise minuciosa do sistema para identificar as necessidades do usuário e entender o processo de gestão de estoque e vendas em uma loja de roupas. Com base nessas informações, a equipe responsável pelo desenvolvimento criou uma estrutura de banco de dados eficiente, que permite a organização e armazenamento de dados importantes para o gerenciamento da loja.

Para a criação da interface do usuário (*front-end*), foram utilizadas ferramentas modernas e de fácil utilização, garantindo uma experiência agradável e intuitiva ao usuário do sistema. Por sua vez, o *back-end* foi desenvolvido com base nas necessidades identificadas na análise do sistema, utilizando linguagens de programação robustas e eficientes.

Entre as rotinas implementadas no *back-end*, destacam-se aquelas relacionadas à gestão do estoque, como o controle de entrada e saída de produtos, a atualização de estoque em tempo real e a geração de relatórios para análise do desempenho da loja. Além disso, foram implementadas rotinas para gestão de vendas, como o controle de vendas realizadas, emissão de notas fiscais eletrônicas e gestão de pedidos de clientes.

2.1 ANÁLISE DO SISTEMA

Para a análise do sistema proposto, foram considerados diversos mecanismos disponíveis, e a escolha foi pela *Unified Modeling Language (UML)*. Essa linguagem de modelagem é uma forma visual de especificar um sistema, o que oferece diversas vantagens em relação a notações textuais. Com a *UML*, é possível obter uma visão geral rápida e simplificada das partes do sistema e de suas relações, o que facilita a compreensão do projeto.

No desenvolvimento do *front-end*, foi escolhida a linguagem *JavaScript* com a biblioteca *React*. Essa escolha se justifica por ser uma das mais populares e

amplamente utilizadas por grandes empresas em todo o mundo, como *Instagram*, *Facebook*, *Netflix*, *Twitter* e *Uber*. Além disso, o *React* trabalha com componentes, o que facilita a manutenção e evolução do código, bem como o reaproveitamento de códigos já desenvolvidos.

Para a estrutura do banco de dados, foram utilizados os conceitos de modelagem de dados, visando a organização eficiente e otimizada das informações do sistema. As principais rotinas implementadas no *back-end* incluem o gerenciamento de estoque, controle de vendas e geração de relatórios.

Com essas ferramentas e tecnologias, foi possível desenvolver um sistema de gerenciamento de loja de roupas eficiente e de fácil uso, que proporciona uma visão ampla e integrada do negócio, auxiliando na tomada de decisões e na otimização dos processos.

2.1.1 ANÁLISE DE REQUISITOS

Nesta fase ocorre a captura das intenções e as necessidades dos usuários a ser desenvolvido através dos requisitos funcionais (RF) que nesse caso serão:

[RF01] O usuário administrador adiciona produtos

[RF02] O usuário administrador altera produtos

[RF03] O usuário administrador exclui produtos

[RF04] O usuário administrador consulta as vendas

[RF05] O usuário administrador cadastra clientes

[RF06] O usuário administrador altera informações dos clientes

[RF07] O usuário administrador exclui clientes

2.1.2 DIAGRAMA DE CASO DE USO

Com os requisitos funcionais estabelecidos, o próximo passo é a criação do diagrama de caso de uso que tem como finalidade a representação gráfica e semântica da interação do usuário e o sistema, segundo Antônio Mendes, autor do livro *Custo de software*, o diagrama de caso de uso é:

“um modelo que descreve o comportamento do sistema sob a ótica da interação de um ator com uma funcionalidade. O ator pode ser um usuário, subsistema (de *software*) ou algum dispositivo de *hardware* (ou equipamento).” (Antônio Mendes, 2014, pág. 17)

Essa análise foi feita a partir dos casos de uso e dos cenários registrados no apêndice A.

O diagrama de caso de uso desse sistema pode ser visualizado no apêndice B.

2.1.3 FLUXO DE EVENTOS

O objetivo do fluxo de eventos é descrever o início e o fim de cada caso de uso, bem como estabelecer a interação dos atores no processo. Para isso, são criados fluxos principais e alternativos, que ilustram o comportamento de cada caso de uso em diagramas que mostram como os dados são inseridos no sistema.

O primeiro fluxo diz respeito ao cadastro de um novo produto. Nesse caso, o sistema verifica se o produto já existe no banco de dados. Caso contrário, um novo produto é cadastrado.

Já o segundo fluxo descreve como ocorre o cadastro de um novo cliente. Nesse caso, o sistema também verifica se já existe um cadastro anterior do cliente, com base no seu CPF (Cadastro de Pessoa Física). Se não houver um cadastro, as informações são inseridas no banco de dados.

Por fim, o terceiro fluxo descreve o cadastro de uma venda no sistema. Nesse caso, o administrador insere as informações relevantes, que são salvas no banco de dados.

Os diagramas dos fluxos de eventos desse sistema podem ser encontrados no Apêndice C, oferecendo uma visualização clara e precisa de cada processo descrito.

2.2 BANCO DE DADOS

Um banco de dados é uma coleção organizada de dados inter-relacionados que representam informações sobre um domínio específico. Existem

diferentes tipos de bancos de dados, sendo os mais comuns o banco de dados relacional e o banco de dados não relacional.

No banco de dados relacional, as informações são organizadas em tabelas, em que cada linha representa um registro com uma chave primária única, e cada coluna representa um campo do registro. Os dados em diferentes tabelas podem ter relacionamentos, usando chaves estrangeiras.

Por outro lado, no banco de dados não relacional, a estrutura é baseada em chave-valor, sendo mais adequado para lidar com grandes volumes de dados, sem precisar de uma estruturação tão rigorosa.

Para gerenciar um banco de dados, existem sistemas de gerenciamento de banco de dados (SGBD), que são softwares que permitem a manipulação dos dados. Alguns exemplos são *Oracle*, *SQL Server*, *DB2*, *PostgreSQL* e *MySQL*.

No projeto em questão, optou-se pelo uso de um banco de dados relacional, com o *PostgreSQL* como SGBD. Para modelar o banco de dados, utilizou-se o Modelo Entidade Relacionamento (MER), um diagrama que permite definir as tabelas e os relacionamentos entre elas.

O ponto central do sistema é a tabela de produtos, que possui variações como cor, tamanho e marca. Identificou-se também a necessidade de uma tabela de vendas, que armazena informações sobre as vendas realizadas, incluindo a lista de produtos relacionados a cada venda e o cliente que realizou a compra.

Com base nessas informações, foi elaborado o Modelo Entidade Relacionamento, que pode ser visualizado no Apêndice D.

2.2.1 TECNOLOGIAS COMPLEMENTARES

Neste projeto, foi optado por utilizar o *Supabase*, uma ferramenta de *BaaS* (*Back-end as a Service*) gratuita e de código aberto que atende aos requisitos necessários para a construção do sistema. O *Supabase* utiliza diversas tecnologias, como *Javascript*, *Typescript*, *Go*, *Elixir*, *Haskell*, *Tailwind*, *React*, *PostgREST*, *Pulumi* e *Kong*, entre outras.

De maneira resumida, o *Supabase* fornece uma interface para criar um banco de dados *PostgreSQL* e disponibiliza uma *RESTful* API (Application Programming Interface) através da ferramenta *PostgREST*. O *PostgREST* é descrito em sua documentação como uma "alternativa à programação manual do CRUD". Isso significa que a ferramenta cria uma API (Application Programming Interface) *RESTful*

(*Representational State Transfer*) com base no banco de dados *PostgreSQL*, fornecendo *endpoints* para realizar operações CRUD (*Create, Read, Update e Delete*).

Com o servidor desenvolvido em *Haskell*, o *Supabase* tem uma ótima performance.

Além da facilidade de uso e de sua performance, o *Supabase* oferece outras vantagens para os desenvolvedores. Ele é altamente escalável e permite que os desenvolvedores foquem em construir a lógica de negócios do sistema, sem se preocupar com a infraestrutura de banco de dados e API.

Outra vantagem do *Supabase* é sua integração com diversas ferramentas e tecnologias populares, como o *React* e o *Tailwind*. Isso permite que os desenvolvedores criem aplicativos web modernos e responsivos de forma rápida e eficiente.

Além disso, o *Supabase* é totalmente gratuito em alguns de seus planos, o que o torna uma opção atrativa para startups e pequenas empresas que precisam de um *backend* eficiente sem gastar muito dinheiro.

Em resumo, o *Supabase* é uma excelente escolha para quem precisa construir um sistema web escalável e eficiente, sem gastar muito tempo e recursos na infraestrutura de *backend*. Com sua interface intuitiva, ferramentas poderosas e integrações populares, o *Supabase* é uma das melhores opções do mercado de *BaaS* atualmente.

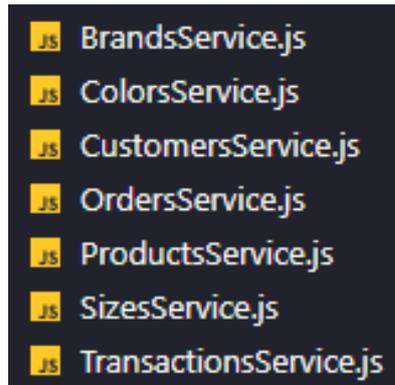
2.3 CAMADA DE NEGÓCIO

No contexto deste projeto, as regras de negócio foram implementadas no *frontend*, enquanto o *backend*, gerenciado pelo *Supabase*, foi utilizado para fornecer os *endpoints* necessários para a aplicação. Para lidar com a comunicação entre a aplicação e o *backend*, foi criada uma camada intermediária chamada *Service Layer*.

A *Service Layer* consiste em um conjunto de arquivos responsáveis por gerenciar a comunicação com o *backend*. É nessa camada que ocorre a integração com a *API RESTful* fornecida pelo *Supabase*. Com essa abstração, o *frontend* pode se comunicar com o *backend* de maneira mais simples e eficiente, sem precisar lidar diretamente com a implementação da API.

Como podemos ver na próxima imagem, essa separação de responsabilidades entre *frontend* e *backend* permite uma melhor organização do código e torna a manutenção e evolução da aplicação mais fácil e escalável. Além disso, a utilização do *Supabase* como ferramenta de *backend as a service*, proporciona uma redução no tempo de desenvolvimento e a possibilidade de focar no desenvolvimento das funcionalidades principais da aplicação.

Figura 01 – Arquivos criados para a organização dos serviços



FONTE: Autores, 2023

Dentro de cada serviço, são disponibilizados métodos que podem ser utilizados pelo restante da aplicação quando necessário. Esses métodos são projetados para atender às necessidades específicas do serviço, o que permite uma melhor organização do código e maior facilidade de manutenção. Além disso, a utilização de serviços permite que as regras de negócio fiquem encapsuladas e centralizadas em um único local, o que ajuda a evitar duplicação de código e aumenta a consistência do sistema como um todo. Na imagem abaixo podemos ver alguns dos métodos criados:

Figura 02 - Lista de métodos disponíveis do serviço de Produtos.

```
class ProductsService {  
  async listProducts(searchTerm = null) { ...  
  }  
  
  async getProduct(productId) { ...  
  }  
  
  async listVariations() { ...  
  }  
}
```

FONTE: Autores, 2023

O serviço de listagem de produtos é fundamental para o funcionamento da aplicação. Na imagem a seguir, pode-se ver como ele funciona. Primeiramente, o método busca todos os registros que estão armazenados na tabela "produtos" no banco de dados *PostgreSQL*. Em seguida, é feita uma cláusula *LEFT JOIN* para adicionar registros de outra tabela que estão relacionados com o produto que foi buscado. Neste caso, busca-se o produto e sua respectiva marca, que está armazenada em outra tabela.

Em seguida, é utilizado o método "*ilike*", fornecido pela ferramenta, para filtrar os resultados com base em palavras-chave digitadas pelo usuário. Por fim, é realizada a ordenação da lista de produtos em ordem crescente, tendo como base o campo "*name*".

Essas etapas são essenciais para fornecer ao usuário uma lista de produtos precisa e organizada, facilitando a busca pelo produto desejado. O serviço de listagem de produtos, que podemos ver na imagem a seguir, é apenas um exemplo de como os serviços da camada *Service Layer* podem ser implementados para facilitar o acesso e o gerenciamento dos dados no *backend* da aplicação.

Figura 03 - Método utilizado para listar os produtos registrados.

```
async listProducts(searchTerm = null) {  
  const { data: products, error } = await supabase  
    .from('products')  
    .select(  
      `*,  
      brands(  
        name  
      )`,  
    )  
    .ilike('name', `%${searchTerm || ''}%`)  
    .order('name', { ascending: true });  
  
  const formattedProducts = products.map((product) => ({  
    id: product.id,  
    name: product.name,  
    price: product.price,  
    desc: product.desc,  
    brand: product.brands.name,  
  }));  
  
  if (error) throw error;  
  
  return formattedProducts;  
}
```

Na camada de visualização, utiliza-se uma técnica chamada *Debounce* para realizar as buscas de forma mais eficiente e melhorar a usabilidade do usuário. Essa técnica consiste em atrasar o envio de informações, ou seja, a aplicação aguarda um certo tempo antes de realizar efetivamente a busca, reduzindo a quantidade de requisições à API.

Na imagem abaixo, pode-se ver o método *Debounce* sendo utilizado como um parâmetro do método "*listProducts*" que vimos anteriormente. Nesse caso, a aplicação irá buscar os registros de produtos filtrando pelas palavras-chave digitadas pelo usuário a cada 500ms enquanto ele digita. Isso permite que o usuário tenha uma experiência mais fluida e sem interrupções enquanto realiza a busca.

Figura 04 - Camada de visualização onde contém a chamada ao método de listagem de produtos.

```
const debounceSearch = useDebounce(searchTerm, 500);
const {
  data: products,
  error,
  isError,
  isLoading,
} = useQuery(['products', { debounceSearch }], () => ProductService.listProducts(debounceSearch));
```

FONTE: Autores, 2023

2.4 CAMADA DE APRESENTAÇÃO

O desenvolvimento da interface do sistema teve como ponto de partida as funcionalidades exigidas pelo projeto, que incluem Gerenciamento de Produtos, Gerenciamento de Clientes e Gerenciamento de Vendas. Com base nesses requisitos, foram criados *wireframes* de baixa fidelidade para validar a experiência do usuário (UX). A partir desses *wireframes*, foi desenvolvido o layout final das telas.

Para a programação da interface, foi escolhida a biblioteca *ReactJS*. Essa escolha se baseou em sua popularidade na comunidade, na possibilidade de criar uma SPA (*Single Page Application*) que melhora a performance da aplicação, uma vez que não necessita o recarregamento total da página a cada ação executada pelo usuário. Além disso, o *ReactJS* possui um ótimo desempenho e facilidade de manutenção, devido ao seu foco em *componentização*.

As telas criadas incluem um *Dashboard* que mostra o total de vendas no mês em quantidade e valor, páginas para o gerenciamento de clientes, produtos e vendas. Para visualizar as telas do sistema, é possível acessar o Apêndice E.

3 RESULTADO

Durante a entrevista com um funcionário de uma loja de roupas que não possuía nenhuma ferramenta de gerenciamento, foram abordadas várias perguntas chave para avaliar a eficácia do sistema proposto. As respostas fornecidas pelo funcionário, que serviu como referência para mensurar a melhoria proporcionada pelo *software*, são apresentadas a seguir:

Em relação à aplicação de forma geral, o funcionário observou uma diferença significativa no tempo médio para concluir uma venda com e sem o sistema. Ele destacou que a ferramenta possibilitou uma melhora considerável no tempo de execução das vendas. Além disso, o sistema ajudou a reduzir possíveis falhas, como erros na marcação de códigos de peças de roupas vendidas.

Quanto à satisfação com a interface da aplicação, o funcionário demonstrou estar muito satisfeito, mencionando que o sistema atendeu às necessidades de modernização e usabilidade que eles tinham anteriormente.

Em relação aos módulos específicos do projeto, foram abordados os seguintes tópicos:

Inventário e Gerenciamento de Estoque:

Ao ser questionado sobre o nível de estoque, o funcionário destacou que o sistema permitiu um controle mais preciso e acurado do estoque, pois eles passaram a ter mais informações sobre as peças disponíveis. Além disso, o gerenciamento de estoque teve um impacto significativo nos produtos mais e menos vendidos, pois agora podem visualizar claramente quais peças são mais populares e quais têm menor saída. Esses dados são fundamentais para as decisões de compra de futuras coleções.

Gerenciamento de Vendas:

O funcionário relatou que o processo de venda utilizando o sistema foi simples, intuitivo e rápido. Comparado ao sistema manual anteriormente utilizado, houve uma

melhora substancial no processo de venda. A ferramenta proporcionou uma experiência mais eficiente e ágil para o funcionário ao realizar vendas.

Atendimento Personalizado:

Em relação ao atendimento personalizado, o funcionário destacou que a ferramenta melhorou significativamente o atendimento ao cliente. Com acesso rápido às informações dos clientes, o funcionário pôde oferecer um atendimento mais focado em suas preferências e necessidades. Isso resultou em vendas mais assertivas, pois tinham todas as informações necessárias prontamente disponíveis. Os clientes também foram beneficiados, pois receberam indicações mais precisas de produtos que se encaixavam em seu estilo e preferências. Além disso, o atendimento foi mais rápido, transmitindo uma sensação de exclusividade ao cliente.

Essas respostas da entrevista destacam os benefícios trazidos pelo sistema de gerenciamento de loja de roupas, incluindo a redução no tempo de vendas, a diminuição de falhas, o melhor controle de estoque, a agilidade no processo de vendas e um atendimento mais personalizado e eficiente.

REFERÊNCIAS BIBLIOGRÁFICAS

AZUL, Conta. **Conta Azul**. Disponível em: <<https://experimente.contaazul.com>>. Acesso em: 10 out 2022

BLING.**Bling**. Disponível em: <<https://bling.com.br/>>. Acesso em: 10 out 2022

Contribuidores da Wikipédia. **Banco de dados**. Disponível em: <https://pt.wikipedia.org/wiki/Banco_de_dados>. Acesso em: 10 out 2022

DESCONHECIDO, Autor. **O que é um Banco de Dados?** Disponível em: <<https://www.oracle.com/br/database/what-is-database/>>. Acesso em: 10 out 2022

DESCONHECIDO, Autor. **SGBD: O QUE É, COMO FUNCIONA E PRINCIPAIS VANTAGENS** Disponível em: <<https://www.fiveacts.com.br/sghbd/>>. Acesso em: 10 out 2022

DESCONHECIDO, Autor. **Qual a diferença entre base de dados relacional e não relacional?** Disponível em: <<https://blog.debugeverything.com/pt/diferenca-base-de-dados-relacional-e-nao-relacional/>>. Acesso em: 10 out 2022

DESCONHECIDO, Autor. **Conceitos Fundamentais de Banco de Dados**. Disponível em: <<https://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>>. Acesso em: 10 out 2022

DARKCRIZT. **PostgREST, uma API REST para qualquer banco de dados PostgreSQL escrito em Haskell**. Disponível em: <<https://blog.desdelinux.net/pt/postgrest-uma-api-rest-para-qualquer-banco-de-dados-postgresql-escrito-em-haskell/>>. Acesso em: 10 out 2022

HIPER. **Hiper**. Disponível em: <<https://hiper.com.br/>>. Acesso em: 10 out 2022

HOUSE, Digital. **Banco de dados: entenda o que é um banco de dados relacional e não relacional**. Disponível em: <<https://www.digitalhouse.com/br/blog/banco-de-dados-relacional-e-nao-relacional/>>. Acesso em: 10 out 2022

KORTH, H.F. e SILBERSCHATZ, A.; **Sistemas de Bancos de Dados**, Makron Books, 2a. edição revisada, 1994.

MENDES, A.; **Custo de software: Planejamento e gestão**, Elsevier, 1. ed., Rio de Janeiro, 2014

POSTGREST. **PostgREST**. Disponível em: <<https://postgrest.org/en/stable/>>. Acesso em: 10 out 2022

RUMPE, B.; **Modeling with UML - Language, Concepts, Methods**, Springer, 2016

SOUZA, Ivan de. **Banco de dados: saiba o que é, os tipos e a importância para o site da sua empresa.** Disponível em: <<https://rockcontent.com/br/blog/banco-de-dados/>>. Acesso em: 10 out 2022

APÊNDICE A:

Tabela 1 - Acessar o sistema

Caso de uso: Acessar o sistema

Descrição: O usuário pode acessar o sistema informando o seu e-mail e senha. Após informações serem confirmadas, o acesso é liberado

FONTE: Autores, 2022

Tabela 2 - Cadastro de produto

Caso de uso: Cadastro de produto

Descrição: O usuário administrador irá fazer o cadastro de um novo produto utilizando as informações necessárias, após isso o produto estará disponível no sistema

FONTE: Autores, 2022

Tabela 3 – Gerenciar produtos

Caso de uso: Gerenciar Produtos

Descrição: O usuário pode gerenciar os produtos, visualizar se está disponível em estoque, quantidade, alterar valores

FONTE: Autores, 2022

Tabela 4 – Cadastro de clientes

Caso de uso: Cadastrar Clientes

Descrição: O usuário pode cadastrar os clientes, inserindo as informações pessoais do cliente, após isso, o cliente estará no banco de dados do sistema

FONTE: Autores, 2022

Tabela 5 – Gerenciar clientes

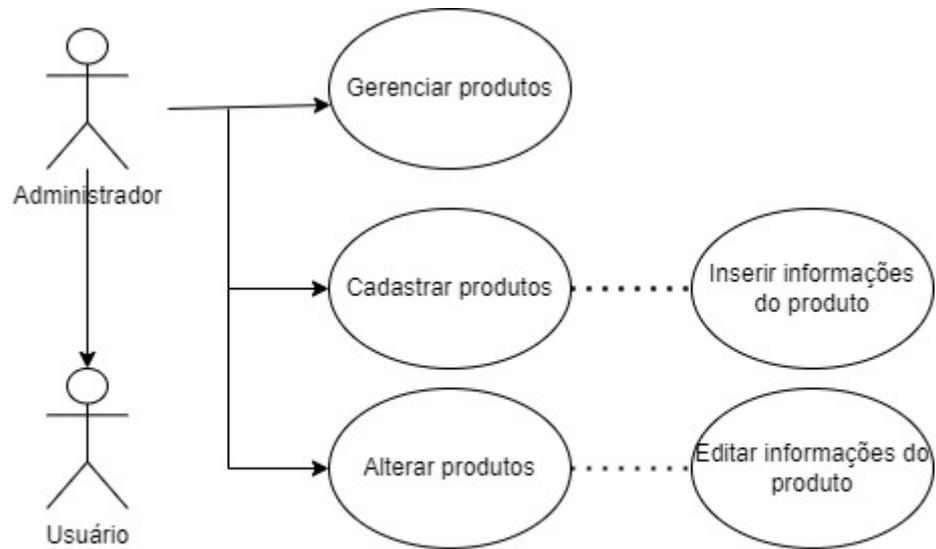
Caso de uso: Gerenciar Produtos

Descrição: O usuário pode gerenciar os clientes, consultar, alterar informações pessoais

FONTE: Autores, 2022

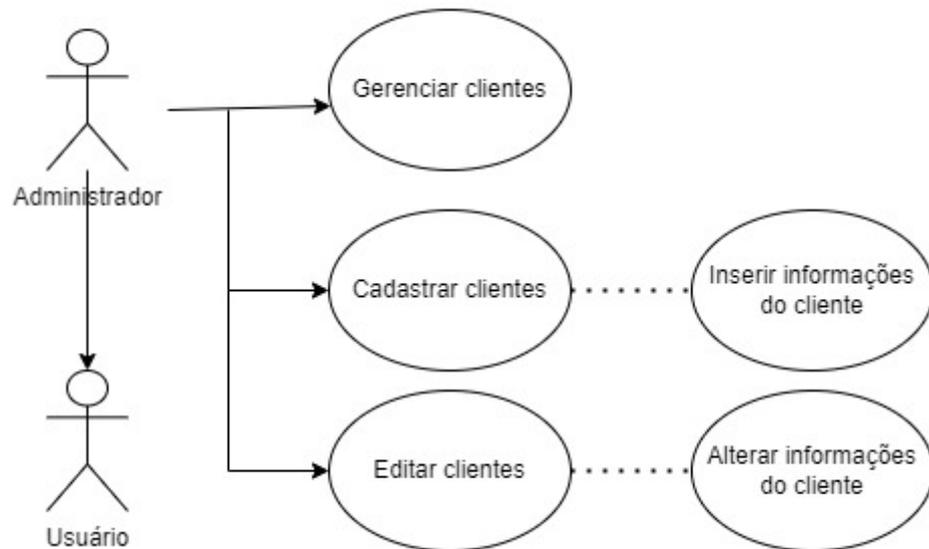
APÊNDICE B:

Ilustração 1 – Diagrama de Produtos



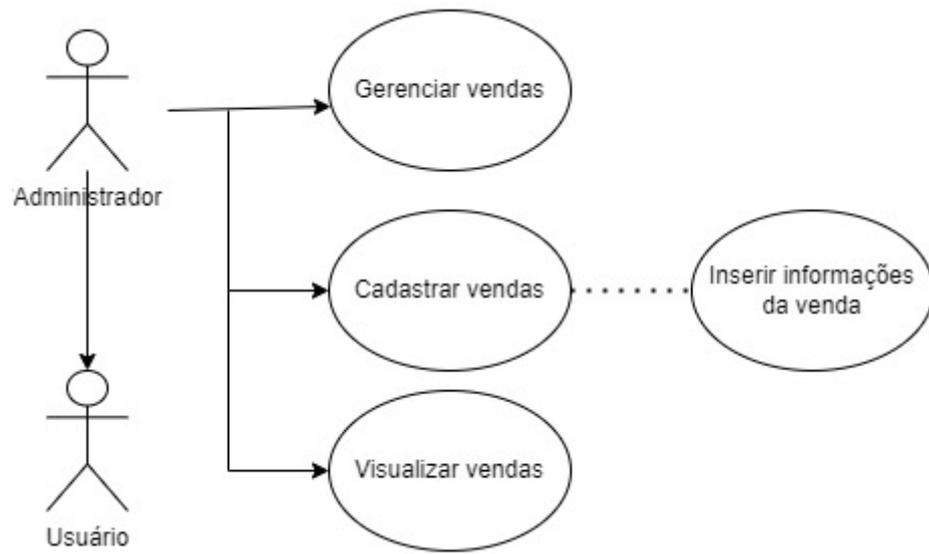
FONTE: Autores, 2022

Ilustração 2 – Diagrama de clientes

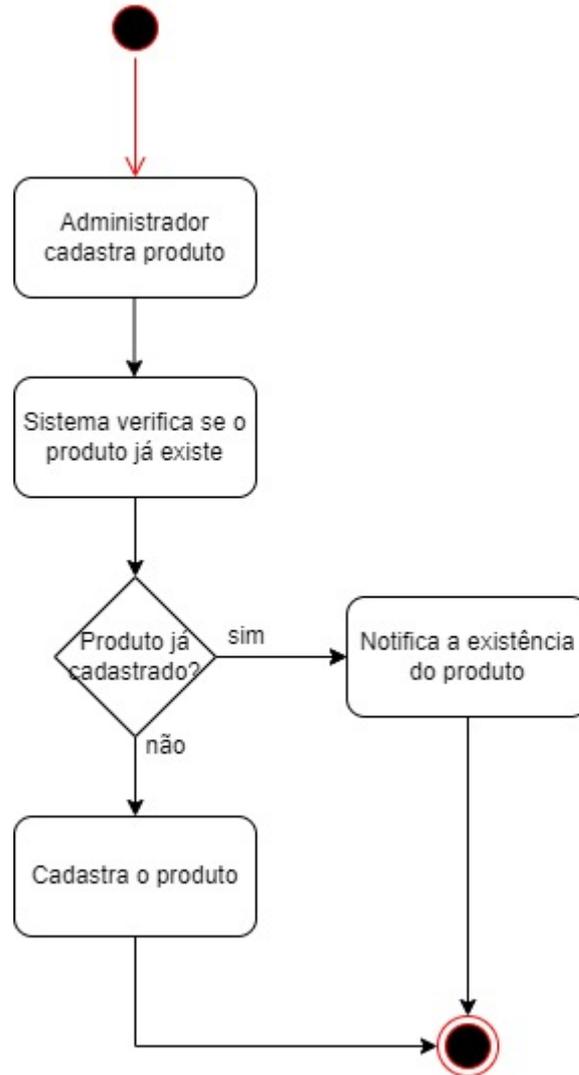


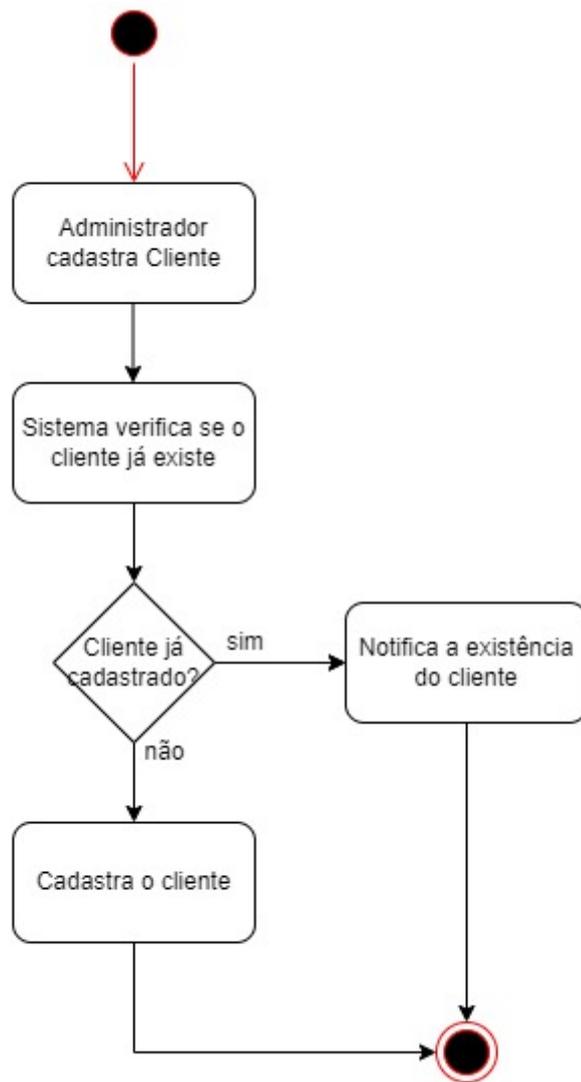
FONTE: Autores, 2022

Ilustração 3 – Diagrama de vendas



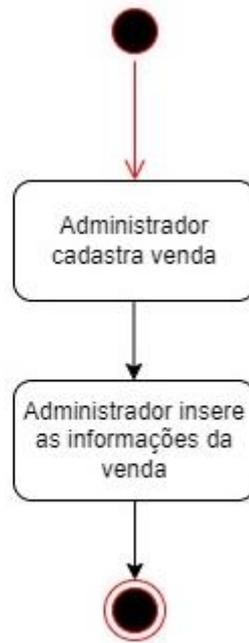
FONTE: Autores, 2022

APÊNDICE C:**Ilustração 4 – Fluxo de cadastro de clientes****FONTE:** Autores, 2022**Ilustração 5 – Fluxo de cadastro de produtos**



FONTE: Autores, 2022

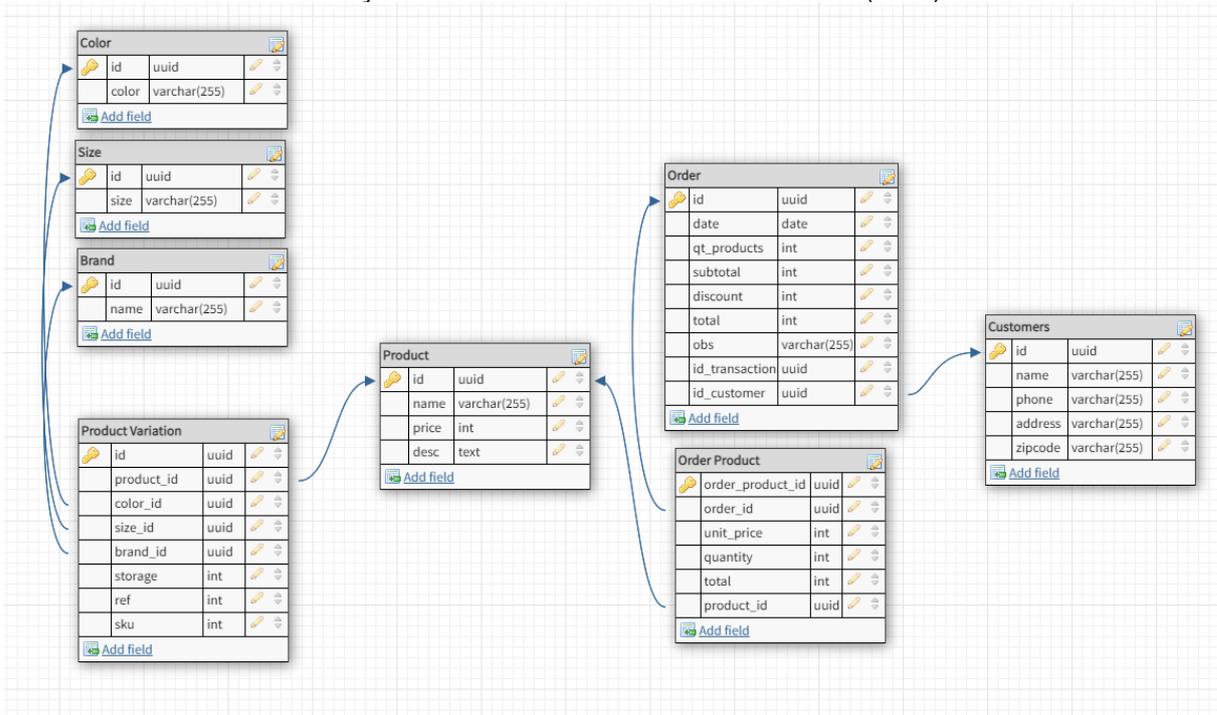
Ilustração 6 – Fluxo de cadastro de vendas



FONTE: Autores, 2022

APÊNDICE D:

Ilustração 7 – Modelo entidade Relacionamento (MER)



FONTE: Autores, 2022

APÊNDICE E:

Ilustração 8 – Tela cadastro de clientes

Novo cliente

Nome

E-mail

Telefone

Endereço

CEP

Data de nascimento
dd/mm/aaaa

CPF

Tamanho do sapato

Tamanho da parte de cima

Tamanho da parte de baixo

FONTE: Autores, 2022

Ilustração 9 – Tela consulta de clientes

Clientes + Novo cliente

Pesquisar
Pesquise pelo nome do cliente:

NOME	TELEFONE	ENDEREÇO	CEP
Adriana Paiva	(13) 98111-1132	Rua Voltemiro Silveira, n 20	11055-150
Adriane Soto	(13) 99718-0135	Rua Egidio Martins, n 85	11030-161
Alice Pierotti	(13) 98131-0214	Rua Machado de Assis, n 365	11050-060
Alessandra Rego	(13) 98145-0550	Rua Minas Gerais, n 04	11055-100
Alexandro Borlenghi	(11) 98266-1673	Rua Carolino Rodrigues, n 10, apto 41	11055-070
Amanda Varzea Boturão	(13) 99613-2449	Rua Dom Pedro I, n 59, apto 63	11075-550
Ana Claudia Hadid	(13) 99676-0607	Rua Dr. Armando Salles de Oliveira, n 15, casa 09	11050-071
Ana Cristina Curado Martins	(13) 97418-3592	Rua Voluntários Santistas, n 11, apto 31	11055-020
Ana Paula Deleuse	(13) 99735-1696	Av. Epitácio Pessoa, n 667, apto 271	11030-603
Andrea Lopes	(13) 98125-8000	Av. Bartolomeu de Gusmão, n 93, apto 141	11045-401
Andrea Monteiro Amador Durães	(13) 98708-9096	Rua Casimiro de Toledo, n 50, apto 102	11045-550

FONTE: Autores, 2022

Ilustração 10 – Tela cadastro de venda

Novo pedido ← Voltar

Cliente
Selecione uma opção

Data
dd/mm/aaaa

Forma de pagamento
Selecione uma opção

Desconto (%)
0

Observações

+ Adicionar produtos

Produtos selecionados

<input type="checkbox"/>	ID	NOME	PREÇO	COR	TAMANHO	MARCA	QUANTIDADE	SUBTOTAL
Finalizar pedido								

FONTE: Autores, 2022

Ilustração 11 – Tela consulta de vendas

Vendas + Nova venda

Pesquisar
Pesquise pelo nome do cliente:

DATA	CLIENTE	QT. PRODUTOS	SUBTOTAL	DESCONTO	TOTAL	OBS	PAGAMENTO
05/07/2022	Victoria Flores Morelli	10	RS 100,00	10	RS 90,00	-	Cartão de crédito

FONTE: Autores, 2022

Ilustração 12 – Tela cadastro de produto

Novo produto

Nome

Preço

Descrição

Marca
Selecione uma marca

[Criar produto](#)

FONTE: Autores, 2022

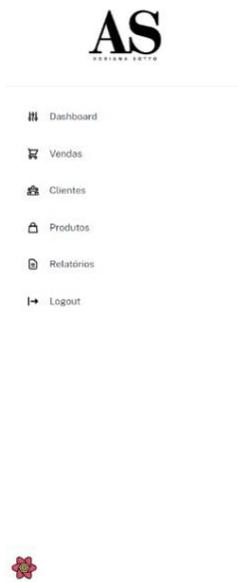
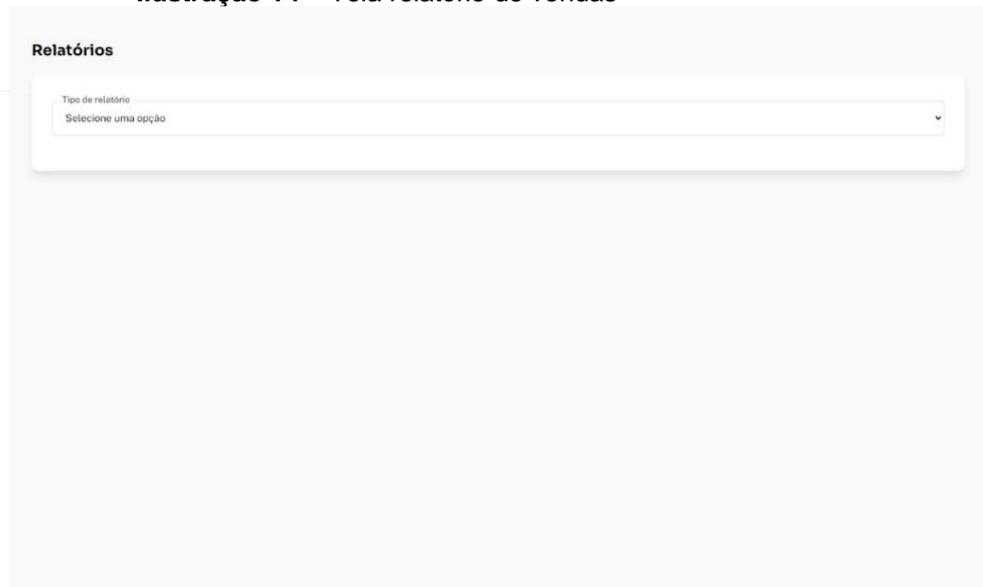
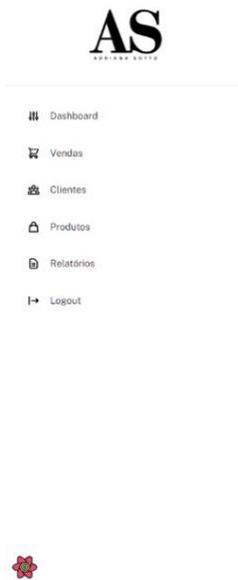
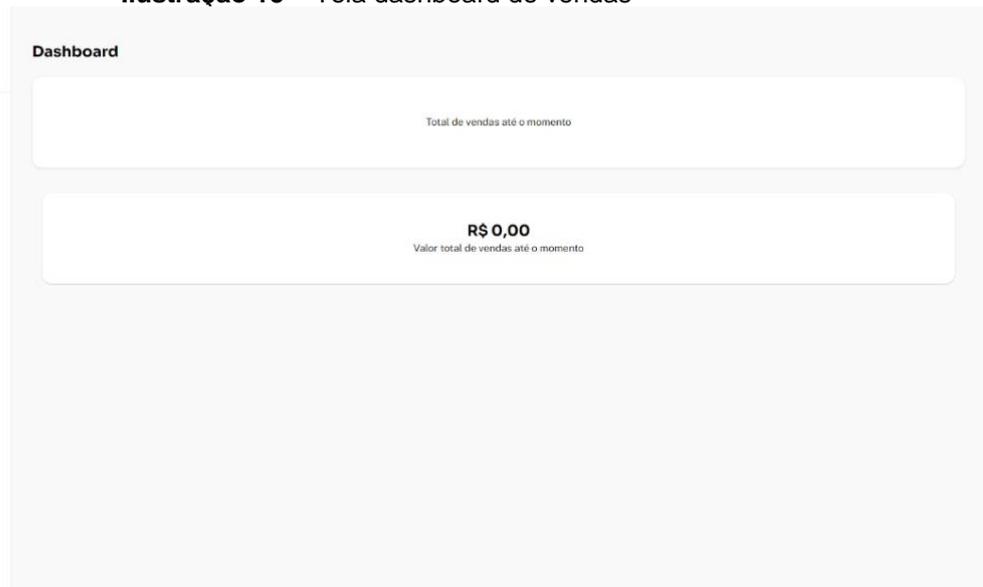
Ilustração 13 – Tela consulta de produtos

Produtos [Novo produto](#)

Pesquisar
Pesquise pelo nome do produto:

NOME	PREÇO	DESCRIÇÃO	MARCA
AAAAAAAAA	123123123	12312	Cris Barros
Blazer Crepe Ventana Maye NKO	955		NK
Blazer Jacquard Concreto Colby NKO	1315	Blazer cropped	NK
Blusa Babados	429	blusa babados-manga longa	L' Cecci
Blusa Cotim Clarice NKO	795		NK
Blusa Crepe Liquido Yuri Nko	635		NK
Blusa Estampada	499	Blusa estampada manga bufante	L' Cecci
Blusa Jacquard	719	Blusa estampada-mangas bufantes	L' Cecci
Blusa Jacquard	619	Blusa jacquard -costas abertas	L' Cecci
Blusa L' Cecci -costas abertas	619	Blusa com abertura nas costas	L' Cecci

FONTE: Autores, 2022

**Ilustração 14 – Tela relatório de vendas****FONTE:** Autores, 2022**Ilustração 15 – Tela dashboard de vendas****FONTE:** Autores, 2022