



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

JOÃO OTAVIO GUIOTTI DA SILVA

**ANÁLISE E DESENVOLVIMENTO DE SOFTWARE PARA
GERENCIAMENTO DE LOJAS DE VAREJO**

**Assis/SP
2021**



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

JOÃO OTAVIO GUIOTTI DA SILVA

**ANÁLISE E DESENVOLVIMENTO DE SOFTWARE PARA
GERENCIAMENTO DE LOJAS DE VAREJO**

Projeto de Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

**Orientando(a): João Otavio Guiotti da Silva
Orientador(a): Prof. Dr. Luiz Ricardo Begosso**

**Assis/SP
2021**

FICHA CATALOGRÁFICA

S586a SILVA, João Otavio Guiotti da
ANÁLISE E DESENVOLVIMENTO DE SOFTWARE PARA GERENCIAMENTO DE LOJAS DE VAREJO / João Otavio Guiotti da Silva. Fundação Educacional do Município de Assis –FEMA – Assis, 2021.
55p.

Trabalho de conclusão do curso (Ciências da Computação). – Fundação Educacional do Município de Assis

Orientador: Dr. Luiz Ricardo Begosso

1. Software-app. 2. Gestão. 3. Varejo

CDD: CDD005.12
Biblioteca da FEMA

ANÁLISE E DESENVOLVIMENTO DE SOFTWARE PARA GERENCIAMENTO DE LOJAS DE VAREJO

JOÃO OTAVIO GUIOTTI DA SILVA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____
Prof. Dr. Luiz Ricardo Begosso

Examinador: _____
Prof. Célio Desiró

RESUMO

Para controlar os dados gerados diariamente, as empresas de varejo aderem a ferramentas ou softwares que auxiliam no controle das informações, mantendo a organização com um melhor controle sobre o fluxo de vendas e disponibilidade de mercadorias. Para elaborar um software de gestão é necessário analisar a fundo os processos relacionados ao ambiente e as regras de negócios impostos pelo varejo e para realizar esta análise é necessário gerar uma documentação seguindo os conceitos de engenharia de software.

Outro fator importante é determinar qual tecnologia deve ser utilizada para o desenvolvimento do software. Neste trabalho foram utilizadas tecnologias web (Web Applications), como o Angular para o Frontend (cliente-side), Lumen para Backend API (server-side), Json Web Token e o conceito HTTP que é base do universo web.

Diante das circunstâncias, o presente trabalho resultou no desenvolvimento de um software, e possibilitou dessa forma aplicar técnicas, linguagens de programação, Frameworks para agilizar o desenvolvimento e fornecer recursos para aumentar a interação do usuário.

Palavras-chaves: Gestão, Varejo, Web Applications, Angular, Frontend, client-side, Lumen, Backend, API, server-side, Json Web Token, HTTP.

ABSTRACT

To control data on a daily basis, retail companies adhere to tools or software that help control information, keeping the organization with better control over the sales flow and availability of goods. In order to create a management software, it is necessary to analyze the processes related to the environment and the business rules imposed by the retail fund, and to carry out this analysis, it is necessary to generate documentation following the concepts of software engineering.

Another important factor is to determine which technology should be used for the development of the software. In this work, web technologies (Web Applications) were used, such as Angular for Frontend (client-side), Lumen for Backend API (server-side), Json Web Token and the HTTP concept that is based on the web universe.

This work resulted in the development of a software, with the possibility to apply techniques, programming languages, Frameworks to streamline the software development and provide resources to increase user interaction.

Keywords: Management, Retail, Web Applications, Angular, Frontend, client-side, Lumen, Backend, API, server-side, Json Web Token, HTTP.

LISTA DE ILUSTRAÇÕES

Figura 1 - Ilustração representativa de aplicações web.	16
Figura 2 - Representação área visual interpretada pelo navegador.	17
Figura 3 - Representação Frontend para Backend fluxo.	18
Figura 4 - Representação de diferentes plataformas se integrando com uma API.	19
Figura 5 - Printscreen debugger Json Web Token.	20
Figura 6 - Estrutura e funcionamento de uma aplicação angular.	23
Figura 7 – Mapa mental.	32
Figura 8 – Diagrama de Caso de Uso.	33
Figura 9 – Modelagem de Banco de Dados.	34
Figura 10 – Diagrama de Classe.	35
Figura 11 – Realizar Login.	36
Figura 12 – Meu Primeiro Acesso.	37
Figura 13 – Página Inicial.	38
Figura 14 – Alterar Usuário.	38
Figura 15 – Dashboard.	39
Figura 16 – Formulário de cliente.	40
Figura 17 – Listagem de clientes.	40
Figura 18 – Formulário de funcionários.	41
Figura 19 – Listagem de funcionários.	42
Figura 20 – Formulário de caixas.	42
Figura 21 – Listagem de caixas.	43
Figura 22 – Formulário de produtos.	43
Figura 23 – Listagem de produtos.	44
Figura 24 – Tela de categoria.	44
Figura 25 – Tela de coleção.	45
Figura 26 – Tela de marca.	46
Figura 27 – Formulário de Venda.	46
Figura 28 – Listagem de Vendas.	47
Figura 29 – Formulário de condicionais.	48
Figura 30 – Listagem de condicionais.	48
Figura 31 – Formulário de condição de pagamento.	49
Figura 32 – Caixa de diálogo.	50

Figura 33 - Gerir Perfis.50

Figura 34 - Realizar Logout.51

LISTA DE TABELAS

Tabela 1 - Manter Usuário.	26
Tabela 2 - Manter Cliente.	26
Tabela 3 - Manter funcionário.	27
Tabela 4 - Manter autorizado.	27
Tabela 5 - Manter caixas de roupas.	28
Tabela 6 - Manter categorias.	28
Tabela 7 - Manter Coleções.	28
Tabela 8 - Manter Marcas.	29
Tabela 9 – Manter Peças.	29
Tabela 10 – Manter Condições de pagamento.	29
Tabela 11 – Manter Condicionais.	30
Tabela 12 – Manter Vendas.	30
Tabela 13 – Painel de Gráficos (Dashboard).	31
Tabela 14 – Multiplataforma.	31

SUMÁRIO

1. INTRODUÇÃO	12
1.1. OBJETIVOS	13
1.2. JUSTIFICATIVAS.....	13
1.3. MOTIVAÇÃO.....	14
1.4. PERSPECTIVAS DE CONTRIBUIÇÃO	14
1.5. METODOLOGIA.....	15
2. REVISÃO BIBLIOGRÁFICA E CONCEITOS.....	16
2.1. WEB APPLICATION	16
2.2. FRONTEND	17
2.3. BACKEND	18
2.4. API.....	19
2.5. JSON WEB TOKEN	19
3. FERRAMENTAS DE DESENVOLVIMENTO.....	21
3.1. VISUAL STUDIO CODE.....	21
3.2. POSTMAN.....	21
3.3. PHPMYADMIN	22
3.4. FRAMEWORK'S	22
3.5. ANGULAR	22
3.6. LUMEN.....	23
4. PROPOSTA DO TRABALHO	24
5. MODELAGEM E DESENVOLVIMENTO	25
5.1. ANÁLISE DE REQUISITOS.....	25
5.2. MAPA MENTAL.....	32
5.3. DIAGRAMA DE CASOS DE USO.....	33
5.4. BANCO DE DADOS	34
5.5. DIAGRAMA DE CLASSE	35
5.6. DESENVOLVIMENTO	36
5.6.1. Tela de Login	36
5.6.2. Tela de Primeiro Acesso.....	37
5.6.3. Tela Inicial.....	38

5.6.4.	Formulário Dados Pessoais	38
5.6.5.	Tela de Dashboard	39
5.6.6.	Formulário de Cliente	40
5.6.7.	Tela de Cliente.....	40
5.6.8.	Formulário de Funcionário.....	41
5.6.9.	Tela de Funcionário	42
5.6.10.	Tela de Funcionário	42
5.6.11.	Formulário de Produto	43
5.6.12.	Tela de Produto.....	44
5.6.13.	Modal de Categoria.....	44
5.6.14.	Modal de Categoria.....	45
5.6.15.	Modal de Marca	46
5.6.16.	Modal de Venda.....	46
5.6.17.	Tela de Vendas.....	47
5.6.18.	Formulário de Condicional	48
5.6.19.	Tela Condicional	48
5.6.20.	Formulário de Condição de Venda.....	49
5.6.21.	Caixa de Diálogo	50
5.6.22.	Tela de Gerir Perfis	50
5.6.23.	Dialogo Realizar Login	51
6.	CONCLUSÃO	52
	REFERÊNCIAS.....	54

1. INTRODUÇÃO

As estratégias do varejo vêm sofrendo grandes transformações ao decorrer dos últimos anos, levando em conta que isto seja apenas uma evolução natural na tentativa de se adaptarem às mudanças deste mercado amplo, onde os concorrentes são bastante competitivos e exigentes. Diante destes fatos vale ressaltar, como Kotler, Kartajaya e Setiawan (2010) que grande parte das empresas de todo o planeta estão passando ou vão passar por constantes mudanças e inovações em seus modelos de negócios.

É importante considerar que o auxílio de um software para gerenciamento do varejo pode trazer inúmeros benefícios e ganhos significativos, sem contar que o fluxo de venda pode se tornar mais ágil e eficiente, assim ocasionando a diminuição de erros, o retrabalho e a redução de custos.

A demanda de softwares para varejo vem aumentando e em meio a tantas disputas, destaca-se que o uso de um bom software de varejo é indispensável para os dias atuais onde percebe-se que o avanço das tecnologias trouxe também um crescimento do mercado de varejo, aumentando-se assim o consumo e consequentemente a empresa de varejo.

O objetivo do contexto abordado neste trabalho, será realizar uma análise e o desenvolvimento de um software para controlar a gestão de uma empresa de varejo de roupas, também enfatizando as tecnologias que serão utilizadas para este aplicativo web, seguindo com o detalhamento que será feito nos próximos tópicos.

1.1. OBJETIVOS

O presente trabalho de conclusão de curso tem como objetivo mostrar etapas para o desenvolvimento de um software para empresas de varejo, desde os levantamentos de requisitos até a instalação da aplicação em um servidor, abordando em seu desenvolvimento algumas técnicas dos frameworks e tecnologias que serão utilizadas e demonstrar seu uso na prática.

Para atingir o objetivo geral deste trabalho destaca-se algumas tecnologias relevantes para o mercado de trabalho, como Angular, Lumen (Micro framework) e conceitos relacionados ao desenvolvimento WEB em geral. O propósito deste software será de garantir o gerenciamento de estoque, controle de fichas, clientes, funcionários, vendas, condicionais e demais necessidades do varejo. Ao final deste trabalho está previsto a apresentação do software finalizado.

1.2. JUSTIFICATIVAS

O tema proposto para este trabalho é de extrema importância para o ramo de empreendedores de varejo de todos os tamanhos, desde os grandes aos pequenos, pois realizar uma boa gerência de seu espaço e vendas sempre apresenta melhores resultados ao decorrer de um período de tempo considerável e conseqüentemente possibilidade de melhorar o entendimento de seus parceiros e clientes.

Além de mostrar o quão eficiente um software como este pode ser, será relatado todo o processo de levantamento de requisitos e desenvolvimento, assim passando um pouco mais de conhecimento sobre tecnologias e ferramentas que agilizam estas etapas de produção da ferramenta de gestão.

Este trabalho justifica-se pela necessidade de as organizações terem um gerenciamento melhor do grande volume de informações geradas no dia-a-dia, é essencial e de extrema importância para uma loja de varejo possuir um software como esse, segundo REZENDE e ABREU (2000, p. 76) um dos principais fatores para precursores do sucesso de um empreendimento é “Gestão da Informação” e isso eleva a potencialidade de atuação das empresas, assim agregando valores de mercado e capacitando a gerência de informações de forma mais eficiente.

1.3. MOTIVAÇÃO

O presente trabalho tem como motivação mostrar o quão um software bem elaborado de gestão de varejo pode ser eficiente no ambiente em que se encontra, além da gestão que a ferramenta proporciona depois de pronta, também levará como motivação algumas técnicas que agilizam e aprimoram o processo de desenvolvimento e será de extrema importância trazer um pouco mais de conhecimento sobre algumas frameworks utilizados.

Um fato a ser ressaltado é que a aplicação será como um web site, porem voltado para o gerenciamento e com algumas funcionalidades extras, isto possibilita o acesso por vários dispositivos que possuem um navegador instalado, de tal forma que tenha uma visão do seu varejo até se não estiver no ambiente da empresa.

E todos os esses aspectos apresentados acima me levaram a escrever este trabalho de conclusão de curso.

1.4. PERSPECTIVAS DE CONTRIBUIÇÃO

O presente trabalho pretende agilizar os processos presentes em um varejo e acima de tudo gerenciar essas informações de forma segura e com uma flexibilidade de acesso para vários tipos de dispositivos, desde um desktop a um mobile.

Com extrema disponibilidade de acesso, perceber mudanças no mercado de varejo se torna algo que realmente pode ser vantajoso para uma empresa, pois um bom gerenciamento é pré-requisito para o sucesso de um negócio.

Espera-se que o presente trabalho possa contribuir para as empresas varejistas e alavancar resultados estabelecidos pelo uso desse software e contribuir para os futuros programadores levar em consideração não somente a mão na massa, mas também todo o processo de levantamento de requisitos e análise de tecnologias que a indústria de frameworks possa oferecer.

1.5. METODOLOGIA

A metodologia empregada para o presente trabalho será realizada utilizando frameworks derivadas de Typescript e PHP, conseqüentemente bem populares no ramo de desenvolvimento WEB, Angular e Lumen. A ideia central é relatar todas as etapas para elaboração de um software como este, onde seguiremos uma receita de desenvolvimento abordada na construção de softwares profissionais. Com todas as etapas realizadas e o app pronto, será realizado um estudo de eficiência no gerenciamento da ferramenta no ambiente de varejo.

O presente trabalho seguirá a seguinte estrutura:

I. Revisão Bibliográfica e conceitos.

Nesta etapa será realizado um estudo sobre as tecnologias que serão utilizadas para a elaboração da aplicação, tecnologias ágeis de produção de software.

II. Ferramentas de desenvolvimento.

Será realizado um direcionamento as aos conceitos e tecnologias, assim como aprofundar os conhecimentos nas ferramentas que serão utilizadas.

III. Proposta de trabalho.

Como proposta o desenvolvimento de um software será realizado utilizando tecnologias e metodologias ágeis.

IV. Modelagem e desenvolvimento.

Após os passos anteriores será dado início ao processo de modelagem e desenvolvimento da aplicação, com tudo iniciando a análise de requisitos.

2. REVISÃO BIBLIOGRÁFICA E CONCEITOS

Este capítulo tem objetivo de apresentar os principais conceitos e ferramentas para o desenvolvimento do software para gerenciamento de empresas de varejo, assim como dar uma introdução aos pilares do desenvolvimento web, utilizando framework's, e ferramenta eficientes de desenvolvimento, e ferramentas que ajudam a testar o funcionamento de aplicações.

2.1. WEB APPLICATION

A princípio uma aplicação web é um software que é instalado em um servidor web e projetado para realizar requisições, essas utilizam um padrão específico o HTTP (Hyper Transfer Protocol) que foi desenvolvido justamente para a comunicação entre navegadores e servidores, ou mais especificamente cliente-servidor. Segundo Loundon (2018) enquanto muitos dos softwares são distribuídos como uma unidade completa, as aplicações web tem muitas de suas partes entregues página por página e operam em um ambiente compartilhado por um grande volume de usuários.

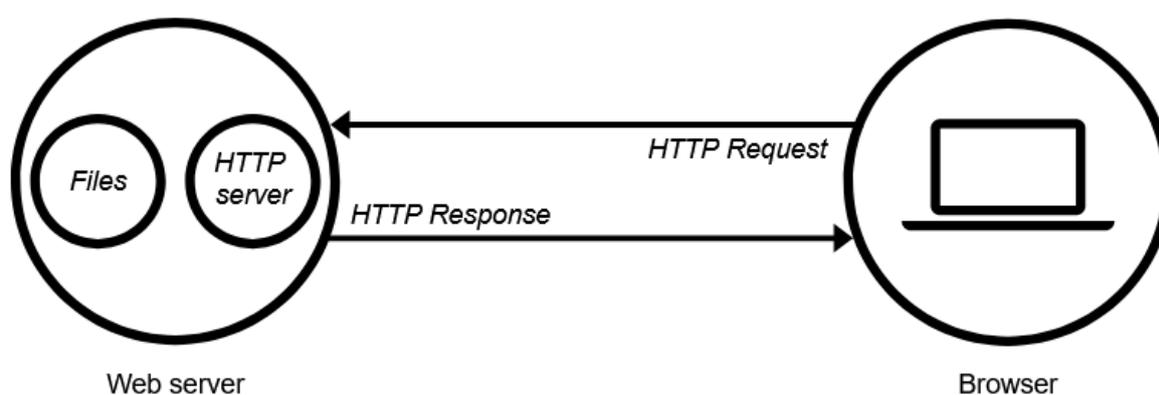


Figura 1 - Ilustração representativa de aplicações web.

Fonte: MDN Web Docs (2020).

Como ilustra a Figura 1, o cliente-sede realiza uma solicitação via http ao servidor, onde neste modelo de aplicação está presente o servidor web, como exemplo o ISS (Internet Information Services), Apache Server e entre outros, a sua função dentro da web é

basicamente receber essas requisições ou arquivos que irão alcançar o servidor e enviar o dado requerido também via HTTP, ou seja, esse retorno é nomeado como HTTP Response.

2.2. FRONTEND

Pode-se classificar o Front-end como uma interface gráfica, ou área de visualização de uma aplicação que emite interações com os usuários, quando ao se referirmos a esse termo são usadas algumas tecnologias base da web como HTML, CSS e Javascript.

Quando se trata de linguagens web, deve-se levar em conta o termo linguagens de client-side, que são responsáveis por fazerem com que o seu código seja interpretado pelo navegador, como ilustra a Figura 2.

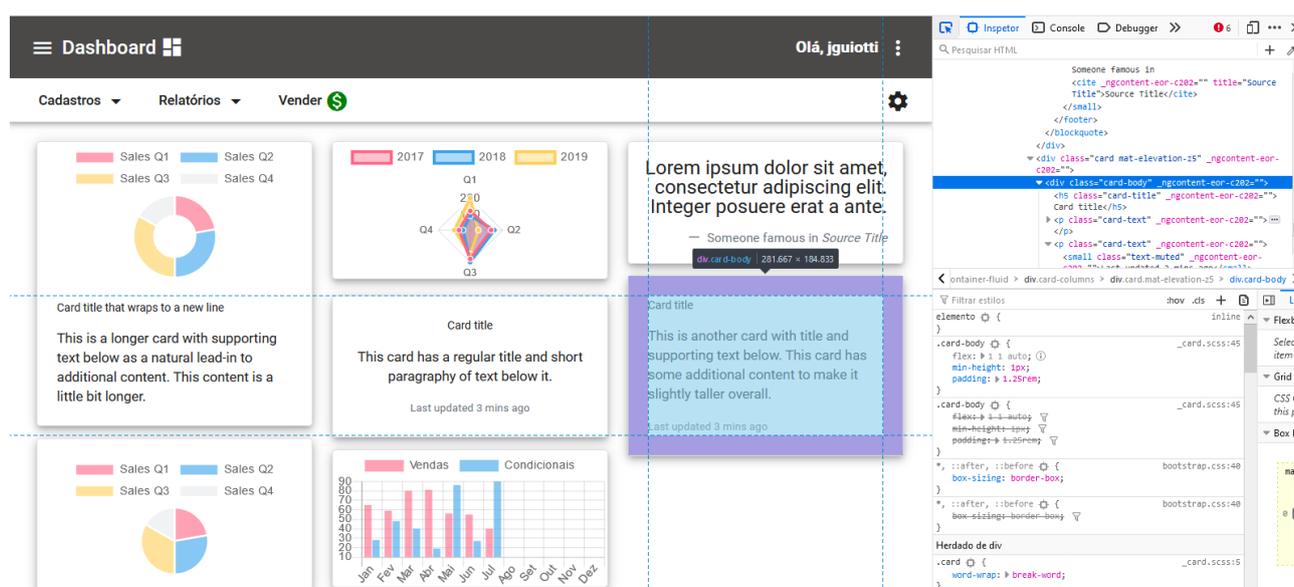


Figura 2 - Representação área visual interpretada pelo navegador.

Fonte: Autoria própria.

Segundo Kaluža (2018) para criar uma boa experiência do usuário é importante escolher uma arquitetura de aplicativo web adequada onde deve-se levar em consideração as necessidades do usuário e os modelos de páginas web, quais são SPA (Single Page Application) e MPA (Multi Page Application).

2.3. BACKEND

O termo Back-end refere-se aos processos realizados no servidor, onde não são visualizados pelos usuários, tais processos podem realizar consultas a banco de dados e trabalhar em cima da informação a ser entregue para o cliente-side, onde as informações já tratadas serão renderizadas na área de visualização (Frontend).

Sendo assim o papel de um bom backend é essencial, pois além de gerir os dados ele também se torna responsável por realizar a segurança da aplicação, atualizações de dados, gerenciamento de conteúdo, estruturação de regras de negócios, etc. Para melhor representar o funcionamento do backend é necessário representá-lo junto ao Frontend como ilustra a figura 3.

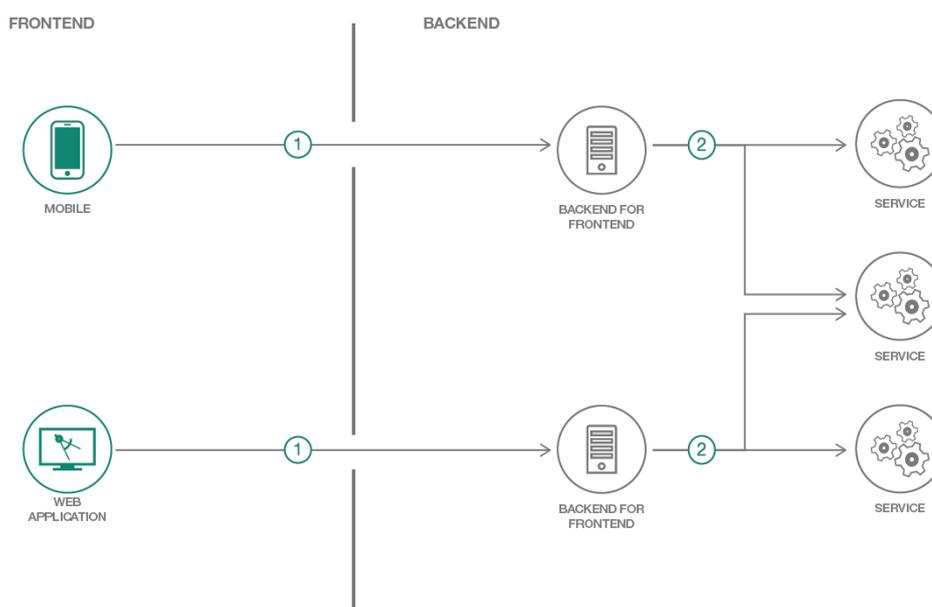


Figura 3 - Representação Frontend para Backend fluxo.

Fonte: Tecmundo (2020).

De forma enxuta resumir o backend segundo STOCKLMAIR et al (2021), o backend ou server-side pode ser considerado como a parte onde o usuário não tem permissão para interagir diretamente, pois lá possui dados sensíveis e confidenciais que são armazenadas em uma base dados e se um usuário tiver comunicação direta pode gerar falhas de segurança a sua aplicação.

2.4. API

Pode-se que uma API (Application Programming Interface) é um conjunto de padrões de programação que possibilitam a integração de outros aplicativos, a utilização de seus recursos vem da necessidade de desenvolvimento, e deve-se levar em consideração que uma API é um conceito relativamente amplo. Um fato claro, analisando este universo de aplicações web, é que na maioria das vezes as API's se comunicam com diversos códigos e funções em um aplicativo. A Figura 4 ilustra a diversidade de plataformas que podem se conectam com uma API.

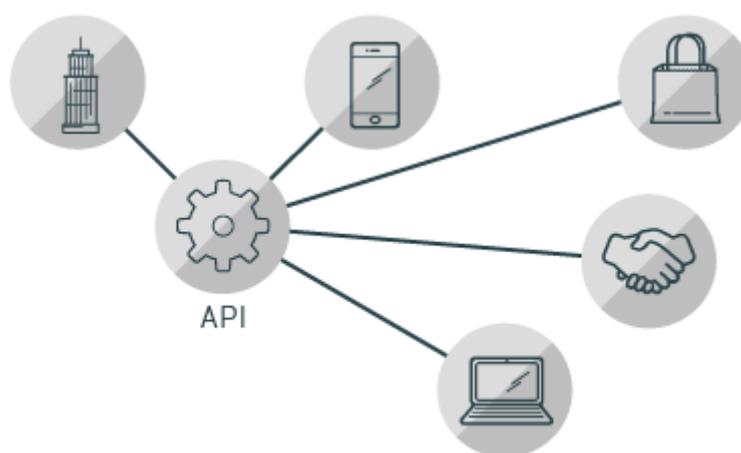


Figura 4 - Representação de diferentes plataformas se integrando com uma API.

Fonte: Banco de imagens Fusio (2021).

Uma especificação de uma API, segundo Red Hat (2021), permite que a solução ou serviço gerado pela API comunique-se com outros produtos e serviços, simplifica o desenvolvimento economizando tempo e dinheiro, pois oferecem a flexibilidade necessária para simplificar o design, a administração e o uso.

2.5. JSON WEB TOKEN

O JWT (Json Web Token) é um padrão (RFC-7519) compacto e muito seguro onde duas aplicações precisam trocar informações, as reivindicações de um JWT são codificadas como um objeto JSON assim permitindo que as declarações sejam digitalmente assinadas e protegidas com um código de autenticação de mensagens (MAC).

Segundo JWT (2021) essas informações estão seguras e confiáveis por que estão assinadas digitalmente, usando um segredo com o algoritmo HMAC ou um par de chaves públicas/privadas usando RSA ou ECDSA.

O JWT pode ser dividido em três partes:

- Header: esta parte é subdividida em duas partes, onde é especificado o tipo e o algoritmo a ser usado.
- Payload: esta parte é onde carrega as informações da entidade, normalmente são carregados dados que identificam um usuário no sistema.
- Signature: Nesta parte pega-se as partes já informadas como, o Header e o Payload já codificados, um segredo e o algoritmo especificado no Header e assiná-lo.

Como é possível observar analisando a Figura 5, o JWT é montado de acordo com as cores do token e a separação por pontos, assim é possível perceber que ele é dividido em Header + . + Payload + . + Signature:

The image shows a JWT debugger interface with two main panels: 'Encoded' and 'Decoded'.

Encoded Panel: Titled 'Encoded' with the subtitle 'PASTE A TOKEN HERE'. It contains a text box with a red border containing the following JWT token: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MzY5MDYyLmF1dG8iLCJpcyI6IjE2MzY5MDYyLmF1dG8iLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpcyI6IjE2MzY5MDYyLmF1dG8iLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpcyI6IjE2MzY5MDYyLmF1dG8iLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9`. A red arrow points from the token to the text 'JWT Montado' below it.

Decoded Panel: Titled 'Decoded' with the subtitle 'EDIT THE PAYLOAD AND SECRET'. It displays the decoded structure of the token in three sections:

- HEADER: ALGORITHM & TOKEN TYPE:**

```
{  "alg": "HS256",  "typ": "JWT"}
```
- PAYLOAD: DATA:**

```
{  "sub": "1234567890",  "name": "John Doe",  "iat": 1516239022}
```
- VERIFY SIGNATURE:** Shows the HMACSHA256 function being used to verify the signature. The input fields are: `base64UrlEncode(header) + "." + base64UrlEncode(payload),` and `your-256-bit-secret`. There is a checkbox for `secret base64 encoded` which is currently unchecked.

Figura 5 - Printscren debugger Json Web Token.

Fonte: JWT (2020).

3. FERRAMENTAS DE DESENVOLVIMENTO

Nesta seção serão descritas as ferramentas principais que serão usadas no processo de criação do software de gerenciamento de empresas de varejo.

3.1. VISUAL STUDIO CODE

Em 2015 foi lançado pelo Microsoft, um editor de código voltado para o desenvolvimento de aplicações web, esta ferramenta caracteriza-se por ser leve e bem funcional comparada a outras existentes. O Visual Studio Code além de possuir extensões para todos tipos de linguagens, é totalmente gratuito e possui uma estrutura open source.

Esta ferramenta segundo o site VISUAL STUDIO CODE (2021), ao instalá-la vem com suporte integrado para algumas linguagens, JavaScript, TypeScript e Node.js, além de possuir um ecossistema bem amplo para outras linguagens, possui disponibilidade para alguns sistemas de forma nativa como Windows, macOS e Linux.

3.2. POSTMAN

O Postman é um aplicativo que foi projetado pra testar requisições e realizar documentações de API's, apresentando um design limpo e bem simples de ser utilizado. Com o auxílio desta ferramenta é possível simular requisições HTTP a rotas de um serviço que responda a protocolos webs.

Neste cenário os serviços de HTTP não possuem cenário gráficos, atuando apenas no recebimento e provimento dos dados através das requisições, seguindo esta linha de raciocínio, Rodrigues (2021) diz que testar as aplicações envolvem testar requisições a partir de cliente conhecido, assim prevenindo e mantendo o controle sobre os dados trafegados.

3.3. PHPMYADMIN

O phpMyAdmin é uma ferramenta de gerenciamento, escrita em PHP e de código livre, se caracteriza por ser um aplicativo web voltado para o gerenciamento de bancos de dados, como MySQL e MariaDB.

Essa ferramenta se torna essencial para o processo de desenvolvimento onde é possível encontrá-la em pacotes off-line, como WAMPServer, XAMPP, EasyPhp e PHP Traid.

3.4. FRAMEWORK'S

Neste tópico serão explanados alguns conceitos de frameworks utilizadas neste desenvolvimento. Um framework resumidamente é conjunto de códigos básico que quando aplicados a uma ferramenta é possível agilizar o processo de desenvolvimento, assim como de evitar o árduo processo de digitação de uma classe do zero, fazendo uma analogia, é como se fossem peças prontas que podem ser utilizadas na criação de um carro.

3.5. ANGULAR

O angular é um framework desenvolvido pela Google, segundo AFONSO (2018) este framework é uma plataforma voltada para a construção de interface de aplicações web, onde utiliza HTML, CSS e Typescript em sua composição. O Angular leva como base o conceito de SPA (Single Page Applications ou Aplicações de página únicas) sendo possível estender seus recursos para dispositivos mobile ou desktops.

Um grande ponto a se ressaltar sobre esse framework é que a Google adotou o Typescript como sua linguagem principal e isso se torna um grande diferencial, pois essa linguagem é um comparativo com o Javascript, porém melhorada, que traz novidades mais recentes da linguagem, sendo assim chamada de supeset.

Os elementos principais que compõem a estrutura de uma aplicação Angular são:

- Módulos
- Componentes
- Templates
- Metadata
- Data Binding
- Diretivas
- Serviços

- Injeção de dependências

Com a análise da Figura 6 é possível ter uma ideia de como a estrutura do Angular opera, assim como cada elemento interage com os outros.

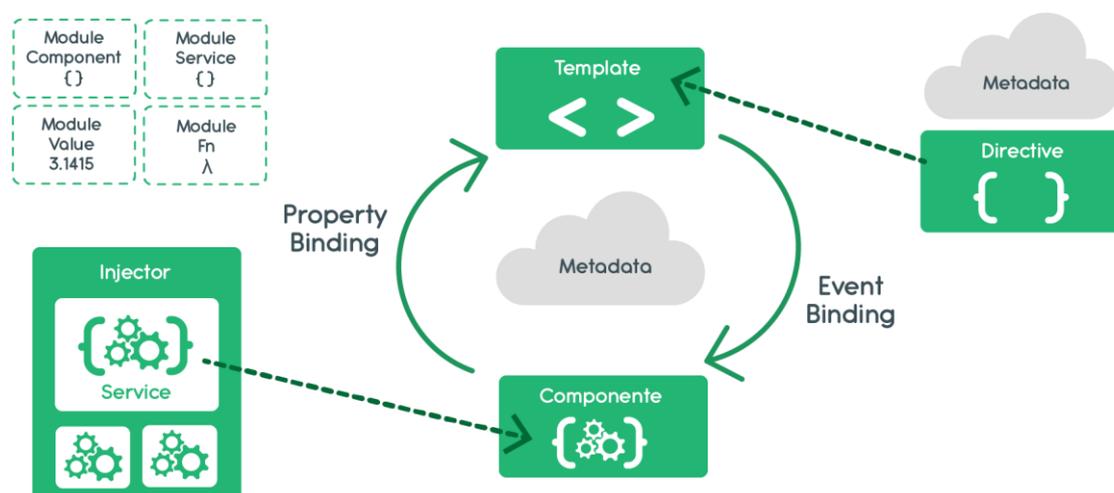


Figura 6 - Estrutura e funcionamento de uma aplicação angular.

Fonte: banco de imagens algaworks.

3.6. LUMEN

O Lumen é um microframework desenvolvido em PHP, construído para desempenhar ótima performance e voltado para o Microserviço e API's, onde aplicações front-end não são o foco, derivado e desenvolvido com recursos mínimos baseados no framework Laravel, que foi desenvolvido pela mesma equipe e mesmo autor Taylor Otwell.

O grande diferencial desta microframework segundo Willians (2015), é que o Lumen possui uma ótima estrutura organizacional, que permite futuramente adicionar recursos do Laravel, apenas descomentando algumas linhas.

A arquitetura utilizada é o MVC (Model View Controller), e para manipular a camada de dados é utilizado o conceito de ORM chamado de Eloquent, a qual faz as relações das tabelas banco de dados com as classes do Model.

4. PROPOSTA DO TRABALHO

No presente projeto será desenvolvida uma aplicação web completa para a gestão de lojas de varejo, de forma fácil e ágil, utilizando técnicas, frameworks, e métodos ágeis de engenharia de software.

O projeto possuirá uma API REST que apresenta soluções para a chamada e gerenciamento das informações tratadas na camada de Frontend, visando aumentar a experiência do usuário com a aplicação. Devido a arquitetura MVC da Framework Lumen, será possível desenvolver uma API de forma simples e rápida, através de conceitos e metodologias empregadas por este Framework.

Na parte de Frontend, a arquitetura e componentes presentes no Framework Angular possibilitarão a criação de telas complexas porem de fácil interação com o usuário, e com o auxílio de outras API públicas para consultas de cep e entre outras, que possibilitarão a diminuição de trabalho ao realizar cadastros dentro da aplicação.

As linguagens que serão usadas são muito relevantes para definir o tipo de software e API que serão desenvolvidos, para o Backend a linguagem utilizada será o PHP, pois possui extrema eficiência em construção de API's e se contextualizam a uma ótima performance em servidores voltados para aplicações web. Na parte do Front serão usados conjuntos de tecnologias web como HTML, CSS e Typescript. Vale ressaltar que o Typescript facilitará muito o trabalho com objetos JSON por ser uma linguagem tipada e muito flexível, possuindo diversos métodos para pré-processar dados para a renderização em tela.

5. MODELAGEM E DESENVOLVIMENTO

Neste capítulo serão detalhadas as etapas do desenvolvimento deste trabalho. Inicialmente será realizado uma entrevista com a empresa de varejo Gato Xadrez Modas, com a finalidade de extrair informações necessárias para a construção da ferramenta.

Em seguida serão apresentados todas as funcionalidades e restrições do sistema identificadas com o usuário e, posteriormente, o mapa mental que possibilita o melhor entendimento da aplicação, a prototipação das telas (Interface de usuários), o diagrama de casos de uso para gerar uma visualização das permissões de usuário que o sistema conterà e o modelo do banco de dados, onde serão detalhadas as características e a modelagem do banco.

5.1. ANÁLISE DE REQUISITOS

Como primeiro passo na modelagem do sistema será realizado o levantamento de requisitos, os requisitos funcionais e não funcionais a fim de determinar características e operações que o sistema deverá conter. Segundo Sommerville (2007, p.80) os requisitos funcionais são:

“São as declarações de serviços que o sistema deve receber, como o sistema deve reagir a entradas específicas e como devem se comportar em determinadas situações. Alguns casos, os requisitos funcionais podem também descrever de forma explícita o que o sistema não deve fazer”.

Já os requisitos não funcionais, de acordo com Sommerville (2007), são conectados com a qualidade de software e suas características, como a finalidade, tempo, resposta, desempenho, entre outros.

Com base nas técnicas citadas acima segue a análise de requisitos do presente desenvolvimento:

Manter usuário			
Descrição: O sistema deve possibilitar a criação edição do perfil do usuário, adicionar ou restringir permissões a este.			
Requisitos Funcionais			
Nome	Restrição	Desejável	Obrigatório
RF1.1 Identificação do usuário no sistema	O usuário será identificado por um código único dentro do sistema	()	(X)
RF1.2 Perfil de usuário	O sistema deve conter níveis de usuários, categorizados por tipos: admin, básico.	()	(X)
RF1.3 Criar usuário (Primeiro acesso)	O sistema solicita informações de cadastro ao usuário, porém usuário é iniciado com nível básico.	()	(X)
RF1.4 Realizar login	O sistema pede o login e a senha do usuário.	()	(X)

Tabela 1 - Manter Usuário.

Manter Cliente			
Descrição: O sistema deve possibilitar o cadastro do cliente e realizar consultas a históricos do mesmo dentro das ações disponíveis pelo sistema.			
Requisitos Funcionais			
Nome	Restrição	Desejável	Obrigatório
RF2.1 Identificação do cliente no sistema	O cliente é identificado por um código único dentro do sistema	()	(X)
RF2.2 Cadastro do usuário, listagem, criação, edição e exclusão.	Ao realizar este processo deve-se levar em conta a regra de negócio da empresa	()	(X)

Tabela 2 - Manter Cliente.

Manter Funcionários			
Descrição: O sistema deve possibilitar o cadastro do funcionário e consultas a históricos do mesmo dentro das ações disponíveis pelo sistema.			
Requisitos Funcionais			
Nome	Restrição	Desejável	Obrigatório
RF3.1 Identificação do Funcionário no sistema	O funcionário é identificado por um código único dentro do sistema	()	(X)
RF3.2 Realizar vendas, comocionais, estornos, consultas e cadastro de peças, formas de pagamento.	Ao realizar este processo deve-se levar em conta a regra de negócio da empresa	()	(X)
RF3.3 Controle de férias.	O sistema precisa realizar a gerencia das férias dos funcionários.	()	(X)

Tabela 3 - Manter funcionário.

Manter Autorizados			
Descrição: O sistema deve possibilitar compra a pessoas autorizadas por clientes já cadastrados no sistema.			
Requisitos Funcionais			
Nome	Restrição	Desejável	Obrigatório
RF4.1 Identificação	O autorizado será reconhecido no sistema com a pessoa próxima ou parente do cliente.	()	(X)
RF4.2 Realizar compras	Ao realizar a venda autorizada, cliente estar ciente.	()	(X)

Tabela 4 - Manter autorizado.

Manter Caixas de roupas			
Descrição: O sistema deve possibilitar o cadastro e alteração de caixas de mercadorias por conjunto de peças, levando em consideração a coleção, modelo e categoria.			
Requisitos Funcionais			
Nome	Restrição	Desejável	Obrigatório
RF5.1 Identificação.	As caixas funcionam como agrupadores de roupas, para melhorar a organização.	()	(X)
RF5.2 Realizar cadastro, alteração e exclusão.	No caso da exclusão a caixa não deve possuir roupas inseridas.	()	(X)

Tabela 5 - Manter caixas de roupas.

Manter Categorias.			
Descrição: O sistema conter um CRUD para as categorias.			
Requisitos Funcionais			
Nome	Restrição	Desejável	Obrigatório
RF6.1 Identificação.	As categorias devem ser únicas no sistema, sem a possibilidade de se repetir.	()	(X)
RF6.2 Realizar consultas, inserção, exclusão e exclusão.	O sistema deve fornecer opções de CRUD para o usuário.	()	(X)

Tabela 6 - Manter categorias.

Manter Coleções.			
Descrição: O sistema conter um CRUD para as coleções.			
Requisitos Funcionais			
Nome	Restrição	Desejável	Obrigatório
RF7.1 Identificação.	As coleções devem ser únicas no sistema, sem a possibilidade de se repetir.	()	(X)
RF7.2 Realizar consultas, inserção, exclusão e exclusão.	O sistema deve fornecer opções de CRUD para o usuário.	()	(X)

Tabela 7 - Manter Coleções.

Manter Marcas.			
Descrição: O sistema conter um CRUD para as marcas.			
Requisitos Funcionais			
Nome	Restrição	Desejável	Obrigatório
RF8.1 Identificação.	As marcas devem ser únicas no sistema, sem a possibilidade de se repetir.	()	(X)
RF8.2 Realizar consultas, inserção, exclusão e exclusão.	O sistema deve fornecer opções de CRUD para o usuário.	()	(X)

Tabela 8 - Manter Marcas.

Manter Peças.			
Descrição: O sistema deve realizar a inserção das peças nas caixas.			
Requisitos Funcionais			
Nome	Restrição	Desejável	Obrigatório
RF9.1 Identificação.	As peças devem ser únicas no sistema, porém podem ser agrupadas por caixas onde as definem.	()	(X)
RF9.2 Realizar inserção, edição, exclusão.	O sistema deve fornecer opções de CRUD para as peças.	()	(X)

Tabela 9 – Manter Peças.

Manter condições de pagamento.			
Descrição: O sistema conter um CRUD para as coleções.			
Requisitos Funcionais			
Nome	Restrição	Desejável	Obrigatório
RF10.1 Identificação.	As condições de pagamento devem ser únicas no sistema, sem a possibilidade de se repetir.	()	(X)
RF10.2 Realizar consultas, inserção, alteração e exclusão.	O sistema deve fornecer opções de CRUD para o usuário.	()	(X)

Tabela 10 – Manter Condições de pagamento.

Manter Condicionais.			
Descrição: O sistema deve controlar todas as peças de roupas levadas em condicionais, assim como a devolução, com análises de históricos e demais opções de filtros.			
Requisitos Funcionais			
Nome	Restrição	Desejável	Obrigatório
RF11.1 Identificação.	As condicionais devem ser únicas no sistema.	()	(X)
RF11.2 Realizar consultas a históricos, gravar condicional e devolução das peças.	O sistema deve controlar as peças levadas e o prazo de devolução.	()	(X)
RF11.3 Transformar condicional em venda direta.	O sistema deve conter a opção de realizar a venda a partir das peças selecionadas na condicional.	(X)	()

Tabela 11 – Manter Condicionais.

Manter Vendas.			
Descrição: O sistema deve controlar todas as vendas realizadas, além de informar as condições de pagamentos a qual o cliente preferir.			
Requisitos Funcionais			
Nome	Restrição	Desejável	Obrigatório
RF12.1 Cada venda possui identificação única no sistema.	As vendas devem ser únicas no sistema.	()	(X)
RF12.2 Dados cliente ou autorizados / vendedor	O funcionário realiza a venda, seleciona o vendedor ou não caso seja si mesmo.	()	(X)
RF12.3 Dados de pagamento.	O sistema fornece todas as formas de pagamento ao cliente, e o funcionário finaliza a venda.	()	(X)

Tabela 12 – Manter Vendas.

Dashboard.			
Descrição: O sistema deve conter visualizações de informações em forma de gráficos, para melhorar as análises gerais.			
Requisitos Funcionais			
Nome	Restrição	Desejável	Obrigatório
RF13.1 Representação.	O sistema deve representar os dados dos cadastros essenciais em visões de gráficos, barra, doughnut, pizza, etc.	()	(X)

Tabela 13 – Painel de Gráficos (Dashboard).

Aplicação Multiplataforma.			
Descrição: O sistema deve ser executado em multiplataformas.			
Requisitos Não-Funcionais			
Nome	Restrição	Desejável	Obrigatório
RNF14.1 Interface Gráfica.	O sistema deve funcionar e manter mesmos aspectos em diferentes plataformas.	(X)	()

Tabela 14 – Multiplataforma.

5.2. MAPA MENTAL

Para representar de forma clara e objetiva as funcionalidades neste sistema, foi utilizado o mapa mental, que é usado para representar conceitos, tarefas, palavras, organizadas ao redor de uma palavra-chave ou requisito central. Segundo Buzan (2009), o mapa mental pode auxiliar o desenvolvimento da criatividade, assim como promove ajuda para gerir novas ideias, e propor soluções novas e criativas.

A Figura 7 ilustra o mapa mental deste projeto, onde a palavra-chave é dada pelo nome da empresa a qual será implementado este software:

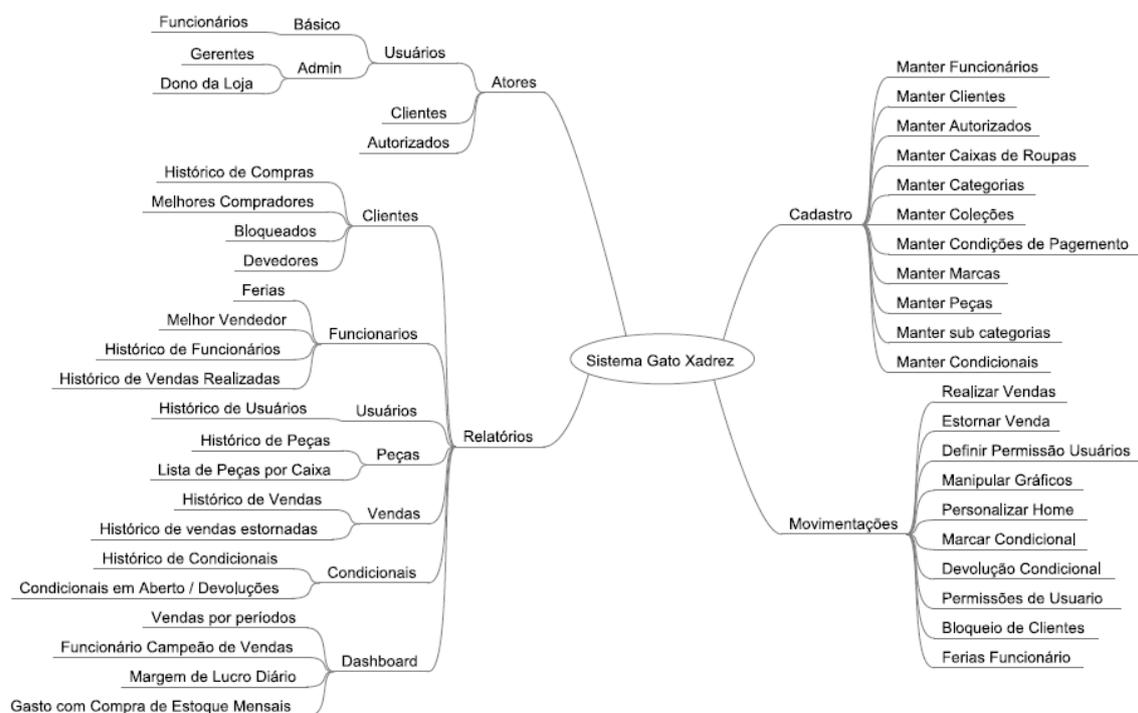


Figura 7 – Mapa mental.

Como é possível observar com o auxílio deste mapa mental, além de visualizar alguns processos que serão de extrema utilidade no desenvolvimento, facilita o entendimento das partes a serem construídas.

5.3. DIAGRAMA DE CASOS DE USO

O Diagrama de casos de uso é usado para descrever as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema, os diagramas são utilizados nas fases de levantamento de requisitos e análise do sistema, porém pode ser consultado durante todo o processo de modelagem e implementação do software. Segundo CONALLEN (2003) um diagrama de caso de uso expressa todos os casos de uso do sistema e a interação que os atores realizam.

O Diagrama é composto por quatro partes, descritas a seguir:

Cenário: Sequência de eventos que ocorrem quando há uma interação do usuário com o sistema.

Ator: Usuário do sistema, ou melhor, tipos de usuários.

Caso de uso: é um método ou funcionalidade que pode ser realizada pelo ator.

Comunicação: é o que realiza a ligação de ator com o caso de uso.

A Figura 8 ilustra a descrição dos casos de usos do presente desenvolvimento do software de gerenciamento de lojas de varejo, expressando todos os casos de usos e comunicações com os atores do presente sistema.

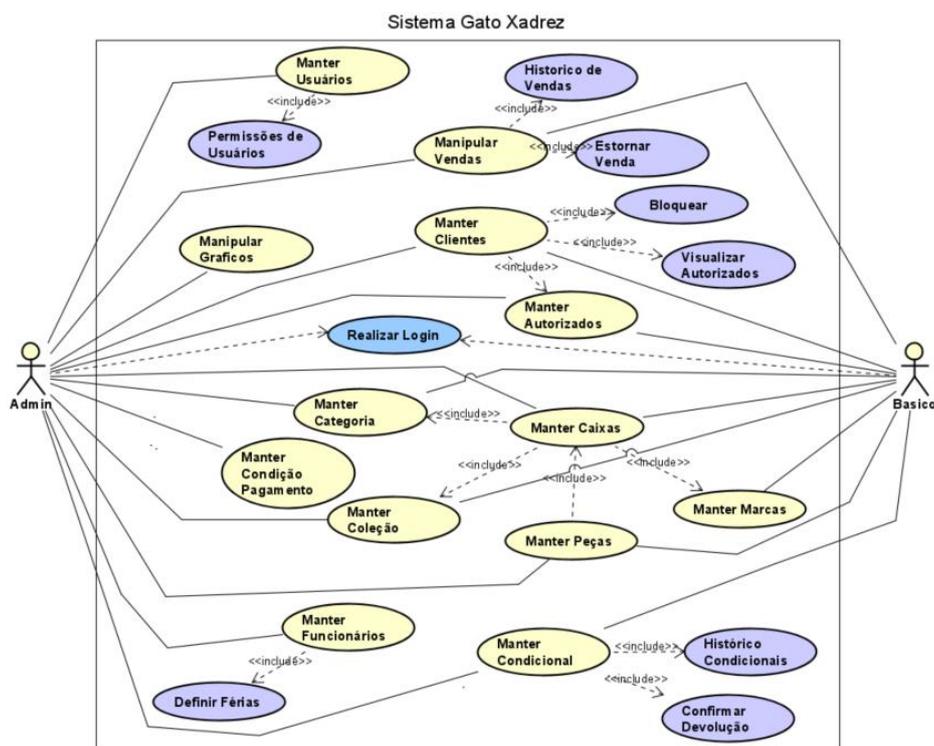


Figura 8 – Diagrama de Caso de Uso.

5.4. BANCO DE DADOS

A modelagem de banco de dados é uma parte essencial que exige bastante da parte cognitiva do analista de sistemas, pois abstrair coisas do mundo real pode ser uma tarefa complexa. No ponto de vista de um desenvolvedor a modelagem de dados é conceitualmente similar à modelagem de classes, porém o foco da modelagem de dados é totalmente voltado para a arquitetura dos dados, já a de classes explora mais o comportamento e domínio dos dados em uma aplicação.

O processo de modelagem em um banco relacional pode ser dividido em três partes segundo HEUSER (2008). Na primeira etapa, a qual é chamada de etapa conceitual, são especificados as entidades e os relacionamentos entre elas, já na segunda, a etapa lógica, são projetados os esquemas relacionais, como a representação da tabela, atributos e os relacionamentos, a última etapa é o projeto físico, com o uso de um SGBD (Sistema Gerenciador de Banco de Dados).

A Figura 9 ilustra a modelagem do banco de dados que será utilizado, havendo a possibilidade de ocorrer alterações até o prazo final de apresentação trabalho.

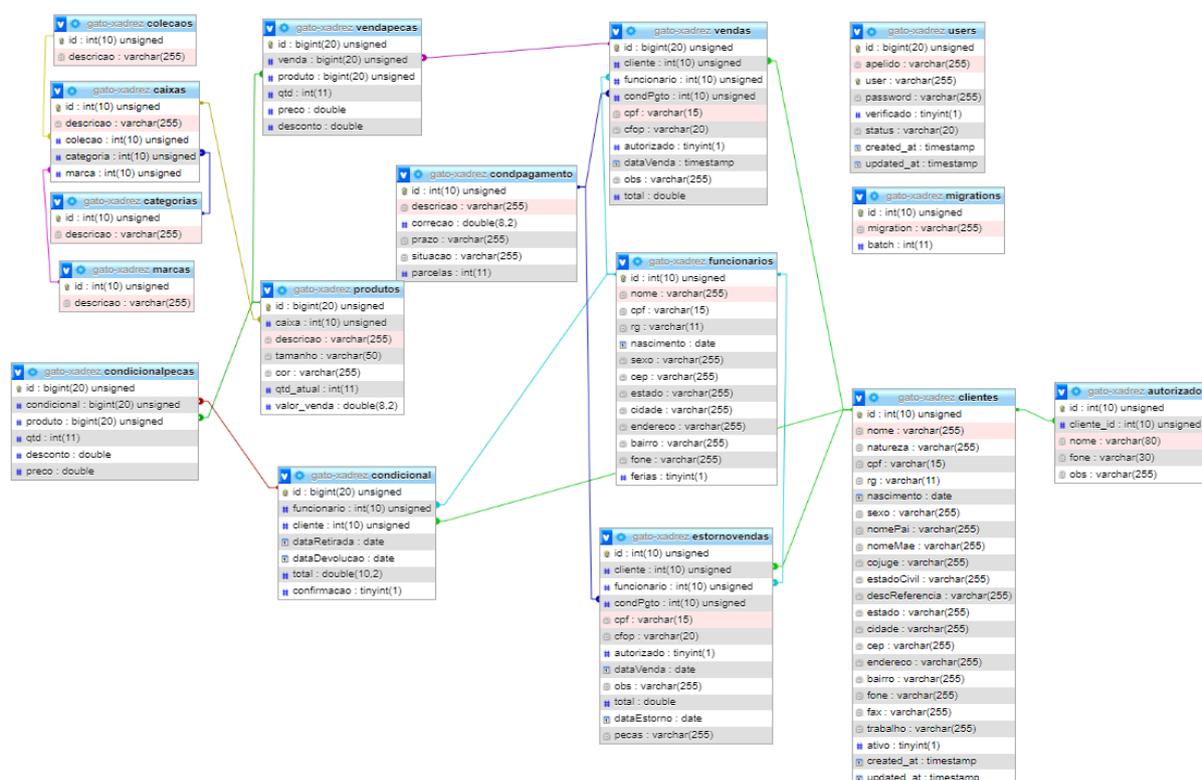


Figura 9 – Modelagem de Banco de Dados.

5.5. DIAGRAMA DE CLASSE

O diagrama de classe é um processo fundamental para a modelagem e desenvolvimento de um software. Classes são descritas segundo Booch, Rumbaugh e Jacobson (2005), como a descrição de um conjunto de objetos que partilham recursos e atributos, relacionamentos e semântica.

Como parte fundamental da modelagem, Melo (2010) afirma que o diagrama de classe é a estrela principal de um sistema orientado a objetos. A Figura 10 ilustra o diagrama realizado para o presente projeto.

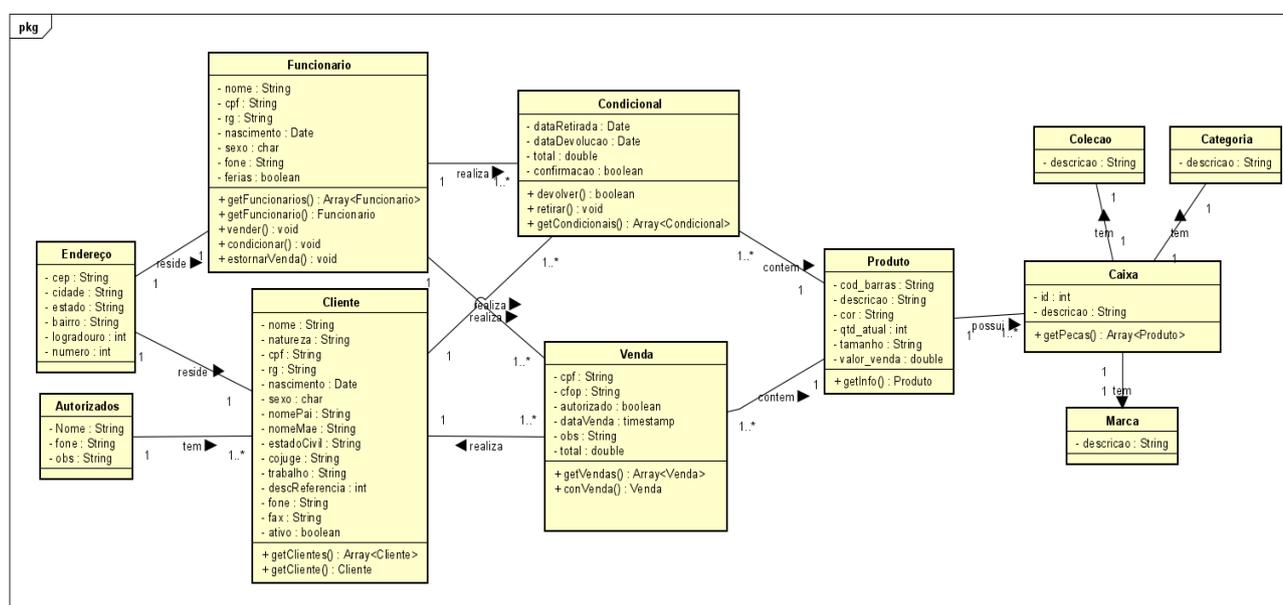


Figura 10 – Diagrama de Classe.

Os atributos usados neste diagrama são baseados nos processos anteriores de modelagem, tudo foi iniciado no levantamento de requisitos, porem umas das referências que propõem mais facilidade para a criação do diagrama de classe é o modelo do banco de dados, onde permite a visualização dos objetos e as entidades de relacionamento.

5.6. DESENVOLVIMENTO

A prototipação de tela, em geral, pode ser compreendida como modelos funcionais construídas a partir de especificações e simulam a aparência e funcionalidades de um software a ser desenvolvido, segundo SOMMERVILLE e Sawyer (1997), o papel de um protótipo pode ser visto como um meio de comunicação entre diversos membros de uma equipe de desenvolvimento ou como caminho para testar as ideias.

Com base na análise de requisitos, localizada na seção 3.1 será explanado superficialmente detalhes dos protótipos de telas desenvolvidos, segundo com a listagem dessas e sua descrição:

5.6.1. Tela de Login

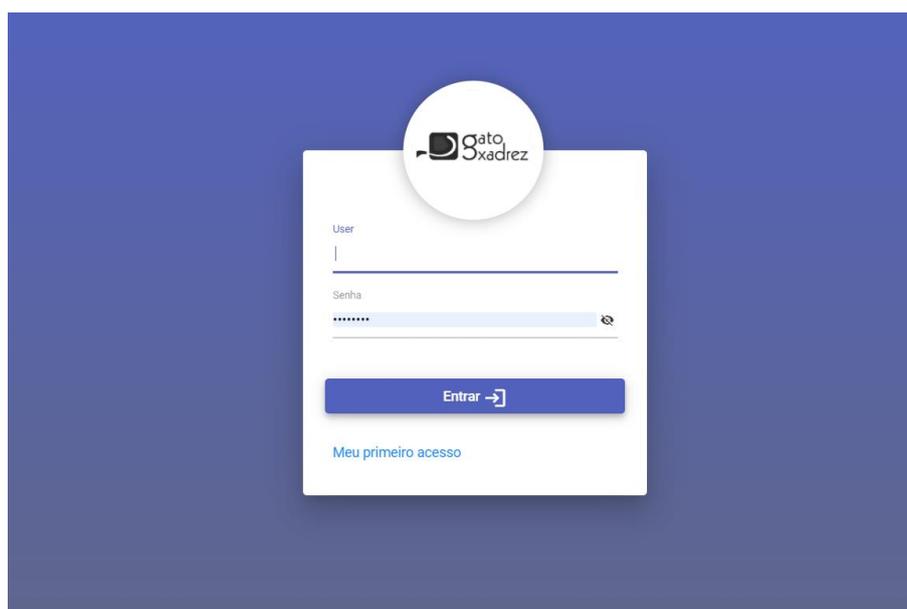
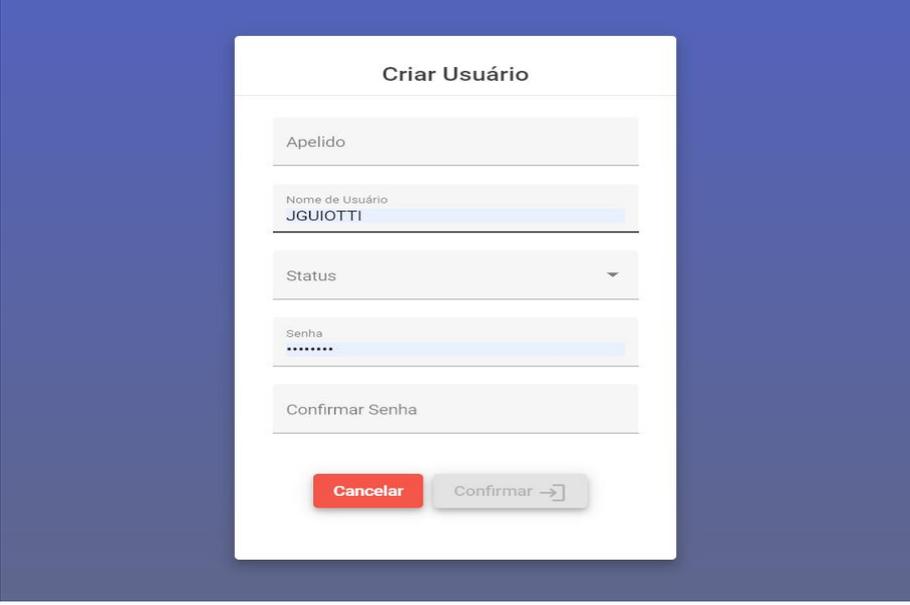


Figura 11 – Realizar Login.

A Figura 11 ilustra o protótipo do RF1.4, que representa a tela de autenticação que segue um layout limpo e funcional, com opções para o usuário entrar no sistema e cadastrar-se, na opção “meu primeiro acesso”.

5.6.2. Tela de Primeiro Acesso



The image shows a web form titled "Criar Usuário" (Create User) centered on a dark blue background. The form is white and contains the following fields and elements:

- A text input field labeled "Apelido" (Surname).
- A text input field labeled "Nome de Usuário" (Username) with the value "JGUIOTTI" entered.
- A dropdown menu labeled "Status".
- A password input field labeled "Senha" (Password) with masked characters ".....".
- A text input field labeled "Confirmar Senha" (Confirm Password).
- At the bottom, there are two buttons: a red "Cancelar" (Cancel) button and a grey "Confirmar →" (Confirm) button.

Figura 12 – Meu Primeiro Acesso.

O Meu primeiro acesso RF1.3 representados visualmente pela Figura 12, apresenta em sua composição um formulário relativamente simples, onde o pré-usuário informa os dados de acesso que deseja ter para os seus futuros acessos. Por padrão validações de telas serão aplicadas para não deixar passar “dados quebrados”.

5.6.3. Tela Inicial

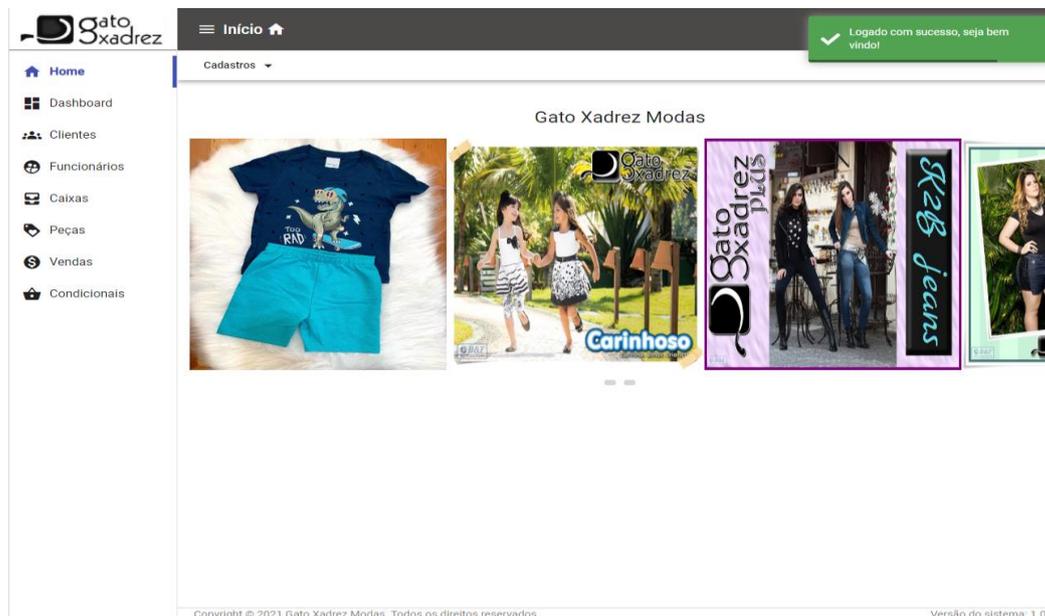


Figura 13 – Página Inicial.

A tela inicial representada pela Figura 13, da sequência ao fluxo de login, aparecendo após a autenticação do usuário, apenas com um carrossel de ilustrações.

5.6.4. Formulário Dados Pessoais

The image shows a web form titled 'Alterar Dados Pessoais' with a close button (X) in the top right corner. The form contains two text input fields: 'Apelido' with the value 'GUIOTTI' and 'User' with the value 'JGUIOTTI'. Below these is a dropdown menu for 'Status' with 'Basico' selected. At the bottom left, there is a checkbox labeled 'Confirmar?'. At the bottom right, there are two buttons: 'Cancelar' and 'Salvar'.

Figura 14 – Alterar Usuário.

Esta tela acima, Figura 14, é onde o usuário autenticado pode alterar suas informações e para acessa-la é só navegar pelo menu do top da tela, no canto superior direito e clicar no seu “avartar”, altear perfil, e para realmente alterar as informações do usuário é preciso confirmar o “checkbox” e clicar no botão salvar.

5.6.5. Tela de Dashboard

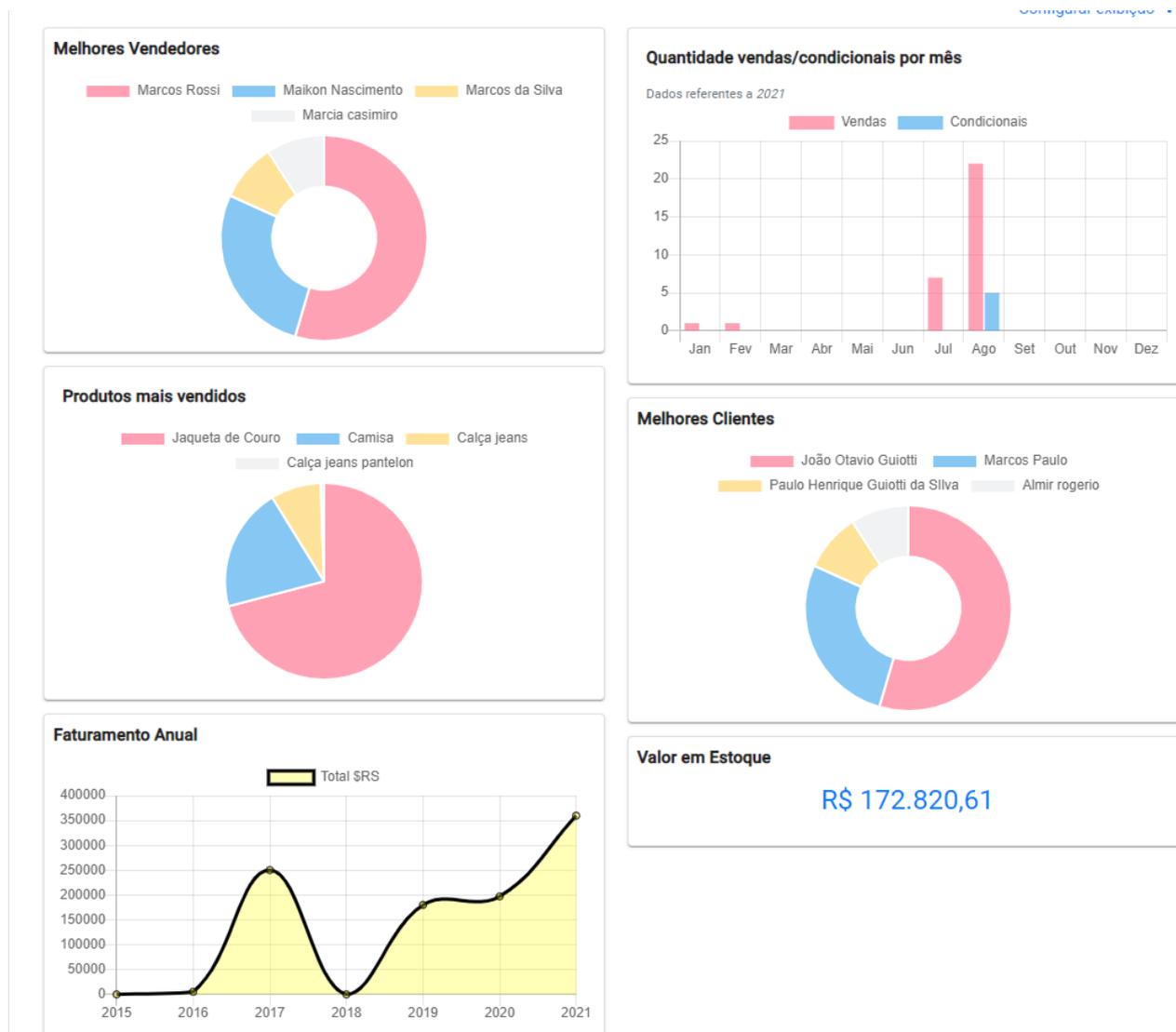


Figura 15 – Dashboard.

A Figura 12 ilustra o “Dashboard” da aplicação RF14.1, é uma página onde os dados são visualizados em gráficos, exibindo dados reais da base de dados e possibilitando a interação com cada gráfico.

5.6.6. Formulário de Cliente

Figura 16 – Formulário de cliente.

A Figura 16 ilustra o formulário de cadastro de clientes, onde possui dois “steps”, no primeiro temos a entrada dos dados pessoais do cliente e ao passar para o segundo o usuário irá informar o endereço atual onde o cliente reside.

5.6.7. Tela de Cliente

Nome	Natureza	CPF/CNPJ	Nascimento	Sexo	Estado Civil	Cidade	CEP	Endereço	Fone	Ativo	Ações
Cliente teste	BR	123.123.123-12	19/11/2001	MASCULINO	SOLTEIRO	Cândido Mota	19.880-556	Rua Gerson Zanini	(18) 99999-9999		0
Joao	BR	123.213.213-11	19/11/2002	MASCULINO	SOLTEIRO	Cândido Mota	19.880-512	Rua Joaquim Guiotti	(18) 99921-3123		0
Almir rogerio	BR	477.747.477-77	19/01/1998	MASCULINO	CASADO	Cândido Mota	19.880-556	Rua Gerson Zanini	(18) 99999-9999		2
teste	BR	123.123.131-22	19/11/2000	MASCULINO	SOLTEIRO	Cândido Mota	19.880-556	Rua Gerson Zanini	(89) 08989-0898		0
Gabriela Kekh Silva	BR	123.213.131-21	10/12/2000	FEMININO	CASADO	Cândido Mota	19.880-512	Rua Joaquim Guiotti	(18) 81218-3212		0
Paulo Henrique Guiotti da Silva	BR	474.985.321-31	03/07/1999	MASCULINO	SOLTEIRO	Cândido Mota	19.880-556	Rua Gerson Zanini	(18) 9985-2094		0
Marcos Paulo	BR	123.132.132-13	19/11/1999	MASCULINO	SOLTEIRO	Cândido Mota	19.880-556	Rua Gerson Zanini	(12) 31232-1321		1
Paulo Roberto da Silva	BR	456.756.757-66	19/11/2000	MASCULINO	SOLTEIRO	Cândido Mota	19.880-556	Rua Gerson Zanini	(18) 99631-2312		0

Figura 17 – Listagem de clientes.

A tela acima ilustrada pela Figura 17 e descrita pelo RF2.2, é área visual onde o usuário terá todos os clientes cadastrados na base de dados podem realizar as operações de CRUD e adicionar uma pessoa autorizada a comprar pelo cadastro do cliente.

5.6.8. Formulário de Funcionário

Cadastrar Funcionário ✕

Dados funcionario Endereço

Nome *	Fone *
CPF/CNPJ *	RG
Nascimento * 	Sexo * 

Cancelar Salvar

Figura 18 – Formulário de funcionários.

A Figura 18 ilustra o formulário de cadastro de funcionários e endereços, o modelo de formulários segue o mesmo para quase todas as outras interfaces de cadastro

5.6.9. Tela de Funcionário

Funcionário
Exportar

Nome	CPF/CNPJ	RG	Nascimento	sexo	CEP	Cidade	Endereço	Bairro	Fone	Ferías	Ações
Marcos da Silva	123.213.121-32	12.312.312-3	19/11/2000	FEMININO	19.880 - 556	Cândido Mota	Rua Gerson Zanini, 19	Jardim Paraíso	(12) 31231-2321	S	
Maikon Nascimento	123.213.123-11	12.312.313-1	19/03/1998	MASCULINO	19.880 - 556	Cândido Mota	Rua Gerson Zanini	Jardim Paraíso	(18) 81311-2312	N	
Marcia casimiro	231.312.312-23	12.456.456-4	19/11/2000	FEMININO	19.880 - 556	Cândido Mota	Rua Gerson Zanini	Jardim Paraíso	(18) 99292-9929	N	
Marcos Rossi	897.898.998-88	89.789.988-8	28/07/2021	MASCULINO	19.880 - 512	Cândido Mota	Rua Joaquim Guiotti	Jardim Santa Lucia	(18) 8978-7879	N	

Items por página 8 1 - 4 of 4 < >

Figura 19 – Listagem de funcionários.

A listagem de funcionários é representada pela Figura 19, possuindo funcionalidades de filtro e exportar para o Excel, os funcionários são organizados em uma grid com opções de exclusão e edição por registro.

Cadastro Caixa
X

Cancelar
Salvar

Figura 20 – Formulário de caixas.

O Formulário de caixas é ilustrado pela Figura 20, funcionalidade simples de criar caixas para agrupar os produtos(peças) assim como pode ser analisado no RF5.2.

5.6.10. Tela de Funcionário

Descrição	Coleção	Categoria	Marca	Ações
sampa	Inverno	Manga Longa	Polo	
Malve	Inverno	Moletom	Mave	

Items por página 8 1 - 2 of 2 < >

Figura 21 – Listagem de caixas.

A listagem de caixas é representada pela Figura 21, possuindo requisitos analisados no item RF5.2

5.6.11. Formulário de Produto

Cadastrar Produto ✕

Descrição

Caixa *

Tamanho

Cor

Qtd. Atual

Valor

Cancelar Salvar

Figura 22 – Formulário de produtos.

O Formulário de produtos, como ilustra a Figura 22 para o cadastro de um novo produto(peça) um modal é aberto ao usuário e os dados são validados em tempo de digitação, com duas opções.

5.6.12. Tela de Produto

<input type="checkbox"/>	#	Tamho	Cor	Qtd. Atual	Valor	Valor Estoque	Caixa	Colecao	Categoria	Obs	Ações
<input type="checkbox"/>	1	GG	Cinza - branco	1386	R\$ 22,87	R\$ 31.697,82	sampa	Inverno	Manga Longa		
<input type="checkbox"/>	2	M	Marrom	108	R\$ 1.000,00	R\$ 108.000,00	sampa	Inverno	Manga Longa		
<input type="checkbox"/>	3	12	preta	196	R\$ 22,00	R\$ 4.312,00	sampa	Inverno	Manga Longa		
<input type="checkbox"/>	4	48	Marrom	197	R\$ 44,67	R\$ 8.799,99	Malve	Inverno	Moletom		
<input type="checkbox"/>	5	M	Azul	146	R\$ 44,50	R\$ 6.497,00	Malve	Inverno	Moletom		
<input type="checkbox"/>	6	42	Marrom	202	R\$ 66,90	R\$ 13.513,80	Malve	Inverno	Moletom		
Total						R\$ 172.820,61					

Figura 23 – Listagem de produtos.

O Para a listagem de produtos como representa a Figura 23 foi utilizado um grid (Tabela) customizado, onde permite a seleção de múltiplas linhas do lado direito com uma caixinha de seleção, e após essa seleção dentro das opções do botão exportar, é exibido a opção de “exportar linhas selecionadas”, também foi adicionado um totalizador na coluna “Valor Estoque”.

5.6.13. Modal de Categoria

ID	Descrição	Ações
1	Manga Longa	
2	Calça	
3	Moletom	
4	Pantakur	
5	Bermuda	

Figura 24 – Tela de categoria.

A tela de categorias é ilustrada pela Figura 24 é aberta sobre um “modal” e fornece opções para o cadastro e visualização de categorias já cadastradas, assim como opções de edição e exclusão.

5.6.14. Modal de Categoria

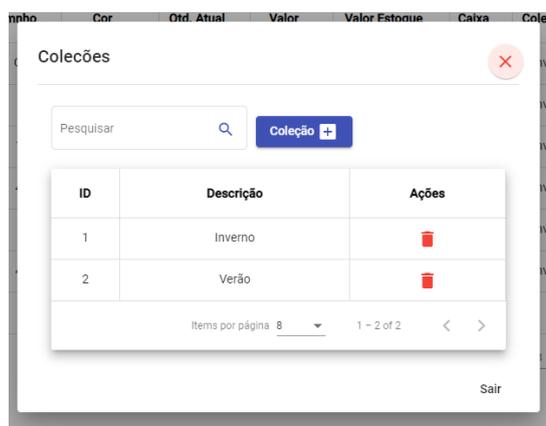


Figura 25 – Tela de coleção.

A Tela de coleção é ilustrada pela Figura 25, aberta sobre um modal que contem todos as coleções e opções para criar e excluir uma coleção.

5.6.15. Modal de Marca

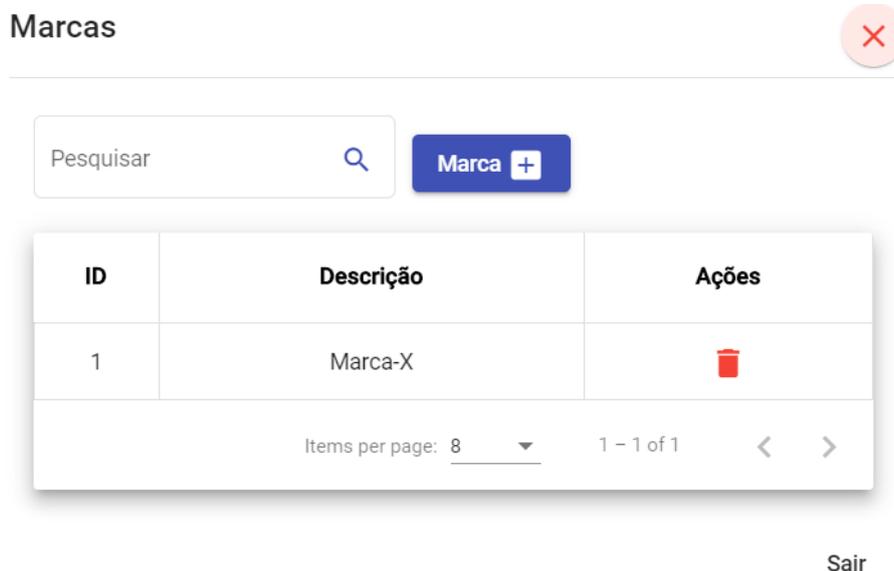


Figura 26 – Tela de marca.

A Tela de marcas é representada pela Figura 26 e também é aberta sobre um modal, permitindo a visualização, criação e edição de uma marca.

5.6.16. Modal de Venda

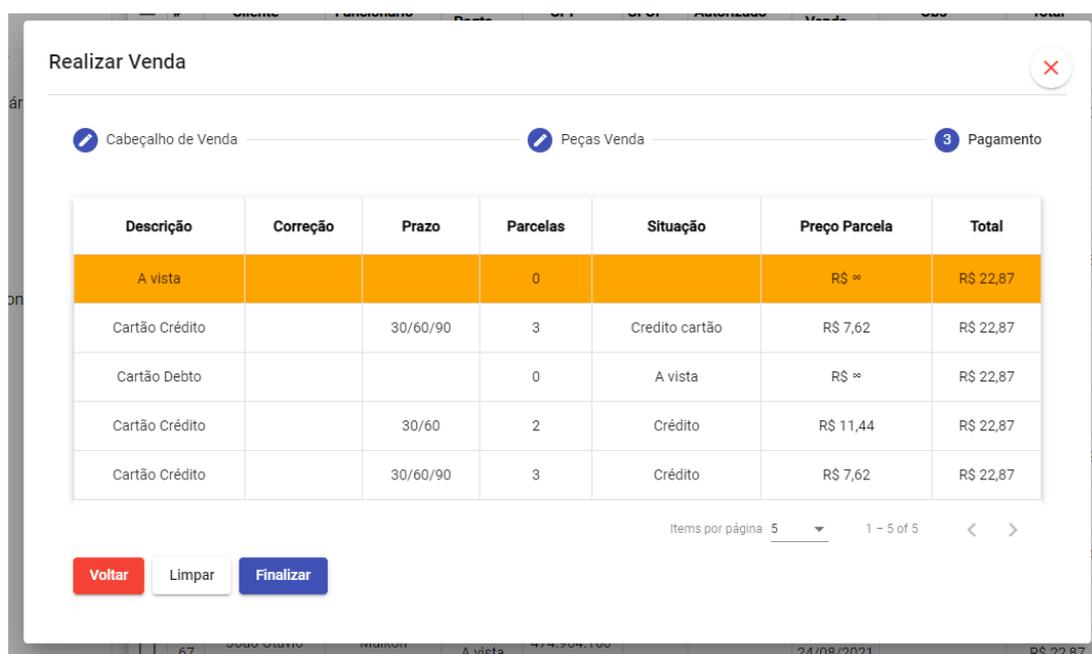


Figura 27 – Formulário de Venda.

O formulário de vendas é ilustrado pela Figura 27, aberto sobre um modal e possui 3 etapas para a realização da venda, a validação abrange as 3 abas para assim poder realizar uma venda.

5.6.17. Tela de Vendas

The screenshot shows a web interface for sales management. At the top, there is a search bar with the text 'Pesquisar'. Below it, there are two filter buttons: 'Data Venda (De):21/08/2021' and 'Data Venda (Até):31/08/2021'. To the right, there are two buttons: 'Vender' with a dollar sign icon and 'Exportar' with a document icon. Below the filters, there is a button for 'Cliente:Guiotti' and a link for 'Novo filtro...'. The main part of the interface is a table with the following columns: #, Cliente, Funcionário, Cond. Pagto, CPF, CFOP, Autorizado, Dt. Venda, Obs, Total, and Ações. The table contains 8 rows of data, including a total row at the bottom. The total value is R\$ 40.506,21. At the bottom right, there is a pagination control showing 'Items por página 13' and '1 - 8 of 8'.

<input type="checkbox"/>	#	Cliente	Funcionário	Cond. Pagto	CPF	CFOP	Autorizado	Dt. Venda	Obs	Total	Ações
<input type="checkbox"/>	79	João Otavio Guiotti	Marcos Rossi	Cartão Crédito	474.984.168-40			27/08/2021		R\$ 22,87	
<input type="checkbox"/>	74	Paulo Henrique Guiotti da Silva	Maikon Nascimento	A vista	474.985.321-31			24/08/2021	dasdad	R\$ 22,87	
<input type="checkbox"/>	70	Paulo Henrique Guiotti da Silva	Marcia casimiro	A vista	474.985.321-31			24/08/2021	Observacao	R\$ 201,06	
<input type="checkbox"/>	67	João Otavio Guiotti	Maikon Nascimento	A vista	474.984.168-40			24/08/2021		R\$ 22,87	
<input type="checkbox"/>	64	João Otavio Guiotti	Marcos Rossi	A vista	474.984.168-40			24/08/2021	dasdas	R\$ 1.922,00	
<input type="checkbox"/>	63	João Otavio Guiotti	Marcos Rossi	A vista	474.984.168-40			24/08/2021		R\$ 37.000,00	
<input type="checkbox"/>	62	Paulo Henrique Guiotti da Silva	Marcos Rossi	Cartão Crédito	474.985.321-31			23/08/2021		R\$ 1.022,87	
<input type="checkbox"/>	60	João Otavio Guiotti	Marcos Rossi	A vista	474.984.168-40		Paulo Henrique	23/08/2021	Compra de autorizado	R\$ 291,67	
Total										R\$ 40.506,21	

Figura 28 – Listagem de Vendas.

A listagem de vendas é ilustrada pela Figura 28 e possui um grid (Tabela) customizado com opções em cima da listagem de filtragem completa, vender e exportar, na parte da listagem possui um totalizador na coluna “Total”.

5.6.18. Formulário de Condicional

Figura 29 – Formulário de condicionais.

O formulário de condicionais representado pela Figura 29 segue o mesmo estilo dos demais que possuem dois ou mais seguimentos, porém neste caso o “step” cabeçalho da condicional, vão os dados de quem abriu a condicional, o cliente, data de devolução e observação. Já na outra aba são informadas as peças escolhidas.

5.6.19. Tela Condicional

Pesquisar

Data Retirada:21/08/2021 ✕
Data Devolução:31/08/2021 ✕

Novo filtro...

Condicional +
Exportar 📄

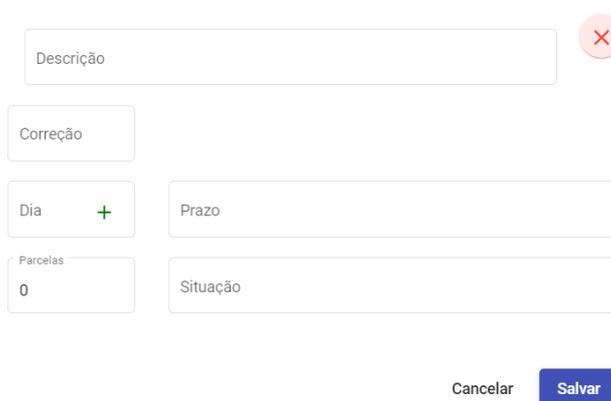
<input type="checkbox"/>	#	Cliente	Funcionário	Qtd. Peças	Dt. Retirada	Previsão Devolução	Total	Ações
<input type="checkbox"/>	19	Paulo Henrique Guiotti da Silva	Marcos da Silva	1	24/08/2021	31/08/2021	R\$ 22,87	\$ 🛒 📄
<input type="checkbox"/>	16	Paulo Roberto da Silva	Maikon Nascimento	12	24/08/2021	27/08/2021	R\$ 762,66	\$ 🛒 📄
<input type="checkbox"/>	14	Gabriela Kekh Silva	Marcos da Silva	2	24/08/2021	31/08/2021	R\$ 67,37	\$ 🛒 📄
<input type="checkbox"/>	11	Paulo Henrique Guiotti da Silva	Marcia casimiro	1	21/08/2021	28/08/2021	R\$ 42,44	\$ 🛒 📄
Total				16			R\$ 895,34	

Items por página 13 ▾ 1 - 4 of 4 < > >>

Figura 30 – Listagem de condicionais.

A Figura 30 ilustra a listagem das condicionais, para esta tela foi usado o grid customizado e com opções de filtros por todas as colunas, também foi adicionado o totalizador nas colunas “Qtd. Peças” e “Total”, para cada registro é possível realizar a venda, estornar e visualização dos produtos (Peças).

5.6.20. Formulário de Condição de Venda



O formulário de condição de pagamento apresenta os seguintes campos e controles:

- Um campo de texto para "Descrição" com um ícone de fechar (X) no canto superior direito.
- Um botão "Correção" para atualizar o conteúdo do campo de descrição.
- Um campo "Dia" com um ícone de "+" verde, seguido por um campo "Prazo".
- Um campo "Parcelas" com o valor "0", seguido por um campo "Situação".
- Dois botões de ação: "Cancelar" e "Salvar" (destacado em azul).

Figura 31 – Formulário de condição de pagamento.

O formulário de condição de pagamento é representado pela Figura 31, é nesta interface onde ocorre os cadastros dos pagamentos, o layout visa ser o mais limpo e eficiente o possível.

5.6.21. Caixa de Diálogo

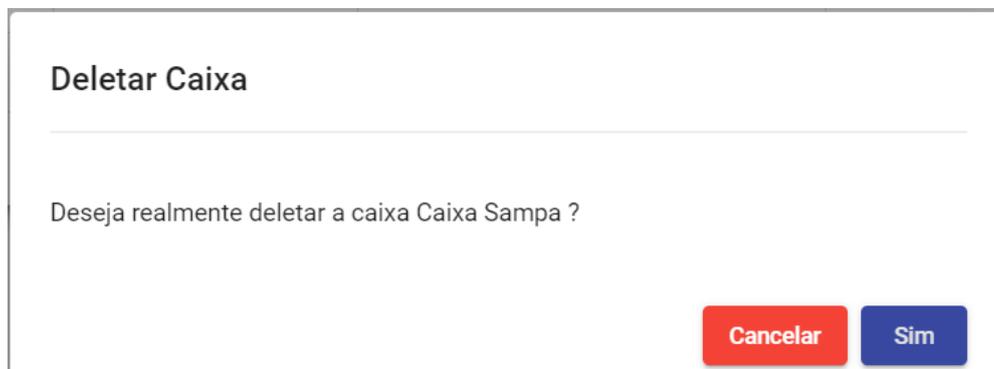


Figura 32 – Caixa de diálogo.

A Caixa de diálogo é um componente de tela compartilhado, que é representado pela Figura 32, o motivo de ser chamado de compartilhado é pelo fato de ser utilizado em quase todo fluxo onde ocorre a confirmação de uma, exclusão, estorno de vendas, confirmação de pagamento, entre outras situações.

5.6.22. Tela de Gerir Perfis

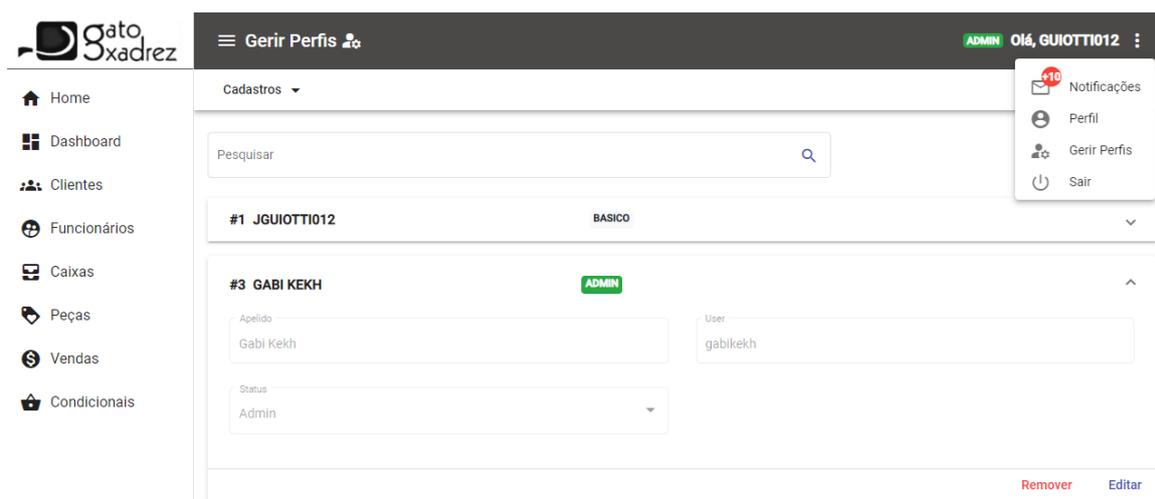


Figura 33 - Gerir Perfis.

A Figura 33 representa a parte de Gerir Perfis, onde somente o usuário com status de admin pode ter acesso. Nesta Tela pode-se filtrar os usuários e realizar edições e exclusões, tanto de status e as informações do mesmo.

5.6.23. Diálogo Realizar Login



Figura 34 - Realizar Logout.

A Figura 34 representa a confirmação para o usuário logado deixar o sistema, esta tela é aberta sobre um modal e possui dois opções, cancelar e confirmar a saída.

6. CONCLUSÃO

O objetivo geral do presente trabalho de conclusão de curso foi realizar a análise e desenvolvimento de um software para gerenciar lojas de varejo, com intuito de construir uma aplicação web utilizando técnicas, metodologias ágeis, frameworks e recursos mais atuais do mercado de desenvolvimento.

Com o propósito de fornecer embasamento teórico e de negócio para esta implementação, foi realizado uma análise de requisitos em uma empresa de varejo, assim obtendo a real necessidade para gerir esse tipo de mercado. Em sequência foi elaborado um mapa mental para representar de forma clara e objetiva as funcionalidades deste sistema. Seguindo as metodologias de desenvolvimento, foi elaborado o diagrama de casos de uso, cujo intuito é descrever as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários; após esses passos foram abstraídos todo esse conjunto para o banco de dados.

Após a criação do banco de dados no início do processo de desenvolvimento do backend, utilizando Lumen, um microframework desenvolvido em PHP, construído para desempenhar ótima performance e voltado para o Microsserviço e API's. Em seguida foram elaborados os protótipos de telas, visando as interações e o uso cotidiano dos atores, como condiz no diagrama de caso de uso, e para o processo de desenvolvimento das interfaces do frontend foi utilizado o framework Angular, um framework voltado para a construção de interface de aplicações web, onde utiliza HTML, CSS e Typescript em sua composição.

Durante o processo de desenvolvimento do backend, utilizando a arquitetura MVC, pude perceber como que é extremamente fácil e rápido mapear as relações do banco de dados, utilizando o ORM (Object relational Mapping), com isso criar os controladores (Controllers) para cada contexto ficou extremamente simples. Com as rotas criadas e apontadas para seus devidos controllers, só bastava criar as interfaces com o Angular que por sua vez permitia realizar chamadas http, utilizando os Observables e a classe HttpClient. Com os serviços criados e buscando os dados na API, com tudo pronto bastava realizar algumas importações de bibliotecas externas como o bootstrap e Angular Material e criar as telas para cada contexto.

No Desenvolvimento do frontend, a parte do Angular CLI (Client Line Interface) agilizou bastante a criação de novos componentes dentro do projeto, assim como o Node que gerencia os pacotes do projeto facilitou a as instalações de novas dependências no projeto.

Como trabalho futuro, no intuito de dar andamento a este sistema, pretende-se efetuar novamente a ampliação e melhoria do mesmo, assim gerando uma nova versão que visa o aperfeiçoamento e melhorias em questões de agilidade nos processos do varejo. Assim possibilitando pesquisas estatísticas, bem como um capítulo explanando a pós implantação do sistema em um ambiente empresarial.

No Desenvolvimento do frontend, a parte do Angular CLI (Client Line Interface) agilizou bastante a criação de novos componentes dentro do projeto, assim como o Node que gerencia os pacotes do projeto facilitou a as instalações de novas dependências no projeto.

Como trabalho futuro, no intuito de dar andamento a este sistema, pretende-se efetuar novamente a ampliação e melhoria do mesmo, assim gerando uma nova versão que visa o aperfeiçoamento e melhorias em questões de agilidade nos processos do varejo. Assim possibilitando pesquisas estatísticas, bem como um capítulo explanando a pós implantação do sistema em um ambiente empresarial.

REFERÊNCIAS

AFONSO, Alexandre. O que é Angular?.**algaworks**, 2018. Disponível em: <https://blog.algaworks.com/o-que-e-angular/>. Acesso em: 07, mar de 2021.

BUZAN, Tony. **Mapas Mentais**. Tradução de Paulo Polzonoff Junior. Rio de Janeiro: Editora Sextante, 2009.

Booch, Grady; James Rumbaugh; e Ivar, Jacobson. **UML: guia do usuário**. 6° Reimpressão. Rio de Janeiro: Elsevier, 2005.

CONALLEN, Jim. **Desenvolvendo Aplicações Web com UML**. Editora Campus, 2003.

HEUSER, C. A. **Projeto de Banco de Dados**. 6. ed. [S.l.]: Bookman, 2008. ISBN 9788577803828.

JWT. Introduction to JSON Web Tokens. Disponível em: <https://jwt.io/introduction>. Acesso em: 07, mar de 2021.

KALUŽA, Troskot, Vukelić, Marin, Krešimir, Bernard. **Comparison of front-end frameworks for web Applications development**, Zbornik Veleučilišta u Rijeci, Vol. 6 (2018), N°. 1, p. (261-282), Jan, 2018. Disponível em: link. <https://hrcak.srce.hr/199922>.

KOTLER, P.; KANTAJAYA, H.; SETIAWAN, I. Marketing 3.0:**As forças que estão definindo o novo marketing centrado no ser humano**. Rio de Janeiro: Elsie-ver, 2010.

LEITE, Marcos. Importância de softwares para loja varejistas. **Blog Artsoft**. São Paulo, 20 fev. 2017. Disponível em: <https://www.artsoftsistemas.com.br/blog/entenda-importancia-software>. Acesso em: 01 mar. 2021.

LOUNDON, Kyle. **Desenvolvimento de Grandes aplicações Web**. Revista Telfrac - nº1/2018. p. 18.

MDN Web Docs. **HTTP** Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP>. Acesso em: 06 mar. 2021.

MDN Web Docs. **O que é um servidor web (web serve)**. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/Common_questions/What_is_a_web_server. Acesso em: 07 mar. 2021.

Melo, Ana Cristina. **Desenvolvendo aplicações com UML 2.2 do conceito à implementação**. 3ª edição. Rio de Janeiro: Brasport, 2010.

O que é API?. **Red Hat**, 2021. Disponível em: <https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>. Acesso em: 07, mar de 2021.

Project Management Institute. **Um Guia do Conhecimento de Projetos**, 6. ed. Newton Square: Project Management Institute, 2017.

REZENDE, Denis Alcides; ABREU, Aline França de. **Tecnologia da informação aplicada a sistemas de informação empresariais: o papel estratégico da informação e dos sistemas de informação nas empresas**. São Paulo: Atlas, 2000.

RODRIGUES, Joel. Testando serviços Web API com Postman?.**Linha de Código**. Disponível em: <https://blog.onedaytesting.com.br/teste-de-api/>. Acesso em: 09, mar de 2021.

SOMMERVILLE, Ian. **Engenharia de Software**. 8ª Ed. Tradução de Selma Shin Shimizu Melnikoff, Reginaldo Arakaki, Edílson de Andrade Barbosa. São Paulo: Editora Pearson, 2007.

SOMMERVILLE, I.; Sawyer P. **Requirements Engineering - A good practice guide**. Editora Wiley, 1997.

STÖCKLMAIR, J. P; Hauquier H, Jakovic M, Landaeta J, Kleine Schaars G. **Elephorum – Web tool of the Catalunya Railway Museum**, <https://upcommons.upc.edu/handle/2117/89940>. Acessado em: 10 mar. 2021.

Tecmundo Web Docs. **Crie uma aplicação com arquitetura Back-end for Front-end** Disponível em: <https://www.tecmundo.com.br/software/155106-crie-aplicacao-arquitetura-back-end-for-front-end.htm>. Acesso em: 07 mar. 2021.

TREINAWEB. **O que é Laravel?**. Disponível em <https://www.treinaweb.com.br/blog/o-que-e-laravel/>. Acesso em 10 de março de 2021.

VISUAL STUDIO CODE. **Docs**. Disponível em <https://code.visualstudio.com/docs>. Acesso em 10 de março de 2021.

WILLIAMS, Wesley. **Lumen, o Micro Framework do Laravel**. Disponível em <https://blog.schoolofnet.com/2015/04/lumen-o-microframework-do-laravel/>. Acesso em 10 de março de 2021.