

**MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA FARROUPILHA
CAMPUS SANTO ÂNGELO**

HAVAÍ PARQUE.RESERVE

TRABALHO DE CONCLUSÃO DE CURSO

Felipe de Oliveira Lyra

**Santo Ângelo, RS, Brasil.
2023**

HAVAÍ PARQUE.RESERVE

por

Felipe de Oliveira Lyra

**Monografia apresentada ao Instituto Federal de Educação Ciência e Tecnologia
Farroupilha, como requisito parcial para obtenção do título de Tecnólogo em Sistemas
para Internet.**

Orientadora: Marta Breunig Loose

Santo Ângelo, RS, Brasil

2023

**Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação Ciência e Tecnologia Farroupilha**

A Comissão Examinadora, abaixo assinada,
aprova a Monografia

HAVAÍ PARQUE.RESERVE

elaborada por
Felipe de Oliveira Lyra

como requisito parcial para obtenção do título de
Tecnólogo em Sistemas para Internet

COMISSÃO EXAMINADORA

Me. Marta Breunig Loose
(Orientadora)

Fábio Weber Albiero, Mestre. (IFFAR)

Juliano Gomes Weber, Mestre. (IFFAR)

Conceito Final: _____

Santo Ângelo, 15 de dezembro de 2023.

Whatever happens, happens.
(AUTOR DESCONHECIDO)

AGRADECIMENTOS

A minha orientadora, que esteve sempre à disposição para me auxiliar quando necessário, do início ao fim do curso.

A minha família, que me apoiou e me incentivou.

Aos meus amigos, que me auxiliaram e também entenderam o motivo da minha ausência em certas ocasiões.

A minha psicóloga que me auxiliou durante este tempo.

E agradeço a todos que me apoiaram e me motivaram até agora.

RESUMO

HAVAÍ PARQUE.RESERVE

AUTOR(A): FELIPE DE OLIVEIRA LYRA

ORIENTADOR(A): MARTA BREUNIG LOOSE

Data e Local da Defesa: Santo Ângelo, 15 de dezembro de 2023.

Com o intuito de estabelecer uma estrutura mais eficiente para as atividades diárias, visando aprimorar a organização de anotações e reservas, apresenta-se o presente Trabalho de Conclusão de Curso. O objetivo é proporcionar um desempenho aprimorado e uma maior organização para todos os envolvidos. Direcionado ao domínio de hospedagem e gestão, o sistema *web* denominado Havaí Parque.Reserve capacita os usuários a exercerem controle sobre hospedagens e a comercialização de pulseiras para visitas diárias. Com o propósito de otimizar o tempo e a gestão de dados, foi desenvolvido o referido sistema *web*.

Por fim, para a concepção desse sistema *web*, foram utilizadas diversas ferramentas com o intuito de auxiliar no processo de programação. A linguagem de programação adotada foi o PHP, sendo integrada ao framework CodeIgniter para facilitar a codificação.

Palavras-chave: Reserva; Programação; Sistema Web.

ABSTRACT

In order to establish a more efficient structure for daily activities, with a view to improving the organization of notes and reservations, this Course Conclusion Paper is presented. The aim is to provide improved performance and greater organization for everyone involved. Aimed at the accommodation and management domain, the web system called Havaí Parque.Reserve enables users to exercise control over accommodation and the sale of wristbands for daily visits. In order to optimize time and data management, this web system was developed.

Finally, to design this web system, various tools were used to help with the programming process. The programming language adopted was PHP, which was integrated with the CodeIgniter framework to facilitate coding.

Keywords: Reserve; Programming; Web system.

LISTAS DE ABREVIATURAS E SIGLAS

ARPA - Advanced Research Projects Agency.

ARPANET - Advanced Research Projects Agency Network.

TCC - Trabalho de Conclusão de Curso.

HTML - HyperText Markup Language.

PHP - Hypertext Preprocessor.

CSS - Cascading Style Sheet.

POO - Programação Orientada a Objeto.

ETC - Et Cetera (abreviatura da expressão em latim, que significa “e assim por diante”).

BD - Banco de Dados.

SGBD - Sistema Gerenciador de Banco de Dados.

MVC - Model-View-Controller.

SIT - Sistemas de Informações Transacionais.

SIG - Sistemas de Informações Gerenciais.

SAD - Sistemas de Apoio a Decisões.

JVM - Java Virtual Machine.

DER - Diagrama de Entidade e Relacionamento.

Diagrama ER - Diagrama de Entidade e Relacionamento.

HQL - Hibernate Query Language.

API - Application Programming Interface.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 - Airbnb..... | 16 |
| Figura 2 - Hurb tela inicial..... | 17 |
| Figura 3 - Hurb tela de seleção..... | 17 |
| Figura 4 - Sesc..... | 18 |
| Figura 5 - Casos de uso..... | 23 |
| Figura 6 - Diagrama de sequência do caso de uso gerenciamento de cadastro... 29 | |
| Figura 7 - Diagrama de sequência do caso de uso gerenciamento de reserva.... 30 | |
| Figura 8 - Diagrama de sequência do caso de uso gerenciamento de venda...31 Fonte: do autor (2023)..... | 31 |
| Figura 9 - Diagrama de sequência do caso de uso emitir relatório..... | 31 |
| Figura 10 - Diagrama de sequência do caso de gerenciar residência..... | 32 |
| Figura 11 - Diagrama ER..... | 33 |
| Figura 12 - Modelo Lógico..... | 34 |
| Figura 13 - Diagrama de classes..... | 36 |
| Figura 14 - Tela inicial..... | 37 |
| Figura 15 - Tela de cadastro de cliente (imagem aproximada)..... | 38 |
| Figura 16 - Tela de cadastro de administrador (imagem aproximada)..... | 39 |
| Figura 17 - Tela de cadastro de residência (imagem aproximada)..... | 39 |
| Figura 18 - Tela de Venda Avulsa (imagem aproximada)..... | 40 |
| Figura 19 - Tela de reserva (imagem aproximada)..... | 41 |
| Figura 20 - Tela de lista de clientes (imagem aproximada)..... | 42 |
| Figura 21 - Tela de listagem administradores (imagem aproximada)..... | 42 |
| Figura 22 - Tela de listagem de vendas (imagem aproximada)..... | 42 |
| Figura 23 - Tela de busca de diárias..... | 43 |
| Figura 24 - Código de busca utilizado..... | 44 |
| Figura 25 - Função de exclusão part.1..... | 44 |
| Figura 26 - Função de exclusão part.2..... | 45 |
| Figura 27 - Função de cálculo part.1..... | 45 |
| Figura 28 - Função de cálculo part.2..... | 46 |
| Figura 29 - Contagem de dias part.1..... | 46 |
| Figura 30 - Contagem de dias part.2..... | 47 |
| 3 RESULTADOS..... | 48 |
| Figura 29 - Primeira pergunta..... | 48 |
| Figura 30 - Segunda Pergunta..... | 49 |
| Figura 31 - Terceira Pergunta..... | 49 |
| 4. CONCLUSÃO..... | 50 |
| REFERÊNCIAS..... | 51 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 - Especificação de caso de uso Gerenciar Cadastro..... | 21 |
| Tabela 2 - Especificação de caso de uso Gerenciar Reserva..... | 22 |
| Tabela 3 - Especificação de caso de uso Gerenciar Reserva..... | 23 |
| Tabela 4 - Especificação de caso de uso Emitir Relatório..... | 24 |
| Tabela 5 - Especificação de caso de uso Gerenciar Residência..... | 24 |

SUMÁRIO

| | |
|--|-----------|
| 1 INTRODUÇÃO..... | 14 |
| 1.1 Objetivos..... | 15 |
| 1.1.1 Objetivo Geral..... | 15 |
| 1.1.2 Objetivos Específicos..... | 15 |
| 1.2 Trabalhos Correlatos..... | 16 |
| 1.2.1 Airbnb..... | 16 |
| 1.2.2 Hurb..... | 16 |
| 1.2.3 Sesc..... | 17 |
| 1.2.4 Comparativo entre os trabalhos correlatos e o trabalho desenvolvido..... | 18 |
| 1.3 Fundamentação teórica..... | 19 |
| 1.3.1 Programação Orientada a Objeto..... | 19 |
| 1.3.2 PHP..... | 19 |
| 1.3.3 MySQL..... | 20 |
| 1.3.4 MVC..... | 21 |
| 1.3.5 HTML e CSS..... | 21 |
| 1.3.7 CodeIgniter..... | 22 |
| 1.3.8 Bootstrap..... | 22 |
| 2. DESENVOLVIMENTO..... | 23 |
| 2.1 Diagrama de casos de uso..... | 23 |
| 2.1.1 Especificação de casos de uso..... | 24 |
| 2.2 Diagrama de sequência..... | 29 |
| 2.3 Banco de dados..... | 32 |
| 2.3.1 Diagrama de entidade e relacionamento..... | 33 |
| 2.3.2 Modelo lógico..... | 34 |
| 2.3.3 Código de criação do banco de dados..... | 35 |
| 2.4 Diagrama de Classes..... | 36 |
| 2.5 Telas do sistema..... | 37 |
| 2.5.1 Tela inicial..... | 37 |
| 2.5.2 Tela de cadastro de cliente/usuário..... | 38 |
| 2.5.3 Tela de cadastro de administrador/funcionário..... | 39 |
| 2.5.4 Tela de cadastro de residência..... | 39 |
| 2.5.4 Tela de venda avulsa..... | 39 |
| 2.5.4 Tela de reserva/locação..... | 41 |
| 2.5.6 Tela de visualização de clientes..... | 41 |
| 2.5.7 Tela de listagem de administradores/funcionários..... | 42 |
| 2.5.8 Tela de listagem de vendas..... | 42 |
| 2.5.9 Tela de busca de diárias disponíveis..... | 42 |
| 2.5.10 Código utilizado para a realização da busca..... | 43 |
| 2.5.11 Função para a confirmação de exclusão..... | 44 |

| | |
|---|-----------|
| 2.5.12 Função de automático na venda..... | 45 |
| 3 Resultados..... | 46 |
| 4. Conclusão..... | 48 |
| REFERÊNCIAS..... | 49 |

1 INTRODUÇÃO

O início da Internet deu-se durante a Guerra Fria, nos Estados Unidos. Com a intenção de criar uma rede para disseminar informações importantes e valiosas. A internet inicialmente foi chamada de ARPANET, com função de criar uma comunicação entre os laboratórios de pesquisa e distribuir informações. A desenvolvedora desta rede foi a ARPA.

As informações eram distribuídas por meio da estrutura de rede, backbone, servindo como transmissor de dados em uma rede local, distribuindo assim, as informações entre todos os dispositivos conectados. Rede esta, que era de uso exclusivo militar. Com o tempo, distribuída para civis e para outros países. O avanço da internet ou sistemas vinculados a ela, vem com o tempo, priorizando sempre a resolução dos novos problemas encontrados, facilitando assim a vida de todos aqueles que a utilizam.

Tendo em vista o avanço tecnológico, foram desenvolvidos diversos tipos de sistemas, sendo eles para *desktop*, *web*, aplicativos, etc. Com o grande número de dados armazenados, neste e em qualquer outro tipo de software ou sistema *web*, são chamados de sistemas de informação.

Segundo (O' Brien, 2000) os sistemas de informações podem ser divididos em quatro tipos: Sistema de Informações Transacionais (SIT), Sistemas de Informações Gerenciais (SIG), Sistema de Apoio a Decisões Executivas (SAD).

SIT são sistemas que contém informações cotidianas, pedidos, compra de mercadorias, emissão de notas fiscais. São informações que são incluídas no banco de dados para futuras consultas. Já no SIG são geradas informações de apoio para a tomada de decisões administrativas, produzindo relatórios para gerência. O SAD é um tipo de sistema que gera um constante apoio interativo nas tomadas de decisões, são decisões complexas como impostos. O SIE possui funções dos sistemas anteriores, Sistema de Apoio à Decisão e o Sistema de Informações Gerenciais.

Dentro deste contexto, possuindo como finalidade facilitar de diversas formas e de resolver novos problemas encontrados, foi observado, em um

parque aquático, a falta de organização e controle de recebimento de seus clientes, os quais possuem reserva no local.

Portanto, pensando em minimizar a falta de controle e organização do processo, para o presente trabalho de conclusão de curso (TCC), será desenvolvido um sistema para gerenciamento de reservas, facilitando assim, para quem possui interesse no local e auxiliando no controle das reservas do parque aquático.

Pensando no desenvolvimento do sistema, a linguagem de programação escolhida, na parte do *back-end*, foi PHP utilizando juntamente o *framework* CodeIgniter, contendo o uso do banco de dados MySQL e para a parte estética, o front-end, HTML, CSS E JavaScript.

1.1 Objetivos

Esta subseção apresenta os objetivos gerais e específicos do presente trabalho.

1.1.1 Objetivo Geral

Desenvolver um Sistema Web para realização de reservas de cabanas, casas e apartamentos, sendo exclusivo para o Havaí Parque Aquático auxiliando no controle e gerenciamento dos dados.

1.1.2 Objetivos Específicos

- Fornecer cadastro de acesso;
- Pesquisar diárias disponíveis para reserva;
- Possibilitar a reserva de residências;
- Realizar vendas de pulseiras para entrada no parque;
- Permitir a visualização de fotos do local e
- Proporcionar interface simples e fácil para o uso.

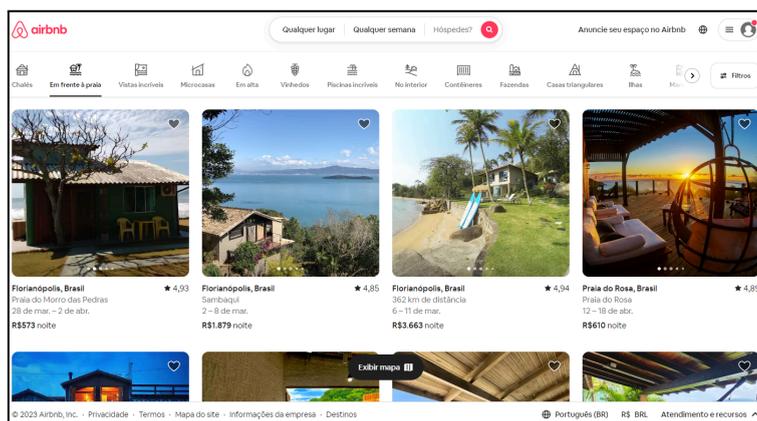
1.2 Trabalhos Correlatos

Nesta subseção será apresentado os trabalhos correlatos referente ao sistema desenvolvido.

1.2.1 Airbnb

Airbnb é uma plataforma de hospedagens, permitindo que pessoas disponibilizem ou realizem reservas de acomodações. Suas hospedagens possuem um grande tipo de variedade, sendo elas, aluguel de casa, apartamentos, quartos entre outros. Segue abaixo na figura 1, um exemplo de tela com a visualização dos locais disponíveis para reservas.

Figura 1 - Airbnb.



Fonte: Airbnb¹

1.2.2 Hurb

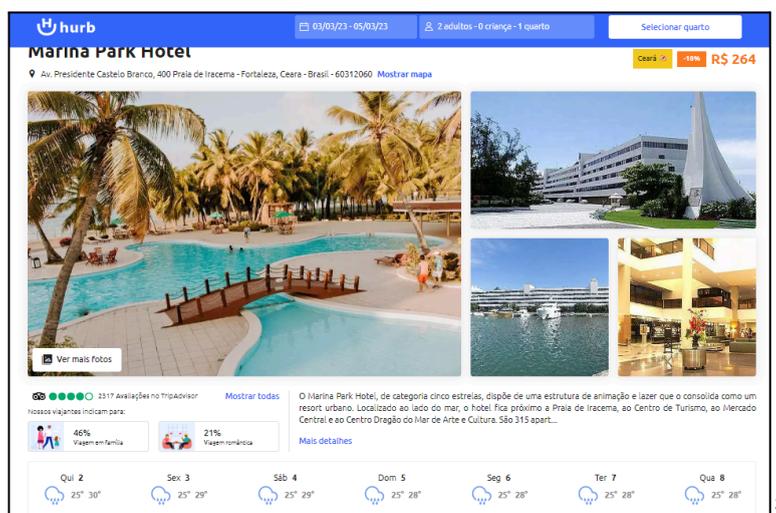
Hurb é uma plataforma de venda de pacotes de viagens e turismos. Possibilita a escolha de diversos pacotes de viagens com diferentes casas, hotéis, resorts entre outros. Seguem abaixo as Figuras 2 e 3, para a visualização do site.

Dentro dos pacotes de viagens, é possível visualizar as datas, refeições inclusas no pacote, como café da manhã, almoço ou jantar, assim como fotos

¹ <https://www.airbnb.com.br>

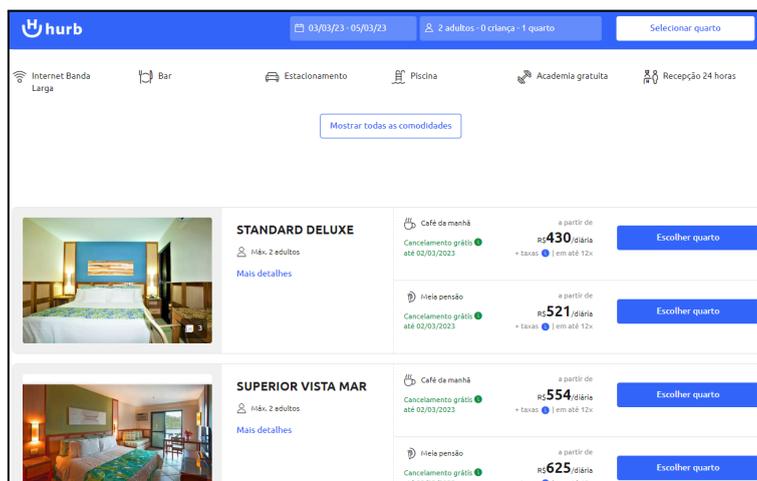
do local desejado. Ao conferir cada pacote de viagem, pode-se verificar a previsão do tempo do local.

Figura 2 - Hurb tela inicial.



Fonte: Hurb²

Figura 3 - Hurb tela de seleção.



Fonte: Hurb²

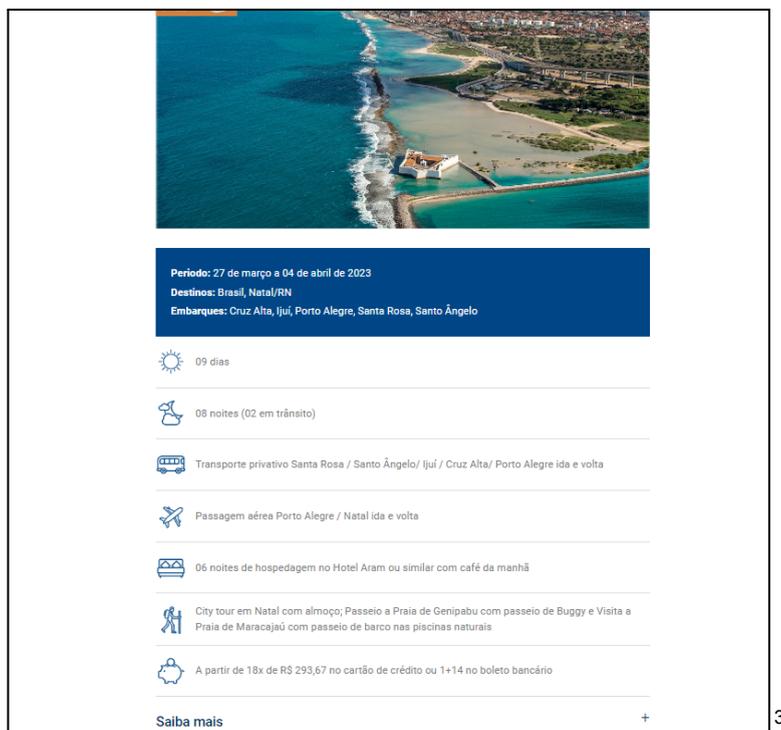
1.2.3 Sesc

Sesc Pacotes de Turismo é um site onde é tem-se a liberdade de efetuar a reserva de pacotes de turismo, possuindo dias pré definidos, como dia de início e final de viagem.

² <https://www.hurb.com/br>

Dentro do pacote pode-se visualizar o que o mesmo tem a oferecer, tal como a duração de viagem, meios de transporte para a chegada do local, entre outras informações situadas no anúncio do pacote selecionado, conforme mostrado na Figura 4.

Figura 4 - Sesc.



Período: 27 de março a 04 de abril de 2023
 Destinos: Brasil, Natal/RN
 Embarques: Cruz Alta, Ijuí, Porto Alegre, Santa Rosa, Santo Ângelo

- ☀️ 09 dias
- 🛏️ 08 noites (02 em trânsito)
- 🚗 Transporte privativo Santa Rosa / Santo Ângelo/ Ijuí / Cruz Alta/ Porto Alegre ida e volta
- ✈️ Passagem aérea Porto Alegre / Natal ida e volta
- 🏠 06 noites de hospedagem no Hotel Aram ou similar com café da manhã
- 👣 City tour em Natal com almoço; Passeio a Praia de Genipabu com passeio de Buggy e Visita a Praia de Maracajá com passeio de barco nas piscinas naturais
- 💰 A partir de 18x de R\$ 293,67 no cartão de crédito ou 1+14 no boleto bancário

Saiba mais +

Fonte: Sesc³

1.2.4 Comparativo entre os trabalhos correlatos e o trabalho desenvolvido

O sistema *web* foi desenvolvido com foco em atender as demandas de um local específico, o Havaí Parque Aquático, tendo assim elementos que facilitam no entendimento do mesmo, como interface simples, de maneira que o usuário não tenha dificuldade em sua utilização. Tendo como vantagem o desenvolvimento em foco para suas necessidades, possuindo um relatório de suas reservas. Além disso, o sistema será disponibilizado gratuitamente, além de ser código aberto.

³ <https://www.sesc-rs.com.br/destinos/rs/>

1.3 Fundamentação teórica

Nesta seção serão apresentados os conceitos básicos utilizados para o desenvolvimento do trabalho.

1.3.1 Programação Orientada a Objeto

Programação Orientada a Objeto ou POO é um paradigma de programação, que tem como idéia central realizar a programação em módulos independentes, onde cada um contém dados e funcionalidades relacionadas a um objeto do mundo real.

A POO está crescendo cada vez mais nos dias de hoje, permitindo programas mais modulares, flexíveis e escaláveis, reduzindo assim a complexidade do código, gerando um entendimento mais fácil para os desenvolvedores.

É considerado um paradigma no qual é preciso projetar o mundo real aos olhos do computador, criando assim objetos, atributos, métodos e etc. Atualmente as linguagens de programação mais utilizadas, mundialmente, possuem o paradigma como base. Entre as vantagens de utilizar este paradigma estão: a reutilização do código, que seria o reaproveitamento do trabalho realizado anteriormente; o encapsulamento, onde permite que o programador esconda detalhes internos dos objetos, fornecendo apenas a funcionalidade; o polimorfismo onde é permitido criar interfaces comuns para objetos distintos; entre outras vantagens (Indaial, 2014).

1.3.2 PHP

O PHP ou *Hypertext Preprocessor*, conhecido como uma linguagem de programação que utiliza o paradigma de programação orientada a objeto. É uma linguagem amplamente utilizada para o desenvolvimento *web*, sendo muito dinâmico e interativo. Foi criado por Rasmus Lerdorf em 1994, apareceu pela primeira vez em um conjunto de binários Common Gateway Interfaces,

escrito em C. Utilizado originalmente para auto monitoramento de currículo online (Dall'Oglio, 2007).

O PHP acabou se tornando uma linguagem muito popular e amplamente utilizada para o desenvolvimento *web*, adotado em diversos projetos, sejam eles grandes ou simples. Uma de suas principais características é o Server-Side Scripting, que em português significa Script do lado do servidor. O PHP é executado no servidor *web* e o resultado é enviado para o usuário, isto é, trabalha de forma camuflada junto ao servidor.

Dentre os principais motivos da utilização do PHP para linguagem de programação está a evolução constante, a flexibilidade, a compatibilidade, o desenvolvimento rápido pois o mesmo possui uma sintaxe simples e por ser de código aberto, sendo assim, gratuito para ser utilizado.

1.3.3 MySQL

Foi desenvolvido na Suécia por David Axmark, Allan Larsson e Michael Widenius. O projeto teve início no ano de 1980, mas teve seu lançamento no ano de 1995. O MySQL é um banco de dados relacional no modelo de cliente-servidor, sendo ele gratuito e de código aberto, atualmente o banco de dados mais popular da Oracle.

Dentre as características do MySQL estão o seu desenvolvimento que foi realizado em C e C++, possuindo funcionalidade em diversos sistemas operacionais diferentes e também sendo um bando de dados relacional. Um banco de dados relacional seus dados são armazenados de forma separada, em tabelas, proporcionando assim uma grande velocidade e flexibilidade (MySQL, 2010).

1.3.4 MVC

MVC é a sigla de *Model View Controller*, é um padrão de arquitetura de software amplamente utilizado no desenvolvimento de aplicativos. O padrão MVC é separado em três camadas distintas, sendo elas:

Model é representada pela apresentação da lógica do negócio e dos dados do aplicativo, responsável também por manipular e armazenar dados do aplicativo, fornecendo também acesso e permitindo a atualização dos dados inseridos anteriormente.

View consiste na camada entregue ao usuário, responsável por exibir os dados já modelados para o usuário e realizar a coleta de dados fornecidas pelo usuário, enviando - as para o controller.

Controller é responsável por realizar a comunicação entre as duas camadas, View e Model. Designado a repassar as informações necessárias para as cada uma das camadas, processando solicitações e respostas.

1.3.5 HTML e CSS

HTML, *HyperText Markup Language* ou em português Linguagem de Marcação de Hipertexto, por sua vez, não é considerado uma linguagem de programação. Utilizado na construção de páginas, utilizando assim elementos para delimitar ou agrupar diferentes tipos de conteúdos.

Tendo como conceito Hipertexto é a forma de como se organiza o conteúdo, conjunto de elementos ou nós. O HTML foi criado para ser utilizado de forma independente, sendo possível visualizar seu conteúdo em diversos *browsers* (Ferreira; Eis, 2012).

CSS *Cascading Style Sheet* ou em português Folha de Estilo em Cascata, é utilizada para estilizar o website, especificando os elementos que irão sofrer a alteração de estilo, alterações estas que podem ser, cores, características, *margins*, *padding*s, entre outras estruturas.

O CSS realiza o trabalho de formatar a informação para realizar a entrega para o HTML, capaz de formatar qualquer elemento. De forma geral, a maior parte dessas formatações são realizadas visíveis (Ferreira; Eis, 2012).

1.3.7 CodeIgniter

O CodeIgniter é um *framework* de aplicação *web* de código aberto escrito em PHP. Projetado para facilitar o desenvolvimento de aplicativos da *web*, seguindo o padrão arquitetural *Model-View-Controller*. O seu principal objetivo é fornecer uma estrutura leve e simples, que permite os desenvolvedores criarem aplicativos *web* de forma rápida e eficiente.

CodeIgniter tem uma documentação completa que auxilia os desenvolvedores a entender e usar suas funcionalidades de maneira correta e eficiente. Possui também uma comunidade ativa, fornecendo assim, suporte e recursos adicionais.

Originalmente desenvolvido por EllisLab. Mantido durante anos pela equipe de desenvolvedores da ExpressionEngine e por um grupo da comunidade denominado Reactor Team (CodeIgniter, 2020).

1.3.8 Bootstrap

O Bootstrap é um *framework* de código aberto, tendo o foco em CSS, utilizado amplamente para o desenvolvimento de sites e aplicativos *web*. Foi criado pela equipe de desenvolvimento do Twitter, mas agora é mantido por uma pequena equipe de desenvolvedores no GitHub. O Bootstrap é atualmente um dos mais populares frameworks, utilizados em projetos de código aberto.

Atualmente seus desenvolvedores vem melhorando a base de códigos, melhorando componentes existentes e seus recursos, adicionando tecnologias melhores, trocando assim o jQuery para JavaScript.

O *framework* fornece um conjunto de ferramentas, estilos e recursos predefinidos para a utilização, facilitando assim a criação de interfaces de usuário, responsivas e modernas.(Bootstrap, 2022).

1.3.9 OpenWeatherMap API

Utilizando esta API é possível realizar a visualização de dados relacionados a previsão de tempo hiperlocal, que são pequenas proximidades. Podendo também realizar consultas como desejar, selecionando os dias que deseja estas informações.

A API do OpenWeatherMap consiste em cálculos de dados meteorológicos utilizando qualquer localização. Para obter os dados meteorológicos, é utilizado diversas fontes de dados, como radares, modelos de agências meteorológicas globais, satélites meteorológicos e uma vasta rede de estações meteorológicas (OpenWeather, 2020).

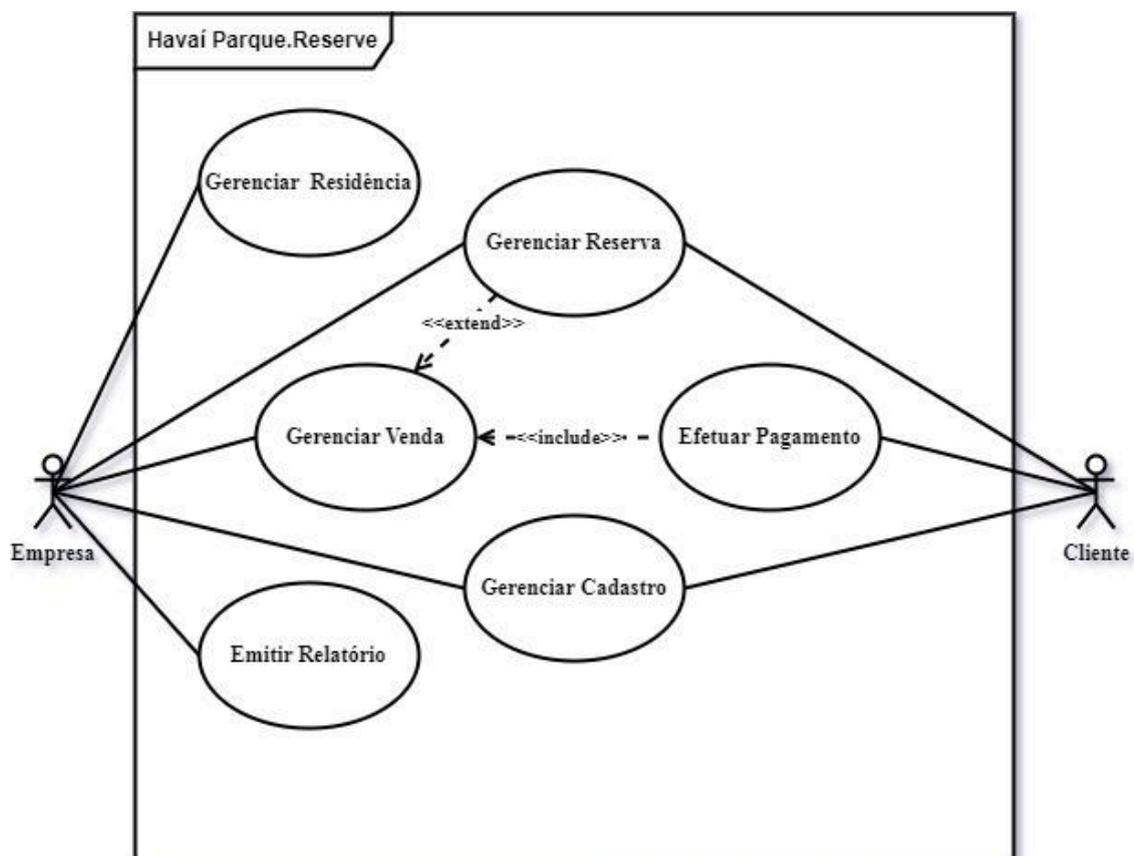
2. DESENVOLVIMENTO

Neste capítulo serão apresentados os seguintes diagramas: casos de uso, especificações de casos de uso, sequência, entidade e relacionamento, lógico e de classes.

2.1 Diagrama de casos de uso

Nesta seção, é realizada a representação das funcionalidades do sistema por meio do diagrama de casos de uso. No mesmo constam as funcionalidades de gerenciar login, reserva para ambos atores (cliente e responsável do parque), possuindo também a gerência de venda, sendo ela exclusiva para o parque, assim como o relatório. Segue abaixo, na figura 5, o diagrama de casos de uso.

Figura 5 - Casos de uso.



Fonte: do autor(2023).

2.1.1 Especificação de casos de uso

Nesta seção, serão apresentadas as especificações dos Casos de Uso, contendo informações mais detalhadas sobre o seu fluxo de funcionamento.

Tabela 1 - Especificação de caso de uso gerenciar cadastro.

| | |
|---|--|
| Identificação do Caso de uso: UC001 | Nome do Caso de uso: Gerenciar cadastro |
| Atores: Empresa, Cliente | |
| Resumo: Descreve os passos realizados para efetuar o cadastro/login | |
| Tipo: Primário | |
| Pré-Condições: Nenhuma | |
| Pós-Condições: Nenhuma | |
| Fluxo Básico | |
| Ações do ator | Ações do sistema |
| 1. Realizar novo cadastro. 2. Inserir informações para efetuar o cadastro, nome, CPF, e-mail, contato, endereço. | 3. Grava informações no banco. |
| Fluxo de exceção I - Dados já cadastrados. | |
| Ações do ator | Ações do sistema |
| | 1. Informar que dados duplicidade de dados. |
| Fluxo alternativo - Recuperação de cadastro | |
| Ações do ator | Ações do sistema |

| | |
|---|---|
| 1. Solicitar a recuperação do cadastro. 3. Informa os dados necessários. | 2. Solicita informações necessárias para a recuperação, CPF e e-mail. 3. Altera informações do cadastro. |
| Fluxo de exceção - Dados em Falta ou Inválidos | |
| Ações do ator | Ações do sistema |
| | 1. Retorna ao formulário com alerta de dados incorretos ou inexistentes. |

Tabela 2 - Especificação de caso de uso gerenciar reserva.

| | |
|--|---|
| Identificação do Caso de uso: UC002 | Nome do Caso de uso: Gerenciar reserva |
| Atores: Empresa, cliente | |
| Resumo: Descreve os passos realizados na gerência de reserva. | |
| Tipo: Primário | |
| Pré-Condições: Possuir um cadastro. | |
| Pós-Condições: | |
| Fluxo Básico | |
| Ações do ator | Ações do sistema |
| 1. Seleciona o local desejado. 3. Informa os dias desejados para a reserva. 4. Informa a quantidade de pessoas que ocuparão a casa. 5. Confirma a reserva. 8. Efetua pagamento | 2. Redireciona para a página da visualização detalhada da residência. 6. Solicita forma de pagamento. 7. Armazena dados no banco de dados. 9. Emite comprovante. |

| | |
|--|--|
| | |
| Fluxo de exceção - Indisponibilidade. | |
| Ações do ator | Ações do sistema |
| | 1. Informa indisponibilidade no dia selecionado. |

Tabela 3 - Especificação de caso de uso gerenciar reserva.

| | |
|---|--|
| Identificação do Caso de uso: UC003 | Nome do Caso de uso: Gerenciar venda |
| Atores: Empresa. | |
| Resumo: Descreve os passos realizados para o gerenciamento de vendas. | |
| Tipo: Primário | |
| Pré-Condições: Possuir uma solicitação pendente do cliente, como venda ou reserva. | |
| Pós-Condições: Gerar comprovante, após o pagamento. | |
| Fluxo Básico | |
| Ações do ator | Ações do sistema |
| 1. Cria uma nova venda. 2. Informa dados necessários para efetuar a venda. | 3. Verifica dados inseridos. 4. Solicita forma de pagamento. 5. Emitir comprovante de pagamento. |
| Fluxo de exceção - Dados em Falta ou Inválidos | |
| Ações do ator | Ações do sistema |
| | 1. Informa que há dados insuficientes para efetuar o cadastro. |

Tabela 4 - Especificação de caso de uso emitir relatório.

| | |
|---|---|
| Identificação do Caso de uso: UC004 | Nome do Caso de uso: Emitir relatório |
| Atores: Empresa. | |
| Resumo: Descreve os passos realizados para a emissão do relatório. | |
| Tipo: Primário | |
| Pré-Condições: Nenhuma | |
| Pós-Condições: Nenhuma | |
| Fluxo Básico | |
| Ações do ator | Ações do sistema |
| 1. Informar o relatório desejado. 2. Insere informações necessárias. | 3. Retorna informações em forma de relatório. |
| Fluxo de exceção - Dados Inválidos. | |
| Ações do ator | Ações do sistema |
| | 1. Dados informados de maneira incorreta. |

Tabela 5 - Especificação de caso de uso gerenciar residência

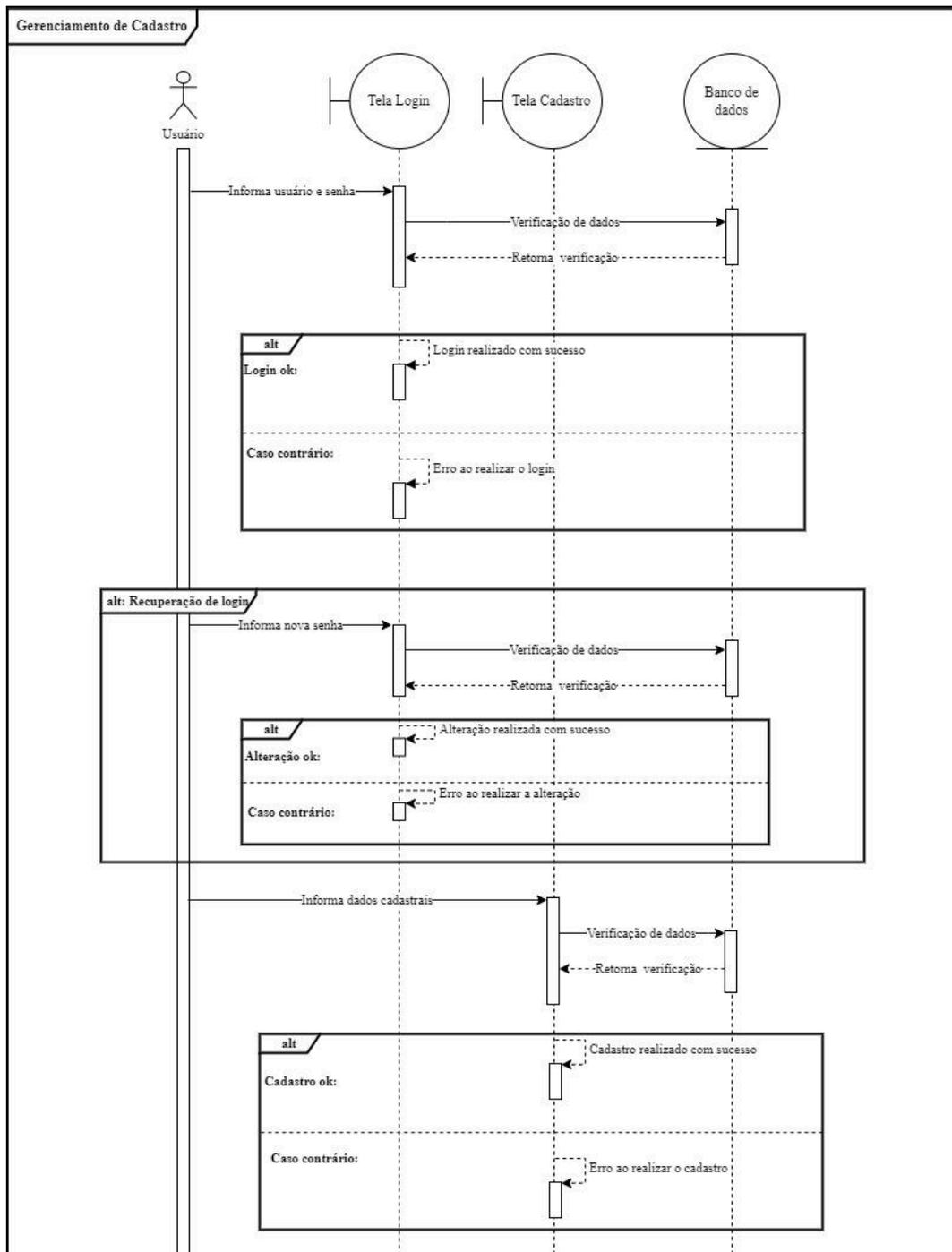
| | |
|--|--|
| Identificação do Caso de uso: UC005 | Nome do Caso de uso: Gerenciar residência |
| Atores: Empresa. | |
| Resumo: Descreve os passos realizados para efetuar o cadastro e atualização das residências disponíveis para locação. | |
| Tipo: Primário | |

| Pré-Condições: Possuir cadastro de ADM | |
|---|--|
| Pós-Condições: | |
| Fluxo Básico | |
| Ações do ator | Ações do sistema |
| 1. Solicita novo cadastro de residência 3. Informa dados necessários para efetuar o cadastro. 4. seleciona a opção “Carregar Fotos”. 5. Seleciona a foto desejada para cadastrar junto à residência. | 2. Solicita dados para realizar o cadastro. 6. Grava informações no banco de dados. |
| Fluxo de exceção - Dados Inválidos. | |
| Ações do ator | Ações do sistema |
| | 1. Informa dados inseridos de maneira incorreta ou ausentes. |

2.2 Diagrama de sequência

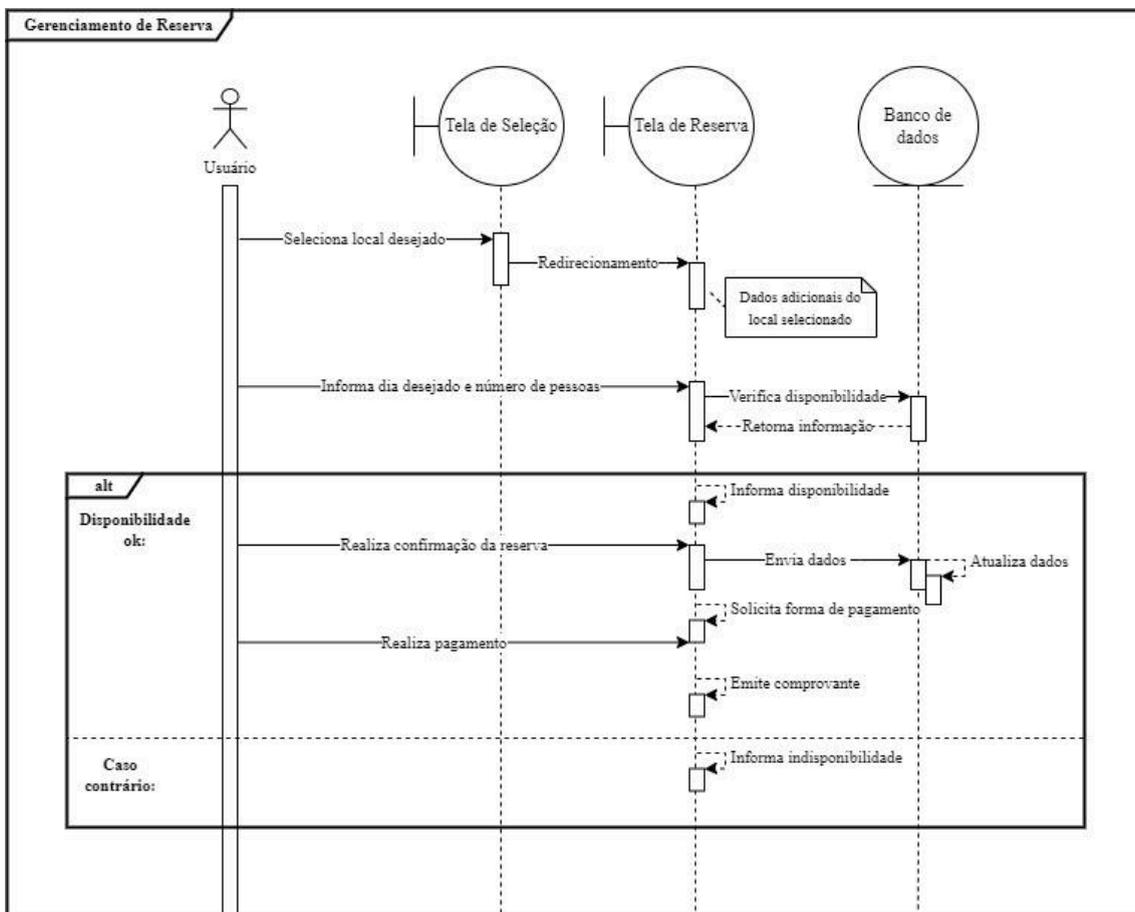
Nesta seção serão apresentados os diagramas de sequência do sistema, que auxiliam no entendimento de funções do sistema.

Figura 6 - Diagrama de sequência do caso de uso gerenciamento de cadastro.



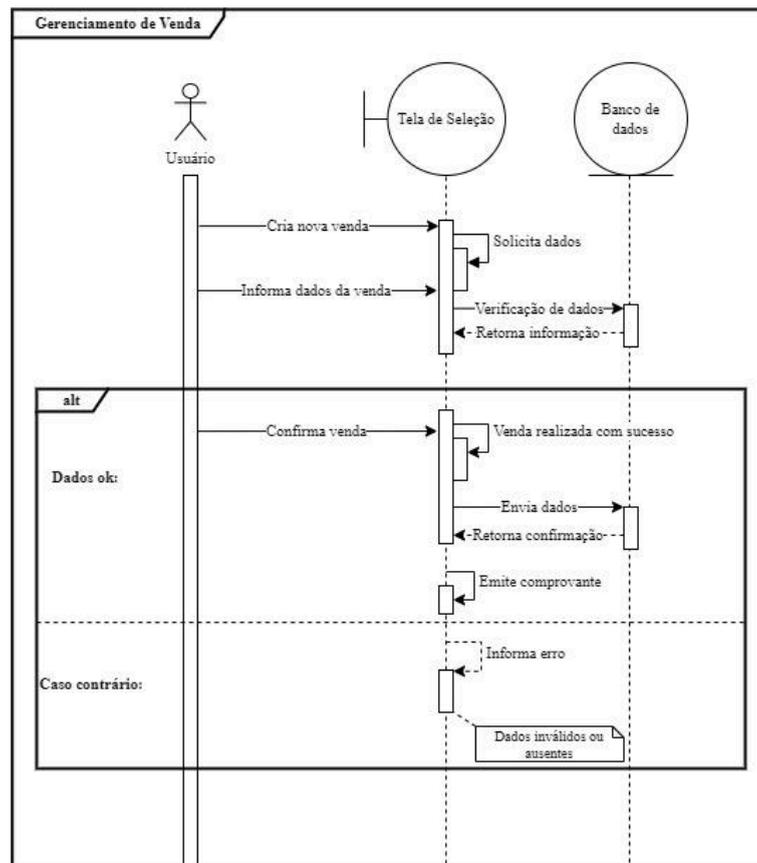
Fonte: do autor (2023).

Figura 7 - Diagrama de sequência do caso de uso gerenciamento de reserva.



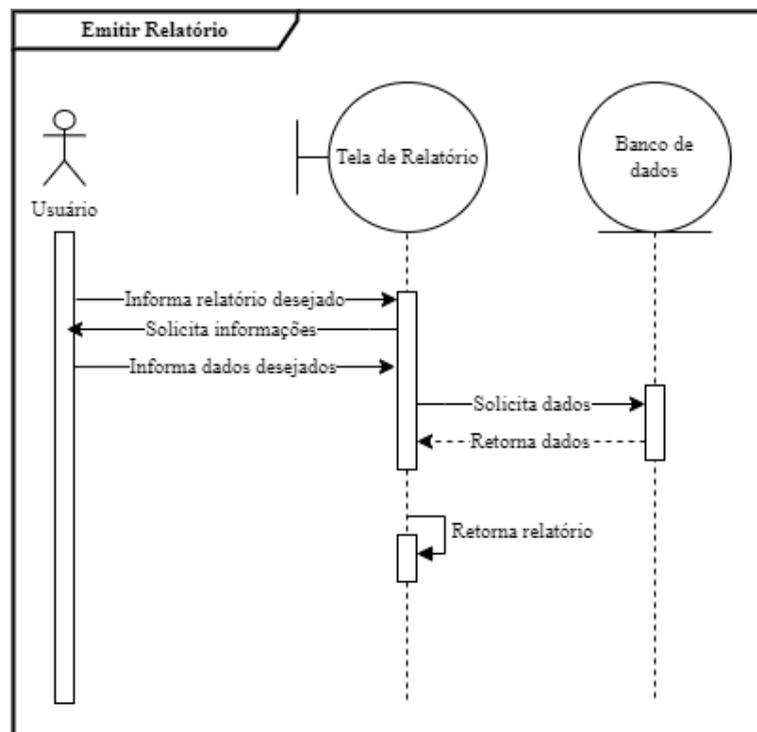
Fonte: do autor (2023).

Figura 8 - Diagrama de seqüência do caso de uso gerenciamento de venda.



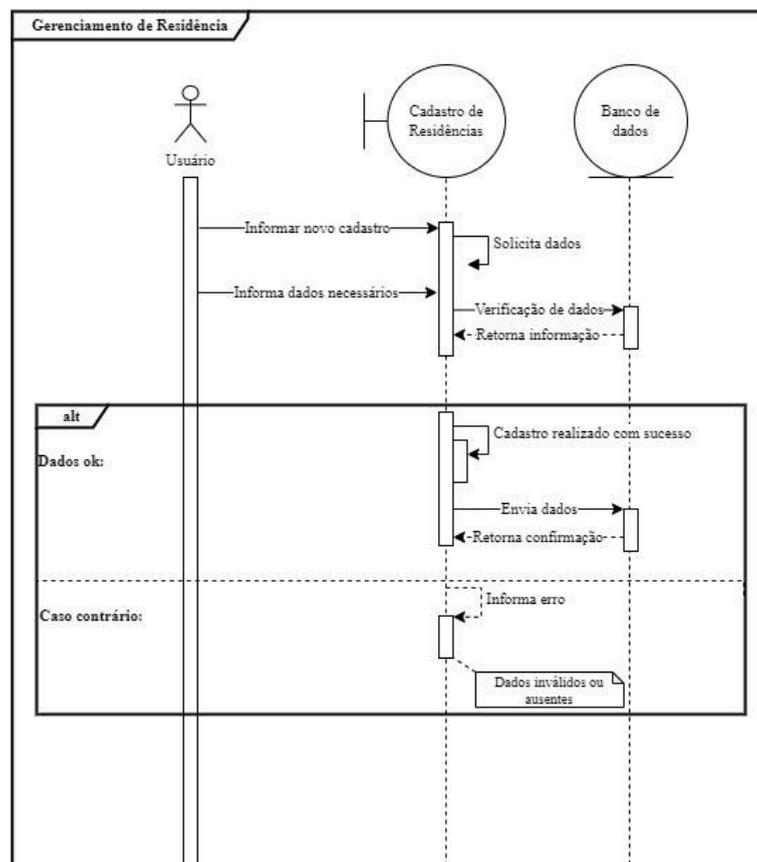
Fonte: do autor (2023).

Figura 9 - Diagrama de seqüência do caso de uso emitir relatório.



Fonte: do autor (2023).

Figura 10 - Diagrama de sequência do caso de gerenciar residência.



Fonte: do autor (2023).

2.3 Banco de dados

O projeto do banco de dados é um processo indispensável para a realização do projeto em um sistema, independente de qual seja.

Após a verificação dos requisitos, precisa ser definido o escopo do projeto, onde deve-se realizar o diagrama de casos de uso, diagrama de sequência e modelagens do banco de dados.

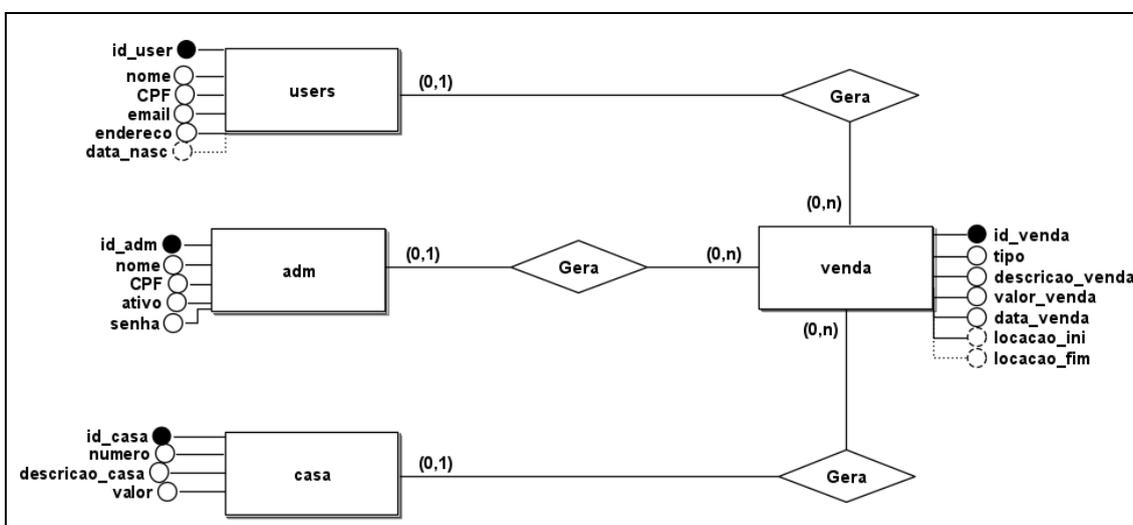
2.3.1 Diagrama de entidade e relacionamento

O diagrama de entidade e relacionamento é uma representação gráfica de um modelo de dados, onde deve aparecer as entidades envolvidas no sistema, bem como as relações entre elas.

O diagrama ER tem como objetivo, auxiliar os programadores a visualizar a relação entre as entidades do sistema que está sendo desenvolvido, para assim, ter um melhor entendimento sobre o mesmo. Segue abaixo o diagrama ER (figura 10).

A entidade “casa” representa os dados sobre as residências disponíveis para locação. As entidades “adm” e “users” representam os dados referentes aos usuários administradores, que são os proprietários da empresa, e os clientes, respectivamente. Já a entidade “venda” representa as vendas de ingressos de entrada para o parque e também as reservas de residências.

Figura 11 - Diagrama ER.



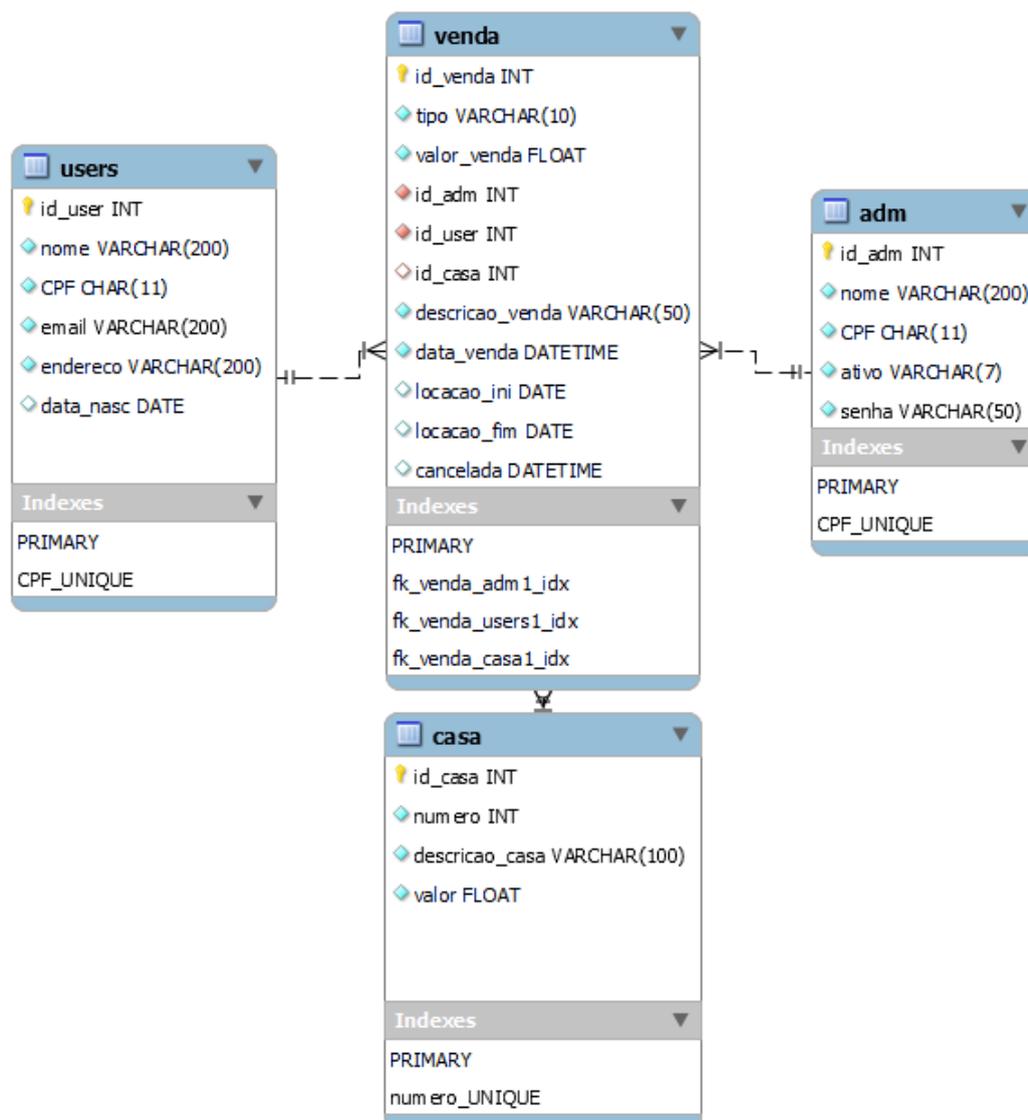
Fonte: do autor (2023).

2.3.2 Modelo lógico

O modelo lógico de um banco de dados é uma representação das tabelas e seus atributos, envolvendo-os em relacionamento no banco de dados.

O diagrama lógico é composto por diversas tabelas, onde são especificadas cada uma delas com dados. Cada tabela possui seu nome contendo dentro dela seus atributos. Segue abaixo o diagrama lógico do banco de dados (figura 11).

Figura 12 - Modelo Lógico.



Fonte: do autor (2023).

2.3.3 Código de criação do banco de dados

Segue abaixo o código para a criação do banco de dados utilizado para este sistema.

```
CREATE DATABASE ci4;
CREATE TABLE adm(
  id_adm int AUTO_INCREMENT PRIMARY KEY,
  nome varchar(200) not null,
  CPF char(11) UNIQUE NOT null,
  ativo varchar(7) not null,
  senha varchar(50) not null
);
CREATE TABLE users(
  id_user int AUTO_INCREMENT PRIMARY KEY,
  nome varchar (200) not null,
  CPF char(11) UNIQUE not null,
  email varchar(200) not null,
  data_nasc date
);
CREATE TABLE casa (
  id_casa int AUTO_INCREMENT PRIMARY KEY,
  numero int UNIQUE not null,
  descricao_casa varchar(100) not null,
  valor float not null
);
CREATE TABLE venda (
  id_venda int AUTO_INCREMENT PRIMARY KEY,
  id_adm int not null,
  id_user int not null,
  id_casa int,
  tipo, varchar(10),
  valor_venda float,
  descricao_venda varchar(50) not null,
  data_venda datetime not null,
  locacao_ini date,
  locacao_fim date,
  cancelada datetime,
  FOREIGN KEY (id_adm) REFERENCES adm(id_adm),
  FOREIGN KEY (id_user) REFERENCES users(id_user),
  FOREIGN KEY (id_casa) REFERENCES casa(id_casa)
);
```

2.4 Diagrama de Classes

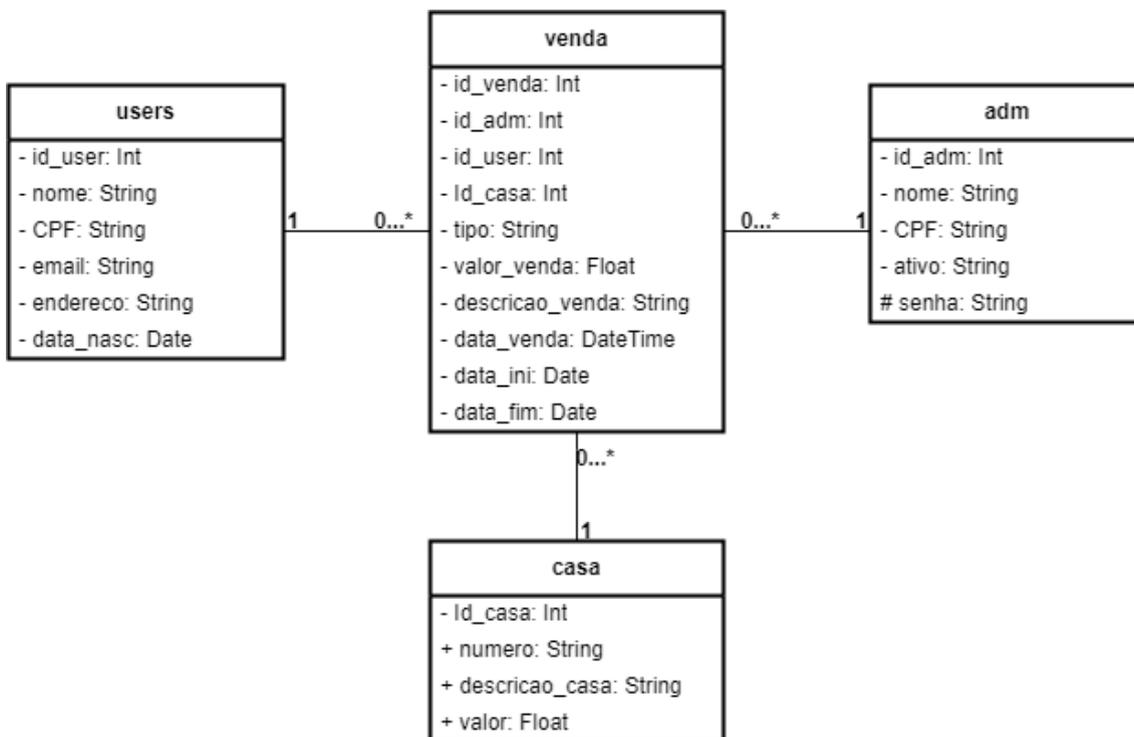
O diagrama de classes tem como objetivo representar, graficamente, as classes e suas relações em um sistema. As classes são representadas por um nome, seguida de seus atributos.

Atributos que podem, por sua vez, serem demarcados como:

- Público (+);
- Privado (-);
- Protegido (#).

Segue abaixo o diagrama de classes (Figura 12).

Figura 13 - Diagrama de classes.



Fonte: do autor (2023).

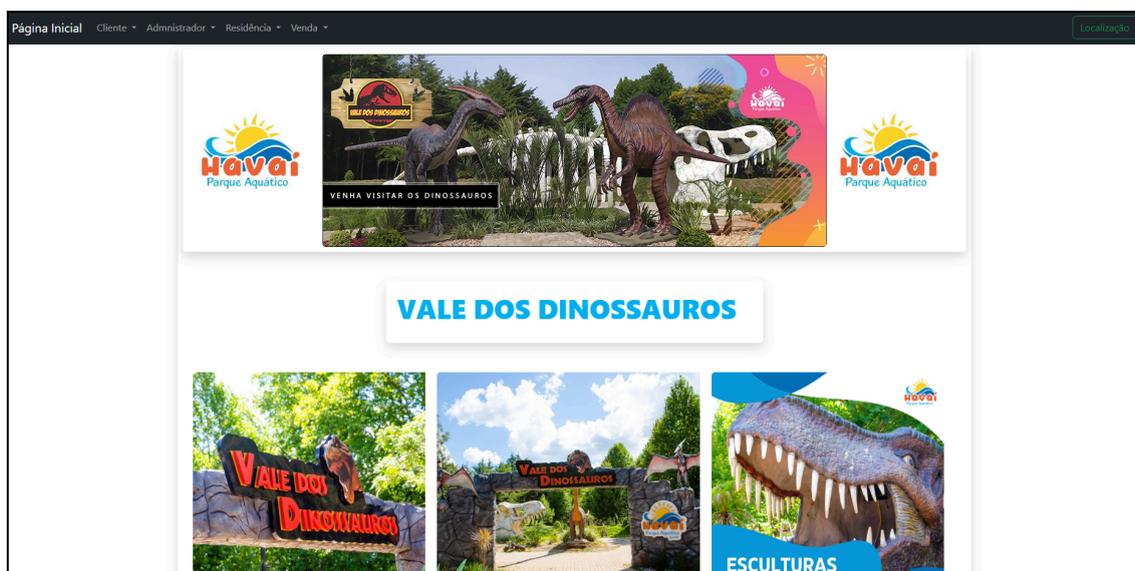
2.5 Telas do sistema

Segue abaixo algumas imagens do sistema desenvolvido, tendo suas funcionalidades e ações funcionando.

2.5.1 Tela inicial

A figura 14 ilustra a tela inicial do administrador, a qual contém os menus e atalhos para as funções desejadas, página inicial, cadastro e lista de clientes, administradores e vendas.

Figura 14 - Tela inicial.



Fonte: do autor (2023).

2.5.2 Tela de cadastro de cliente/usuário

A figura 15 ilustra a tela de cadastro de clientes, sendo que de forma simples o cliente pode informar seu nome, CPF, endereço, e-mail e data de nascimento para efetuar o cadastro.

Figura 15 - Tela de cadastro de cliente (imagem aproximada).

A imagem mostra uma interface web para o cadastro de clientes. O formulário é intitulado "Cadastro de Clientes" e contém os seguintes campos:

- Nome: Campo de texto simples.
- CPF: Campo de texto simples.
- Endereço: Campo de texto simples.
- E-mail @: Campo de texto com um ícone de envelope à esquerda.
- Data de Nascimento: Campo de texto com o formato "dd/mm/aaaa" e um ícone de calendário à direita.

Um botão azul com o texto "Gravar" está localizado no canto inferior direito do formulário.

Fonte: do autor (2023).

2.5.3 Tela de cadastro de administrador/funcionário

Já na figura 16 apresenta-se a tela de cadastro de administradores, que está disponível apenas para os administradores já cadastrados no sistema.

Figura 16 - Tela de cadastro de administrador (imagem aproximada).

Cadastro de Administrador

Nome

CPF

Situação:
 Ativo
 Inativo

Senha

Gravar

Fonte: do autor (2023).

2.5.4 Tela de cadastro de residência

Na figura 17, mostrada a seguir, temos a tela de cadastro de novas residências.

Figura 17 - Tela de cadastro de residência (imagem aproximada).

Cadastro de Residência

Número da Residência

Descrição

Valor da Diária R\$

Gravar

Fonte: do autor (2023).

2.5.4 Tela de venda avulsa

Por sua vez, na figura 18, segue a tela de venda avulsa, sendo a pulseira disponibilizada para passar o dia no parque, onde o valor total é calculado automaticamente ao inserir o valor e a unidade.

Figura 18 - Tela de Venda Avulsa (imagem aproximada).

Venda Avulsa

Funcionário
arthur

Cliente
pedro

Descrição da Venda | Pulseira

Valor | 15 | UN | 1

Total R\$ | 15

Finalizar Venda

Fonte: do autor (2023).

2.5.4 Tela de reserva/locação

Na figura 19 apresenta-se a tela de cadastro de novas locações, utilizando a mesma tabela no banco de dados que a venda avulsa, ambas com informações necessárias escondidas no *back-end*. Também possui o cálculo de dias ao selecionar as datas de entrada e saída, após isso, será necessário informar o valor, posteriormente tudo será multiplicado, gerando o valor total.

Figura 19 - Tela de reserva (imagem aproximada).

| Locação | | | | | |
|-------------------------|-----------------|------|-------|---------------------------------|---|
| Vendedor | arthur | | ▼ | | |
| Cliente | Pedro Rodrigues | | ▼ | | |
| Numeração da Residência | 155 | | ▼ | | |
| Entrada | 16/12/2023 | 📅 | Saída | 23/12/2023 | 📅 |
| Valor R\$ | 200 | Dias | 7 | | |
| Total R\$ | 1400.00 | | | | |
| | | | | Finalizar Venda | |

Fonte: do autor (2023).

2.5.6 Tela de visualização de clientes

A seguir, na figura 20, a tela com a listagem de usuários cadastrados no sistema, com suas informações sendo exibidas.

Figura 20 - Tela de lista de clientes (imagem aproximada).

| Lista de Usuarios | | | | | | |
|-------------------|-------|-------------|--------------------|-----------------|------------------------------|--|
| # | Nome | CPF | Data de Nascimento | E-mail | Endereço | Ações |
| 1 | pedro | 12345696385 | 2015-09-02 | teste@gmail.com | rua do teste | Editar Excluir |
| 3 | Maria | 12364789632 | 2023-09-12 | Maria@gmail.com | Rua Teste, 40 - Bairro Teste | Editar Excluir |

Fonte: do autor (2023).

2.5.7 Tela de listagem de administradores/funcionários

A tela de listagem de administradores, abaixo mostrada na figura 21, contém as informações dos mesmos, além da função de editar.

Figura 21 - Tela de listagem administradores (imagem aproximada).

| Lista de Administradores | | | | |
|--------------------------|--------|-------------|----------|------------------------|
| # | Nome | CPF | Situação | Ações |
| 18 | novo | 95163247851 | Inativo | Editar |
| 19 | arthur | 12365498752 | Ativo | Editar |

Fonte: do autor (2023).

2.5.8 Tela de listagem de vendas

A seguir na figura 22, é apresentada a tela de listagem de vendas realizadas, sendo elas locações e diárias, possuindo somente ação de cancelamento de venda. Sendo exibida de forma decrescente.

Figura 22 - Tela de listagem de vendas (imagem aproximada).

| Vendas Realizadas | | | | | | | | | | | |
|-------------------|---------|-----------|--------|---------------------|-------------------|----------------|-----------------|----------|---------|--------------------------|--------------------------|
| # | Tipo | Descrição | Valor | Data Venda | Início da Reserva | Fim da Reserva | Cliente | Vendedor | ID Casa | Ações | Cancelada |
| 3 | Locação | Diária | R\$200 | 2023-12-14 02:57:26 | 2023-12-16 | 2023-12-17 | Pedro Rodrigues | arthur | 3 | Cancelar | NÃO |
| 1 | Avulsa | Pulseira | R\$15 | 2023-12-14 02:18:38 | | | Pedro Rodrigues | arthur | | Cancelar | SIM: 2023-12-14 02:59:10 |

Fonte: do autor (2023).

2.5.9 Tela de busca de diárias disponíveis

Na figura 23, pode-se visualizar a tela de busca de diárias disponíveis, sendo uma busca entre datas, processo realizado pelo cliente antes de realizar a nova locação.

Figura 23 - Tela de busca de diárias.

Busca por Data

Entrada dd/mm/aaaa

Saída dd/mm/aaaa

Pesquisar

Lista de Residências Disponíveis

Nova Locação

| # | Número da Casa | Descrição | Valor |
|----|----------------|--|---------|
| 7 | 15 | Casa com 2 camas, ar, banheiro e vista para o parque | R\$ 500 |
| 10 | 333 | sem locação | R\$ 123 |

Fonte: do autor (2023).

2.5.10 Código utilizado para a realização da busca.

A seguir, na figura 24, é possível visualizar o código utilizado para realizar a consulta no banco de dados que retorna quais residências disponíveis para locação em um determinado período de datas informadas pelo cliente. Foi necessário utilizar recursos específicos do *framework* para construir a consulta que utiliza comandos como *subquery* e *not in* na cláusula *where*.

Figura 24 - Código de busca utilizado.

```

$data = filter_input_array(INPUT_POST, FILTER_DEFAULT); ?>

<form action="" method="post">
  <div class="input-group mb-3">
    <span class="input-group-text">Entrada</span>
    <input type="date" name="ini" class="form-control" required>
  </div>
  <div class="input-group mb-3">
    <span class="input-group-text">Saída</span>
    <input type="date" name="fim" id="fim" class="form-control" required>
  </div>
  <button type="submit" class="btn btn-primary" value="busca" name="busca_">Pesquisar</button>
</form>

<?php if (!empty($data['busca_'])) {
    $db = db_connect();
    $query = $db->table('casa c');
    $query->select('*');

    $subQuery = $db->table('venda')->select('id_casa')->where('locacao_ini >=', $data['ini'])->where('locacao_fim <=', $data['fim']);
    $builder = $db->table('casa')->whereNotIn('id_casa', $subQuery);
    $query = $builder->get();

    $query->getResultArray();
}
?>

```

Fonte: do autor (2023).

2.5.11 Função para a confirmação de exclusão

As figuras 25 e 26 apresentam a função que foi utilizada para gerar o aviso de confirmação para a exclusão de dados no sistema.

Figura 25 - Função de exclusão part.1.

```

foreach ($users as $user) : ?>
  <tbody>
    <tr>
      <th scope="row"><?php echo $user['id_user'] ?></th>
      <td><?php echo $user['nome'] ?></td>
      <td><?php echo $user['CPF'] ?></td>
      <td><?php echo $user['data_nasc'] ?></td>
      <td><?php echo $user['email'] ?></td>
      <td><?php echo $user['endereco'] ?></td>
      <td>
        <?php echo anchor('usercontroller/edit/' . $user['id_user'], 'Editar ') ?>
        |
        <?php echo anchor('usercontroller/delete/' . $user['id_user'], 'Excluir', ['onclick' => 'return confirma()']) ?>
      </td>
    </tr>
  </tbody>
</foreach ?>

```

Fonte: do autor (2023).

Figura 26 - Função de exclusão part.2.

```

<script>
  function confirma() {
    if (!confirm('Deseja mesmo excluir este registro?')) {
      return false;
    }
    return true;
  }
</script>

```

Fonte: do autor (2023).

2.5.12 Função de cálculo automático na venda.

Nas figuras 27 e 28, é ilustrada a função utilizada para realizar o cálculo automático na parte da venda.

Figura 27 - Função de cálculo part.1.

```

<div class="input-group mb-3">
  <span class="input-group-text">Valor</span>
  <input type="text" class="form-control" placeholder="R$val" oninput="Calc()" value="15" class="form-control">
  <span class="input-group-text">UN</span>
  <input type="text" class="form-control" placeholder="Quantidade" oninput="Calc()" value="1" class="form-control">
</div>
<div class="input-group mb-3">
  <span class="input-group-text">Total R$</span>
  <input type="text" value="<?php echo isset($venda['valor_venda']) ? $venda['valor_venda'] : '15' ?>">
</div>

```

Fonte: do autor (2023).

Figura 28 - Função de cálculo part.2.

```

<script>
function Calc() {
  var calc = 0;
  var ipts = document.querySelectorAll('input[oninput="Calc()"]');
  for (var x = 0; x < ipts.length; x++) {
    var valorItem = parseFloat(ipts[x].value);
    if (calc === 0)
      calc = valorItem;
    else {
      !isNaN(valorItem) ? calc *= parseFloat(valorItem) : null;
      console.log(calc);
    }
  }
  document.querySelector('#valor_venda').value = calc.toFixed(2);
}
</script>

```

Fonte: do autor (2023).

2.5.13 Função de contagem de dias.

Nas figuras 29 e 30, é ilustrada a função utilizada para realizar a contagem dos dias entre as datas selecionadas, data de início e data de saída, na tela de nova locação.

Figura 29 - Contagem de dias part.1.

```

<div class="input-group mb-3">
  <span class="input-group-text">Entrada</span>
  <input type="date" name="locacao_ini" id="locacao_ini" class="form-control"
  class="form-control" onchange="chamar()">
  <span class="input-group-text">Saída</span>
  <input type="date" name="locacao_fim" id="locacao_fim" class="form-control"
  class="form-control" onchange="chamar()">
</div>

```

Fonte: do autor (2023).

Figura 30 - Contagem de dias part.2.

```
<script>
function difDias() {
  var data_ini = new Date(document.getElementById("locacao_ini").value);
  var data_fim = new Date(document.getElementById("locacao_fim").value);
  return parseInt((data_fim - data_ini) / (24 * 3600 * 1000));
}

function chamar() {
  document.getElementById("dias").value = isNaN(difDias()) ? "Selecione a outra data" : difDias();
}
</script>
```

Fonte: do autor (2023).

3 RESULTADOS

A seguir será apresentado de forma breve o resultado de alguns questionários, realizados de maneira simples e sucinta, para assim possuir uma avaliação sobre o sistema desenvolvido.

As perguntas a seguir foram respondidas por usuários que realizaram o teste do sistema, avaliando tanto seu layout quanto a usabilidade. Além disso, essa etapa proporciona uma avaliação do sistema, além de auxiliar na coleta de algumas opiniões referente ao mesmo, a fim de melhorá-lo futuramente.

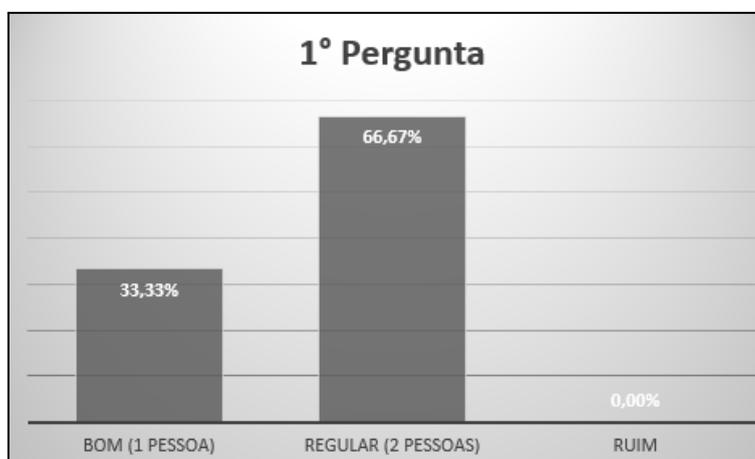
Para que o questionário fosse simples e sucinto, a fim de não tomar muito tempo de quem as responde, foram designados níveis de resposta, sendo: bom, regular ou ruim. As perguntas foram feitas a três usuários.

Segue abaixo as perguntas realizadas aos usuários:

1. Proporcionado as seguintes funções mostradas no sistema até o momento presente, responda com a sua opinião sobre as mesmas.
2. Tendo em vista o layout do sistema, responda com a sua opinião sobre o mesmo.
3. Seguindo a necessidade de organização e controle de vendas ou locações, o que você acha do sistema até agora?

A seguir a figura 29 contém as respostas da primeira pergunta. Sendo 33,33% (uma resposta) “Bom” e 66,67% (duas respostas) como “Regular”.

Figura 29 - Primeira pergunta.



Fonte: do autor (2023).

Na figura 26, podemos visualizar o gráfico com as respostas realizadas referente a segunda pergunta. Onde os três usuários acharam o *layout* da página bom. Totalizando 100% das respostas como “Bom”.

Figura 30 - Segunda Pergunta.



Fonte: do autor (2023).

Já na figura 27, segue a resposta da terceira e última pergunta realizada aos usuários. Sendo dois usuários (66,67%) com a resposta “Bom” e somente um usuário (33,33%) com a resposta “Regular”.

Figura 31 - Terceira Pergunta.



Fonte: do autor (2023).

4. CONCLUSÃO

Portanto, possuindo como objetivo principal fornecer a organização e o melhor atendimento ao público, este sistema teve como foco a administração do local. Sendo restringido, por hora, somente para a parte administrativa, sendo assim funcionários e superiores.

O objetivo de manter o cadastro de vendas e controle de locações, possuindo seus cadastros de clientes, administrador e residência foi concluído. Além disso, foram desenvolvidas telas de visualizações de clientes, administrador, vendas, e ainda, a possibilidade de consultas às residências disponíveis para locação em um determinado período de datas. Por outro lado, algumas funções como o *login* e o gerenciamento de venda, possuindo função de pagamento e geração de comprovante após a confirmação do pagamento, encontra-se pendente. Devido a estes itens faltantes, o sistema foi restringido a parte administrativa.

Por fim, o presente trabalho de conclusão de curso foi de suma importância para o aprendizado, proporcionando uma experiência única para o desenvolvimento prévio de um sistema *web*.

REFERÊNCIAS

Academia. *PHP programando com orientação a objetos*. Disponível em: https://www.academia.edu/40632903/Livro_php_programando_com_orientação_a_objetos. Acesso em: 09 de Março de 2023.

Betrybe. *O que é PHP: o guia básico desta linguagem de programação*. Disponível em: <https://blog.betrybe.com/desenvolvimento-web/php/>. Acesso em: 25 de Agosto de 2023

Bootstrap. *Get Started*. Disponível em: <https://getbootstrap.com/>. Acesso em: 16 de Março de 2023.

CIn UFPE. *HTML5 e CSS3 com farinha e pimenta*. Disponível em: <https://www.cin.ufpe.br/~dfop/Arquivos/Pacote%20Web/HTML5%20e%20CSS3%20com%20Farinha%20e%20Pimenta%20Diego%20Eis%20e%20Elcio%20Ferreira.pdf>. Acesso em: 22 de Novembro de 2023.

CodeIgniter 4. *User Guide*. Disponível em: https://codeigniter.com/user_guide/index.html. Acesso em: 09 de Março de 2023.

DevMedia. *Padrão MVC*. Disponível em: <https://www.devmedia.com.br/padrao-mvc-java-magazine/21995>. Acesso em: 22 de Março de 2023

Le Wagon. *O que é padrão MVC? Entenda arquitetura de softwares*. Disponível em: <https://www.lewagon.com/pt-BR/blog/o-que-e-padrao-mvc>. Acesso em: 22 de Março de 2023

Melhores Hospedagens. *O que é PHP e como funciona: guia básico*. Disponível em: <https://www.melhoreshospedagem.com/o-que-e-php/>. Acesso em: 25 de Agosto de 2023

MySQL. *Reference Manual*. Disponível em: <https://dev.mysql.com/doc/refman/8.2/en/>. Acesso em: 22 de Novembro de 2023.

PHP. *História do PHP*. Disponível em: https://www.php.net/manual/pt_BR/history.php.php#:~:text=Criado%20em%201994%20por%20Rasmus.em%20linguagem%20de%20programa%C3%A7%C3

[%A3o%20C](#). Acesso em: 25 de Agosto de 2023

Uniasselvi. *Programação Orientada a Objetos*. Disponível em:

<https://www.uniasselvi.com.br/extranet/layout/request/trilha/materiais/livro/livro.php?codigo=24670>. Acesso em: 22 de Março de 2023