

**CENTRO UNIVERSITÁRIO PARA O DESENVOLVIMENTO DO ALTO VALE DO
ITAJAÍ - UNIDAVI**

PAULO HENRIQUE FÜLLE

**DUOCICLO: PROTÓTIPO DE APLICAÇÃO WEB PARA GERENCIAMENTO DE
ALUGUÉIS**

**RIO DO SUL
2023**

**CENTRO UNIVERSITÁRIO PARA O DESENVOLVIMENTO DO ALTO VALE DO
ITAJAÍ - UNIDAVI**

PAULO HENRIQUE FÜLLE

**DUOCICLO: PROTÓTIPO DE APLICAÇÃO WEB PARA GERENCIAMENTO DE
ALUGUÉIS**

Trabalho de Conclusão de Curso a ser apresentado ao curso de Sistemas da Informação, da Área das Ciências Naturais, da Computação e das Engenharias, do Centro Universitário para o Desenvolvimento do Alto Vale do Itajaí, como condição parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Orientador: M.e Jullian Hermann Creutzberg

**RIO DO SUL
2023**

**CENTRO UNIVERSITÁRIO PARA O DESENVOLVIMENTO DO ALTO VALE DO
ITAJAÍ - UNIDAVI**

PAULO HENRIQUE FÜLLE

**DUOCICLO: PROTÓTIPO DE APLICAÇÃO WEB PARA GERENCIAMENTO DE
ALUGUÉIS**

Trabalho de Conclusão de Curso a ser apresentado ao curso de Sistemas da Informação, da Área das Ciências Naturais, da Computação e das Engenharias, do Centro Universitário para o Desenvolvimento do Alto Vale do Itajaí- UNIDAVI, a ser apreciado pela Banca Examinadora, formada por:

Professor Orientador: M.e Jullian Hermann Creutzberg

Banca Examinadora:

Prof. M.e Fernando Andrade Bastos

Prof. Sandro Alencar Fernandes

Rio do Sul, 27 de novembro de 2023.

DEDICATÓRIA

Dedico especialmente a minha mãe, meu pai, minha vó e minha namorada que sempre me apoiaram em todos os momentos.

AGRADECIMENTOS

Agradeço primeiramente ao professor M.e Jullian Hermann Creutzberg pela compreensão e apoio durante o desenvolvimento do projeto em questão.

Aos professores do curso de Sistemas de Informação que auxiliaram no meu conhecimento e desenvolvimento da minha carreira.

Aos colegas da classe onde obtive diversas amizades ao longo destes quatro anos.

As demais pessoas que me ajudaram e me apoiaram nesse ciclo da minha vida.

RESUMO

Durante a pandemia e suas restrições, notou-se um aumento relevante da necessidade de serviços de *delivery*, principalmente com a criação de novos aplicativos para suprir a necessidade de entregas em diversos segmentos. Diante disto, também se percebeu o aumento de profissionais nesta área em todo país, gerando empregos para a profissão no ramo de *delivery*. Este trabalho tem como objetivo principal desenvolver um protótipo de uma aplicação web que possibilite lojistas gerenciarem aluguéis de veículos de duas rodas. Quanto a metodologia, é caracterizada como uma pesquisa aplicada e descritiva. Para atender todos os objetivos propostos, inicialmente foram realizadas pesquisas referentes as tecnologias e ferramentas viáveis para o desenvolvimento, destacadas no capítulo da revisão da literatura. Foram também levantadas ferramentas com funcionalidades e propósitos similares, que são apresentadas na seção do estado da arte. Antes do desenvolvimento do protótipo foram avaliadas todas as questões relacionadas ao negócio e ao funcionamento do protótipo, realizando o levantamento de requisitos funcionais, não funcionais, regras de negócio, além do desenho dos diagramas, mostrando uma parte mais ampla do funcionamento, sendo estes destacados na seção de análise. No capítulo de implementação são apresentados os detalhes referentes ao desenvolvimento e funcionamento do protótipo. Espera-se que ao utilizar o protótipo em questão, o lojista tenha a possibilidade de gerenciar a sua frota e definir quais veículos que serão disponibilizados para aluguéis. Além de cadastrá-los, o lojista ficará responsável pelas solicitações de reservas que os clientes fazem, podendo aceitar e assim gerando aluguéis. O lojista também ficará ciente dos lucros e gastos do negócio, através de gráficos e informações visuais. Outra vantagem do protótipo é que o cliente poderá ver quais veículos são disponibilizados pelas lojas, podendo escolher os melhores planos com os melhores preços. O principal benefício esperado é o potencial de gerar lucro para ambos, clientes e lojistas através do protótipo, incentivando ambas as partes a utilizarem o aluguel em seu dia a dia. O principal foco do protótipo é dar circulação a veículos parados em lojas e dar uma oportunidade a clientes de realizarem o aluguel destes veículos.

Palavras-Chave: Duociclo, desenvolvimento web, sistemas de informação.

ABSTRACT

During the pandemic and its restrictions, there was a significant increase in the need for delivery services, mainly with the creation of new applications to meet the need for deliveries in different segments. In view of this, an increase in professionals in this area across the country was also noticed, generating jobs for the profession in the delivery sector. This work's main objective is to develop a prototype of a web application that allows retailers to manage two-wheeled vehicle rentals. As for the methodology, it is characterized as applied and descriptive research. To meet all proposed objectives, research was initially carried out regarding viable technologies and tools for development, highlighted in the literature review chapter. Tools with similar functionalities and purposes were also identified, which are presented in the state of the art section. Before developing the prototype, all issues related to the business and the functioning of the prototype were evaluated, carrying out a survey of functional and non-functional requirements, business rules, in addition to drawing the diagrams, showing a broader part of the functioning, these being highlighted in the analysis section. The implementation chapter presents details regarding the development and operation of the prototype. It is expected that by using the prototype in question, the retailer will have the possibility of managing their fleet and defining which vehicles will be available for rental. In addition to registering them, the retailer will be responsible for booking requests made by customers, being able to accept them and thus generating rentals. The store owner will also be aware of the business's profits and expenses, through graphs and visual information. Another advantage of the prototype is that the customer will be able to see which vehicles are available in the stores, being able to choose the best plans at the best prices. The main expected benefit is the potential to generate profit for both customers and retailers through the prototype, encouraging both parties to use the rental in their daily lives. The main focus of the prototype is to circulate vehicles parked in stores and give customers an opportunity to rent these vehicles.

Keywords: Duociclo, web development, information systems.

LISTA DE FIGURAS

Figura 1 - A arquitetura MVC.....	19
Figura 2 - Loca9motos – Aplicação Web.....	38
Figura 3 - Benefícios – Loca9motos Aplicação Web	39
Figura 4 - Minha Conta – Loca9motos Aplicação Web	40
Figura 5 - Reserva Online – Roxmoto Aplicação Web	41
Figura 6 - Aluguel de Motos – Roxmoto Aplicação Web	41
Figura 7 - Informações do Tour – Roxmoto Aplicação Web.....	42
Figura 8 - Opções para Tour – Roxmoto Aplicação Web	42
Figura 9 - Localiza – Aplicação Web	43
Figura 10 - Opções de Acessórios na Reserva – Localiza Aplicação Web	44
Figura 11 - Opções de Carros – Localiza Aplicativo Mobile	45
Figura 12 - Diagrama de Caso de Uso - Cliente.....	55
Figura 13 - Diagrama de Caso de Uso - Lojista	56
Figura 14 - Diagrama de Caso de Uso - Administrador.....	57
Figura 15 - Diagrama de Atividade – Solicitação de Reserva (Cliente).....	58
Figura 16 - Diagrama de Atividade – Solicitação de Reserva (Lojista).....	59
Figura 17 - Modelagem Física – Banco de Dados	60
Figura 18 - Diagrama de Classes.....	61
Figura 19 - Login (RF01)	64
Figura 20 - Cadastre-se (RF03)	65
Figura 21 - Menu Cliente (RF07, RF08, RF09)	66
Figura 22 - Meu Perfil (RF04).....	66
Figura 23 - Consulta de Meus Aluguéis (RF05)	67
Figura 24 - Visualizar Aluguel (RF05)	68
Figura 25 - Consulta Reserva (RF06)	68
Figura 26 - Visualizar Solicitação de Reserva (RF06).....	69
Figura 27 - Veículos para Reserva (RF06).....	70
Figura 28 - Visualizar Loja do Veículo para Reserva (RF06)	70
Figura 29 - Visualizar Veículo disponível para Reserva (RF06).....	71
Figura 30 - Solicitar Reserva do Veículo (RF06).....	71
Figura 31 - Menu Lojista (RF02, RF10, RF18, RF19, RF20, RF21, RF22, RF23).....	72
Figura 32 - Rotina de Marcas (RF11).....	73

Figura 33 - Cadastro de Marcas (RF11).....	74
Figura 34 - Rotina de Veículos (RF12).....	74
Figura 35 - Cadastro de Veículos (RF12).....	75
Figura 36 - Rotina de Planos (RF17).....	76
Figura 37 - Cadastro de Planos (RF17)	76
Figura 38 - Rotina de Manutenções (RF14)	77
Figura 39 - Cadastro de Manutenções (RF14).....	78
Figura 40 - Rotina de Reservas (RF15)	78
Figura 41 - Visualização da Solicitação de Reserva (RF15)	79
Figura 42 - Rotina de Aluguéis (RF13).....	79
Figura 43 - Visualizar Aluguel (RF13)	80
Figura 44 - Consulta das Parcelas do Aluguel (RF16)	81
Figura 45 - Minha Loja (RF22)	81
Figura 46 - Meu Perfil (RF23).....	82
Figura 47 - Menu do Administrador (RF02, RF24, RF27, RF28, RF29, RF30)	83
Figura 48 - Rotina de Lojas (RF26)	84
Figura 49 - Cadastro da Loja (RF26).....	84
Figura 50 - Rotina de Usuários (RF25)	85
Figura 51 - Cadastro de Usuário (Lojista) (RF25)	86
Figura 52 - Tela Responsiva	86

LISTA DE QUADROS

Quadro 1 - Tags HTML	22
Quadro 2 - Seletores CSS.....	23
Quadro 3 - Eventos de Mouse.....	25
Quadro 4 - Operadores Lógicos	29
Quadro 5 - Tipos de Modelagem.....	34
Quadro 6 - Tabela de Preços (Planos) – Loca9motos	39
Quadro 7 - Comparativos dos Recursos do Duociclo com Estado da Arte	48
Quadro 8 - Requisitos Funcionais - Geral	48
Quadro 9 - Requisitos Funcionais - Cliente.....	48
Quadro 10 - Requisitos Funcionais - Lojista.....	49
Quadro 11 - Requisitos Funcionais - Administrador	51
Quadro 12 - Requisitos Funcionais Opcionais	52
Quadro 13 - Requisitos Não Funcionais.....	53
Quadro 14 - Regras de Negócio.....	54

SUMÁRIO

1. INTRODUÇÃO	13
1.1 PROBLEMA DE PESQUISA.....	14
1.2 OBJETIVOS.....	14
1.2.1 Geral	14
1.2.2 Específicos	14
1.3 JUSTIFICATIVA.....	14
2. REVISÃO DA LITERATURA	16
2.1 ENGENHARIA DE SOFTWARE	16
2.1.1 Levantamento de Requisitos	16
2.1.1.1 Requisitos Funcionais.....	17
2.1.1.2 Requisitos Não Funcionais	17
2.1.1.3 Regras de Negócio	18
2.1.2 Projeto de Arquitetura	18
2.1.2.1 MVC.....	19
2.1.3 Modelagem de Sistemas	20
2.2 HTML	21
2.3 CSS.....	23
2.3.1 Bootstrap	24
2.3.2 Materialize	25
2.4 JAVASCRIPT	25
2.4.1 JQuery	26
2.4.2 Ajax	27
2.4.3 Chart.js	27
2.5 PHP	28
2.5.1 Laravel	31
2.6 BANCO DE DADOS.....	33
2.6.1 Sistema de Gerenciamento de Banco de Dados (SGBD)	33
2.6.2 Modelagem de Dados	34
2.6.3 MySQL	35
3. METODOLOGIA DA PESQUISA	37
3.1 ESTADO DA ARTE	38
3.1.1 Loca9motos	38
3.1.2 Roxmoto	40
3.1.3 Localiza – Aluguel de Carros	43

4. DUOCICLO: PROTÓTIPO DE APLICAÇÃO WEB PARA GERENCIAMENTO DE ALUGUÉIS	46
4.1 ANÁLISE	46
4.1.1 Visão Geral do Protótipo	46
4.1.2 Comparação do Protótipo com o Estado da Arte	47
4.1.3 Requisitos	48
4.1.4 Diagramas	54
4.2 IMPLEMENTAÇÃO	62
4.2.1 Ferramentas e Técnicas Utilizadas	62
4.2.2 Utilização e Funcionamento	63
4.2.2.1 Rotinas Cliente	64
4.2.2.2 Rotinas Lojista	72
4.2.2.3 Rotinas Administrador	82
5. CONSIDERAÇÕES FINAIS	88
5.1 RECOMENDAÇÕES DE TRABALHOS FUTUROS.....	89
REFERÊNCIAS	92

1. INTRODUÇÃO

É notável o crescimento de estabelecimentos e softwares de *delivery* em todo o Brasil. Novos aplicativos são criados para facilitar a vida dos comerciantes e clientes, gerando cada vez mais emprego nesse ramo, através de novas tecnologias.

Segundo Faust (2021), o crescimento do *delivery* no Brasil tem sido evidente, especialmente impulsionado pela pandemia e pelas restrições impostas. Esse processo de expansão, que já estava em curso, acelerou significativamente. Como resultado, houve uma transformação completa no modo de operação da maioria das empresas gastronômicas do país, que agora oferecem opções além do tradicional *fast food*. Devido ao isolamento social causado pelo Covid-19, todos os tipos de restaurantes e lanchonetes tiveram que se adaptar e reestruturar seus serviços de entrega para sobreviver. Conseqüentemente, o mercado de *delivery*, que já era promissor, tornou-se a principal escolha tanto para consumidores quanto para empreendedores que desejam se manter no mercado.

Conforme afirma Stochero (2021) o Sindicato dos Mensageiros Motociclistas, Ciclistas e Mototaxistas do Estado de São Paulo (SindimotoSP) alega que, embora essencial, a categoria ainda não tem recebido a devida atenção por parte das autoridades e das empresas que contratam seus serviços. De acordo com a entidade, em março de 2020, quando a epidemia começou no estado, havia 220 mil profissionais autônomos na capital. Esse número aumentou para mais de 305 mil em fevereiro deste ano.

Pensando nisso, surgiu a proposta de desenvolver um protótipo de sistema visando disponibilizar para os comerciantes do ramo de veículos de duas rodas, como motocicletas, bicicletas, ciclomotores, patinetes, entre outros, a possibilidade de alugar suas frotas e utilizar uma ferramenta para gerenciar os aluguéis realizados. Com isso, tem-se como objetivo manter ambos os lados lucrando. O comerciante que irá disponibilizar os veículos, podendo lucrar através de frotas que ficarão paradas em seu estoque. E o cliente que caso tenha interesse, poderá entrar no ramo de entregas ou aquele que teve seu veículo pessoal avariado, poderá alugar para continuar trabalhando.

1.1 PROBLEMA DE PESQUISA

Como gerenciar de forma eficiente o aluguel de veículos de duas rodas na região do Alto Vale do Itajaí?

1.2 OBJETIVOS

1.2.1 Geral

- Desenvolver protótipo de uma aplicação web que possibilite lojistas gerenciarem aluguéis de veículos de duas rodas.

1.2.2 Específicos

- Detalhar ferramentas que serão empregadas no desenvolvimento do protótipo por meio de uma revisão da literatura;
- Demonstrar software similares ao protótipo por meio de um estado da arte;
- Definir a análise do protótipo por meio das regras de negócio, os requisitos, diagramas e modelagem de banco de dados;
- Construir o protótipo empregando as tecnologias escolhidas.

1.3 JUSTIFICATIVA

Pensando no crescimento dos serviços de *delivery*, o objetivo principal deste projeto é apresentar uma ideia às lojas que vendem veículos de duas rodas. Devido à pandemia, houve um aumento de 40% no número de profissionais de moto frete em um ano na cidade de São Paulo. Esse crescimento é resultado da maior demanda durante o período de isolamento social e do aumento de 78% no número de novos restaurantes cadastrados em plataformas de entrega. Em março de 2020, de acordo com a plataforma *iFood*, o número de entregadores ativos chegou a 170 mil. (STOCHERO, 2021).

O objetivo do protótipo é possibilitar o aluguel de veículos de duas rodas para pessoas que não têm condições de comprar ou que preferem não usar seu próprio veículo para o trabalho de entregas. As lojas que vendem esses veículos poderão gerenciar e disponibilizar a sua própria frota para aluguel, definir o valor diário do aluguel e cadastrar planos semanais e mensais para sua frota. O produto também fornecerá uma função de consulta, onde será possível verificar informações relevantes para o comerciante, como o número de aluguéis realizados e para quais veículos.

Com o protótipo e a ideia apresentada pela empresa, as lojas serão mais reconhecidas na cidade e pessoas interessadas em trabalhar no setor de entregas/*delivery* terão a oportunidade de atuar na área através do aluguel desses veículos de duas rodas. Além da venda dos veículos, os comerciantes desse ramo poderão lucrar com os aluguéis. Considerando o aumento do serviço de *delivery*, esta proposta é apresentada como inovadora para a região do Alto Vale do Itajaí.

2. REVISÃO DA LITERATURA

A revisão da literatura deste trabalho aborda os temas de Engenharia de Software, HTML, CSS, JavaScript, PHP e Banco de Dados. São explorados conceitos e metodologias relacionadas a engenharia de software, como levantamento de requisitos e definição das regras de negócio, a importância do banco de dados no desenvolvimento web, destacando o sistema de gerenciamento de banco de dados e modelagem de dados, os fundamentos do HTML, características do CSS, Bootstrap e Materialize, a linguagem JavaScript e suas bibliotecas JQuery e Chart.js, a linguagem de programação PHP e o *framework* Laravel.

2.1 ENGENHARIA DE SOFTWARE

Por anos a Engenharia de Software tem sido empregada na criação de software de acordo com os prazos estipulados pelo cliente, focando na qualidade do software e balanceando os custos e prazos, evitando os desperdícios de tempo, mão de obra e prevenindo riscos e atrasos. Contudo, a partir da definição da Organização do Tratado do Atlântico Norte (OTAN) sobre a Engenharia de Software, explicando que ela é um conjunto de metodologias para adequar o processo de desenvolvimento, que os profissionais e empresas de Tecnologia da Informação começaram a se preocupar com os setores de desenvolvimento de software, como a análise de requisitos, testes, análise de sistemas, desenvolvimento e implantação. (MORAIS, 2020).

2.1.1 Levantamento de Requisitos

O engenheiro de software é responsável por criar a documentação que descreve os requisitos e as regras de negócio, detalhando as necessidades e atributos do sistema. Além disso, é importante classificar os requisitos de acordo com sua prioridade (fundamental, significativo e opcional), para que o desenvolvimento do sistema relevância primeiramente aos requisitos essenciais para seu funcionamento. (MORAIS, 2020).

Morais (2020) explica que o documento de requisitos deve ser elaborado levando em consideração as especificações do software em desenvolvimento.

Documentos elaborados para uma determinada aplicação nunca serão aplicáveis em outro sistema pelo fato de ter suas próprias regras, metodologias, ferramentas e detalhes. No entanto, o que pode ser compartilhado entre diversos projetos são os princípios gerais da estrutura do documento.

A identificação dos requisitos na inicialização do projeto é de extrema importância para um desenvolvimento eficaz e produtivo da aplicação. Um requisito pode assumir diversas formas, como uma condição, uma capacidade, uma função, um objetivo, uma característica, ou uma limitação, todas destinadas a definir as características de um sistema e garantir o cumprimento das regras de negócio ou contratos estabelecidos. (MORAIS, 2020).

2.1.1.1 Requisitos Funcionais

Segundo Reinehr (2020), um requisito funcional, é uma funcionalidade necessária para atender a um objetivo de negócio definido pelos stakeholders. Ele descreve o que os desenvolvedores devem implementar para permitir que os usuários executem suas tarefas.

“Geralmente, os requisitos funcionais são expressos em frases do tipo “o sistema deve”. Por exemplo: “O sistema deve permitir que o usuário pague com cartão de débito ou crédito”.”. (REINEHR, 2020, p.35).

2.1.1.2 Requisitos Não Funcionais

Conforme Paula Filho (2019), os requisitos não funcionais englobam aspectos relacionados ao desempenho e outros atributos de qualidade do produto. Além disso, incluem requisitos lógicos relacionados aos dados persistentes e requisitos de natureza técnica, como limitações no design e na implementação. Os requisitos não funcionais devem ser descritos de maneira precisa e, apesar de estabelecer requisitos concretos na fase inicial do projeto, é importante que isso seja feito.

“A determinação correta dos requisitos não funcionais aplicáveis é essencial para o sucesso de um produto, pois eles costumam ter grande impacto na percepção que os usuários terão da qualidade e utilidade do produto.”. (PAULA FILHO, 2019, p.144).

Paula Filho (2019) diz que para sistemas grandes e importantes, os requisitos passam a influenciar a arquitetura ainda mais, sendo necessários métodos de formulação, conferência e análise cada vez mais aprimorados para a aplicação em questão.

2.1.1.3 Regras de Negócio

“Regras de negócio são regras que definem ou restringem aspectos dos processos de negócio. Elas são elementos do modelo de negócio, e devem ser levantadas durante a modelagem de negócio, se essa atividade for executada.”. (PAULA FILHO, 2019, p.139).

Paula Filho (2019) afirma que um modelo de requisitos deve incorporar todas as regras que sejam relevantes para o produto especificado. É importante a aplicação de sessões *workshop* e entrevistas de requisitos para que a definição das regras de negócio seja sempre clara e explícitas, evitando falta de detalhes delas. Por isso muitas vezes os usuários somente ficam cientes das regras de negócio após *feedbacks* de um protótipo ou até mesmo após o lançamento do produto.

“Regras de negócio são muitas vezes embutidas em fluxos de casos de uso ou na documentação de elementos de modelagem como classes, operações, relacionamentos, estados e atividades.”. (PAULA FILHO, 2019, p.143).

2.1.2 Projeto de Arquitetura

Pressman (2021) afirma que a arquitetura consiste em 'inúmeras decisões, tanto significativas quanto insignificantes'. Algumas dessas decisões são feitas logo no início do projeto e podem influenciar todas as etapas posteriores. Outras são adiadas o máximo possível, eliminando, assim, restrições que poderiam resultar em uma implementação inadequada do estilo arquitetural.

Um projeto de software pode ser visto como uma manifestação de uma arquitetura de software específica. No entanto, os elementos e estruturas estabelecidos como parte dessa arquitetura formam a base de todo o projeto. (PRESSMAN, 2021).

2.1.2.1 MVC

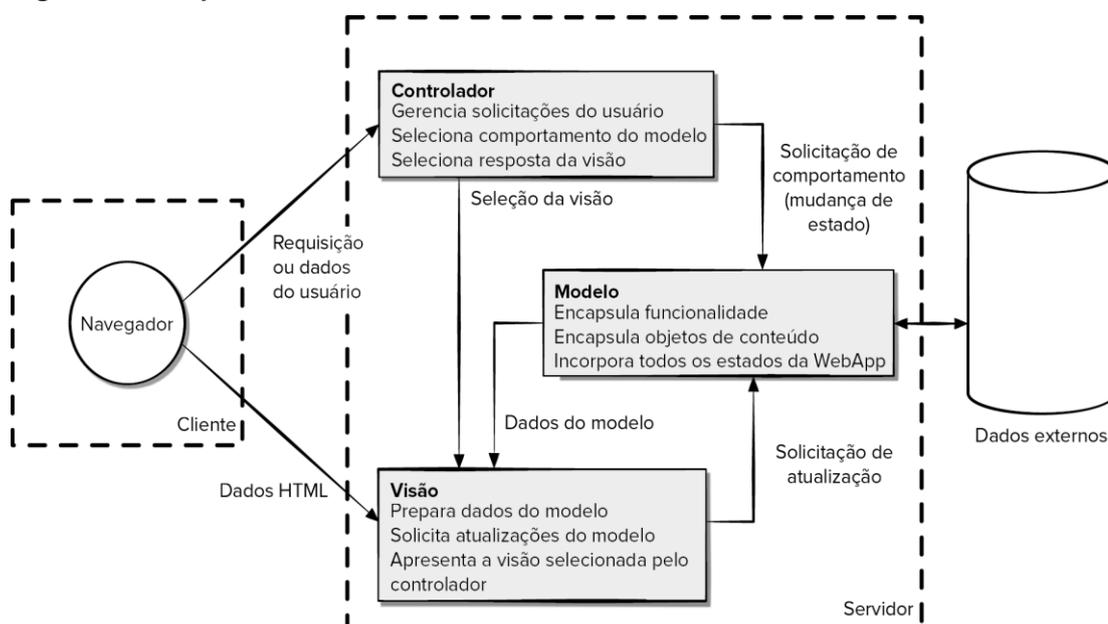
“A arquitetura Modelo-Visão-Controlador (MVC, do inglês *Model-View-Controller*) é uma de vários modelos de infraestrutura usados no desenvolvimento para a Web.”. (PRESSMAN, 2021, p.189).

O modelo contém todo o conteúdo e a lógica de processamento específicos à aplicação. A visão contém todas as funções específicas à interface e possibilita a apresentação do conteúdo e lógica de processamento exigido pelo usuário. O controlador gerencia o acesso ao modelo e à visão e coordena o fluxo de dados entre eles. (PRESSMAN, 2021, p.189).

Pressman (2021) afirma que a arquitetura MVC aplicada no projeto de WebApp4 pode ser compreendida como a fusão de duas estruturas de problemas distintas: o controle de comportamento e a apresentação de informações. No padrão MVC, a ação do usuário é encaminhada da janela do navegador para um processador de comandos (controlador), que supervisiona o acesso aos dados (modelo) e direciona as instruções de apresentação de informações (visão) para formatá-las de maneira apropriada para a exibição através do software do navegador.

Na Figura 1, uma representação da arquitetura MVC, segundo Pressman (2021).

Figura 1 - A arquitetura MVC



Fonte: Pressman (2021, p.191).

O controlador também seleciona o objeto de visualização adequado em resposta à solicitação do usuário. Uma vez determinado o tipo de solicitação, uma solicitação de ação é encaminhada ao modelo, que executa a funcionalidade necessária ou recupera os dados requeridos para atender à solicitação. O objeto-modelo pode acessar informações armazenadas em um banco de dados corporativo, seja parte de um repositório de dados local ou de um conjunto de arquivos independentes. Os dados gerados pelo modelo devem ser formatados e organizados pelo objeto de visualização apropriado e, em seguida, enviados do servidor de aplicativos de volta para o navegador instalado no cliente, onde serão exibidos na máquina do usuário. (PRESSMAN, 2021).

2.1.3 Modelagem de Sistemas

O modelo de requisitos deve cumprir três objetivos principais: (1) documentar as necessidades do cliente; (2) servir como fundamento para o desenvolvimento de um projeto de software; e (3) estabelecer um conjunto de requisitos que possam ser validados após a conclusão da construção do software. (PRESSMAN, 2021).

As duas primeiras tarefas da engenharia de requisitos – concepção e levantamento – fornecem as informações necessárias para começarmos a escrever casos de uso. As reuniões para levantamento de requisitos e outros mecanismos de engenharia de requisitos são utilizadas para identificar os envolvidos, definir o escopo do problema, especificar as metas operacionais gerais, estabelecer as prioridades, descrever todos os requisitos funcionais conhecidos e descrever os itens (objetos) manipulados pelo sistema. (PRESSMAN, 2021, p.131).

Segundo Pressman (2021), para iniciar a elaboração de um conjunto de casos de uso, é necessário listar as funcionalidades ou ações executadas por um ator específico. Essas informações podem ser derivadas de uma lista de funcionalidades presentes nos requisitos do sistema, obtidas por meio de diálogos com as partes interessadas, ou resultantes de uma análise de diagramas de atividades desenvolvidos como parte do processo de modelagem de requisitos.

“Podemos começar a identificar classes examinando os cenários de uso desenvolvidos como parte do modelo de requisitos e realizando uma “análise

sintática” dos casos de uso desenvolvidos para o sistema a ser construído.”. (PRESSMAN, 2021, p.137).

Pressman (2021) afirma que as classes são identificadas destacando cada substantivo ou frase nominal (que inclui substantivos) e registrando-os em uma tabela simples. É importante registrar também os sinônimos. Se uma classe (substantivo) precisa implementar uma solução, ela faz parte do domínio da solução; caso contrário, se uma classe é necessária apenas para descrever uma solução, ela faz parte do domínio do problema.

O modelo funcional lida com dois elementos de processamento da aplicação, cada um representando um nível diferente de abstração procedural: (1) funcionalidade observável pelo usuário, fornecida pela aplicação aos usuários; e (2) as operações contidas nas classes de análise que implementam comportamentos associados à classe. (PRESSMAN, 2021, p.145).

“O diagrama de sequência da UML pode ser usado para modelagem comportamental. Os diagramas de sequência também podem ser usados para mostrar como os eventos provocam transições de objeto para objeto.”. (PRESSMAN, 2021, p.147).

2.2 HTML

“Os documentos HTML são formados por textos puros, que podem ser criados até por aplicativos de edição de texto, como o Bloco de Notas do Windows. Eles são formados por um grupo de seções, sendo que cada uma delas é destinada a abrigar um tipo de conteúdo específico.”. (ALVES, 2021, p.7).

Assim como são necessárias linguagens de programação para o desenvolvimento de programas e aplicativos, também é necessária uma linguagem para criação de páginas de sites. Essa linguagem é denominada HTML (*HyperText Markup Language*, em português Linguagem de Marcação de Hipertexto). Porém, em vez de comandos para execução de cálculos ou processamento de dados, ela possui marcadores, conhecidos como *tags*, que são utilizados para formatar um texto, uma imagem ou qualquer outro objeto que faça parte da estrutura de um documento HTML. (ALVES, 2021, p.7).

“A HTML5 é o resultado da quinta maior revisão da linguagem HTML, considerada o novo padrão para Web, que une o que há de melhor na HTML versão

4, XHTML e HTML DOM, e adicionará um conjunto de novos recursos”. (TERUEL, 2014, p.17).

A força da HTML5 não está apenas nos novos elementos e atributos, mas também na possibilidade de acessá-los, manipulá-los e formatá-los com JavaScript e CSS utilizando o *Document Object Model* (DOM), que é a interface entre a linguagem JavaScript e os objetos/ elementos da linguagem HTML e que oferece uma maneira padrão para a construção de sites ricos e interativos com HTML, CSS e JavaScript. O DOM vê a página HTML como uma árvore de elementos e atributos hierarquizados que podem ser acessados com instruções JavaScript, normalmente a partir de eventos em elementos HTML, como cliques, mudanças, foco etc. (TERUEL, 2014, p.19).

Teruel (2014, p.19) afirma que “A linguagem HTML5 permite a utilização de uma série de APIs que ajudam na criação de aplicações web. Essas APIs podem ser acessadas por meio de scripts JavaScript e usadas em conjunto com os novos elementos HTML.”.

Miletto (2014) menciona as principais *tags* HTML, conforme apresentado no Quadro 1.

Quadro 1 - Tags HTML

html	Marca o início e o fim da página Web, informando ao navegador que o texto contido no documento está escrito em HTML.
head	Marca o início e o fim do cabeçalho, a área onde serão descritos os cabeçalhos e o título da página. Pode ser utilizado, ainda, para declarar scripts em JavaScript ou definir formatações CSS.
title	Marca o início e o fim do título da página, que sempre está posicionado na barra superior do browser.
body	Marca o início e o fim do corpo da página, que contém imagens, textos, títulos, links etc.

Fonte: Elaborado a partir de Miletto (2014).

“A HTML5 traz um conjunto de novos tipos de campos que fazem a validação automática dos dados de entrada ao submeter o formulário, sem a necessidade de criar complexos scripts de validação.”. (TERUEL, 2014, p.34).

Quando o navegador (*browser*) recebe do servidor a página HTML para ser exibida, ele utiliza a codificação de caracteres definida explicitamente na página, ou a codificação padrão do idioma do navegador do usuário (nos navegadores em português a codificação padrão é ISO- 8859-1). Essa codificação de caracteres determina como os caracteres suportados pelo navegador serão exibidos na tela. (TERUEL, 2014, p.22).

“Além de novos elementos e atributos, HTML5 não traz diversos atributos de estilização e formatação que são melhor utilizados com a linguagem CSS. Além disso, não traz elementos que comprometem a usabilidade e a acessibilidade.”. (TERUEL, 2014, p.33).

2.3 CSS

As folhas de estilo possibilitam o desenvolvedor personalizar sua aplicação, definindo tamanhos, cores, fontes em suas *tags* de títulos, listas, imagens, etc; dando uma aparência personalizada para sua aplicação web. (MILETTO, 2014, p.80).

As folhas de estilo em cascata (ou CSS – *Cascading Style Sheets*) mudam a forma de organização das páginas. O HTML passa a ser utilizado somente como elemento para estruturar as páginas, e o CSS é utilizado na formatação da aparência das páginas. Com o CSS, é possível definir em um único local a formatação que será utilizada por cada TAG. Com isso, apenas um arquivo é alterado, sendo que a mudança é automaticamente propagada a todas as páginas que compõem o site. (MILETTO, 2014, p.80).

Conforme afirma Miletto (2014, p.84), “No CSS, existem vários tipos de seletores que você pode usar para personalizar uma página: seletor de TAGs, seletor de classes, seletor de IDs, seletor de atributos, seletores pseudoclasse e pseudoelemento.”.

Miletto (2014), apresenta alguns seletores disponíveis no CSS, conforme Quadro 2.

Quadro 2 - Seletores CSS

Seletor	Descrição	Exemplo
De grupo	Utilizado para formatar um grupo de TAGs.	h1, h2, h3, h4, h5, h6 {color:#9cf;}
Universal	Utilizado para formatar todas as TAGs de uma página.	*{color:#9cf;}
Descendente	Utilizado para formatar uma TAG que está aninhada em outra TAG.	div a {background-color:#9cf;}

Fonte: Miletto (2014).

“Uma das principais propriedades CSS utilizadas na introdução de recursos mais visuais nas páginas Web é a *font-family*. Ela é utilizada para definir quais fontes

podem ser aplicadas a parágrafos, listas, cabeçalhos e outros elementos.”. (MILETTO, 2014, p.90).

As folhas de estilo em cascata (ou CSS – *Cascading Style Sheets*) mudam a forma de organização das páginas. O HTML passa a ser utilizado somente como elemento para estruturar as páginas, e o CSS é utilizado na formatação da aparência das páginas. Com o CSS, é possível definir em um único local a formatação que será utilizada por cada TAG. Com isso, apenas um arquivo é alterado, sendo que a mudança é automaticamente propagada a todas as páginas que compõem o site. (MILETO, 2014, p.70).

2.3.1 Bootstrap

Zabot (2020) afirma que o Bootstrap é um *framework front-end* de código aberto para aplicações web, tendo como base o HTML e CSS, sendo possível personalizar formulários, botões, tabelas, listas, barras de navegação, entre outros elementos.

O Bootstrap fornece elementos predefinidos em CSS e plug-ins JavaScript. Por isso, temos de incluir um arquivo CSS e um arquivo JavaScript em nosso código HTML. Além disso, o Bootstrap utiliza uma versão reduzida do jQuery e a biblioteca Popper2, de código aberto. O Bootstrap também permite o uso de um CDN ou fazer download da versão compilada ou do código-fonte. (ZABOT, 2020, p.102).

“É fácil de utilizar e, para começar, basta ter conhecimentos básicos sobre HTML E CSS. O Bootstrap disponibiliza recursos responsivos que se ajustam a telefones, *tablets* e *desktops* na abordagem *mobile-first*.”. (ZABOT, 2020, p.101).

O Bootstrap funciona com HTML5 e CSS3 e, por isso, requer a declaração de *doctype* HTML5 no início do documento HTML. Além disso, dado que o Bootstrap foi concebido na perspectiva *mobile-first* (ou seja, o código fica otimizado para dispositivos móveis e escalonável para dispositivos maiores), precisa estabelecer renderização e *touch* adequado a sistemas móveis utilizando o meta *tag viewport*. (ZABOT, 2020, p.102)

“O Bootstrap disponibiliza os *badges* (crachás), que são pequenas etiquetas que adicionam informação a um elemento. Nesse caso, estes serão utilizados com um elemento *span* dentro de botões.”. (ZABOT, 2020, p.113).

Zabot (2020) afirma que um aplicativo é frequentemente complementado por uma barra de navegação (*navbar*), que constitui uma parte essencial da interface

gráfica, auxiliando os usuários na obtenção de informações. O Bootstrap oferece a capacidade de desenvolver uma *navbar* simples, incluindo um título e um logotipo, bem como uma *navbar* com um menu responsivo.

2.3.2 Materialize

“Materialize é um *framework front-end* moderno baseado na filosofia de *Design Material* do Google que auxilia os desenvolvedores a construir e projetar sites imersivos.”. (PRABHU, 2018, p.7).

Prabhu (2018) afirma que o Materialize apresenta uma notável biblioteca de componentes de interface do usuário (UI) com um toque criativo, assegurando compatibilidade com diversos navegadores e a capacidade de desenvolver sites atraentes e consistentes, independentemente do dispositivo.

2.4 JAVASCRIPT

JavaScript é uma das linguagens mais utilizadas no desenvolvimento web na atualidade, se caracteriza pelo fato de ter uma tipagem dinâmica, baseada em objetos e uma porção de eventos (como movimentos do mouse, pressionar botão, arrastar e soltar etc.) que auxiliam nas páginas web. (MILETTO, 2014).

Miletto (2014), menciona eventos de mouse realizados através do JavaScript, conforme apresentado no Quadro 3.

Quadro 3 - Eventos de Mouse

Propriedade	Ocorre quando...
<i>onClick</i>	O usuário clica em um elemento.
<i>onDbClick</i>	O usuário clica duas vezes em um elemento.
<i>onMouseDown</i>	O usuário pressiona o botão do mouse sobre um elemento.
<i>onMouseMove</i>	O ponteiro do mouse está em movimento sobre o elemento.
<i>onMouseOver</i>	O ponteiro do mouse está sobre o elemento.
<i>onMouseOut</i>	O usuário move o ponteiro do mouse para fora do elemento.
<i>onMouseUp</i>	O usuário solta o botão do mouse sobre um elemento.

Fonte: Miletto (2014).

Linguagens para desenvolvimento de sites podem ser divididas em duas categorias básicas: as que rodam no lado cliente e as que rodam do lado servidor. As linguagens do lado cliente vistas até o momento (HTML, CSS e agora o JavaScript) são aquelas executadas utilizando apenas o navegador do computador do usuário. Uma vez carregadas, não necessitam de novas requisições ao servidor Web. (MILETTO, 2014, p. 106).

“É possível alterar a cor, a fonte e o tamanho por meio da combinação de CSS com JavaScript. Ou seja, ao invocar um evento do JavaScript, é possível acessar um elemento e, então, solicitar que seu estilo seja modificado.”. (MILETTO, 2014, p.132).

2.4.1 JQuery

A jQuery é uma biblioteca que define uma única função chamada `jQuery()`. Essa função é tão utilizada em aplicações que a biblioteca define “\$” como um atalho para essa função. (FLANAGAN, 2014).

A jQuery torna fácil encontrar os elementos importantes de um documento e então manipulá-los, adicionando conteúdo, editando atributos HTML e propriedades CSS, definindo rotinas de tratamento de evento e fazendo animações. Ela também tem utilitários Ajax para fazer requisições HTTP dinamicamente e funções utilitárias de uso geral para trabalhar com objetos e *arrays*. (FLANAGAN, 2014, p.524).

Flanagan (2014, p.526) afirma que “A função `jQuery()` (também conhecida como `$()`) é a mais importante da biblioteca jQuery. Contudo, ela é bastante sobrecarregada e existem quatro maneiras diferentes de chamá-la.”.

A função jQuery é o valor de `jQuery` ou de `$`. Essa é a função que cria objetos jQuery, registra rotinas de tratamento para serem chamadas quando o DOM estiver pronto e também serve como espaço de nomes da jQuery. Normalmente, me refiro a ela como `$()`. Já que serve como espaço de nomes, a função jQuery também poderia se chamar “o objeto global jQuery”, mas é muito importante não confundir isso com “um objeto jQuery”. (FLANAGAN, 2014, p.528).

“A biblioteca jQuery também usa a função `jQuery()` como espaço de nomes e define várias funções utilitárias e propriedades embaixo dela. A função `jQuery.noConflict()`, mencionada anteriormente, é uma função utilitária.”. (FLANAGAN, 2014, p.528).

2.4.2 Ajax

“O Ajax (*Asynchronous JavaScript and XML*) mudou a forma de interação dos usuários com os sistemas na Web. Antes dele, os sistemas eram baseados somente em formulários simples para entrada de dados.”. (MILETTO, 2014, p.223).

Miletto (2014) afirma que com o seu advento, tornou-se viável criar aplicações web que oferecessem um nível de interatividade comparável ao das aplicações desktop. Graças ao Ajax, atualmente é factível encontrar sistemas web complexos, como editores de texto, clientes de e-mail, planilhas eletrônicas, e muito mais.

Ajax não é uma linguagem de programação, mas uma maneira interessante de utilizar os padrões existentes na Web. Com o Ajax, é possível enviar e receber dados de um servidor Web de forma assíncrona (em segundo plano). Esse método permite, enquanto uma requisição é processada no servidor, não interferir na interação do usuário com a página. Além disso, possibilita a atualização de partes de uma página sem a necessidade de recarregar todo o documento HTML. (MILETTO, 2014, p. 224).

Miletto (2014), diz que a incorporação do Ajax traz consigo diversas vantagens significativas. Ao possibilitar a atualização de seções específicas de um documento HTML, há uma redução na quantidade de dados transmitidos entre o cliente e o servidor, resultando em tempos de resposta mais rápidos e melhorando o desempenho na construção da interface web para o usuário.

2.4.3 Chart.js

O Chart.js oferece uma variedade de tipos de gráficos, plugins e opções de personalização frequentemente utilizados. Além dos tipos de gráficos incorporados que estão disponíveis, também pode aproveitar tipos de gráficos adicionais mantidos pela comunidade. Além disso, é possível combinar vários tipos de gráficos em um único gráfico misto, permitindo a fusão de diversos tipos de gráficos em uma única tela. (CHART.JS, 2023, n.p.).

“Entre muitas bibliotecas de gráficos para desenvolvedores de aplicativos JavaScript, Chart.js é atualmente a mais popular de acordo com estrelas do GitHub (~60.000) e downloads npm (~2.400.000 semanalmente).” (CHART.JS, 2023, n.p.).

Segundo Chart.js (2023, n.p.), “Chart.js foi criado e anunciado em 2013, mas já percorreu um longo caminho desde então. É de código aberto, licenciado sob a licença muito permissiva do MIT e mantido por uma comunidade ativa”.

2.5 PHP

Pode-se afirmar que “Um dos destaques do PHP está na sua apresentação em conjunto com as marcações de linguagem HTML, o que possibilita a adição de dinamicidade às páginas desenvolvidas nessa linguagem.”. (MILETTO, 2014, p. 172).

O PHP é uma linguagem criada por Ramus Ledorf, em 1995, que explora a criação de *scripts* que são normalmente interpretados em um servidor Web no qual esses *scripts* estejam armazenados. O pré-requisito para que isso ocorra é que o servidor tenha o interpretador PHP devidamente configurado. No entanto, *scripts* PHP também podem ser executados localmente via linha de comando, mediante a presença de um interpretador. (MILETTO, 2014, p.172).

“A interpretação dos *scripts* PHP possibilita a geração de códigos HTML, JavaScript, além de documentos PDF, XML, imagens ou textos, os quais podem ser enviados ao cliente ou simplesmente armazenados no servidor.”. (MILETTO, 2014, p.172).

Miletto (2014) diz que para obter a configuração necessária para executar o PHP em um computador, uma opção é instalar pacotes AMP (sigla para Apache, MySQL e PHP) pré-configurados, que incluem conjuntos de servidores web, PHP e sistemas de gerenciamento de banco de dados. A escolha do pacote adequado dependerá do contexto (plataforma) do computador em que o servidor será instalado.

PHP não requer que as variáveis sejam declaradas antes de sua utilização, fato que a difere de outras linguagens tradicionais. Diante disso, uma variável será criada na primeira vez que um valor for atribuído a ela. Uma variável também não precisa ser inicializada no PHP, mas essa prática pode ser recomendável em situações em que desejarmos ter a garantia de que essa variável não foi utilizada previamente. (MILETTO, 2014, p.175).

Miletto (2014), menciona os operadores lógicos na linguagem PHP, conforme apresentado no Quadro 4.

Quadro 4 - Operadores Lógicos

Exemplo	Nome	Resultado
$\$a \text{ and } \b	E	Verdadeiro (<i>true</i>) se tanto $\$a$ quanto $\$b$ são verdadeiros.
$\$a \text{ or } \b	OU	Verdadeiro se $\$a$ ou $\$b$ são verdadeiros.
$\$a \text{ xor } \b	XOR	Verdadeiro se $\$a$ ou $\$b$ são verdadeiros, mas não ambos.
$!\$a$	NÃO	Verdadeiro se $\$a$ não é verdadeiro.
$\$a \ \&\& \ \b	E	Verdadeiro se tanto $\$a$ quanto $\$b$ são verdadeiros.
$\$a \ \ \b	OU	Verdadeiro se $\$a$ ou $\$b$ são verdadeiros.

Fonte: Elaborado a partir de Miletto (2014).

“O PHP não suporta sobrecarga de funções, e também não permite cancelar ou alterar a definição de funções previamente declaradas. Os nomes de funções são *case-insensitive*.”. (MILETTO, 2014, p. 186).

No PHP, assim como em grande parte das linguagens de programação, as estruturas de controle são utilizadas para agrupar conjuntos de instruções que devem ser executadas dada uma determinada condição da execução do *script* ou um determinado número de vezes. Assim, as estruturas de controle são divididas em dois grupos: estruturas condicionais e estruturas de repetição. (MILETTO, 2014, p.181).

Conforme diz Miletto (2014), ao criar funções personalizadas, é interessante pré-definir os valores de alguns dos parâmetros. Quando isso ocorre, é possível fazer uma chamada à função sem passar esse argumento, utilizando o valor pré-definido em sua declaração. A invocação com todos os parâmetros também continua funcionando, utilizando o valor informado na chamada da função, em vez do valor pré-definido.

Dentre todos esses bancos de dados, o mais utilizado é o MySQL (em seguida temos o PostgreSQL). E para esse gerenciador o PHP dispõe de duas bibliotecas para acesso ao banco de dados. A primeira e mais antiga, conhecida simplesmente como *mysql*, provê o suporte a versões do MySQL inferiores à versão 4.1.3, porém funciona também com as versões mais recentes, mas sem todas as funcionalidades disponíveis nessas versões. (SOARES, 2013, p.409).

“Uma das mais importantes mudanças no PHP, e provavelmente a principal, foi a total remodelação da linguagem para uma melhor aderência ao conceito de OO (Orientação a Objetos).” (SOARES, 2013, p.291).

No PHP existem três grupos de operadores. São eles os operadores unários, os quais como o nome sugere, manipulam um único valor, por exemplo os operadores! (negação) e ++ (incremento). Existem ainda os

operadores binários, que manipulam dois valores ou expressões e retornam um valor, por exemplo OR (ou) e >= (maior ou igual), sendo a maioria dos operadores no PHP desta categoria. Finalmente existe o operador ternário, o qual retorna um valor entre dois possíveis, conforme o resultado de um terceiro valor ou expressão. (SOARES, 2013, p.78).

Soares (2013) diz que inicialmente concebido para o desenvolvimento de sites dinâmicos, o PHP tinha como foco principal a integração com HTML e bancos de dados. No entanto, ao longo do tempo, foram adicionadas funcionalidades adicionais para aprimorar a potência e a abrangência da linguagem. Entre essas funcionalidades, destaca-se a capacidade de manipular imagens de diversos formatos, como JPEG, PNG e TIFF.

A partir da versão 5.3, o PHP disponibiliza o recurso de *namespace*, o qual permite ao desenvolvedor diferenciar sua biblioteca das desenvolvidas por outros e até mesmo do pacote principal do PHP. Com esse novo recurso não precisamos mais nos preocupar se o nome de uma ou mais classes que criarmos está repetido dentro de todas as bibliotecas carregadas na execução do sistema. Com *namespace* precisamos apenas definir uma assinatura (um prefixo) para as classes (antes desse recurso era possível criar nomes únicos, mas em geral as classes ficavam com nomes extensos e complicados) e dentro do escopo do *namespace* não teremos mais problemas de duplicidade de nomes (é lógico que o nome do *namespace* deve ser único; caso contrário, teremos o mesmo problema). (SOARES, 2013, p.394).

Um dos pontos importantes da linguagem PHP é a sua comunicação fácil e simples com o HTML, sendo possível realizar *websites* dinâmicos e responsivos com facilidade e rapidez. Estes *websites* dinâmicos têm como característica a interação com os usuários, através de formulários, cadastros, buscas, etc. (SOARES, 2013, p.117).

Para que os programas fiquem organizados e fáceis de manter - e até mesmo para que seja possível implementar a lógica definida para o programa, precisamos, muitas vezes, executar partes de códigos em vários pontos do *script*. Caso a linguagem não disponibilize uma forma de reaproveitamento desses pedaços de código, geralmente com uma tarefa específica, teremos de repeti-los sempre que necessário. O PHP resolve isso com funções. Uma função é um subprograma que executa uma série de instruções e pode retornar ou não um valor como resultado. (SOARES, 2013, p.101).

“Toda linguagem, e o PHP não está fora desta categoria, necessita de mecanismos para controlar o fluxo do programa, sem os quais seria impossível (ou

praticamente) implementar uma lógica qualquer, seja ela simples ou complexa.”. (SOARES, 2013, p.87).

“No PHP *arrays* são mapas ordenados de chaves e valores, ou seja, você pode atribuir a um elemento do *array* uma chave e um valor. Desta forma é possível representar listas, *hashtables*, coleções, dicionários, pilhas, filas, etc.”. (SOARES, 2013, p.53).

Segundo Soares (2013, p.56), “Recursos são variáveis especiais no PHP, as quais referenciam recursos externos ao PHP. Os recursos são criados e utilizados por funções especiais no PHP, por exemplo: manipulação de bancos de dados, imagens, arquivos,etc.”.

2.5.1 Laravel

Cardoso (2021) diz que Laravel surgiu em 2011, conforme necessidade de um *framework* mais completo para linguagem PHP em aplicações web, disponibilizando recursos para o desenvolvimento, além de um ponto de partida para criação do software.

Essa plataforma proporciona para o desenvolvedor executar o aplicativo em escala, quase ilimitada na mais recente tecnologia sem servidor da *Amazon Web Services (AWS)*, plataforma de serviços de computação em nuvem. Outra vantagem do Laravel é combinar os melhores pacotes do ecossistema PHP, com objetivo de proporcionar o *framework* robusto e *friendly*, ou seja, considerado amigável para os desenvolvedores e com diversas opções para execução e criação de projetos. (CARDOSO, 2021, p. 39).

“Ele dispõe de diversas ferramentas e estruturas para o desenvolvimento de aplicativos da web, facilitando a criação de aplicativos robustos e moderno utilizando os recursos Laravel, que possui uma extensa biblioteca de documentação.”. (CARDOSO, 2021, p.38).

Para Cardoso (2021, p.46), “Todos os arquivos de configuração do *framework* Laravel são armazenados na pasta config, em que cada opção é documentada. Dessa maneira, é importante examinar os arquivos e se familiarizar com as opções disponíveis.”.

Cardoso (2021) diz que é fascinante como o Laravel pode desempenhar o papel de um *framework full stack*. Para uma melhor compreensão do termo "*full*

stack", refere-se à capacidade do Laravel de lidar com o roteamento de requisições e a construção da aplicação em desenvolvimento. Além disso, o Laravel oferece opções para renderizar o *frontend* do projeto, seja através de *templates* Blade ou aproveitando uma tecnologia híbrida de aplicação de página única, como o Inertia.js.

[...] o Laravel dispõe de ferramentas completas que permitem de serem configuradas, ou seja, personalizadas para os diversos projetos. O Laravel é escalonável devido as suas características de escalonamento do PHP, ao suporte embutido para sistemas de cache rápido e distribuído como o Redis, facilitando o escalonamento horizontal. (CARDOSO, 2021, p.38).

“Os projetos podem ser desenvolvidos no próprio computador do desenvolvedor ou utilizar os recursos como a que recebe o nome de Sail, que se trata de uma solução embutida para executar projetos Laravel usando o Docker.”. (CARDOSO, 2021, p.39).

Uma das maneiras é cercar-se das melhores ferramentas possíveis. Aí entra o Laravel. Entregar projetos de qualidade em tempo reduzido minimizando a escrita manual de código de forma organizada para que uma equipe possa trabalhar em conjunto, testar as soluções implementadas e dar manutenção em tempo hábil quando necessário. De uma vasta lista de *frameworks* PHP, o Laravel recebeu grande atenção nos anos recentes. Atualmente, é um dos *frameworks* PHP mais populares. (GABARDO, 2017, p.16).

Gabardo (2017, p.17) afirma que “O Laravel tem boa documentação e comunidade bastante ativa. Esses entre outros vários fatores fazem do Laravel uma ótima escolha para desenvolvimento web.”.

Laravel Sail é uma interface de linha de comando que se destaca pela sua simplicidade. Seu principal propósito é facilitar a interação com a configuração padrão do Docker do Laravel. Com o Laravel Sail, é possível iniciar rapidamente o desenvolvimento de aplicativos Laravel, utilizando Redis, PHP e MySQL, sem a necessidade de possuir conhecimentos avançados em Docker. Ele serve como um excelente ponto de partida para construir seus projetos Laravel de forma descomplicada. (CARDOSO, 2021).

2.6 BANCO DE DADOS

“Os bancos de dados surgiram principalmente em função da grande necessidade de armazenar dados de forma permanente e organizada.”. (PICHETTI, 2021, p.11).

“Um sistema de banco de dados não teria muita utilidade se não fosse possível localizar e recuperar de forma rápida as informações que desejamos.”. (ALVES, 2020, p.42).

Praticidade, eficiência, rapidez na busca e confiabilidade das informações foram os principais motivos que impulsionaram o desenvolvimento de bancos de dados informatizados. Porém, atualmente ainda é possível encontrar várias formas de bancos de dados não informatizados. (ALVES, 2014).

Conforme explica Alves (2014), sistemas de gerenciamento de bancos de dados podem ser considerados entre os mais antigos desenvolvidos para computadores. Antes da criação desse tipo de software aplicativo, os programas operavam com arquivos sequenciais, fazendo uso dos recursos fornecidos pelo sistema operacional para armazenar e recuperar dados em disco.

2.6.1 Sistema de Gerenciamento de Banco de Dados (SGBD)

Para Alves (2020), um sistema de banco de dados não seria muito útil se não fosse possível acessar os dados de forma rápida, conforme a necessidade. Vários métodos foram criados para agilizar as operações de busca, durante anos conforme evolução, proporcionando algoritmos eficientes, nos ajudando a facilitar o dia a dia.

“Um SGBD é um sistema de software de uso geral, que facilita o processo de definição, construção, manipulação e compartilhamento de dados, simplificando a experiência do usuário.”. (PICHETTI, 2021, p.12).

Com o surgimento dos sistemas de gerenciamentos de banco de dados (SGBDs), tornou-se possível obter diversas informações importantes que auxiliam na tomada de decisão em questão de minutos, auxiliando para a tomar a melhor decisão em determinado assunto. (PICHETTI, 2021).

2.6.2 Modelagem de Dados

A modelagem de dados é um agrupamento de conceitos que consiste em estruturas lógicas e físicas de um banco de dados. Uma das principais abordagens da modelagem do banco de dados é que ele apresenta informações relevantes ao usuário final, construindo e organizando os dados de determinado contexto. (MACHADO, 2020).

Pichetti (2021), menciona três tipos de modelagem: modelagem conceitual, modelagem lógica e modelagem física, conforme apresentado no Quadro 5.

Quadro 5 - Tipos de Modelagem

Modelagem Conceitual	A modelagem conceitual é a representação que considera exclusivamente o ponto de vista do usuário criador dos dados, levando em consideração fatores técnicos para sua implementação. O nível conceitual especifica como os dados são armazenados e relacionados, independentemente de como serão implementados no banco de dados.
Modelagem Lógica (representativa ou de implementação)	O modelo lógico só deve ser inicializado após a conclusão do modelo conceitual. Diferentemente do modelo conceitual, o modelo lógico será criado com base em um tipo de banco de dados, como SQL Server, Oracle, MySQL, dentre outros.
Modelagem física (baixo nível)	O modelo físico é concebido por meio do modelo lógico. É nesse modelo que serão definidos os tipos de dados que serão armazenados, e ocorre a implementação da estrutura lógica em um sistema gerenciador de banco de dados (SGBD), que administra fisicamente os dados armazenados.

Fonte: Elaborado a partir de Pichetti (2021).

Para Machado (2020), a refatoração é um dos métodos ágeis, mantendo um foco no projeto e nas funcionalidades de cada versão dele, sem prever possíveis necessidades futuras. É pensando nos métodos ágeis que devemos nos concentrar no desenvolvimento do projeto de banco de dados.

Analisando o impacto sobre a implantação dos bancos de dados, observamos que refatorar qualquer estrutura de tabelas de um deles é uma tarefa complexa e que pode, caso exista uma administração de dados independente do desenvolvimento com os métodos ágeis, criar um ponto de perda de produtividade dos ganhos obtidos com a aplicação desses métodos. (MACHADO, 2020, p.333).

“As estruturas hierárquicas são aparentemente mais complexas de serem resolvidas, com Diagramas Entidade-Relacionamento (DER) e em ambientes relacionais, ou seja, bancos de dados relacionais.”. (MACHADO, 2020, p.213).

Machado (2020, p.337) afirma que “Administração de Dados é a função responsável por administrar estratégias, práticas e procedimentos para o processo de gerência dos dados de uma organização, seus ativos digitais.”

Conforme Machado (2020), o processo de administração de dados tem com intuito disseminar a existência e o significado dos dados, independente da área da organização. Isso permite que administradores da empresa tenham uma visão ampla para definir um passo a passo referente ao projeto.

“É fundamental para o projetista de BD que ele possua capacitação na navegação do banco de dados proposto e implementado e, principalmente, compreensão do modelo, conhecendo seus caminhos de navegação.”. (MACHADO, 2020, p.26).

2.6.3 MySQL

“O programa MySQL é um sistema de gerenciamento de banco de dados relacional que utiliza a linguagem de consulta estruturada SQL como interface de acesso e extração de informações do banco de dados em uso.”. (MANZANO, 2011, p.21).

Como curiosidade, o logotipo do gerenciador de banco de dados MySQL é a imagem de um golfinho de nome *Sakila* que, segundo os desenvolvedores do programa, representa a imagem dos valores da empresa MySQL AB, e também a imagem do sistema de gerenciamento de banco de dados em si, uma vez que o programa MySQL é uma ferramenta rápida, precisa, com grande potencial e facilidade de uso. (MANZANO, 2011, p.23).

“O MySQL é um dos sistemas de gerenciamento de bancos de dados mais populares e usados no mundo. É rápido, multitarefa e multiusuário.”. (MANZANO, 2011, p.21).

O nome do produto MySQL sugere que a linguagem de consulta usada pelo servidor de banco de dados é a SQL. O programa MySQL segue boa parte dos padrões SQL normalizados e propostos segundo a entidade ANSI, mas também acrescenta outros recursos particulares que não são previstos segundo o padrão ANSI SQL. A versão 5.1 dá suporte ao padrão ANSI SQL3 e possui a capacidade de uso de *triggers* e *stored procedures*. (MANZANO, 2011, p.23).

Conforme afirma Manzano (2011, p.22), “São muitas as empresas que usam o produto MySQL em aplicações críticas. Entre elas podem ser destacadas NASA, Silicon Graphics, Motorola, Texas Instruments, Yahoo! Finance e MP3.com.”.

O MySQL a cada dia torna-se um produto apreciado por várias empresas, entidades e pessoas, pois possui um servidor confiável, rápido e de fácil utilização, que pode ser utilizado com grandes bancos de dados, considerando inclusive aplicações voltadas para a Internet. Aliás, parte de seu sucesso é devido à fácil integração com a linguagem de script PHP. (MANZANO, 2011, p.22).

“Função é um recurso (ferramenta auxiliar) que facilita muitas tarefas corriqueiras do programa MySQL, que disponibiliza diversas funções divididas em grupos de características operacionais.”. (MANZANO, 2011, p.106).

3. METODOLOGIA DA PESQUISA

O presente trabalho caracteriza-se como do tipo aplicada e descritiva, pois ele tem o objetivo de desenvolver uma aplicação web que possibilite comerciantes gerenciarem aluguéis de veículos de duas rodas.

A ideia busca responder as seguintes questões: É possível realizar aluguel de veículos de duas rodas no Alto Vale do Itajaí? O que o proprietário de uma loja de motocicletas usadas faz com as motos que estão paradas no seu estoque? Se a motocicleta de um entregador tem algum problema, é possível que ele alugue outro veículo para trabalhar?

O desenvolvimento deste protótipo também tem como objetivo auxiliar os comerciantes de veículos de duas rodas usadas que buscam inovação no comércio. O objetivo é permitir ao comerciante alugar sua frota e gerenciar seus aluguéis. Com a implementação do protótipo e a ideia inovadora, a loja do comerciante recebe uma atenção maior na sua região de atuação.

Para o trabalho proposto, foram realizadas pesquisas em tecnologias para o desenvolvimento dele através da revisão da literatura, aprofundando cada conceito e detalhando cada uma das tecnologias.

Além da pesquisa sobre as tecnologias para construção do protótipo, foram realizadas pesquisas em aplicações similares ao protótipo proposto neste trabalho. Os detalhes referentes as ferramentas encontradas são destacadas no Estado da Arte.

Para o desenvolvimento do protótipo as ferramentas que auxiliaram foram o Visual Studio Code para utilização do *framework back-end* Laravel, a linguagem PHP, utilização de HTML e CSS com o *framework front-end* Bootstrap, implementação do JavaScript e sua respectiva biblioteca jQuery, além da implementação da arquitetura MVC. O MySQL foi utilizado para definição e gerenciamento do Banco de Dados. Para o levantamento de requisitos através de diagramas de caso de uso, diagramas de atividade e diagramas de classe foi utilizado o Draw.io, desenhando o funcionamento do protótipo. Para o gerenciamento entre servidor, aplicação e banco de dados foi utilizado a ferramenta Laragon.

3.1 ESTADO DA ARTE

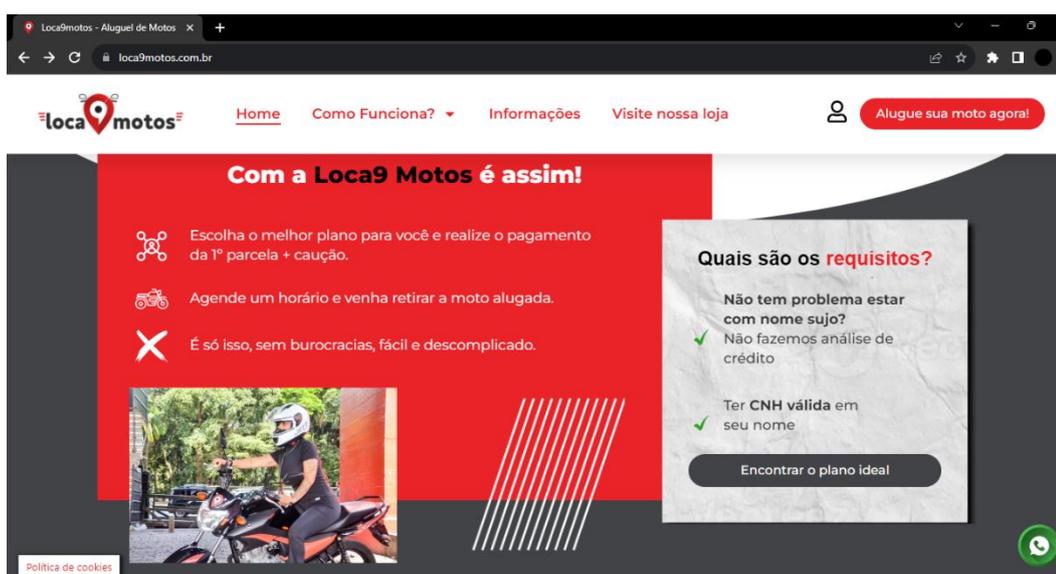
Neste tópico são apresentadas três ferramentas disponíveis no mercado semelhantes ao protótipo proposto neste trabalho. São demonstradas as principais funções e objetivos de cada aplicação visando também subsidiar a análise dos requisitos para a construção do protótipo.

Entre as três aplicações pesquisadas, as aplicações Loca9motos e Roxmoto disponibilizam recursos para aluguel de motocicletas da sua própria frota para o cliente final. Já a aplicação da empresa Localiza disponibiliza apenas o aluguel de carros de sua própria frota. Até o momento não foram encontradas ferramentas que possibilite um comerciante escolher quais frotas da sua loja deseja disponibilizar para aluguel, conforme o protótipo proposto neste trabalho, tornando-o assim um diferencial no mercado.

3.1.1 Loca9motos

O *web site* do Loca9motos (Figura 2), disponibiliza informações referente a aluguel de motocicletas. A aplicação permite o cliente solicitar uma motocicleta para aluguel, conforme planos oferecidos pela empresa através do WhatsApp.

Figura 2 - Loca9motos – Aplicação Web



Fonte: Loca9motos (2022).

Conforme Figura 3, a Loca9motos em seu *web site* apresenta alguns benefícios como Suporte Personalizado para emergência, Atendimento Emergencial, Proteção e Segurança, Quilômetros Livre, Agentes Dedicados e Revisões Preventivas. (LOCA9MOTOS, 2022).

Figura 3 - Benefícios – Loca9motos Aplicação Web



Fonte: Loca9motos (2022).

Os planos (Quadro 6) e solicitações de aluguel, bem como consulta de multas e consulta a financeiro são disponibilizados apenas pelo WhatsApp.

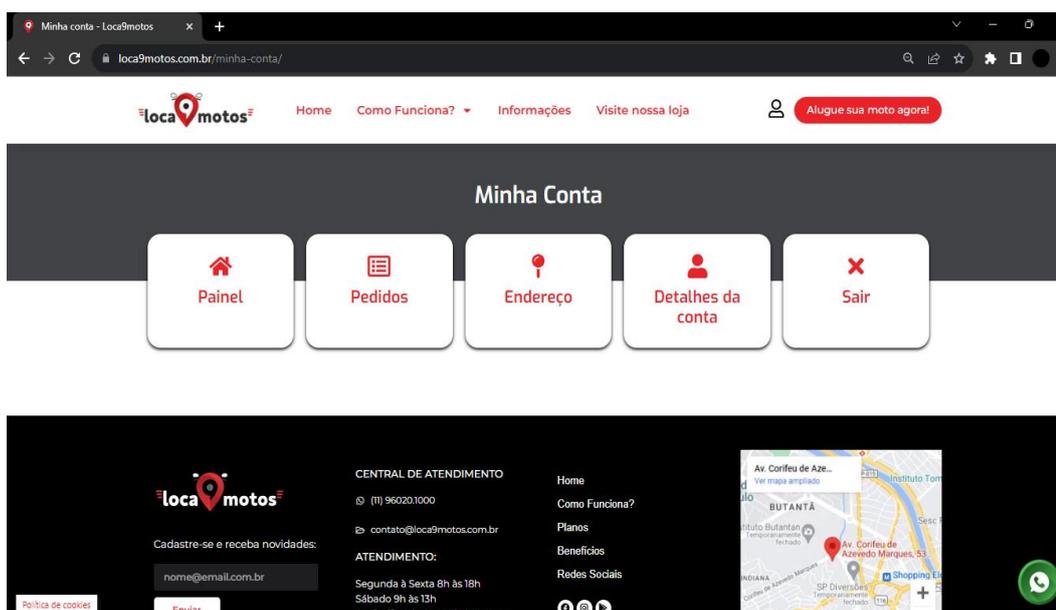
Quadro 6 - Tabela de Preços (Planos) – Loca9motos

	Mensal	Semestral	Anual
Valor Diária	R\$40,00	R\$37,14	R\$35,00
Valor Semanal	R\$280,00	R\$260,00	R\$245,00
Valor Caução	R\$790,00	R\$790,00	R\$790,00

Fonte: Loca9motos (2023).

Além das informações na página inicial, a aplicação disponibiliza a opção “Minha Conta”, conforme Figura 4, onde traz algumas opções para o usuário sobre informações referente aos pedidos, endereços e detalhes da conta.

Figura 4 - Minha Conta – Loca9motos Aplicação Web



Fonte: Loca9motos (2023).

Porém, ao selecionar alguma das opções, somente aparece “Página não encontrada”. A aplicação não disponibiliza exatamente um cadastro e um login para o usuário. Basicamente o site fica limitado, trazendo apenas informações da empresa, porém os veículos e planos para os aluguéis são disponibilizados somente via WhatsApp.

3.1.2 Roxmoto

A Roxmoto é uma empresa que disponibiliza motocicletas de alta cilindrada para aluguel, conforme informações disponibilizadas pelo seu web site. Além da possibilidade de aluguel, é possível verificar *tours* realizados pela empresa, o usuário podendo avaliar preços e categorias diferentes, conforme disponibilidade da empresa no seu web site.

Nesta aplicação é possível realizar o aluguel de motocicletas de alta cilindrada, indicadas para viagens de semanas. Além disso, é possível realizar uma busca das opções de acordo com o local de retirada, data e hora de retirada, local de devolução e data e hora de devolução do veículo (Figura 5).

Figura 5 - Reserva Online – Roxmoto Aplicação Web

Fonte: Roxmoto (2023).

No site também é possível consultar qual o custo de cada veículo de acordo com a quilometragem que o cliente deseja realizar na sua reserva. No tópico de aluguel de motos é possível ver as informações de cada veículo que é disponibilizado pela empresa para aluguel. Conforme na Figura 6, o site traz informações como o preço da diária de acordo com a quilometragem que o cliente irá fazer, altura do assento, torque, potência, motor e os preços da diária de acordo com os dias da semana.

Figura 6 - Aluguel de Motos – Roxmoto Aplicação Web

HONDA CB 500X	HONDA CB 500F	Triumph Bonneville T100	Triumph Street Scrambler 900
<p>Motor: 471 cc</p> <p>Potência: 50,2 cv @ 8.500 rpm</p> <p>Torque: 4,54 kgf.m @ 7.000 rpm</p> <p>Altura do assento: 834 mm</p> <p>1 Diária (250 km): R\$ 331,00</p> <p>1 Diária (350 km): R\$ 414,00</p> <p>Final de semana (sáb. a seg.): a partir de R\$ 662,00</p> <p>Final de semana (sex. a seg.): a partir de R\$ 854,00</p> <p>Caução (pré-autorização): R\$ 4.000,00</p> <p>Km excedente: R\$ 1,30</p>	<p>Motor: 471 cc</p> <p>Potência: 50,2 cv @ 8.500 rpm</p> <p>Torque: 4,54 kgf.m @ 7.000 rpm</p> <p>Altura do assento: 780 mm</p> <p>1 Diária (250 km): R\$ 331,00</p> <p>1 Diária (350 km): R\$ 414,00</p> <p>Final de semana (sáb. a seg.): a partir de R\$ 662,00</p> <p>Final de semana (sex. a seg.): a partir de R\$ 854,00</p> <p>Caução (pré-autorização): R\$ 4.000,00</p> <p>Km excedente: R\$ 1,30</p>	<p>Motor: 900 cc</p> <p>Potência: 65 cv @ 7.400 rpm</p> <p>Torque: 80 Nm @ 3.750 rpm</p> <p>Altura do assento: 790 mm</p> <p>1 Diária (250 km): R\$ 419,00</p> <p>1 Diária (350 km): R\$ 523,00</p> <p>Final de semana (sáb. a seg.): a partir de R\$ 837,00</p> <p>Final de semana (sex. a seg.): a partir de R\$ 1.092,00</p> <p>Caução (pré-autorização): R\$ 4.500,00</p> <p>Km excedente: R\$ 1,30</p>	<p>Motor: 900 cc</p> <p>Potência: 65 cv @ 7.250 rpm</p> <p>Torque: 80 Nm @ 3.250 rpm</p> <p>Altura do assento: 750 mm</p> <p>1 Diária (250 km): R\$ 469,00</p> <p>1 Diária (350 km): R\$ 587,00</p> <p>Final de semana (sáb. a seg.): a partir de R\$ 938,00</p> <p>Final de semana (sex. a seg.): a partir de R\$ 1.224,00</p> <p>Caução (pré-autorização): R\$ 4.500,00</p> <p>Km excedente: R\$ 1,30</p>

Fonte: Roxmoto (2023).

Além da disponibilidade para aluguel de motocicletas, a empresa disponibilizou no seu site a possibilidade de adquirir pacotes de viagens em todo o mundo. É possível visualizar quais são os *tours* realizados no mês e participar dos mesmos. Conforme Figura 7 e Figura 8, nota-se as informações sobre a viagem escolhida e quais são as opções para os hotéis e opções de motos para alugar para a viagem.

Figura 7 - Informações do Tour – Roxmoto Aplicação Web

The screenshot displays the 'URUGUAI ABSOLUTO 2023' tour page. At the top, there is a navigation bar with the Roxmoto logo and links for 'MINHA CONTA', 'ROX MOTO', 'ASSINE NOSSA NEWSLETTER', and 'SUPPORTE'. Below the navigation, there are links for 'VIAGENS', 'CALENDÁRIO DE VIAGENS', 'PASSEIOS', and 'ALUGUEL DE MOTOS'. The main heading is 'URUGUAI ABSOLUTO 2023'. Below this, there are four key details: 'TEMPO DE VIAGEM' (11 dias), 'PERÍODO' (06/09/2023 a 16/09/2023), 'CONTINENTE' (América do Sul), and 'PAÍSES' (Brasil | Uruguai). A prominent yellow button says 'TENHO INTERESSE NESTE TOUR'. Below this is the 'ROAD BOOK' section, which includes a description of the tour and a 'Seja bem-vindo à ROXMOTO!' chat bubble. The chat bubble contains the text: 'Meu nome é Ivan. Como posso te ajudar?' and a WhatsApp icon.

Fonte: Roxmoto (2023).

Figura 8 - Opções para Tour – Roxmoto Aplicação Web

The screenshot displays the 'OBSERVAÇÕES' (Observations) section of the tour page. It lists several conditions: 'Viagem sujeita ao número mínimo participantes.', 'Realização da viagem atrelada a condições climáticas', 'Uso obrigatório de equipamento completo de proteção: Capacete, Botas, Luvas, Jaqueta e Calças com proteções.', and 'Não é permitido utilizar calça jeans, tênis, etc.'. Below the observations is a social sharing section with the text 'Compartilhar com um amigo' and icons for Facebook, Twitter, Email, and WhatsApp. The bottom section is titled 'VALORES E CONDIÇÕES' and contains a table with two columns: 'Tipo de Acomodação' and 'Valor do Aluguel'. The table lists three accommodation options and two motorcycle options.

Tipo de Acomodação	Valor do Tour	Modelo da Moto	Valor do Aluguel
<input type="radio"/> Apenas Piloto em Quarto Duplo (compartilhado)	R\$18.214,23	<input type="radio"/> Moto própria	Grátis
<input type="radio"/> Apenas Piloto em Quarto Single (individual)	R\$20.383,68	<input type="radio"/> Triumph Tiger 900	+ R\$6.111,11
<input type="radio"/> Piloto com Garupa	R\$21.634,79	<input type="radio"/> Triumph Tiger 1200	+ R\$8.655,56

Fonte: Roxmoto (2023).

A aplicação web possui diversas informações referente as viagens durante o ano e os aluguéis das motocicletas. Nota-se que o intuito da empresa é disponibilizar motocicletas de alta cilindrada para viagens. Além disso, no site são bem detalhadas as informações referentes aos *tours* que a empresa realiza, deixando vários opcionais para a escolha do usuário. A aplicação deixa bem clara quais são as opções disponíveis, como abas relacionadas a viagens, calendário de viagens, passeios, aluguéis de motos, além das informações sobre a empresa e a possibilidade de acessar a “Minha Conta”.

3.1.3 Localiza – Aluguel de Carros

A aplicação da Localiza (Figura 9) tem como intuito fornecer uma frota de veículos de diversas classes para que pessoas possam alugar, sendo possível verificar quais são as opções no site ou no aplicativo IOS ou Android (Play Store). A ferramenta apresenta quais são as opções do seu estoque e opções a mais para que a locação do cliente seja confiável e segura. (LOCALIZA, 2023).

Figura 9 - Localiza – Aplicação Web



Fonte: Localiza (2023).

Na ferramenta, é possível selecionar o local de retirada e devolução, e as datas e hora para retirada e devolução. Após as escolhas de Local, Data e Hora da

reserva, são apresentados opcionais para complementar a reserva do cliente. Durante o processo de reserva é possível selecionar opções extras como Proteção do Carro, Cobertura para Terceiros, Proteção de Vidros Premium e Pneus e acessórios, além de outros adicionais como Limpeza Garantida, Neutraliza (Compensação de Carbono), Bebê Conforto, Assento de Elevação e Cadeira de Bebê, conforme Figura 10. (LOCALIZA, 2023).

Figura 10 - Opções de Acessórios na Reserva – Localiza Aplicação Web

The screenshot displays the 'Proteções do carro' (Car Protections) section of the Localiza website. It features several selectable options for car protection and accessories, each with a price per day and a checkbox for selection. A summary table on the right provides a breakdown of the total cost for the reservation.

Proteções do carro

Mais segurança na sua jornada

- Proteção do carro: R\$ 29,95 / dia
- Proteção Total Avarias: R\$ 38,95 / dia
- Cobertura para terceiros: R\$ 14,95 / dia

Que tal uma proteção complementar? Selecione uma das opções:

- Proteção de Vidros Premium e Pneus: R\$ 12,95 / dia

Adicionais e Acessórios

Fique livre de preocupações! Conheça e inclua os nossos adicionais na sua reserva:

- Conductor Jovem: R\$ 29,95 / dia
- Limpeza Garantida: R\$ 34,95 *valor único

Resumo da Reserva

Retirada
22 de Agosto de 2023 às 08:00
Agência Centro Rio Do Sul

Devolução
22 de Agosto de 2023 às 18:00
Agência Centro Rio Do Sul

Grupo
Grupo B - Compacto Com Ar
Fiat Mobi 1.0 ou similar

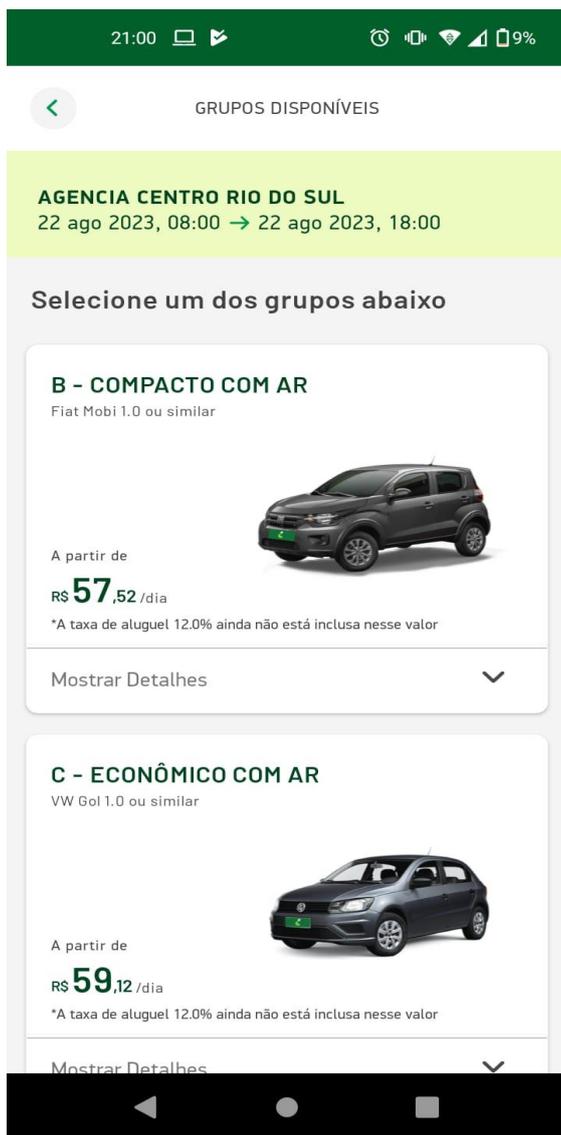
Oferta Especial

Diária	Total
1x R\$ 71,90	R\$ 71,90
Proteções	
Proteção do carro 1 diária x R\$ 29,95	R\$ 29,95
Cobertura para terceiros 1 diária x R\$ 14,95	R\$ 14,95
Limpeza Garantida Valor único R\$ 34,95	R\$ 34,95
Taxa de aluguel (12%)	R\$ 18,21

Fonte: Localiza (2023).

Na Figura 11, é possível verificar que as opções disponibilizadas no aplicativo são as mesmas disponibilizadas no site. Assim como na aplicação web, as opções dos veículos, escolha do local e data/hora de retirada e devolução, opcionais adicionais mais na reserva, cadastro e login de usuário, entre outras funções do site, também são encontradas no aplicativo, facilitando o aluguel para o usuário.

Figura 11 - Opções de Carros – Localiza Aplicativo Mobile



Fonte: Localiza (2023).

4. DUOCICLO: PROTÓTIPO DE APLICAÇÃO WEB PARA GERENCIAMENTO DE ALUGUÉIS

Neste capítulo, são discutidos os aspectos técnicos envolvidos no processo de análise e implementação do protótipo.

4.1 ANÁLISE

Neste capítulo são apontados os requisitos e regras de negócio para o desenvolvimento do protótipo em questão. A visão geral do protótipo é descrita e explica o porquê do protótipo e qual a sua utilidade para o usuário, destacando suas principais vantagens e comparando-o com as aplicações apresentadas no Estado da Arte. Além da visão sobre o protótipo, é apresentado o levantamento de requisitos, sendo eles requisitos funcionais, requisitos funcionais opcionais, requisitos não funcionais e as regras de negócio e a funcionalidade do protótipo através de diagramas, como os diagrama de caso de uso, diagramas de atividade, diagrama de classe e a modelagem física do banco de dados, assim mostrando visualmente uma visão ampla do protótipo proposto.

4.1.1 Visão Geral do Protótipo

O protótipo em questão tem como objetivo possibilitar que lojistas no ramo de veículos de duas rodas (motocicletas, bicicletas, ciclomotores, patinetes, etc.) possam colocar a sua frota para aluguel, assim gerando lucro para veículos que podem estar parados há meses e ganhando reconhecimento com a ideia inovadora na região. O protótipo tem como principal função auxiliar no gerenciamento desses aluguéis e fornecer as frotas que cada lojista definiu para clientes que necessitam ou queiram trabalhar no ramo de *delivery*, tendo diversas opções de veículos para alugar.

Ao utilizar essa aplicação o lojista poderá cadastrar quais veículos ele deseja fornecer para aluguel. No cadastro de veículo é possível fornecer informações dele como cor, quilometragem, ano, marca, entre outras informações, detalhando o máximo possível e até mesmo adicionando uma foto. Assim os clientes podem

visualizar quais são as frotas que cada loja tem disponível e definir qual o melhor plano para a utilização do veículo no seu dia a dia.

Após a escolha do veículo, o cliente irá fazer uma reserva informando qual será o plano escolhido. Essa solicitação chega até o lojista, que analisa a solicitação e decide se irá alugar o veículo para o cliente solicitante. Após a aceitação e inclusão de um novo aluguel, tanto o cliente como o lojista ficam cientes do novo aluguel e é possível acompanhar o andamento dele, através das parcelas geradas e datas definidas para retirada e devolução do veículo. As parcelas são definidas pelo lojista na criação do plano, onde ele informa a quantidade de parcelas que o cliente final poderá escolher ao selecionar o determinado plano. Ao selecionar o plano, irá mostrar ao cliente as parcelas possíveis referente ao plano e o valor total, sendo possível escolher o número de vezes que deseja pagar o plano do aluguel. As parcelas têm sua data de vencimento para o dia 10 dos próximos meses.

A principal vantagem na utilização desta aplicação é que os benefícios podem ser observados tanto do lado lojista quanto do lado do cliente, ambos saem ganhando com os aluguéis, com isso vale destacar que várias lojas podem disponibilizar os veículos de suas lojas para aluguel. O lojista através das definições dos veículos, gestão dos aluguéis de cada cliente e das manutenções da sua frota, fica reconhecido em sua cidade/região pela ideia inovadora. Além da ideia, a ferramenta apresenta gráficos e rotinas para que o gerenciamento desses aluguéis seja completo e intuitivo. Já para o cliente final, a ideia de alugar é bastante eficiente, principalmente para aqueles que desejam entrar para o ramo de *delivery*. Através da aplicação o cliente poderá verificar qual a melhor loja para realizar o aluguel e quais planos estão alinhados ao seu orçamento. Além disso, caso o cliente tenha seu veículo pessoal avariado, ele possui a opção de alugar, evitando a perda do dia de trabalho. Com a aplicação o cliente fica ciente do aluguel ativo em seu nome e quais parcelas ele ainda necessita pagar referente a esse aluguel.

4.1.2 Comparação do Protótipo com o Estado da Arte

No Quadro 7 é possível analisar as comparações realizadas entre os recursos do protótipo proposto com as aplicações pesquisadas no estado da arte. O objetivo é demonstrar quais recursos estão presentes em cada aplicação.

Quadro 7 - Comparativos dos Recursos do Duociclo com Estado da Arte

Recurso	Duociclo	Loca9moto	Roxmoto	Localiza
Listagem de Veículos	X		X	X
Aplicativo para Dispositivos Móveis		X		X
Listagem de veículos de diversas lojas	X			
Aplicação Web Responsiva	X	X	X	X
Aluguel de veículos de duas rodas	X	X	X	
Consulta das reservas	X	X		X
Consulta de manutenções dos veículos	X			

Fonte: Acervo do autor (2023).

4.1.3 Requisitos

Nesta seção estão listados todos os requisitos funcionais, não funcionais e regras negócio identificadas para o desenvolvimento da aplicação. Para que a aplicação funcione conforme necessidade do negócio, os requisitos devem ser desenvolvidos conforme planejados e listados.

No Quadro 8, são apresentados os requisitos funcionais para o desenvolvimento do protótipo, bem como as regras de negócio na coluna mais à direita. Esses requisitos são encontrados para todos os módulos disponíveis no protótipo (cliente, lojista e administrador).

Quadro 8 - Requisitos Funcionais - Geral

Número	Nome	Descrição	Regras de Negócio
RF01	Login	O protótipo deverá conter a tela de Login para o usuário efetuar o login no sistema, acessando os recursos de acordo com o tipo do usuário. Para efetuar o login deverá utilizar e-mail e senha.	RN01
RF02	Logout	O protótipo deverá ter a opção de logout, para o usuário encerrar sua sessão.	

Fonte: Acervo do autor (2023).

No Quadro 9, são apresentados os requisitos funcionais para o Módulo do Cliente.

Quadro 9 - Requisitos Funcionais - Cliente

Número	Nome	Descrição	Regras de Negócio
RF03	Cadastre-se	O protótipo deverá possibilitar que o cliente final	RN04

		se cadastre no site. Para realização do cadastro, o protótipo deverá contar com os seguintes campos: Nome (obrigatório), CPF (obrigatório), Data de Nascimento (obrigatório), Complemento do Endereço, N° do Endereço, CEP (obrigatório), E-mail (obrigatório), Telefone (obrigatório) e Senha (obrigatório). Quando inserido por essa rotina, por padrão deverá o usuário ser do tipo cliente.	RN06
RF04	Meu Perfil (Cliente)	O protótipo deverá possibilitar que o cliente final edite seus dados do perfil. Deverá ser possível alterar o Telefone, Senha, CEP, N° do Endereço e Complemento do Endereço.	RN04 RN06
RF05	Meus Aluguéis	O protótipo deverá conter uma consulta dos aluguéis relacionados ao cliente. Os campos apresentados na consulta são: Veículo, Data de Início, Data de Término, Valor e Situação. O aluguel terá uma situação que poderá ser: Em Execução ou Finalizado.	RN04 RN12
RF06	Solicitar Reserva	O protótipo deverá conter uma rotina para solicitar uma reserva. Irá apresentar uma listagem em card com todos os veículos disponíveis para aluguel. Para cada card irá apresentar um botão para visualizar as informações da loja, um botão para informações do veículo e um botão para realização da solicitação da reserva. O cliente final deverá informar os seguintes campos para solicitar uma reserva: veículo (obrigatório), data de início (obrigatório), data de término (obrigatório) e o plano (obrigatório).	RN04 RN11
RF07	Consulta das Parcelas do Aluguel no Menu Principal	O protótipo deverá conter uma consulta das parcelas de determinado aluguel. Deverá ser apresentado a situação da parcela, valor e a data de validade. Para o cliente deverá apresentar a listagem das parcelas do aluguel ativo na tela central.	RN04 RN08
RF08	Menu (Cliente)	O protótipo deverá ter o menu do cliente, contendo as opções no menu: Home, Solicitar Reserva, Meus Aluguéis, Meu Perfil e Sair.	RN04
RF09	Card Aluguel Ativo	O protótipo deverá ter um <i>card</i> onde apresente informações sobre o aluguel ativo do cliente. Como informações deverá ter a data de devolução do veículo e quantidade de parcelas a pagar.	RN04

Fonte: Acervo do autor (2023).

No Quadro 10, são apresentados os requisitos funcionais para o Módulo do Lojista.

Quadro 10 - Requisitos Funcionais - Lojista

Número	Nome	Descrição	Regras de Negócio
RF10	Menu (Lojista)	O protótipo deverá ter o menu do lojista, contendo as opções no menu: Home, Cadastros, Gestão, Minha Loja, Meu Perfil e Sair. Para o	RN03

		menu de Cadastros deverá conter os seguintes submenus: Veículos, Marcas e Planos. Para o menu de Gestão deverá conter os seguintes submenus: Aluguéis, Manutenções e Reservas.	
RF11	Rotina de Marcas	O protótipo deverá ter o cadastro de marcas dos veículos. Os campos necessários são: nome (obrigatório). Além da possibilidade de cadastro, deverá ter a consulta das marcas já cadastradas, podendo alterar, excluir ou visualizar os registros.	RN03
RF12	Rotina de Veículos	O protótipo deverá conter o cadastro de veículos. Os campos necessários são: marca (obrigatório), descrição (obrigatório), cor (obrigatório), quilometragem (obrigatório), Ano (obrigatório), Placa (obrigatório) e Imagem do Veículo. Além do cadastro de veículos, deverá ter a consulta dos veículos já cadastrados, podendo alterar, excluir ou visualizar os registros.	RN03
RF13	Gestão de Aluguéis	O protótipo deverá conter a gestão dos aluguéis, sendo possível cadastrar novos aluguéis e consultar os aluguéis que já foram cadastrados. Os campos necessários para cadastro de um novo aluguel são: Veículo (obrigatório), Cliente (obrigatório), Data de Início (obrigatório), Data de Término (obrigatório), Valor (obrigatório). O aluguel terá uma situação que poderá ser: Em Execução ou Finalizado.	RN03 RN13
RF14	Gestão de Manutenções	O protótipo deverá conter uma gestão de manutenções dos veículos, sendo possível realizar cadastro dos veículos em manutenção e poder consultar os veículos que estão na manutenção. Para o cadastro de uma manutenção são necessárias as seguintes informações: Veículo (obrigatório), Descrição (obrigatório), Valor (obrigatório), Data de Início (obrigatório) e Observação. Deverá ser possível finalizar a manutenção através de uma ação. Ao finalizar o lojista deverá informar a data término da manutenção do veículo. A consulta das manutenções deverá listar os registros e conter ainda as ações de alterar (quando o registro estiver com a situação 'Em Execução'), excluir e visualizar.	RN03 RN10
RF15	Gestão das Reservas	O protótipo deverá conter uma rotina para gestão das reservas solicitadas. Deverá listar as reservas solicitadas pelos clientes e ter as opções de aceitar, recusar e visualizar. Após ser aceita, deverá gerar um novo aluguel e as outras reservas para o mesmo veículo devem ser recusadas (somente são recusadas as reservas que estiverem entre as datas da reserva aceita). Depois de ser aceita o registro só poderá ser visualizado.	RN03 RN11
RF16	Consulta das Parcelas do Aluguel	O protótipo deverá conter uma consulta das parcelas de determinado aluguel. Deverá ser apresentado a situação da parcela, valor e a data de validade. Para o lojista irá ficar disponível ao acessar o aluguel do cliente. Deverá ter as ações Paga e Aberta, para que o	RN03 RN08

		lojista tenha o controle das parcelas pagas ou não. Caso a parcela esteja atrasada deverá pintar a linha do registro de vermelho.	
RF17	Rotina de Planos	O protótipo deverá ter um cadastro de planos, sendo possível determinar quantos dias o plano poderá ter. Além disso o lojista tem a possibilidade de configurar quantas parcelas pode ter o plano, onde o cliente escolhe na solicitação da reserva em quantas vezes ele deseja parcelar o valor total do plano. O lojista ao aceitar a solicitação da reserva, gera o aluguel e suas respectivas parcelas conforme escolha do cliente. As parcelas são geradas com a data de vencimento no dia 10 do próximo mês. Para o cadastro deverá ter os seguintes campos: Descrição (obrigatório), Quantidade de Dias (obrigatório), Valor (obrigatório) e Quantidade de Parcelas Possíveis (obrigatório).	RN03
RF18	Gráfico de Aluguéis	O protótipo deverá ter em seu menu inicial um gráfico com os dados referente aos aluguéis no ano. Esse gráfico irá mostrar a quantidade de aluguéis em cada mês do ano, contabilizando apenas os aluguéis finalizados.	RN03 RN09
RF19	Card de Aluguéis	O protótipo deverá ter em seu menu inicial um <i>card</i> com o lucro total com os aluguéis (contabilizando apenas os finalizados).	RN03 RN09
RF20	Gráfico de Manutenções	O protótipo deverá ter em seu menu inicial um gráfico que conste as manutenções no ano. Esse gráfico deverá mostrar a quantidade de manutenções em cada mês do ano, contabilizando apenas as manutenções finalizadas.	RN03 RN10
RF21	Card de Manutenções	O protótipo deverá ter em seu menu inicial um <i>card</i> com o gasto total (contabilizando apenas manutenções finalizadas).	RN03 RN10
RF22	Minha Loja	O protótipo deverá ter uma rotina para que o lojista possa alterar as informações referente a loja. Poderá alterar os seguintes dados: Complemento do Endereço, N° do Endereço, Telefone, E-mail e CEP.	RN03
RF23	Meu Perfil (Lojista)	O protótipo deverá possibilitar que o lojista edite seus dados do perfil. Deverá ser possível alterar o Telefone, Senha, CEP, N° do Endereço e Complemento do Endereço.	RN03

Fonte: Acervo do autor (2023).

No Quadro 11, são apresentados os requisitos funcionais para o Módulo do Administrador.

Quadro 11 - Requisitos Funcionais - Administrador

Número	Nome	Descrição	Regras de Negócio
RF24	Menu (Administrador)	O protótipo deverá conter o menu do administrador, que deverá ter as opções no menu: Home, Lojas, Usuários e Sair. Essas rotinas ficam responsáveis pelo usuário	RN02

		administrador para que ele defina quais são as lojas que utilizam o sistema e seus respectivos usuários, vinculando-os em sua loja.	
RF25	Rotina de Usuários	O protótipo deverá dispor de uma rotina para realizar a manutenção dos usuários. Para realização do cadastro, o protótipo deverá contar com os seguintes campos: Nome (obrigatório), CPF (obrigatório), Data de Nascimento (obrigatório), Complemento do Endereço, N° do Endereço, CEP (obrigatório), E-mail e Telefone. A opção de tipo de usuário ficará disponível apenas para o usuário administrador para que ele possa distinguir quem é comerciante e quem é o cliente final. Além disso, deverá conter um campo lista com as lojas disponíveis, sendo possível o administrador vincular o usuário a uma loja, sendo assim o novo usuário cadastrado será um lojista.	RN02 RN06
RF26	Rotina de Lojas	O protótipo deverá conter o cadastro de lojas, que ficará responsável pelo usuário Administrador no menu inicial do sistema, onde ele poderá cadastrar as lojas que utilizam o sistema para o gerenciamento de seus aluguéis. Neste cadastro deverá conter os seguintes campos: Nome (obrigatório), Complemento do Endereço, N° do Endereço, CEP (obrigatório) e CNPJ (obrigatório).	RN02
RF27	Card de Usuários	O protótipo deverá ter em seu menu inicial um card com o total de usuários cadastrados.	
RF28	Gráfico de Usuários	O protótipo deverá ter em seu menu inicial um gráfico que conste os usuários cadastrados no ano, apresentando a quantidade de cada mês.	
RF29	Card de Lojas	O protótipo deverá ter em seu menu inicial um card com o total de lojas cadastradas.	
RF30	Gráfico de Lojas	O protótipo deverá ter em seu menu inicial um gráfico que conste as lojas cadastradas no ano, apresentando a quantidade de cada mês.	

Fonte: Acervo do autor (2023).

No Quadro 12, são apresentados os requisitos funcionais que poderão ser desenvolvidos e implementados em versões futuras do protótipo, sendo que não fazem parte da versão inicial da aplicação.

Quadro 12 - Requisitos Funcionais Opcionais

Número	Nome	Descrição
RF31	Rastreamento do Veículo	O protótipo poderá dispor uma opção em cada aluguel para que o lojista possa rastrear o veículo alugado.
RF32	Pagamento Integrado	O protótipo poderá dispor de integração para pagamento através da aplicação.
RF33	Chat para Suporte	O protótipo poderá disponibilizar uma opção para que os usuários possam tirar dúvidas com o suporte através de um chat de atendimento.
RF34	Expiração de Senha	O protótipo poderá estabelecer um período para que expire a senha do usuário, fazendo com que ele atualize, mantendo uma

		maior segurança para a aplicação.
RF35	Versão Mobile	O protótipo poderá dispor de uma versão mobile, disponibilizado para Android e IOS.
RF36	Envio de E-mail	O protótipo poderá realizar envio de e-mail referente a atividades realizadas no software, atualizando o usuário além da aplicação. Também auxiliará para recuperação de senha.
RF37	Cadastro de Usuário (Lojista)	O protótipo poderá ter um cadastro de usuário para os lojistas cadastrarem o cliente quando ele for na loja e não tenha cadastro ainda.
RF38	Cadastro de aluguel	O protótipo poderá ter um cadastro de aluguéis para os lojistas cadastrarem estes aluguéis realizados pessoalmente com o cliente, sem passar pelo ciclo de reserva.
RF39	Taxa de Juros	O protótipo poderá ter uma configuração para que o lojista defina uma taxa de juros para parcelas atrasadas dos aluguéis.
RF40	Redefinir senha (lojista)	O protótipo poderá ter uma tela para obrigar o lojista a alterar sua senha no seu primeiro acesso, pois por padrão no cadastro dos lojistas, eles terão uma senha "padrão", por isso é importante a troca no primeiro acesso do lojista.
RF41	Notificação Lojista	O protótipo poderá ter um ícone que apresente a quantidade de notificações e que seja possível verificar essas notificações, apresentando as entradas de novas solicitações de reserva, atrasos de parcelas em determinado aluguel, etc.
RF42	Notificação Cliente	O protótipo poderá ter um ícone que apresente a quantidade de notificações e que seja possível verificar essas notificações, apresentando as aceitações ou rejeições das solicitações de reserva, parcelas próximas da validade, etc.
RF43	Aluguel dinâmico	O protótipo poderá disponibilizar um aluguel dinâmico, sendo possível selecionar a data de início e data de término do aluguel. Neste caso o lojista deverá configurar um plano diário, estipulando um valor por dia.
RF44	Cancelamento da solicitação da reserva	O protótipo poderá disponibilizar uma opção de cancelamento da solicitação da reserva para o cliente. Além disso, caso tenha passado da data de início da solicitação da reserva, o protótipo deverá cancelar automaticamente.
RF45	Laudo de Vistoria check-in e check-out	O protótipo poderá ter um registro para controle de retirada (check-in) e devolução (check-out) do veículo, para que o lojista tenha controle e possa registrar as datas de retirada e devolução.

Fonte: Acervo do autor (2023).

O Quadro 13, apresenta os requisitos não funcionais, esses que são pontos gerais da aplicação, envolvendo questões como desempenho, usabilidade e segurança.

Quadro 13 - Requisitos Não Funcionais

Número	Descrição
RNF01	O protótipo deverá ser desenvolvido com a versão do PHP 7 ou superior.
RNF02	O protótipo deverá ser desenvolvido com MySql com versão 5.7 ou superior.
RNF03	O protótipo deverá garantir que apenas usuários autorizados tenham acesso às funcionalidades apropriadas e aos dados sensíveis.
RNF04	O protótipo deverá responder às solicitações do usuário dentro de um limite de tempo de 20 segundos ou menos.
RNF05	O protótipo deverá ser fácil de usar, com uma interface intuitiva que não exija treinamento extenso para os usuários.
RNF06	As telas relacionadas a cadastro deverão ter validações no <i>front-end</i> , de acordo

	com a definição da regra de negócio.
RNF07	As telas do protótipo deverão ser totalmente responsivas, mantendo o padrão em diversas dimensões diferentes.

Fonte: Acervo do auto (2023).

O Quadro 14 apresenta detalhes das regras de negócio do protótipo, definindo regras e comportamentos necessários para o funcionamento da aplicação.

Quadro 14 - Regras de Negócio

Número	Descrição
RN01	Os usuários deverão ser de um tipo entre os 3 disponíveis: administrador, lojista ou cliente.
RN02	Para o Administrador: ficará disponível as rotinas de Lojas e Usuários.
RN03	Para o Lojista: ficará disponível as rotinas de Veículos, Marcas, Planos, Aluguéis, Manutenções, Reservas, Minha Loja e Meu Perfil.
RN04	Para o Cliente: ficará disponível as rotinas de Meus Aluguéis, Reservas, consulta das Parcelas do Aluguel ativo e Meu Perfil.
RN05	O cliente poderá ter apenas um aluguel ativo.
RN06	Para realização do cadastro a senha deverá ter entre 6 a 20 caracteres.
RN07	Para as informações de Ruas, Bairros, Cidades e Estados deverá ser utilizado os dados da API dos Correios, carregando-os de acordo com o CEP.
RN08	Na consulta de parcelas do cliente, deverá apenas apresentar as parcelas do aluguel ativo do cliente.
RN09	As informações de lucro com aluguéis no menu do lojista, deverá contabilizar somente aluguéis já finalizados, contabilizando o total no ano.
RN10	As informações de gasto em manutenções no menu do lojista, deverá contabilizar somente as manutenções já realizadas, contabilizando o total no ano.
RN11	Os clientes podem reservar um veículo com no máximo 7 dias de antecedência.
RN12	O aluguel não pode ser finalizado se tiver parcelas pendentes.

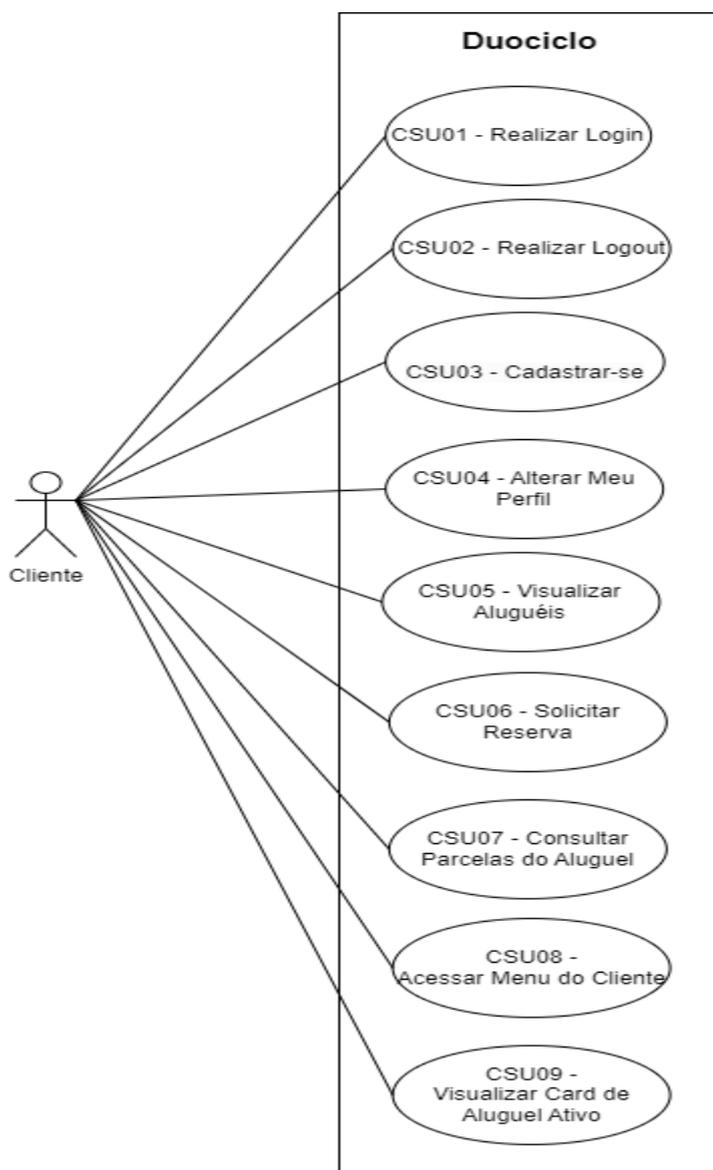
Fonte: Acervo do autor (2023).

4.1.4 Diagramas

Para facilitar a leitura referente aos requisitos mencionados e o desenvolvimento da aplicação foram elaborados alguns diagramas. Os diagramas apresentam os recursos disponíveis para os usuários e o passo a passo para execução da atividade principal proposta na aplicação.

Conforme Figura 12, para o usuário do tipo cliente, fica disponível as rotinas de reserva e de aluguéis. Na rotina de reservas, fica disponível para o cliente as lojas e os veículos disponibilizados para solicitação de reserva de cada loja. Além dessas rotinas, fica para visualização do cliente um *card* com informativo do aluguel ativo e uma consulta com as parcelas referente a este aluguel, também terá a opção “Meu Perfil”, onde o cliente poderá alterar suas informações.

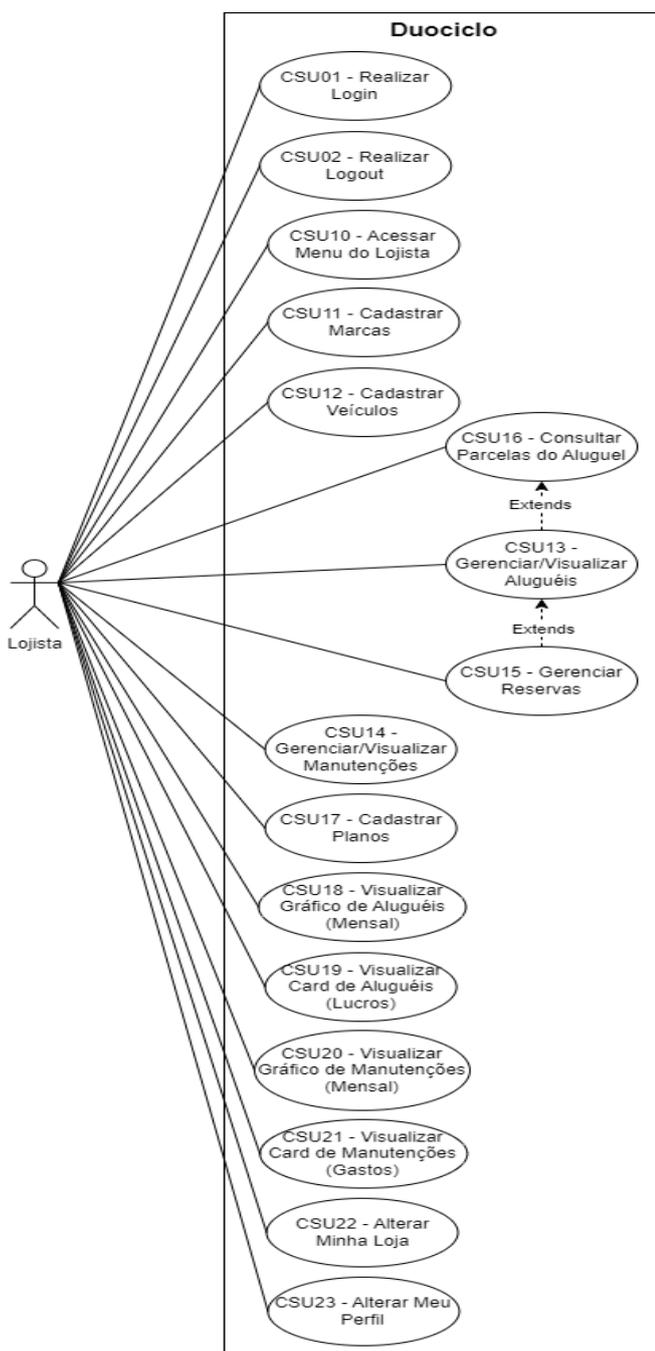
Figura 12 - Diagrama de Caso de Uso - Cliente



Fonte: Acervo do autor (2023).

Conforme Figura 13, o usuário do tipo lojista tem acesso a cadastros de veículos, marcas, usuários, planos, além de poder gerenciar/visualizar os aluguéis, manutenções e reservas. Para o lojista fica disponível o *card* de aluguéis apresentando o total de aluguéis finalizados e o lucro total, o gráfico dos aluguéis realizados por mês também fica disponível. Além das informações de aluguéis, fica visível para o lojista o *card* de manutenções, apresentando o total de manutenções realizadas e o gasto, além disso apresenta também um gráfico com as informações das manutenções, terá também a opção “Minha Loja”, para alterar as informações da loja.

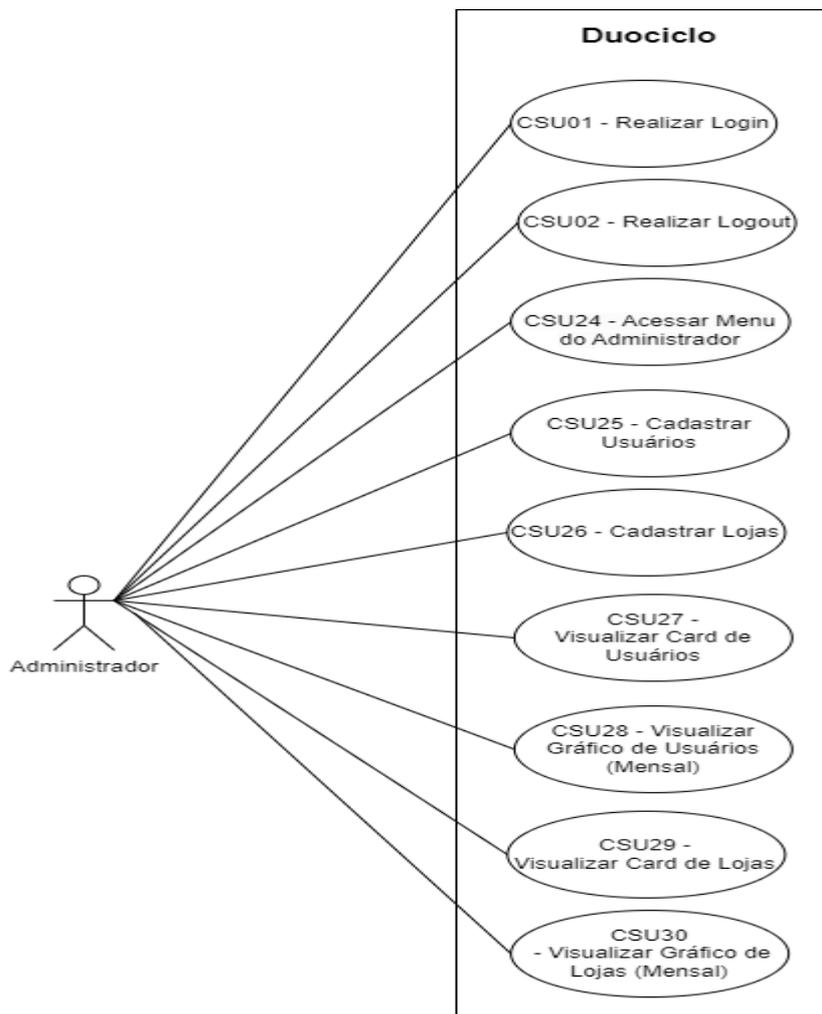
Figura 13 - Diagrama de Caso de Uso - Lojista



Fonte: Acervo do autor (2023).

Conforme Figura 14, o usuário do tipo administrador terá acessos como cadastro de lojas e cadastro de usuários. Além disso, o administrador terá como informações para visualização o *card* de usuários, que apresenta o número total de usuários cadastrados no sistema, o gráfico de usuários mostrando quantos usuários foram cadastrados em cada mês do ano. Além dessas informações ele poderá ver o *card* e o gráfico de lojas.

Figura 14 - Diagrama de Caso de Uso - Administrador

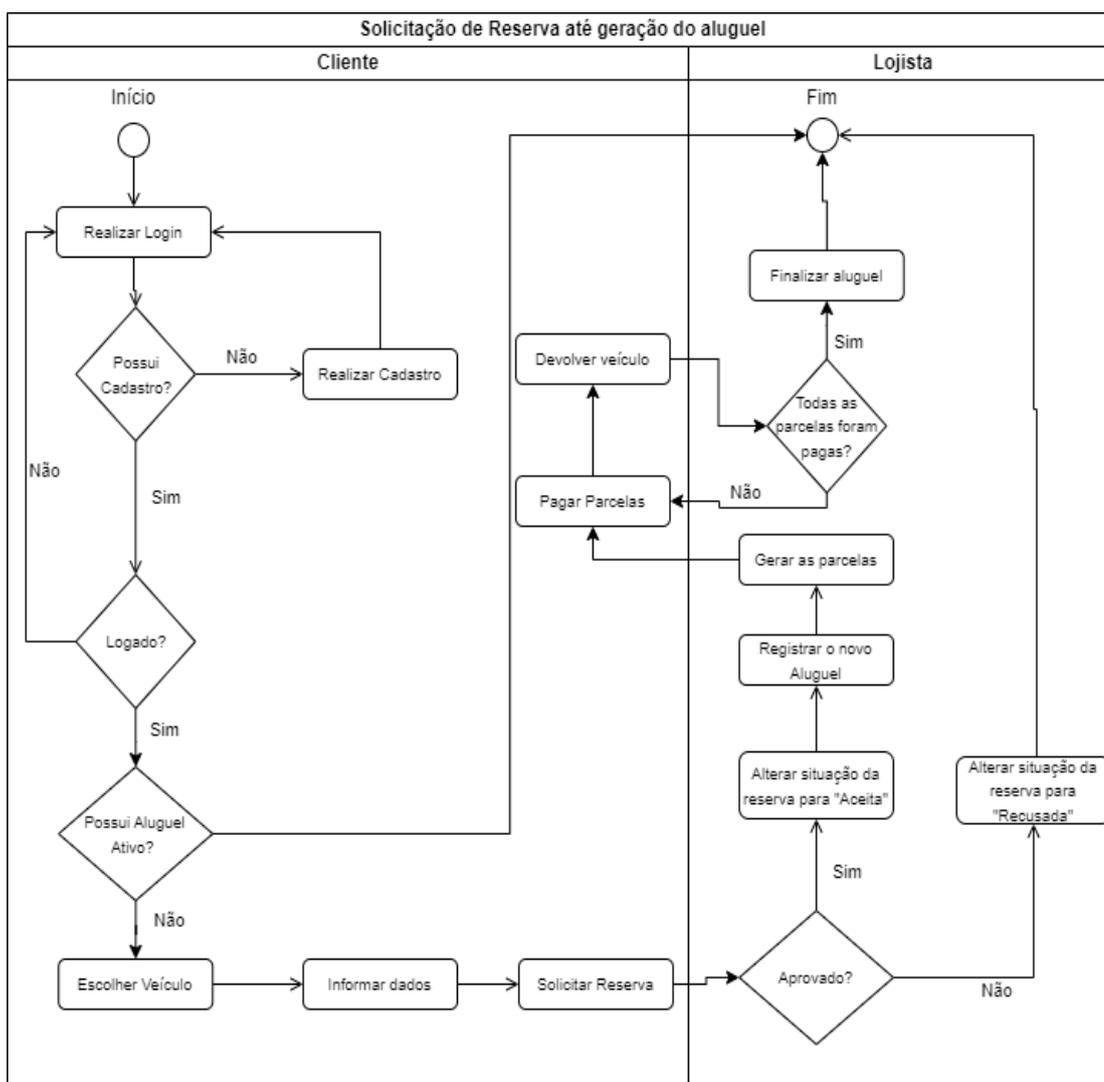


Fonte: Acervo do autor (2023).

Nota-se no diagrama de caso de uso que todos os tipos de usuário podem fazer *login* e *logout* no sistema e tem suas respectivas telas, específicas de acordo com o tipo de usuário.

Na aplicação é possível que o cliente solicite uma reserva para o lojista, escolhendo algum veículo disponibilizado pela loja, conforme detalhamento na Figura 15.

Figura 15 - Diagrama de Atividade – Solicitação de Reserva (Cliente)



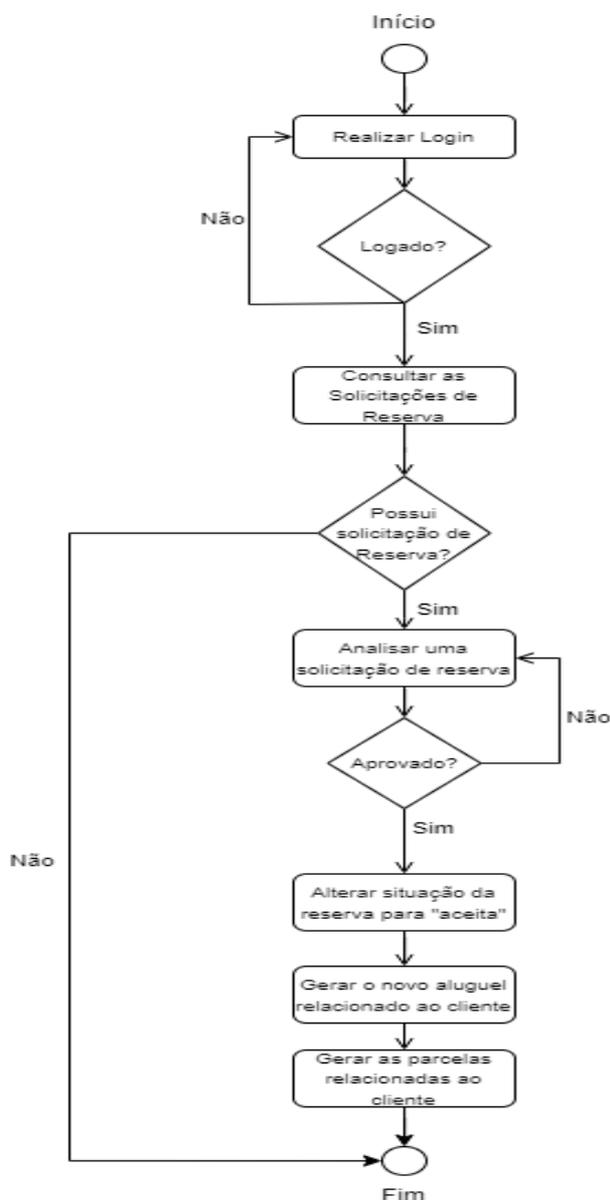
Fonte: Acervo do Autor (2023).

O Cliente precisa estar logado no software para a solicitação de reserva de um veículo para aluguel. Primeiramente o usuário precisa realizar o login no sistema, caso ele não esteja cadastrado, o mesmo deve realizar o cadastro e após a conclusão deverá efetivar o login no sistema. Após o login, o cliente irá escolher a loja para verificar as opções de veículos disponíveis, após escolha da loja o cliente deverá escolher o veículo no qual deseja alugar. Após a escolha do veículo, o cliente informa o restante dos dados e solicita a reserva, caso o cliente já possua um aluguel ativo, não poderá solicitar uma nova reserva, emitindo uma mensagem de validação antes mesmo de apresentar os veículos disponíveis, porém se o usuário não tiver um aluguel ativo, fica por conta de o lojista aceitar a solicitação de reserva do cliente. Caso o lojista aceite é alterado a situação da reserva para "Aceita" e é

gerado um novo aluguel para o cliente, gerando assim também as parcelas referente a este aluguel. Para parte da devolução, o cliente precisará pagar todas as parcelas, assim o lojista poderá finalizar o aluguel. Porém se a solicitação da reserva for recusada, a situação da reserva é alterada para “Recusada”.

Além da visão do cliente, o lojista também possui um passo a passo referente a solicitações de reserva, conforme Figura 16.

Figura 16 - Diagrama de Atividade – Solicitação de Reserva (Lojista)



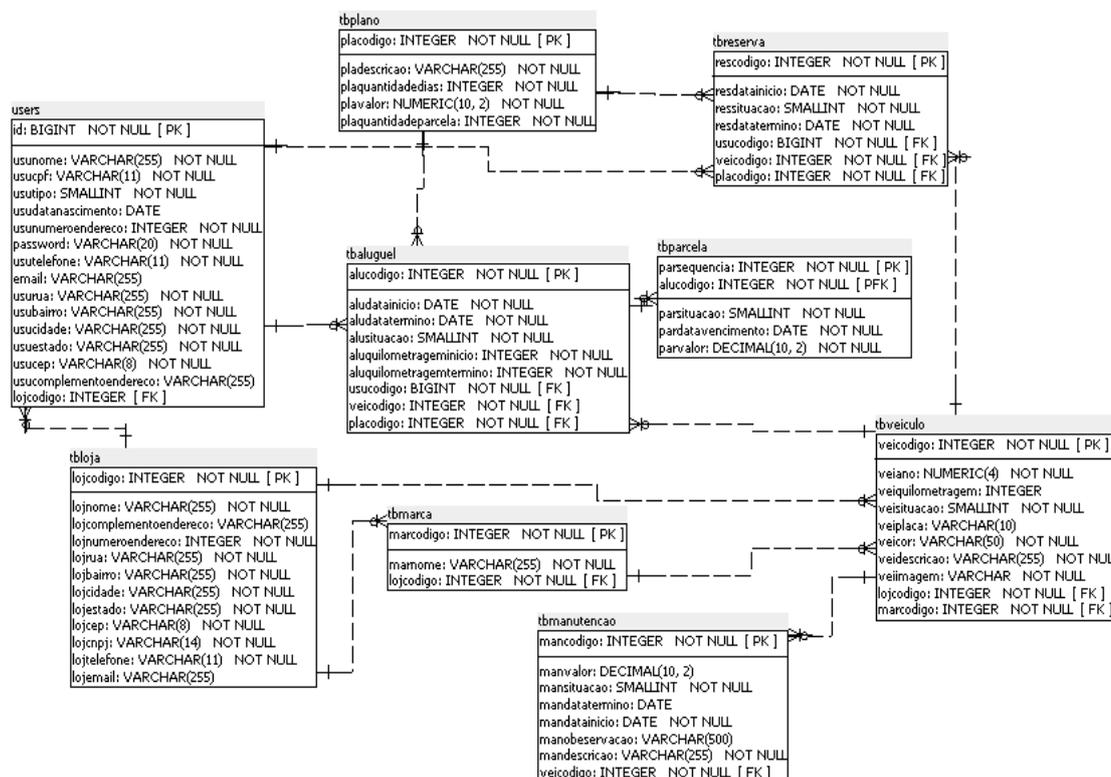
Fonte: Acervo do Autor (2023).

Conforme Figura 16, é possível verificar qual o passo a passo do lojista para as solicitações de reserva para aluguéis de veículos da sua loja. Primeiramente o

lojista deverá realizar o login no sistema. Após o login, o lojista acessa a rotina de solicitações de reservas, verificando todas as solicitações para a sua frota. Caso possua solicitação de reserva, o lojista irá analisar os dados e decidirá se irá aceitar ou recusar o pedido. Caso aceite irá gerar um novo aluguel para o cliente solicitante e consequentemente irá gerar as parcelas referente a esse aluguel. Caso o lojista recuse a solicitação, irá retornar para a consulta das solicitações, apresentando os demais clientes solicitantes referentes a sua frota de veículos.

Para melhor visualização sobre o protótipo foi desenvolvido uma modelagem do banco de dados, apresentando todos os relacionamentos e as tabelas, conforme Figura 17.

Figura 17 - Modelagem Física – Banco de Dados

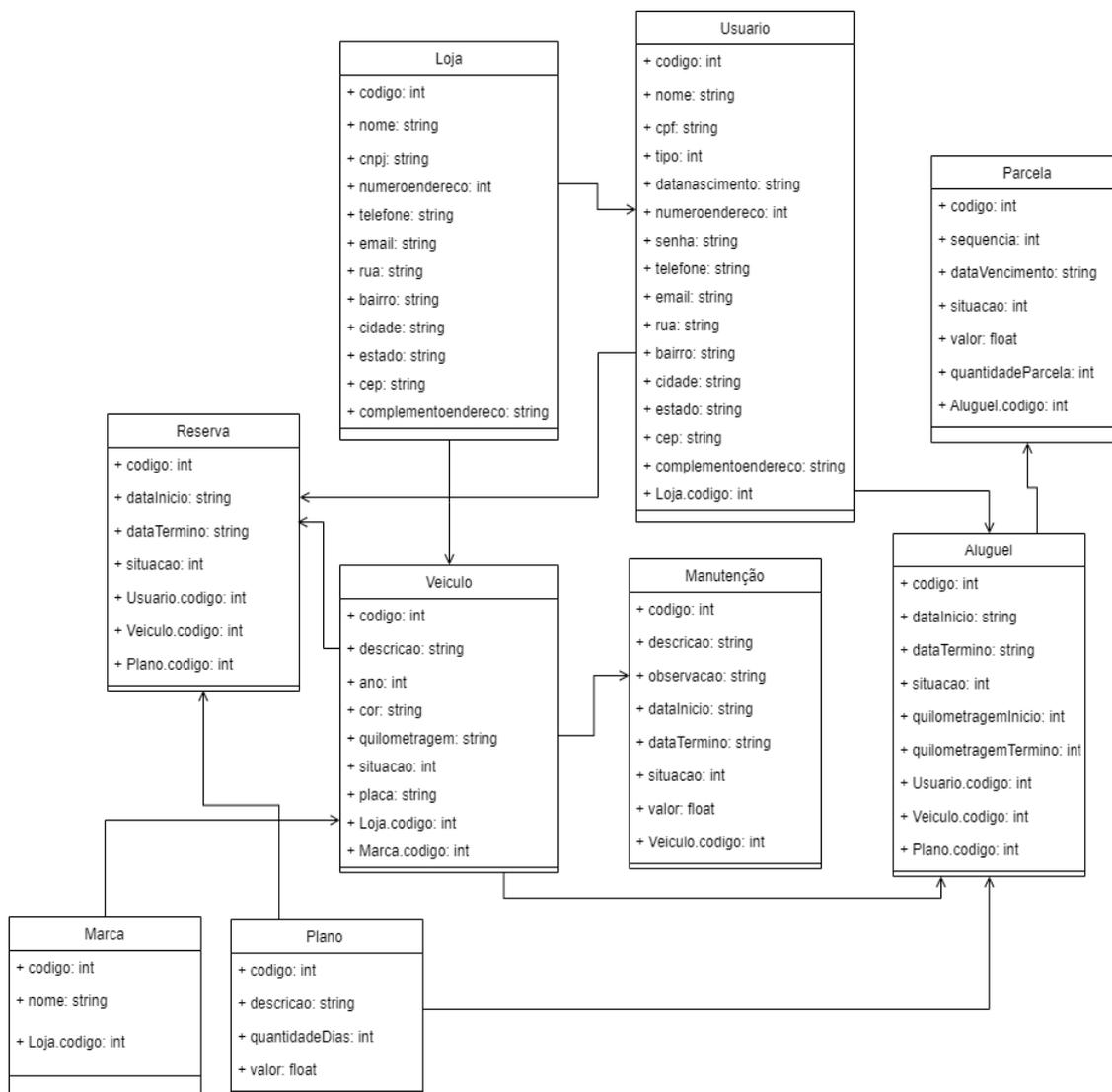


Fonte: Acervo do Autor (2023).

A modelagem apresenta a necessidade do protótipo e é formada por doze tabelas: planos, usuários, reservas, aluguéis, parcelas, lojas, veículos, marcas e manutenções. Todas as tabelas possuem chaves primárias e possui tabelas associativas para melhor entendimento do protótipo.

Como padrão de desenvolvimento foi utilizado o padrão MVC, ou seja, o desenvolvimento foi feito orientado a objeto. Conforme Figura 18, foi construído o diagrama de classes, para melhor entendimento.

Figura 18 - Diagrama de Classes



Fonte: Acervo do Autor (2023).

No próximo capítulo são apresentadas todas as informações referentes a implementação do protótipo, detalhando todos os recursos desenvolvidos.

4.2 IMPLEMENTAÇÃO

Neste capítulo são apresentadas as ferramentas e tecnologias utilizadas para o desenvolvimento do protótipo, assim como são detalhadas as rotinas implementadas para atenderem os requisitos funcionais.

4.2.1 Ferramentas e Técnicas Utilizadas

Para a etapa de análise (apresentada no capítulo 4.1) foram utilizadas algumas ferramentas para elaboração dos diagramas. O *Draw.io* foi utilizado para construção dos diagramas de caso de uso, diagramas de classe e diagramas de atividade. Já a ferramenta *SQL Power Architect* foi utilizada para a construção da modelagem física do banco de dados, auxiliam na definição de tabelas e relacionamentos.

Para o desenvolvimento das funcionalidades definidas nos requisitos, foram utilizadas as seguintes tecnologias web: HTML, CSS, JavaScript e PHP.

Como linguagem de marcação para a construção das páginas web foi utilizado o HTML5, ele é compatível com a maioria dos navegadores atuais e sua implementação é simples de ser aplicada. O HTML ficou responsável pela apresentação dos componentes em tela conforme definições nos requisitos, sendo a parte visual do protótipo.

Para auxiliar a estilização das telas foi utilizado o CSS, que através de folhas de estilo os componentes HTML são estilizados para melhor visualização e usabilidade. Com a ajuda do *framework* Materialize foi possível facilitar essa estilização, ajudando até mesmo na questão responsiva das telas e estilização de componentes. Também foi utilizado a biblioteca do Bootstrap, que auxiliou na estilização das telas de login e cadastro do protótipo.

O *JavaScript* foi utilizado para os comportamentos nas telas do protótipo, ele nos permite adicionar eventos que são executados em tempo de execução, auxiliando na implementação das regras de negócio definidas. Além do *JavaScript* foi utilizada a biblioteca jQuery para melhorar a usabilidade utilizando Ajax, fazendo buscas no banco e poupando tempo para o usuário. O Ajax auxiliou para buscas no banco de dados e seu tempo de busca foi definido para que não ultrapasse 20 (vinte

segundos), conforme requisitos não funcionais. Além disso foi utilizado o Chart.js para a apresentação dos dados em formato de gráficos nos menus dos usuários.

Para a comunicação com o servidor foi utilizado a linguagem PHP na versão 8.2.8. Para manipulação da linguagem foi utilizado o *framework* Laravel que facilitou o processo de desenvolvimento do protótipo. Através das definições das rotas, o Laravel direciona para o *Controller* responsável, isso facilita o desenvolvimento, tornando até mais fácil o código-fonte para manutenção. Para manipulação do banco de dados também se torna fácil, através das *migrations* é simples fazer a distribuição das tabelas, criando suas respectivas colunas, chaves e tipagens dos dados.

Para o armazenamento dos dados foi utilizado o banco de dados *MySQL*, que é bem conhecido e utilizado em diversos desenvolvimentos. Além de ser gratuito, é básico para manipular suas tabelas e colunas.

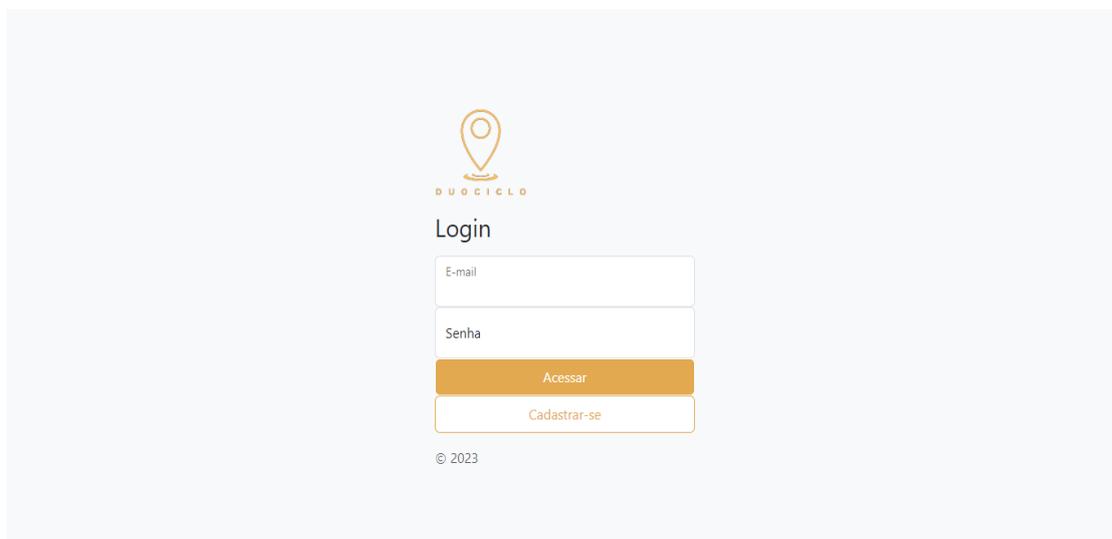
O desenvolvimento do protótipo foi feito utilizando a IDE *Visual Studio Code*, que disponibiliza diversas extensões que auxiliam no desenvolvimento do protótipo.

Além do *Visual Studio Code*, foi utilizado o *Laragon* para manipulação do *MySQL* e a auxiliou na comunicação com o servidor, pois ele tem o Apache integrado junto. O *Laragon* engloba e facilita a utilização de todas as ferramentas e ajuda com a manipulação do Laravel também.

4.2.2 Utilização e Funcionamento

Conforme citado anteriormente, o protótipo tem o intuito de um gerenciamento dos aluguéis e solicitações dos clientes, possuindo três atores para utilização do software. Os dois principais atores são os lojistas e os clientes, que interligam através das solicitações de reserva. As funcionalidades para estes dois atores são diferentes, com opções diferentes no menu e rotinas. Para cada ator foram separados capítulos, especificando o que cada um possui de acesso, detalhando suas rotinas.

Para acesso as funcionalidades do protótipo são necessárias que o usuário realize login (Figura 19), informando seu e-mail e senha.

Figura 19 - Login (RF01)

The image shows a login interface for 'DUOCICLO'. At the top center is a logo consisting of a location pin icon above the text 'DUOCICLO'. Below the logo, the word 'Login' is displayed. There are two input fields: the first is labeled 'E-mail' and the second is labeled 'Senha'. Below these fields are two buttons: 'Acessar' (highlighted in orange) and 'Cadastrar-se'. At the bottom of the form area, there is a small copyright notice '© 2023'.

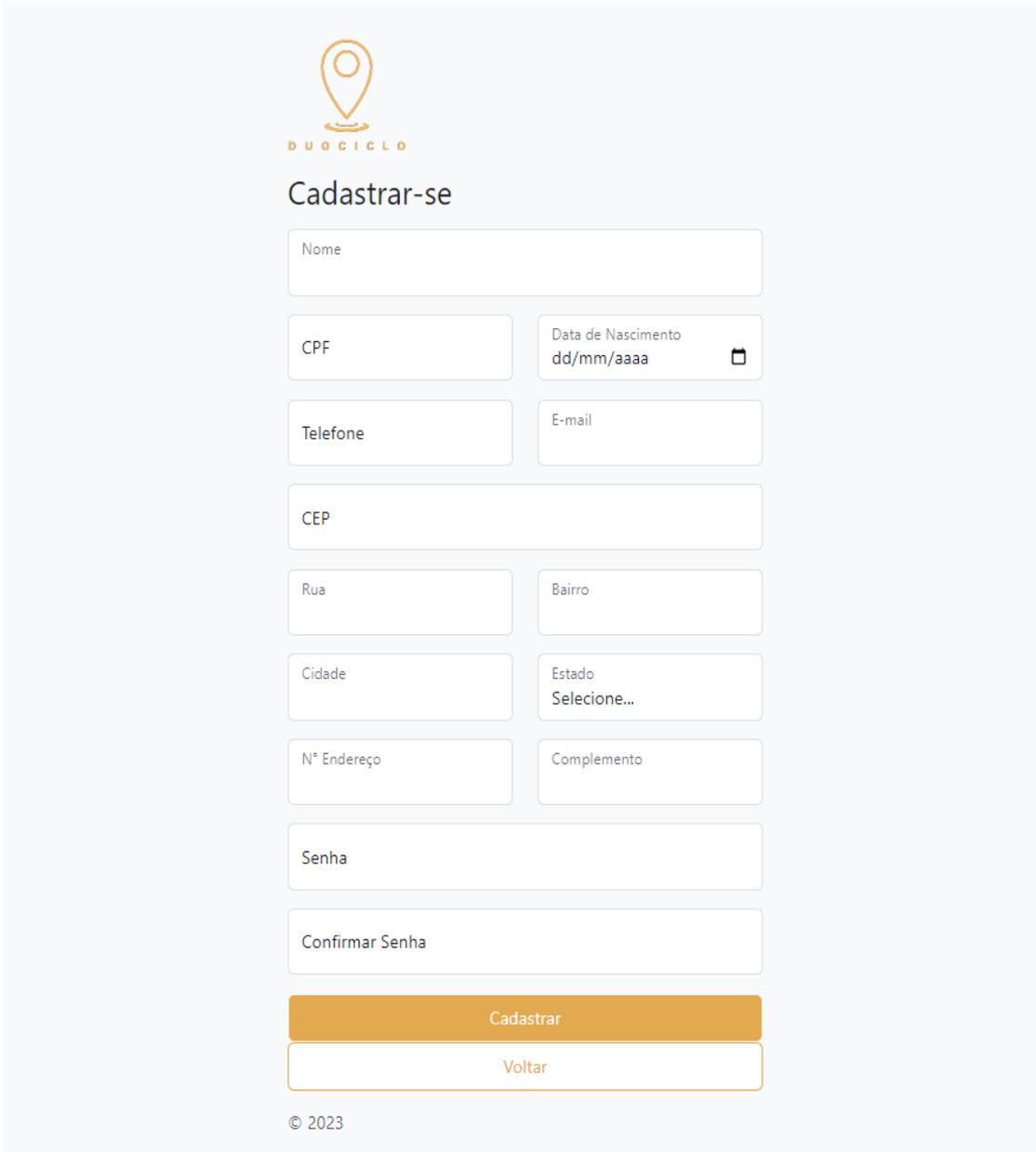
Fonte: Acervo do Autor (2023).

4.2.2.1 Rotinas Cliente

Neste capítulo é apresentada as rotinas do cliente, como ele tem o acesso e quais informações aparecem referente ao seu login. Estas rotinas apareceram apenas para o cliente final e ele tem uma comunicação direta com o lojista, através das rotinas de solicitar reserva e meus alugueis, podendo acompanhar os dados dos registros.

Para o cadastro dos clientes finais, basta acessar a opção “Cadastre-se” na tela de login, assim o usuário é redirecionado para a tela de cadastro, informando dados pessoais (Figura 20).

Figura 20 - Cadastre-se (RF03)



The image shows a registration form titled "Cadastrar-se" for the application "DUOCICLO". The form is set against a light gray background. At the top, there is a logo consisting of a location pin icon above the text "DUOCICLO". Below the logo, the title "Cadastrar-se" is displayed in a bold, dark font. The form contains several input fields: a single-line text field for "Nome"; two single-line text fields for "CPF" and "Data de Nascimento" (with a calendar icon and the format "dd/mm/aaaa"); two single-line text fields for "Telefone" and "E-mail"; a single-line text field for "CEP"; two single-line text fields for "Rua" and "Bairro"; two single-line text fields for "Cidade" and "Estado" (with a dropdown menu labeled "Selecione..."); two single-line text fields for "N° Endereço" and "Complemento"; a single-line text field for "Senha"; and a single-line text field for "Confirmar Senha". At the bottom of the form, there are two buttons: a solid orange button labeled "Cadastrar" and a white button with an orange border labeled "Voltar". Below the buttons, the copyright notice "© 2023" is visible.

Fonte: Acervo do Autor (2023).

A efetivação do cadastro leva em conta a informação dos campos de Nome, CPF, Data de Nascimento, Telefone, E-mail, CEP, Rua, Bairro, Cidade, Estado, N° Endereço, Complemento do Endereço, Senha e Confirmação de Senha. Após o cadastro o usuário é direcionado para tela de login.

A funcionalidade do cliente final traz um menu básico com as informações referente ao aluguel ativo, bem como as parcelas deste aluguel e algumas rotinas

para que o cliente possa solicitar reserva ou verificar os aluguéis em seu nome. Conforme Figura 21, o menu contém as rotinas de Solicitar Reserva e Meus Aluguéis, além da possibilidade de alteração das informações pessoais na opção “Meu Perfil”.

Figura 21 - Menu Cliente (RF07, RF08, RF09)

The screenshot shows the Duociclo client interface. At the top, there is a navigation bar with 'Home', 'Solicitar Reserva', and 'Meus Aluguéis'. The user is identified as 'Olá Paulinho!'. A dropdown menu for 'Meu Perfil' is open, showing 'Meu Perfil' and 'Sair' options. Below the navigation, there is a green card for an active rental: 'ALUGUEL ATIVO', 'Veículo: CG 125 ES', 'Plano: Mensal', 'Data de Início: 02/11/2023', 'Data de Término: 02/12/2023', and 'Valor: R\$500,00'. Underneath, a section titled 'Parcelas' contains a table with one row of installment data.

Parcela	Valor	Data de Validade	Situação
1	R\$500,00	10/11/2023	Aberta

Fonte: Acervo do Autor (2023).

Para editar os dados pessoais o cliente pode acessar a opção “Meu Perfil” no canto superior direito da tela. As alterações consistem em dados que auxiliam o lojista para contato com o cliente. Também é possível realizar a alteração da senha (Figura 22).

Figura 22 - Meu Perfil (RF04)

The screenshot shows the 'Meu Perfil' page in the Duociclo system. The user is identified as 'Paulinho Cliente'. The page contains several input fields for personal information, organized in two columns. At the bottom, there are fields for 'Senha Atual' and 'Nova Senha', with a 'Confirmar Nova Senha' label. A 'SALVAR' button is located at the bottom right.

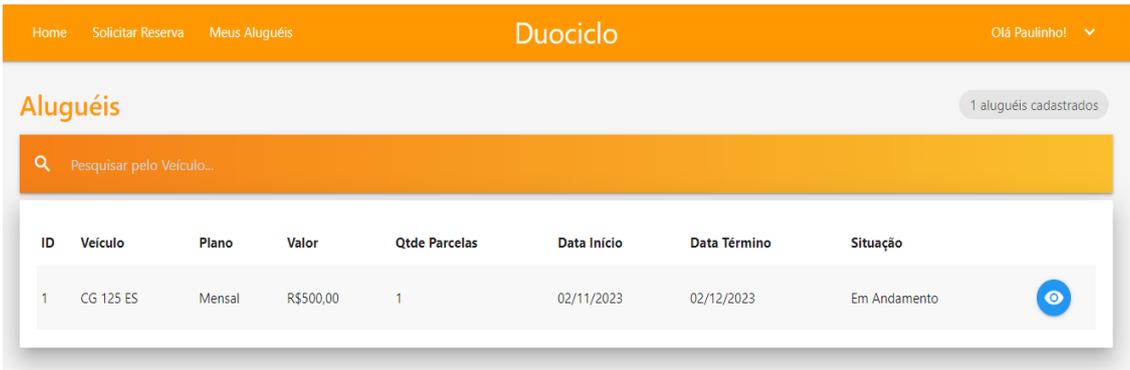
Nome	Paulinho Cliente
CPF	10029170934
E-mail	paulocliente@gmail.com
Telefone	47988607112
Data de Nascimento	06/08/2001
CEP	89163467
Rua	Guido Vota
Bairro	Fundo Canoas
Cidade	Rio do Sul
Estado	Santa Catarina
Nº Endereço	62
Complemento Endereço	casa
Senha Atual	
Nova Senha	Confirmar Nova Senha

Fonte: Acervo do Autor (2023).

Nessa tela o cliente pode realizar a alteração das informações referente a Telefone, CEP, Rua, Bairro, Cidade, Estado, N° do Endereço, Complemento do Endereço, além de poder alterar a senha, tendo que informar a senha atual e a nova senha.

Na rotina de Aluguéis, o cliente pode verificar quais aluguéis foram realizados em seu nome. Nessa consulta são apresentados os aluguéis que foram gerados após a solicitação de reserva do cliente ser aceita pelo lojista. Basicamente os registros são criados conforme informações da solicitação. Na consulta aparece a situação do aluguel, se está em andamento ou finalizado, além de outras informações referente ao serviço. A rotina consiste em uma consulta que lista os aluguéis, sendo possível visualizar através da ação, conforme Figura 23.

Figura 23 - Consulta de Meus Aluguéis (RF05)



ID	Veículo	Plano	Valor	Qtde Parcelas	Data Início	Data Término	Situação
1	CG 125 ES	Mensal	R\$500,00	1	02/11/2023	02/12/2023	Em Andamento

Fonte: Acervo do Autor (2023).

A consulta traz como informações as colunas de Identificador (ID), Veículo, Plano, Valor, Quantidade de Parcelas, Data Início, Data Término e Situação do Aluguel. Além disso é possível filtrar pelo nome do veículo e contém a ação de visualizar o aluguel (Figura 24).

Figura 24 - Visualizar Aluguel (RF05)



Fonte: Acervo do Autor (2023).

Na visualização traz os dados referente ao aluguel selecionado. Algumas informações como Veículo, Plano, Valor, Quantidade de Parcelas, Data de Início, Data de Término e Situação.

Para rotina de Solicitar Reserva, é apresentada uma consulta com as solicitações de reserva em andamento de acordo com o cliente logado. Os registros apresentados são referentes as solicitações de reserva do cliente, apresentando sua situação. Nela é possível acompanhar se o lojista aceitou ou recusou a solicitação. A consulta exibe informações básicas referente as reservas, conforme mostra a Figura 25.

Figura 25 - Consulta Reserva (RF06)

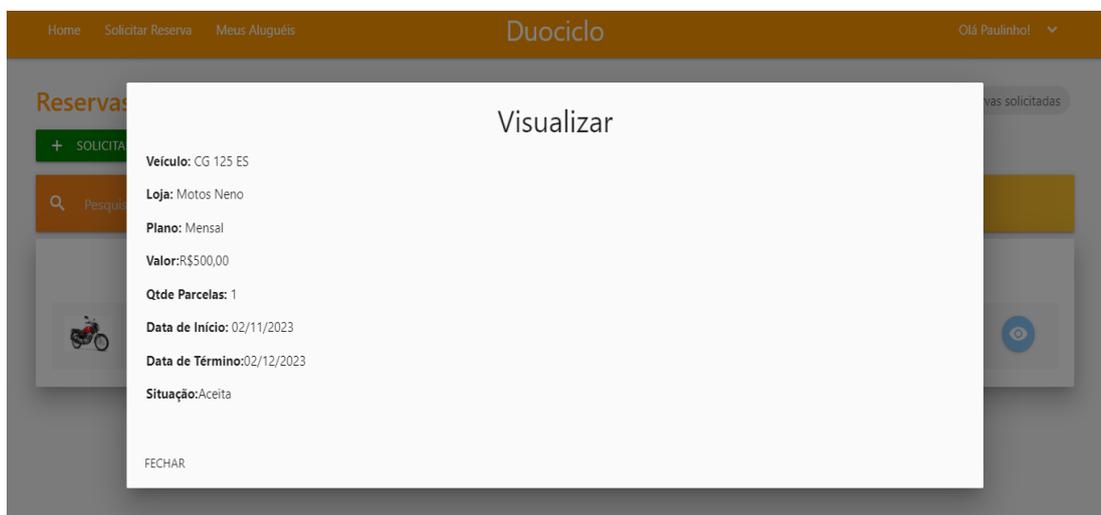


Fonte: Acervo do Autor (2023).

A consulta de solicitações de reserva apresenta todas as reservas solicitadas pelo cliente, apresentando as informações através das colunas de Identificador (ID), Veículo, Loja, Plano, Valor, Quantidade de Parcelas, Data Início, Data Término e

Situação. Além das colunas, a consulta apresenta um campo para filtrar através dos veículos e uma ação de visualizar para cada registro. A visualização traz os dados referente a solicitação (Figura 26).

Figura 26 - Visualizar Solicitação de Reserva (RF06)



Fonte: Acervo do Autor (2023).

A tela de visualização exibe as seguintes informações: Veículo, Loja, Plano, Valor, Quantidade de Parcelas, Data de Início, Data de Término e Situação do registro.

Além da ação de visualizar, é possível realizar solicitações, através da ação de "Solicitar" o usuário é direcionado para uma consulta de todos os veículos de todas as lojas, onde o cliente consegue visualizar as informações dos veículos, informações das lojas e solicitar a reserva do veículo que deseja. Conforme Figura 27, os veículos disponíveis são apresentados em cards, com três ações em cada um deles, a de visualizar loja, visualizar veículo e a de solicitar reserva. Além disso, existe o botão para voltar para a consulta de solicitações de reservas.

Figura 27 - Veículos para Reserva (RF06)



Fonte: Acervo do Autor (2023).

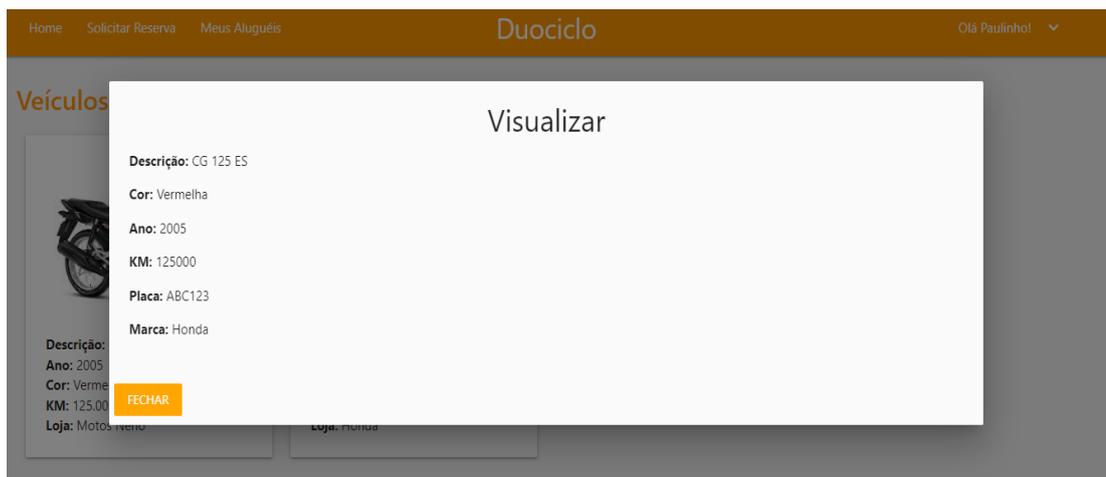
A visualização da loja traz as seguintes informações: Nome, CNPJ, Telefone, E-mail, CEP, N° Endereço e Complemento do Endereço (Figura 28).

Figura 28 - Visualizar Loja do Veículo para Reserva (RF06)



Fonte: Acervo do Autor (2023).

Já a visualização do veículo exibe as informações de Descrição, Cor, Ano, KM, Placa, Marca, conforme Figura 29.

Figura 29 - Visualizar Veículo disponível para Reserva (RF06)

Fonte: Acervo do Autor (2023).

A ação de solicitar a reserva do veículo exige que o cliente selecione o Plano, e informe a data de início, data de término e a quantidade de parcelas (Figura 30).

Figura 30 - Solicitar Reserva do Veículo (RF06)

Fonte: Acervo do Autor (2023).

Desta forma, ao clicar em solicitar irá gerar uma nova solicitação que ficará visível tanto para o cliente quanto para o lojista. O lojista irá decidir se irá recusar ou aceitar a solicitação. Caso aceite irá gerar um novo aluguel e suas parcelas para o cliente, através das informações da solicitação da reserva. As parcelas são geradas de acordo com a quantidade de parcelas escolhidas pelo cliente, irá dividir o valor total do plano na quantidade escolhida, sendo que a data de vencimento de cada

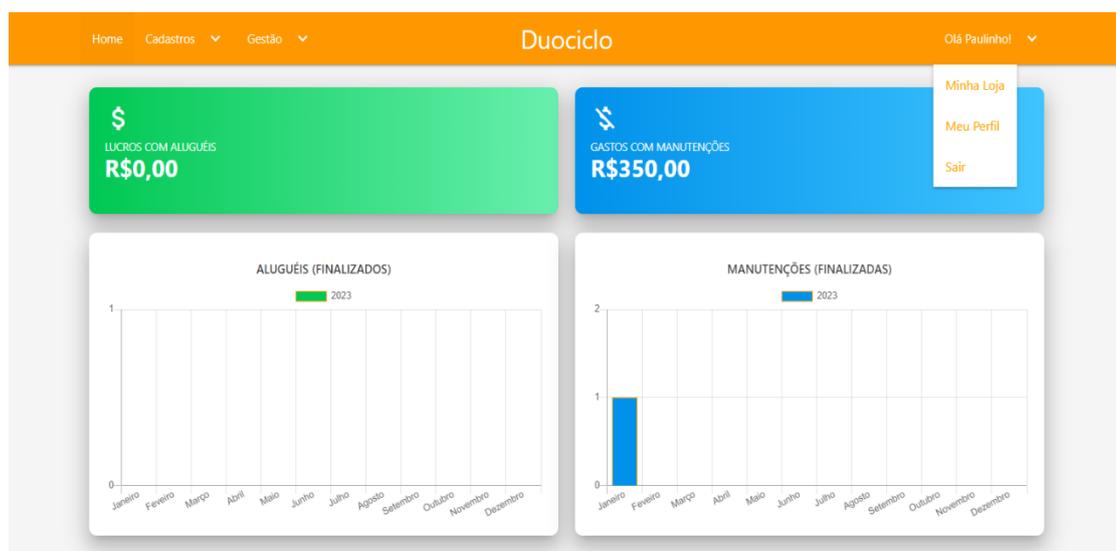
parcela é gerada no dia 10 do próximo mês. As informações do aluguel ativo ficam visíveis no menu principal do cliente, conforme citado anteriormente. Além disso, após a solicitação, na consulta de reservas ficará disponível para o cliente todas as reservas solicitadas por ele e sua devida situação.

4.2.2.2 Rotinas Lojista

As rotinas do lojista aparecem de acordo com seu tipo de login. Nelas é possível verificar as rotinas de marcas, veículos, etc. As rotinas de lojista influenciam diretamente ao cliente final, através dos registros é possível realizar solicitações de reserva e até mesmo acompanhar o andamento dos aluguéis.

Após o login, o Lojista terá acesso ao seu Menu e painel inicial, conforme Figura 31.

Figura 31 - Menu Lojista (RF02, RF10, RF18, RF19, RF20, RF21, RF22, RF23)



Fonte: Acervo do Autor (2023).

No menu do lojista são apresentados os menus de cadastro e gestão, localizados na parte superior esquerda da tela. Para o menu de cadastro são apresentadas as rotinas de Marcas, Veículos e Planos. Já no menu de gestão são apresentadas as rotinas de Manutenções, Reservas e Aluguéis. Além disso o lojista pode alterar as suas informações através do atalho “Minha Loja” localizado na parte superior direita da tela e poderá também alterar os dados referente ao seu perfil através do atalho “Meu Perfil” e o usuário poderá fazer logout na opção “Sair”.

Na área central do menu ficam os cards e gráficos referente a lucros e gastos. No card de lucros com aluguéis irá apresentar o valor total somando todos os aluguéis que já foram finalizados e logo abaixo apresenta a quantidade de aluguéis finalizados em cada mês do ano. Já no card da direita apresenta o valor total de gastos com manutenções já finalizadas e logo abaixo apresenta a quantidade de manutenções já finalizadas nos meses do ano.

Referente a rotina de Marcas, acessando pelo menu Cadastros > Marcas, abrirá a consulta de marcas já cadastradas relacionadas a loja do usuário logado conforme Figura 32.

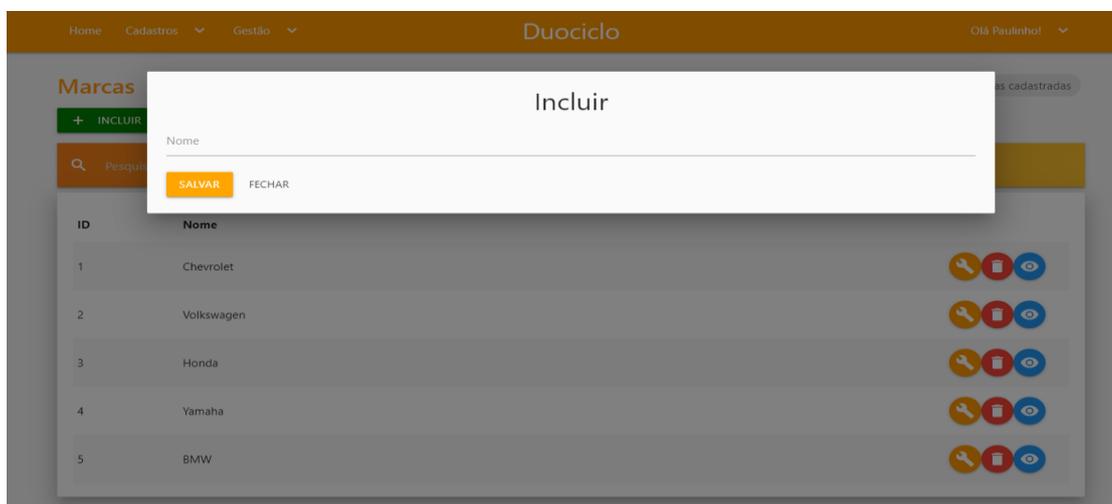
Figura 32 - Rotina de Marcas (RF11)



Fonte: Acervo do Autor (2023).

Esta rotina apresenta as marcas cadastradas para a loja do usuário logado. Como informação exibe as colunas identificador (ID) e nome. Além disso, é possível filtrar de acordo com o campo pesquisar, filtrando de acordo com o nome da marca. As ações de manutenção são a de incluir, alterar, excluir e visualizar. Ao excluir emite uma mensagem de confirmação para o usuário. As demais ações seguem o padrão (Figura 33).

Figura 33 - Cadastro de Marcas (RF11)

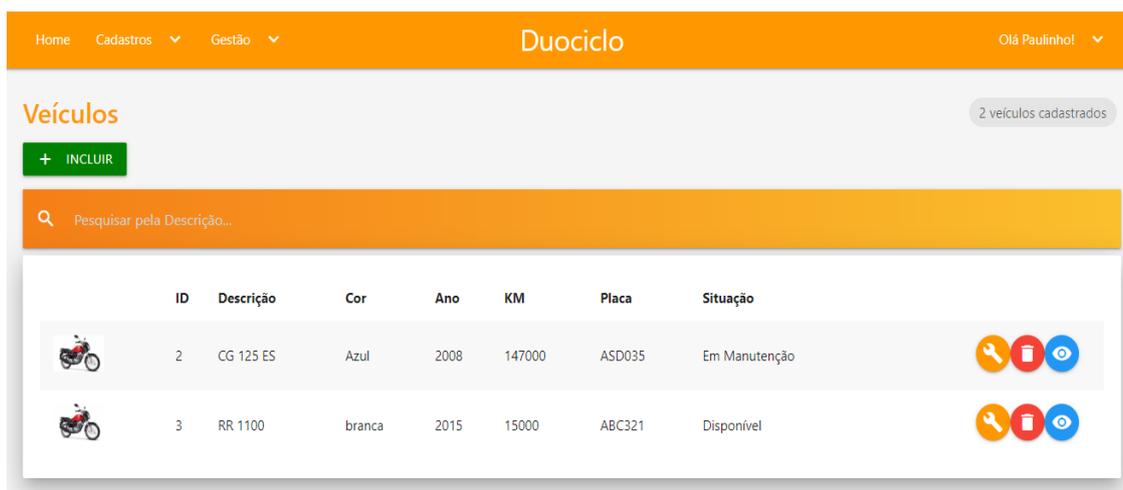


Fonte: Acervo do Autor (2023).

Para realizar o cadastro da marca o lojista deve informar o Nome da Marca. As marcas são utilizadas no cadastro dos veículos.

Já a rotina de Veículos pode ser acessada no caminho Cadastros > Veículos e traz a consulta de veículos cadastrados na loja do usuário logado (Figura 34).

Figura 34 - Rotina de Veículos (RF12)

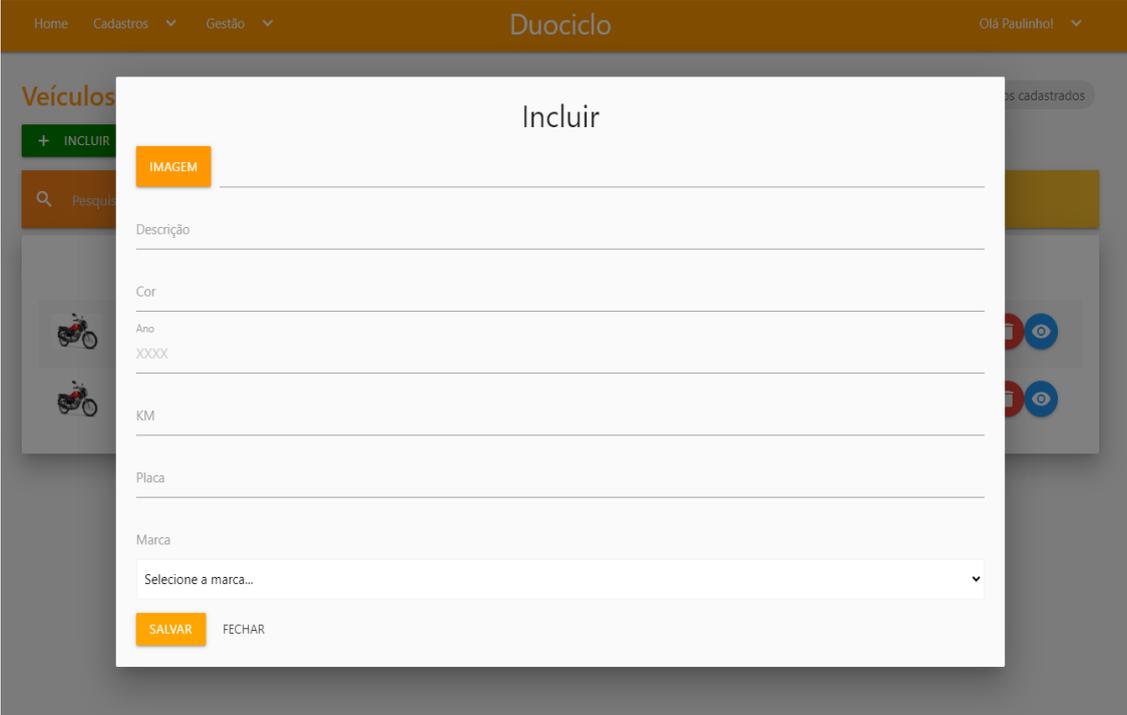


Fonte: Acervo do Autor (2023).

A consulta irá listar todos os veículos relacionados a loja do lojista logado. Exibe as colunas de Identificador (ID), Descrição, Cor, Ano, KM, Placa e Situação, além da imagem do veículo caso cadastrado. Pelo campo de pesquisa é possível

buscar através da descrição do veículo. As ações de manutenção são a de incluir, alterar, excluir e visualizar. A ação de excluir emite uma mensagem de alerta, porém só poderá excluir caso o veículo esteja disponível. As ações de incluir, alterar e visualizar seguem o mesmo padrão conforme Figura 35.

Figura 35 - Cadastro de Veículos (RF12)



A imagem mostra a interface de usuário do sistema Duociclo, especificamente a tela de inclusão de um veículo. O formulário é intitulado "Incluir" e contém os seguintes campos:

- IMAGEM**: Botão para upload de imagem.
- Descrição**: Campo de texto para a descrição do veículo.
- Cor**: Campo de texto para a cor do veículo.
- Ano**: Campo de texto com máscara "XXXX" para o ano.
- KM**: Campo de texto para o quilômetro rodado.
- Placa**: Campo de texto para a placa do veículo.
- Marca**: Menu suspenso com o texto "Selecione a marca..." para selecionar a marca do veículo.

Na base do formulário, há dois botões: **SALVAR** (em laranja) e **FECHAR** (em cinza).

Fonte: Acervo do Autor (2023).

Para a realização do cadastro de um veículo o lojista precisa informar a Descrição do Veículo, Cor, Ano, KM, Placa e selecionar de qual marca o veículo é, através da lista de marcas.

Além do cadastro de veículos, o lojista pode definir os planos para que os clientes finais possam escolher para o aluguel. A rotina de planos fica em Cadastros > Planos e ao acessar irá trazer a consulta de planos (Figura 36).

Figura 36 - Rotina de Planos (RF17)



Fonte: Acervo do Autor (2023).

A consulta lista todos os planos cadastrados para a loja do lojista logado. Como informação traz as colunas de Identificador (ID), Descrição, Quantidade de Dias, Valor e Quantidade de Parcelas. É possível buscar pela descrição através do campo de pesquisa. A rotina contém as ações de incluir, alterar, excluir e visualizar. Para excluir irá emitir uma mensagem de confirmação, porém só poderá excluir caso o plano não esteja vinculado a nenhum aluguel em andamento. As ações de incluir, alterar e visualizar seguem o padrão conforme Figura 37.

Figura 37 - Cadastro de Planos (RF17)



Fonte: Acervo do Autor (2023).

Para cadastrar um novo plano o lojista precisa informar os campos de Descrição, Quantidade de Dias, Valor e Quantidade de Parcelas. O campo de

quantidade de dias irá influenciar de acordo com a data de início do aluguel selecionado pelo cliente, calculando os dias a mais para verificar qual dia será o término do aluguel. O campo de quantidade de parcelas irá servir para que o cliente possa parcelar o valor total do plano do aluguel, gerando as parcelas conforme solicitação da reserva, com data de vencimento no dia 10 do próximo mês da solicitação.

O lojista também tem acesso a consulta e cadastros de manutenções dos veículos. O lojista fica responsável por cadastrar veículos que passaram por manutenções, sendo que eles não ficarão disponíveis para aluguel durante esse período. Acessando pelo menu Gestão > Manutenções, o lojista será redirecionado para consulta (Figura 38).

Figura 38 - Rotina de Manutenções (RF14)

ID	Descrição	Data Início	Data Término	Valor	Situação
2	Teste manutencao	24/10/2023	26/10/2023	350,00	Finalizado
3	Teste 2	02/11/2023	-	580,00	Em Andamento

Fonte: Acervo do Autor (2023).

A consulta segue o padrão das demais do protótipo, trazendo como informação as colunas Identificador (ID), Descrição, Data de Início, Data de Término, Valor e Situação, sendo possível filtrar de acordo com a descrição da manutenção. A consulta possui as ações incluir, alterar, excluir, visualizar e finalizar. A ação de excluir emite uma mensagem de confirmação para o lojista. A ação de finalizar só ficará disponível para manutenções com a situação em andamento, ao clicar em finalizar irá exibir uma tela para informar a data término. Ao salvar irá alterar a situação da manutenção para finalizado. Desta forma, ao finalizar uma manutenção ela irá contabilizar no valor de gastos total no menu do lojista. A ação de incluir, alterar e visualizar seguem o padrão conforme Figura 39.

Figura 39 - Cadastro de Manutenções (RF14)

The screenshot shows a web application interface for recording maintenance. The main header is 'Duociclo' with navigation links for 'Home', 'Cadastros', and 'Gestão'. A user profile 'Olá Paulinho!' is visible in the top right. The page title is 'Manutenções' and a '+ INCLUIR' button is present. A modal form titled 'Incluir' is open, containing the following fields:

- Descrição:** A text input field.
- Valor:** A text input field containing '0,00'.
- Data de Início:** A date picker field showing 'dd/mm/aaaa'.
- Observação:** A text area containing 'Ex: trocado óleo do motor...'.
- Veículo:** A dropdown menu with the placeholder text 'Selecione o veículo...'.

At the bottom of the modal are two buttons: 'SALVAR' (orange) and 'FECHAR' (grey).

Fonte: Acervo do Autor.

Para realizar o cadastro da manutenção o lojista precisa informar a Descrição, Valor, Data de Início, Observação e Veículo. A lista de veículos só apresentará os veículos disponíveis.

Além de poder gerenciar as manutenções, o lojista pode gerenciar as reservas solicitadas, sendo possível aceitar ou recusar uma reserva na listagem de registros da consulta (Figura 40).

Figura 40 - Rotina de Reservas (RF15)

The screenshot displays the 'Reservas' (Reservations) routine. The header includes 'Duociclo' and 'Olá Paulinho!'. The page title is 'Reservas' and a badge indicates '1 reservas cadastradas'. A search bar is labeled 'Pesquisar pelo Veículo...'. Below is a table with the following data:

ID	Veículo	Cliente	Plano	Data Início	Data Término	Qtde Parcelas	Situação	
1	CG 125 ES	Paulinho Cliente	Mensal	02/11/2023	30/11/2023	1	Pendente	

Fonte: Acervo do Autor (2023).

A consulta de reservas exibe todas as solicitações para os veículos relacionados a loja do lojista logado. Como informação apresenta as colunas de

veículo, cliente, plano, data de início, data de término, quantidade de parcelas e situação da reserva. Caso a situação da solicitação esteja 'Pendente', então deve aparecer as opções para o lojista decidir se irá aceitar ou não. Caso ele clique em aceitar ou recusar irá emitir uma mensagem de confirmação, caso confirme irá realizar a ação conforme selecionado. A consulta poderá ser filtrada de acordo com o nome do veículo no campo de pesquisar. Também é possível visualizar a solicitação da reserva (Figura 41).

Figura 41 - Visualização da Solicitação de Reserva (RF15)



Fonte: Acervo do Autor (2023).

Na visualização da solicitação de reserva exibe as seguintes informações: Veículo, Cliente, Plano, Data de Início, Data de Término, Quantidade Parcelas e Situação. Essas informações são relevantes para o lojista decidir se deverá aceitar a solicitação ou não.

Após aceitar uma solicitação é realizada a geração de um aluguel, podendo ser encontrado no caminho Gestão > Aluguéis, ao clicar o lojista irá ser direcionado a consulta de aluguéis, conforme Figura 42.

Figura 42 - Rotina de Aluguéis (RF13)



Fonte: Acervo do Autor (2023).

Na consulta de aluguéis é possível verificar todos os aluguéis relacionados a loja do lojista logado. Como informação exibe as seguintes colunas: Identificador (ID), Veículo, Cliente, Plano, Valor, Quantidade de Parcelas, Data de Início, Data de Término e Situação. Dispõe ainda das ações de visualizar, parcelas e finalizar. A ação de finalizar dispara uma mensagem de confirmação para o lojista, porém só irá emitir essa mensagem caso todas as parcelas já estejam pagas. A ação de visualizar irá apresentar os dados do aluguel, conforme Figura 43.

Figura 43 - Visualizar Aluguel (RF13)



Fonte: Acervo do Autor (2023).

A ação de visualizar apresenta os dados referente ao aluguel, tais como Veículo, Cliente, Plano, Valor, Quantidade de Parcelas, Data de Início, Data de Término e Situação.

A ação de Parcelas da consulta de aluguel exibe as parcelas relacionadas ao registro selecionado, conforme Figura 44.

Figura 44 - Consulta das Parcelas do Aluguel (RF16)

Sequência	Valor	Data de Vencimento	Situação
1	R\$500,00	10/11/2023	Aberta

Fonte: Acervo do Autor (2023).

A consulta traz como informação as colunas de Sequência, Valor, Data de Vencimento e Situação da Parcela. Possui duas ações em que o lojista pode marcar que a parcela foi paga ou ainda está em aberto. Ao selecionar uma das opções, deve emitir uma mensagem de confirmação para o lojista confirmar. Também deve ter um botão de salvar, onde o lojista pode retornar a consulta de aluguéis.

Além das rotinas de cadastros e gestão, o lojista consegue alterar as informações da sua loja pela opção “Minha Loja” localizada na parte superior direita (Figura 45).

Figura 45 - Minha Loja (RF22)

Fonte: Acervo do Autor (2023).

O lojista pode alterar as informações da loja, como Telefone, E-mail, CEP, Rua, Bairro, Cidade, Estado, N° do Endereço e Complemento do Endereço. O nome

da loja e o CNPJ não ficam habilitados para edição. Estas informações são exibidas para o cliente durante a solicitação da reserva.

Além de alterar os dados da loja, o lojista pode alterar os seus dados, acessando a opção “Meu Perfil” na parte superior direita do menu (Figura 46).

Figura 46 - Meu Perfil (RF23)

Meu Perfil	
Nome	Paulinho Lojista
CPF	10029170933
E-mail	paulolojista@gmail.com
Telefone	47988956613
Data de Nascimento	06/08/2001
CEP	89163467
Rua	Augusto Perfoill
Bairro	Fundo Canoas
Cidade	Rio do Sul
Estado	Santa Catarina
N° Endereço	93
Complemento Endereço	casa
Senha Atual	
Nova Senha	
Confirmar Nova Senha	
SALVAR	

Fonte: Acervo do Autor (2023).

Nessa tela o lojista pode alterar suas informações pessoais, tais como Telefone, CEP, Rua, Bairro, Cidade, Estado N° do Endereço, Complemento do Endereço e a senha. Para alteração da senha é necessário informar a senha atual, após isso deve informar a nova senha e confirmar a nova senha.

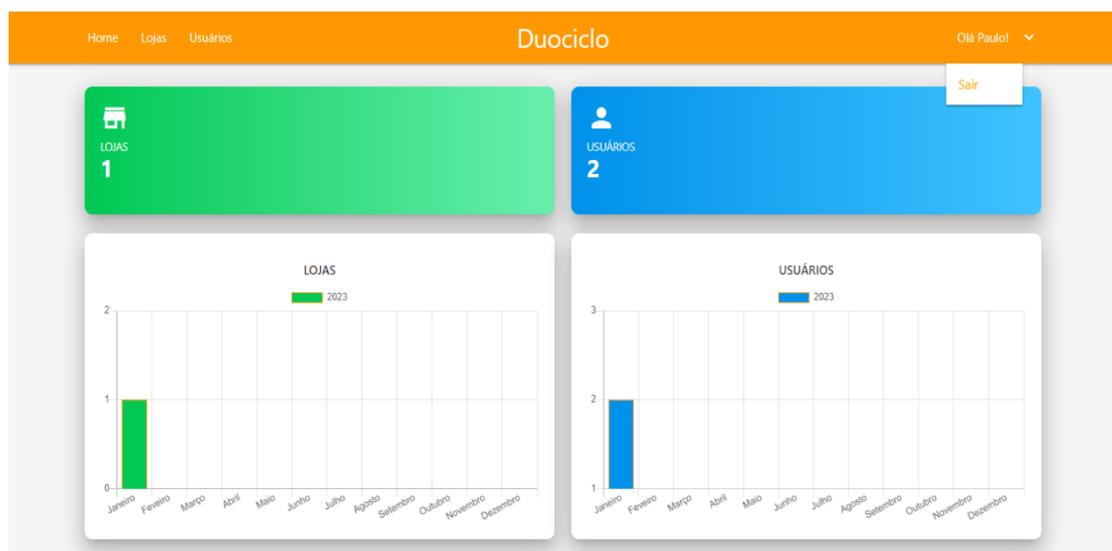
4.2.2.3 Rotinas Administrador

As rotinas do administrador basicamente ficam responsáveis pelo cadastro das lojas e seus respectivos lojistas. Essas rotinas aparecem de acordo com o tipo de login do usuário. Estas rotinas influenciam diretamente no funcionamento do sistema.

Para acessar o protótipo e suas funcionalidades o usuário deve informar o e-mail e senha, com isso as informações são validadas no banco de dados. Caso os dados informados existam no banco de dados, o usuário é direcionado para o menu

principal de acordo com o seu tipo de usuário que está cadastrado, podendo ser Administrador, Lojista ou Cliente. Caso o usuário logado seja do tipo Administrador, ele será direcionado para o menu principal do Administrador, conforme demonstrado na Figura 47.

Figura 47 - Menu do Administrador (RF02, RF24, RF27, RF28, RF29, RF30)



Fonte: Acervo do Autor (2023).

No menu do administrador, na parte superior da tela estão dispostos os acessos para as rotinas de lojas e usuários. O administrador pode fazer logout acessando o atalho pelo seu nome na parte superior direita da tela. Na parte central o administrador encontra os dados referente a quantidade de lojas e usuários que utilizam o protótipo. No card de lojas traz a quantidade total de lojas cadastradas e logo abaixo traz um gráfico apresentando a quantidade de lojas cadastradas por mês, durante o ano atual. A mesmas informações são apresentadas no card e gráfico referente aos usuários da aplicação.

Acessando no menu a rotina de lojas, o administrador poderá cadastrar e visualizar quais são as lojas que contrataram a aplicação para utilização. A rotina de loja pode ser conferida na Figura 48.

Figura 48 - Rotina de Lojas (RF26)

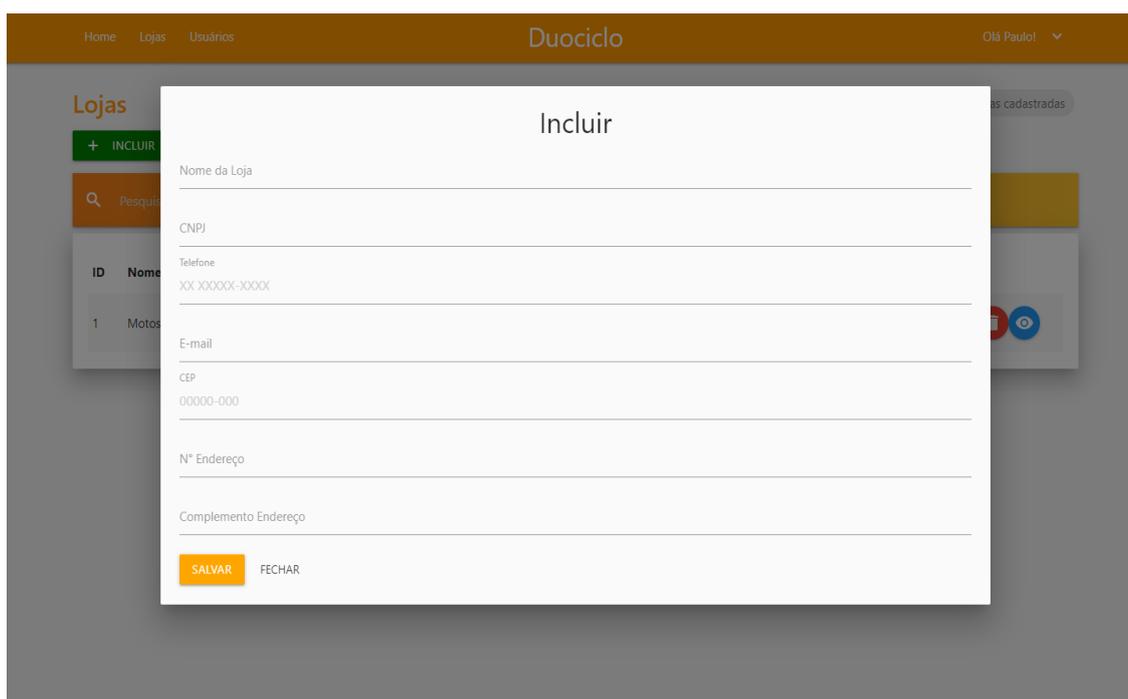


Fonte: Acervo do Autor (2023).

Na consulta de Lojas é possível verificar quais lojas estão utilizando o protótipo, listando uma por uma. Como informação a consulta traz as colunas de identificador (ID), Nome da loja, CNPJ, Telefone, E-mail e CEP da loja. Além dessas informações, o administrador pode utilizar o campo para filtrar as lojas de acordo com o nome.

Para manutenção de lojas, o administrador terá as ações de incluir, alterar, excluir e visualizar. A ação de excluir irá emitir uma mensagem de confirmação antes de executar a exclusão. Já as demais ações seguem o padrão (Figura 49).

Figura 49 - Cadastro da Loja (RF26)



Fonte: Acervo do Autor (2023).

Para realizar o cadastro da loja o administrador deverá informar o Nome da Loja, CNPJ (14 números), Telefone, E-mail (opcional), CEP (8 números), N° do Endereço e Complemento do Endereço.

Além da rotina de Loja, o administrador fica responsável pelo cadastro dos lojistas, que são os vendedores destas lojas. Estes são os responsáveis em aceitar ou recusar as solicitações de reservas, cadastrar pagamento de parcelas etc. A consulta de usuários segue o padrão (Figura 50).

Figura 50 - Rotina de Usuários (RF25)

ID	Nome	Telefone	E-mail	CEP
2	Paulinho Lojista	47988956613	paulolojista@gmail.com	89163467

Fonte: Acervo do Autor (2023).

A consulta de usuários segue o padrão de interface, listando todos os usuários do tipo lojista para que o administrador possa realizar alterações. As colunas que apresentam os dados dos usuários são identificador (ID), nome, telefone, e-mail e CEP. É possível filtrar a consulta através do campo de pesquisa, filtrando pelo nome. As ações disponíveis nessa rotina são a de incluir, alterar, excluir e visualizar. A ação de excluir irá emitir uma mensagem de confirmação para o administrador. As ações de incluir, alterar e visualizar seguem um padrão conforme Figura 51.

Figura 51 - Cadastro de Usuário (Lojista) (RF25)

The image shows a web application interface for user registration. At the top, there is a navigation bar with 'Home', 'Lojas', and 'Usuários' links, the 'Duociclo' logo, and a user profile 'Olá Paulol'. Below this, a modal window titled 'Incluir' is open, displaying a form with the following fields: 'Nome', 'CPF', 'Data de Nascimento' (with a 'dd/mm/aaaa' placeholder and a calendar icon), 'Telefone' (with a 'XX XXXXX-XXXX' placeholder), 'CEP' (with a '00000-000' placeholder), 'Nº Endereço', 'Complemento Endereço', 'E-mail', and 'Loja' (a dropdown menu with 'Selecione a loja...' as the selected option). At the bottom of the form are two buttons: 'SALVAR' (orange) and 'FECHAR' (green). In the background, a table with columns 'ID' and 'Nome' is partially visible, showing a row with '2' and 'Paulo'.

Fonte: Acervo do Autor (2023).

Para que o administrador consiga cadastrar um usuário lojista é necessário informar o campo de Nome, CPF, Data de Nascimento, Telefone, CEP, N° Endereço, Complemento Endereço, E-mail e informar de qual loja o usuário é responsável pela gestão dos aluguéis.

Conforme citado nos tópicos anteriores, o protótipo é responsivo, portanto, as telas se adequam ao tamanho da tela do usuário, sem perder a essência da usabilidade, como pode ser verificado na Figura 52.

Figura 52 - Tela Responsiva



Fonte: Acervo do Autor (2023).

Desta forma, as informações ainda são visíveis, sem atrapalhar a usabilidade do usuário.

5. CONSIDERAÇÕES FINAIS

O protótipo desenvolvido tem como objetivo principal auxiliar o lojista a gerenciar as solicitações de reservas e aluguéis referente aos veículos que ele disponibilizou para os clientes alugarem, permitindo um gerenciamento facilitado e intuitivo para os envolvidos.

Para o desenvolvimento, as tecnologias escolhidas foram suficientes e adequadas para esta primeira versão do protótipo. Para auxiliar no desenvolvimento foi utilizado a *framework* Laravel para o *back-end*. O Laravel possui diversos métodos e um estrutura bem sofisticada para a linguagem em PHP, facilitando em diversos pontos para o desenvolvimento. Para o *front-end* foi utilizado a biblioteca Bootstrap e Materialize, ficando responsáveis pelo CSS das páginas, estilizando as *tags* HTML para melhor usabilidade das telas. Para os comportamentos da tela foi utilizado a linguagem JavaScript, executando comportamentos em tempo de execução. Para o armazenamento das informações foi utilizado o banco de dados MySql.

Dentre os objetivos propostos para este trabalho, o primeiro visava apresentar as tecnologias utilizadas no desenvolvimento do protótipo, sendo detalhado e descrito no capítulo da revisão da literatura.

O segundo objetivo era demonstrar aplicações similares ao do protótipo em questão. Durante as pesquisas foram identificadas três aplicações com funcionalidades similares que foram apresentadas na seção do estado da arte: Loca9motos, Roxmoto e Localiza.

O terceiro objetivo também foi cumprido com o levantamento das funcionalidades e requisitos para o protótipo. Realizado através dos levantamentos de requisitos funcionais, não funcionais, além da definição das regras de negócio, foi possível definir os principais pontos do funcionamento do software. Além destes levantamentos foram elaborados diagramas de classes, diagramas de atividades, diagramas de casos de uso e modelo de dados, todos eles sendo encontrados na seção de Análise.

O quarto e último objetivo visava a construção do protótipo através das tecnologias escolhidas para o desenvolvimento do projeto. O objetivo foi também cumprido e consta detalhado na seção de Implementação, onde são demonstradas todas as rotinas criadas.

O protótipo tem como um de seus benefícios possibilitar o cadastro dos veículos bem como suas marcas e até mesmo os planos disponibilizados pela loja. Traz também informações relevantes ao lojista, como os lucros com os aluguéis durante o ano e os gastos com as manutenções já realizadas nos veículos durante o ano.

Além do ponto de vista do lojista, deve-se destacar as diversas opções que o cliente terá para decidir qual a melhor opção para realizar o seu aluguel, sendo possível escolher a loja, o veículo, o plano e até mesmo a quantidade de parcelas para o pagamento do aluguel. Traz como informativo também os dados referentes ao aluguel em andamento do cliente.

O principal benefício do protótipo é gerar lucros para ambas as partes, lojista e cliente, sendo utilizado principalmente por profissionais no ramo de *delivery*, alugando veículos para que tenham normalmente seu dia de trabalho em ocasiões em que seu veículo pessoal estragou por exemplo. Outro benefício é que o protótipo não fica focado apenas para aluguel de motocicletas, mas sim para qualquer tipo de veículo, pois suas rotinas são genéricas. O principal foco do protótipo é dar circulação a veículos parados em lojas e dar uma oportunidade a clientes de realizarem o aluguel destes veículos.

Por fim, nota-se que os objetivos do trabalho foram atendidos, incluindo o objetivo geral. No entanto, durante o desenvolvimento do protótipo novas ideias e sugestões foram surgindo para melhorar a aplicação, porém não foram implementados devido ao limite de escopo e tempo pré-definidos. Portanto, estas sugestões ficaram como requisitos opcionais e são apresentados como sugestão de trabalhos futuros.

5.1 RECOMENDAÇÕES DE TRABALHOS FUTUROS

Para melhor usabilidade e segurança do software foram levantados alguns requisitos opcionais, apresentados brevemente no Quadro 12 na seção de Análise.

O primeiro requisito opcional (RF31) levantado seria a implementação de rastreamento dos veículos para o lojista, assim podendo acompanhar por onde estão os veículos de suas frotas em tempo real.

O segundo requisito opcional (RF32) seria a opção de pagamento dentro do protótipo, facilitando para o lojista e para o cliente, sendo possível até ter um controle melhor sobre os pagamentos referente aos aluguéis.

O terceiro requisito opcional (RF33) seria a construção de um suporte para a aplicação, auxiliando para eventuais dúvidas para cliente e lojista.

O quarto requisito opcional (RF34) seria um controle para expiração de senha, assim mantendo sempre a segurança dos dados dos clientes e lojistas.

Como quinto requisito opcional (RF35), sugere-se o desenvolvimento de um aplicativo mobile, apresentando as mesmas funcionalidades da aplicação web, porém na palma da mão dos usuários da aplicação.

O sexto requisito opcional (RF36) visa a implementação de envio de e-mail para os usuários da aplicação, informando sobre aluguéis, parcelas e solicitações de reservas. Além disso seria importante para caso de redefinição de senha.

O sétimo requisito opcional (RF37) seria especialmente para os lojistas, que poderiam cadastrar os clientes pessoalmente em sua loja. Seria para aqueles clientes que desejam alugar, porém ainda não tem cadastro na aplicação.

O oitavo requisito opcional (RF38), visa a implementação de um cadastro de aluguéis para que os lojistas possam realizar estas contratações pessoalmente, sem que o cliente precise realizar uma solicitação de reserva.

Para o nono requisito opcional (RF39) sugere-se o desenvolvimento de uma configuração para que o lojista defina uma taxa de juros por dia para parcelas atrasadas de um contrato de aluguel.

O décimo requisito opcional (RF40) pretende desenvolver uma tela para que o lojista altere sua senha “padrão” no seu primeiro acesso ao sistema, visando maior segurança para seus usuários.

O décimo primeiro requisito opcional (RF41) tem como intuito criar um atalho para notificações do lojista, apresentando-o informações como novas solicitações de reserva, parcelas atrasadas de determinado aluguel, etc.

O décimo segundo requisito opcional (RF42) pretende disponibilizar notificações ao cliente, através de um atalho apresentando dados como aceite e ou rejeições das solicitações de reserva, proximidade de vencimento de parcelas, etc.

Como décimo terceiro requisito opcional (RF43) visa a implementação de uma configuração para um plano de aluguel diário, onde o lojista irá definir o preço diário

do aluguel e o cliente poderá realizar o aluguel dinâmico, definindo uma data de início e data de término.

O décimo quarto requisito opcional (RF44) seria a opção de cancelamento da solicitação da reserva, onde caso a solicitação não tenha sido aceita ainda o cliente poderá cancelar o pedido. Outro cenário será quando a data de início da solicitação tenha passado da data atual, o sistema deverá cancelar automaticamente a solicitação.

O décimo quinto e último requisito opcional (RF45) visa o controle do lojista referente a retirada e devolução do veículo para o cliente. O lojista poderá cadastrar a data de retirada e na finalização do aluguel irá definir a data de devolução do veículo alugado.

Como melhoria, as consultas poderão ter filtros específicos, podendo escolher qual coluna o usuário deseja filtrar, deixando mais dinâmico e prático ao filtrar os registros.

Para melhorar o cadastro do cliente, disponibilizar um campo para anexar a imagem da carteira de motorista, com isso o lojista fica ciente que o cliente é habilitado e já confere se as informações do cadastro batem com o documento do cliente.

Outra melhoria seria implementar para que o cliente possa ser Pessoa Física ou Pessoa Jurídica, ajustando para que o cliente possa alugar mais que um veículo. Importante para casos em que estabelecimentos que trabalham com *delivery*, como lanchonetes por exemplo, possam alugar os veículos para seus funcionários efetuarem as entregas.

Além disto, sugere-se a realização de testes e homologação da aplicação, sendo realizados com o auxílio de lojistas parceiros e clientes.

REFERÊNCIAS

- ALVES, William Pereira. **Banco de dados: teoria e desenvolvimento**. 2. São Paulo Erica 2020.
- ALVES, William Pereira. **Banco de dados**. São Paulo Erica 2014.
- ALVES, William Pereira. **Html & Css: aprenda como construir páginas web**. São Paulo Expressa 2021.
- CARDOSO, Leandro da Conceição. **Frameworks back end**. São Paulo Platos Soluções Educacionais 2021.
- CHART.JS. **Chart.js: Simple yet flexible JavaScript charting for designers & developers**. 2023. Disponível em: <<https://www.chartjs.org/>>. Acesso em: 02 nov. 2023.
- FAUST, Dianna. **O Crescimento do Delivery no Brasil em 2021**. (2021). Disponível: <<https://help.grandchef.com.br/o-crescimento-do-delivery-no-brasil-em-2021/>>. Acessado em: 22 jun. 2023.
- FLANAGAN, David. **JavaScript: o guia definitivo**. 6. Porto Alegre Bookman 2014.
- GABARDO, Ademir Cristiano. **Laravel para Ninjas**. São Paulo Novatec Editora Ltda 2017.
- LOCALIZA. **Aluguel de Carros com a maior frota do Brasil**. 2023. Disponível em: <https://www.localiza.com/brasil/pt-br>. Acesso em: 21 ago. 2023.
- LOCA9MOTOS. **Home**. 2022. Disponível em: <https://loca9motos.com.br/>. Acesso em: 20 ago. 2023.
- MACHADO, Felipe Nery Rodrigues. **Banco de dados: projeto e implementação**. 4. São Paulo Erica 2020.
- MANZANO, José Augusto N. G. **MySQL 5.5 interativo: guia essencial de orientação e desenvolvimento**. São Paulo Erica 2011.
- MILETTO, Evandro Manara. **Desenvolvimento de software ii: introdução ao desenvolvimento web com html, css, javascript e php**. Porto Alegre Bookman 2014.
- MORAIS, Izabelly Soares de. **Engenharia de software**. Porto Alegre SAGAH 2020
- PAULA FILHO, Wilson de Pádua. **Engenharia de software, v. 1: produtos**. 4. Rio de Janeiro LTC 2019.
- PICHETTI, Roni Francisco Vida. **Banco de dados**. Porto Alegre SAGAH 2021.

PRABHU, A; SHENOY, A. **CSS framework alternatives**. Nova York: Apress, 2018.

PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. 9. Porto Alegre AMGH 2021.

REINEHR, Sheila. **Engenharia de requisitos**. Porto Alegre SAGAH 2020

ROXMOTO. **Roxmoto**. 2023. Disponível em: <https://www.roxmoto.com.br/rox-moto/>. Acesso em 20 ago. 2023.

SOARES, Wallace. **Php 5: conceitos, programação e integração com banco de dados**. 7. São Paulo Erica 2013.

STOCHERO, Tahiane. **Com pandemia, número de profissionais de motofrete cresce 40% em um ano na cidade de SP**. (2021). Disponível: <<https://g1.globo.com/sp/sao-paulo/noticia/2021/03/07/apos-um-ano-de-pandemia-numero-de-profissionais-de-motofrete-cresce-40percent-em-um-ano-na-cidade-de-sp.ghtml>>. Acessado em: 22/06/2023.

TERUEL, Evandro Carlos. **Html 5: guia prático**. 2. São Paulo Erica 2014.

ZABOT, Diego. **Aplicativos com bootstrap e angular: como desenvolver APPs responsivos**. São Paulo Erica 2020.