

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
ETEC DA ZONA LESTE
NOVOTEC DESENVOLVIMENTO DE SISTEMAS

Júlio Da Silva Neves
Maria Eduarda Silva Werlang
Victor Rafael Ferreira De Roma

SR's: Reconhecimento de Placas Veiculares

São Paulo
2023

Júlio Da Silva Neves
Maria Eduarda Silva Werlang
Victor Rafael Ferreira De Roma

SR's: Reconhecimento de Placas Veiculares

Trabalho de Conclusão de Curso apresentado ao Novotec Desenvolvimento de Sistemas da Etec Zona Leste, orientado pelo Prof. Esp. Jeferson Roberto de Lima, como requisito parcial para a obtenção do título de técnico em Desenvolvimento de Sistemas.

São Paulo

2023

AGRADECIMENTO

Gostaríamos de agradecer aos nossos responsáveis, cujo apoio e incentivo foram fundamentais durante todo o processo de produção deste projeto. Também queremos agradecer aos nossos professores, cujas aulas ao longo dos últimos três anos foram valiosas para nossa formação. Em especial, gostaríamos de agradecer a orientação dos coordenadores e orientadores do nosso curso, que, apesar das exigências, desempenharam um papel crucial no nosso desenvolvimento teórico e prático do TCC.

*“Exige muito de ti e espera pouco dos outros.
Assim, evitarás muitos aborrecimentos.”*

CONFÚCIO

RESUMO

Este artigo busca levantar a problemática decorrente do aumento no número de veículos em circulação pelo mundo, que vem apresentando desafios cada vez mais complexos em relação ao tráfego desses veículos. A fim de enfrentar essa questão, o projeto SR's foi projetado, propondo soluções inovadoras e automatizadas por meio da implementação de um dispositivo equipado com um sistema de reconhecimento de placas veiculares em condomínios, acompanhado por um sistema de gerenciamento de dados específicos para essa localidade. Para alcançar esse objetivo, serão aplicados conceitos de IoT (Internet das Coisas) na criação do dispositivo, utilizando do minicomputador Raspberry Pi, e serão empregadas tecnologias como Python, Java, MySQL, OpenCV, entre outras.

Palavras-chave: Veículos. Tráfego. Reconhecimento de placas. Condomínios. IoT.

ABSTRACT

This article seeks to raise the problem deriving from the increase in the number of vehicles in circulation around the world, which has been presenting more complex challenges in relation to the traffic of these vehicles. To resolve this issue, the SR's project was designed, proposing innovative and automated solutions through the implementation of a device equipped with a vehicle's license plate recognition system in condominiums, accompanied by a data management system specific to that location. To achieve this goal, IoT (Internet of Things) concepts will be applied in the creation of the device, using the Raspberry Pi minicomputer, and technologies such as Python, Java, MySQL, OpenCV, among others, will be employed.

Keywords: Vehicles. Traffic. License plate recognition. Condominiums. IoT.

LISTA DE FIGURAS

Figura 1 - Exemplo do código Java	16
Figura 2 - Resultado do código Java	17
Figura 3 - As tecnologias mais populares de 2022.....	18
Figura 4 - Exemplo de código Python	18
Figura 5 - Inserindo os dados após a execução do código	19
Figura 6 - Resultado do código Python após a inserção de dados	20
Figura 7 - Exemplo da implementação do OpenCV	21
Figura 8 - Composição de uma tabela de banco de dados	23
Figura 9 - Exemplo de uma placa Raspberry Pi B+	25
Figura 10 - Diagrama de Caso de Uso geral da aplicação	30
Figura 11 - Diagrama de classe da aplicação	44
Figura 12 - DER da aplicação	45
Figura 13 - MER da aplicação	45
Figura 14 - Diagrama de atividade do gerenciamento de condomínio	46
Figura 15 - Diagrama de atividade do gerenciamento de morador	47
Figura 16 - Diagrama de atividade do gerenciamento de porteiro.....	48
Figura 17 - Diagrama de atividade do gerenciamento de veículos.....	49
Figura 18 - Diagrama de atividade para ativação de condomínios.....	50
Figura 19 - Diagrama de atividades para ativação de porteiros	51
Figura 20 - Diagrama de atividade para listar entrada	51
Figura 21 - Diagrama de sequência do login do administrador e do porteiro	52
Figura 22 - Diagrama de sequência do cadastro de condomínio	53
Figura 23 - Diagrama de sequência do cadastro de morador	53
Figura 24 - Diagrama de sequência do cadastro de porteiro	54
Figura 25 - Diagrama de sequência do cadastro de veículo	54
Figura 26 - Diagrama de sequência da atualização de condomínio.....	55
Figura 27 - Diagrama de sequência da atualização de morador	55
Figura 28 - Diagrama de sequência da atualização de porteiro	56
Figura 29 - Diagrama de sequência da atualização de veículo.....	56
Figura 30 - Diagrama de sequência da exclusão do cadastro de morador	57
Figura 31 - Diagrama de sequência da exclusão do cadastro de veículo	57
Figura 32 - Diagrama de sequência para a inativação de condomínio.....	58

Figura 33 - Diagrama de sequência para a inativação de porteiro	58
Figura 34 - Diagrama de sequência para a listagem de entrada.....	59
Figura 35 - Diagrama de sequência para ativação de condomínios.....	59
Figura 36 - Diagrama de sequência para ativação de porteiros	60
Figura 37 - Wireframe de baixa fidelidade da interface de “Login”	61
Figura 38 - Wireframe de baixa fidelidade da interface “Menu Administrador”	61
Figura 39 - Wireframe de baixa fidelidade da interface “Gerenciamento de Condomínios”	62
Figura 40 - Wireframe de baixa fidelidade da interface “Gerenciamento de Porteiros”	63
Figura 41 - Wireframe de baixa fidelidade da interface “Condomínios Inativados” ...	64
Figura 42 - Wireframe de baixa fidelidade da interface "Pré-Gerenciamento de Dados do Porteiro"	65
Figura 43 - Wireframe de baixa fidelidade da interface "Porteiros Inativados"	66
Figura 44 - Wireframe de baixa fidelidade da interface “Menu Porteiro”	66
Figura 45 - Wireframe de baixa fidelidade da interface “Gerenciamento de Moradores”	67
Figura 46 - Wireframe de baixa fidelidade da interface "Gerenciamento de Veículos"	68
Figura 47 - Wireframe de baixa fidelidade da interface "Listar Entradas"	69
Figura 48 - Wireframe de alta fidelidade da interface "Login".....	70
Figura 49 - Wireframe de alta fidelidade da interface "Menu Administrador"	70
Figura 50 - Wireframe de alta fidelidade da interface "Gerenciamento de Condomínios"	71
Figura 51 - Wireframe de alta fidelidade da interface "Gerenciamento de Porteiros".....	72
Figura 52 - Wireframe de alta fidelidade da interface "Condomínios Inativados"	73
Figura 53 - Wireframe de alta fidelidade da interface "Pré-Gerenciamento de Dados do Porteiro"	74
Figura 54 - Wireframe de alta fidelidade da interface "Porteiros Inativados"	75
Figura 55 - Wireframe de alta fidelidade da interface "Menu Porteiro"	75
Figura 56 - Wireframe de alta fidelidade da interface "Gerenciamento de Moradores"	76
Figura 57 - Wireframe de alta fidelidade da interface "Gerenciamento de Veículos"	77

Figura 58 - Wireframe de alta fidelidade da interface "Listar Entradas"	78
Figura 59 - Interface "Login"	79
Figura 60 - Interface "Menu Administrador"	80
Figura 61 - Interface "Gerenciamento de Condomínios"	81
Figura 62 - Interface "Gerenciamento de Porteiros"	82
Figura 63 - Interface "Condomínios Inativados"	83
Figura 64 - Interface "Pré-Gerenciamento de Dados dos Porteiros"	84
Figura 65 - Interface "Porteiros Inativados"	85
Figura 66 - Interface "Menu Porteiro"	86
Figura 67 - Interface "Gerenciamento de Moradores"	87
Figura 68 - Interface "Gerenciamento de Veículos"	88
Figura 69 - Interface "Listar Entradas"	89

LISTA DE QUADROS

Quadro 1 - Descrição do Caso de Uso geral a partir da perspectiva do administrador	31
Quadro 2 - Descrição do Caso de Uso Geral da aplicação a partir da perspectiva do porteiro	32
Quadro 3 – Descrição do Caso de Uso Geral da aplicação a partir da perspectiva do SRs	32
Quadro 4 – Descrição do Caso de Uso “Realizar Login” (administrador).....	33
Quadro 5 – Descrição do Caso de Uso “Realizar Login” (porteiro)	33
Quadro 6 – Descrição do Caso de Uso “Listar Entrada”	34
Quadro 7 – Descrição do Caso de Uso “Cadastrar Morador”.....	34
Quadro 8 – Descrição do Caso de Uso “Listar Morador”	35
Quadro 9 – Descrição do Caso de Uso “Atualizar Morador”	35
Quadro 10 – Descrição do Caso de Uso “Excluir Morador”	36
Quadro 11 – Descrição do Caso de Uso “Cadastrar Veículo”	36
Quadro 12 – Descrição do Caso de Uso “Listar Veículo”	37
Quadro 13 – Descrição do Caso de Uso “Atualizar Veículo”.....	37
Quadro 14 - Descrição do Caso de Uso “Excluir Veículo”	38
Quadro 15 - Descrição do Caso de Uso “Autenticar Placa”	38
Quadro 16 - Descrição do Caso de Uso “Liberar Acesso”	39
Quadro 17 - Descrição do Caso de Uso “Cadastrar Condomínio”	39
Quadro 18 - Descrição do Caso de Uso “Listar Condomínio”	40
Quadro 19 - Descrição do Caso de Uso “Atualizar Condomínio”	40
Quadro 20 - Descrição do Caso de Uso “Inativar Condomínio”	41
Quadro 21 - Descrição do Caso de Uso “Cadastrar Porteiro”	41
Quadro 22 - Descrição do Caso de Uso “Listar Porteiro”	42
Quadro 23 - Descrição do Caso de Uso “Atualizar Porteiro”	42
Quadro 24 - Descrição do Caso de Uso “Inativar Porteiro”	43

LISTA DE ABREVIATURAS E SIGLAS

Diagrama Entidade-Relacionamento (DER)

General Purpose Input/Output (GPIO)

Internet of Things (IoT)

Java Development Kit (JDK)

Java Virtual Machine (JVM)

Modelo Entidade-Relacionamento (MER)

Open-Source Computer Vision Library (OpenCV)

Random Access Memory (RAM)

Secretaria Nacional de Trânsito (SENATRAN)

Sistema de Gerenciamento de Banco de Dados Relacional (SGBDR)

Structured Query Language (SQL)

Unified Modeling Language (UML)

Universal Serial Bus (USB)

World Wide Web (WWW)

SUMÁRIO

1	INTRODUÇÃO.....	12
2	REFERENCIAL TEÓRICO	13
2.1	Reconhecimento de Placas.....	13
2.2	Visão Computacional.....	13
2.3	Tecnologias Utilizadas	14
2.3.1	UML	14
2.3.2	Java.....	15
2.3.3	Python.....	17
2.3.4	OpenCV.....	20
2.3.5	Banco de Dados.....	21
2.3.5.1	SQL	22
2.3.5.2	MySQL.....	23
2.3.5.3	Diagrama Entidade-Relacionamento.....	23
2.3.5.4	Modelo Entidade-Relacionamento	24
2.3.6	IoT	24
2.3.7	Raspberry Pi.....	24
2.3.8	Figma	25
2.3.8.1	Wireframe	25
3	DESENVOLVIMENTO.....	27
3.1	Documentação do Caso de Uso	27
3.1.1	Requisitos funcionais.....	27
3.1.2	Requisitos não-funcionais	28
3.1.3	Regras de negócio.....	29
3.1.4	Diagrama de Caso de Uso.....	29
3.1.4.1	Quadros de descrição dos casos de uso	30
3.2	Diagrama de Classe.....	43
3.3	Diagrama Entidade-Relacionamento da Aplicação.....	44
3.4	Modelo Entidade-Relacionamento da Aplicação	45
3.5	Diagramas de Atividade	46
3.6	Diagramas de Sequência.....	52
3.7	Representação Gráfica das Interfaces	60

3.7.1	Wireframe de Baixa Fidelidade	60
3.7.2	Wireframe de Alta fidelidade	69
3.7.3	Aplicação no Java	78
4	CONSIDERAÇÕES FINAIS	90
	REFERÊNCIAS	91

1 INTRODUÇÃO

Devido ao aumento constante do número de veículos em circulação pelo mundo, surgem desafios na gestão de tráfego, especialmente no gerenciamento do trânsito e no controle de acesso em estacionamentos, tanto em locais públicos quanto privados. Isso é especialmente presenciado em regiões densamente povoadas, como o estado de São Paulo.

Segundo dados da Secretaria Nacional de Trânsito (SENATRAN), em 2022, a frota de veículos registrados no país ultrapassava 115 milhões, sendo que aproximadamente 35 milhões desses veículos estavam no estado de São Paulo.

Essa quantidade excessiva de veículos em uma única região justifica os problemas de congestionamento e dificuldades no controle de acesso. Diante dos dados apresentados, questiona-se: o que podemos desenvolver para solucionar esses problemas relacionados a gestão de tráfego?

Este projeto tem como objetivo o desenvolvimento de uma simulação de um sistema de reconhecimento de placas veiculares, visando automatizar o controle de acesso em estacionamentos de condomínios. Permitindo assim a otimização do processo de entrada, e uma melhoria no gerenciamento de tráfego.

O sistema possui uma aplicação para controle de dados relacionados ao condomínio. Além de um dispositivo que realiza a identificação das placas veiculares e a possível liberação do veículo para acessar o local. Para desenvolver o projeto, utilizamos tecnologias como: Java, Python, Open Source Computer Vision Library (OpenCV) e MySQL. Além disso, foram aplicados conceitos de IoT (Internet of Things) para construir um dispositivo utilizando o minicomputador Raspberry Pi.

2 REFERENCIAL TEÓRICO

Capítulo dedicado ao embasamento teórico que apresentará os principais tópicos que permeiam o trabalho, além de apresentar as tecnologias utilizadas para o desenvolvimento do projeto.

2.1 Reconhecimento de Placas

Dentre as principais tecnologias utilizadas para o reconhecimento automático de placas de veículos, destacam-se as técnicas de Visão Computacional. Uma das vantagens fundamentais dessas técnicas é sua natureza não intrusiva, uma vez que não exigem qualquer tipo de intervenção nos veículos fiscalizados. O único requisito para seu funcionamento é a obtenção de imagens digitais desses veículos, o que pode ser realizado por meio de câmeras controladas por dispositivos com comandos predefinidos.

Esse processo pode ser dividido em três etapas: localização da placa, segmentação dos caracteres presentes na placa e o reconhecimento da placa. No entanto, encontrar uma solução única para esse problema é uma tarefa complexa, considerando que as imagens capturadas podem apresentar variações significativas em relação aos modelos de placas, além de estarem sujeitas a diversas condições climáticas, e variações de iluminação.

Então, é possível utilizar a tecnologia para realizar o reconhecimento de placas por meio de um dispositivo com comandos predefinidos, além de um sistema que gerencie os dados obtidos do controle de acesso a um determinado local.

2.2 Visão Computacional

Visão Computacional é área que estuda e desenvolve tecnologias que permitem que máquinas enxerguem e extraiam características de imagens capturadas por diferentes meios (BARELLI, 2018).

O processo para o desenvolvimento dessas tecnologias requer um conjunto predefinido de etapas, sendo realizadas desde a aquisição da imagem até a extração das informações que ela possui (GONZALEZ, WOODS, 2009).

2.3 Tecnologias Utilizadas

Para o desenvolvimento do SR's, foram empregadas as seguintes tecnologias:

2.3.1 UML

A UML (Unified Modeling Language ou Linguagem de Modelagem Unificada), é uma linguagem visual de modelagem de software, arquitetada a partir do padrão de desenvolvimento em orientação a objeto (FOWLER, 2005).

Não sendo uma linguagem de programação, mas sim uma linguagem de modelagem, a UML é utilizada pela Engenharia de Software, para facilitar a definição de características do sistema, através de seus requisitos, seus comportamentos e lógica (GUEDES, 2011).

Os diagramas são um dos métodos utilizados para visualizar os sistemas de forma que auxilie em seu desenvolvimento (BOOCH, 2006). Embora existam diferentes tipos de diagramas, cada um deles possuem suas próprias características, o que os tornam individualmente únicos (GUEDES, 2011). Entre os treze diagramas existentes, utilizaremos os Diagramas de Caso de Uso, Classe, Atividade e Sequência.

O caso de uso possui um fácil entendimento do sistema, sem necessariamente mostrar como tais funcionalidades serão implementadas (BOOCH, 2006). Ele normalmente é utilizado na fase de levantamento de requisitos e análise de dados.

O diagrama de classe permite a visualização das classes que compõem o sistema, juntamente com os seus respectivos métodos e atributos. É utilizado na modelagem de aplicações orientada a objetos (FOWLER, 2005).

O diagrama de atividade procura descrever os passos que o sistema deve percorrer na execução de uma determinada atividade, podendo ser representada por um método, um algoritmo ou um processo completo (GUEDES, 2011).

O diagrama de sequência trata-se de um diagrama comportamental que visa representar a sequência temporal das mensagens trocadas entre os objetos participantes em um processo específico (FOWLER, 2005).

2.3.2 Java

O Java é uma linguagem de programação multiplataforma orientada a objeto, projetada para ser robusta, portátil e segura, como afirma Junior (2021, p.10). Segundo Caelum (2003, p.4), o Java foi desenvolvido com o objetivo de ser implementado em pequenos dispositivos, entretanto, com o avanço da internet, ele foi modificado para atender as demandas Web através de pequenas aplicações.

Como afirma Schildt (2015, p.6), o segredo para o Java ser de alta portabilidade e segurança é a utilização de um compilador que não seja um código executável, mas sim, um bytecode. Um bytecode é um conjunto de alta otimização projetado para ser executado por uma Máquina Virtual Java (JVM, Java Virtual Machine). A JVM permite que um programa seja executado em diversos ambientes, pois requer apenas sua implementação para cada plataforma específica. Uma vez que, a JVM estiver presente em um determinado sistema, qualquer programa Java poderá ser executado nele.

Além da JVM, de acordo com Caelum (2003) existem mais dois grandes responsáveis, tanto pela execução quanto pelo desenvolvimento de uma aplicação Java, sendo eles o JRE (Java Runtime Environment) e o JDK (Java Development Kit).

JRE é um ambiente de execução Java, constituído pela JVM e bibliotecas, o essencial para executar qualquer aplicação Java. O JDK possui todas as ferramentas para o desenvolvimento de uma aplicação Java. Ou seja, inclui um compilador, um depurador (identifica e corrige erros) e o JRE para executar os programas.

Figura 1 - Exemplo do código Java

```

1  package exemplotcc;
2
3  import java.util.Scanner;
4
5  public class ExemploTCC {
6
7      public static void main(String[] args) {
8          double n1, n2, n3, n4, media;
9
10         Scanner entrada = new Scanner (source: System.in);
11
12         System.out.println("Digite sua primeira nota:");
13         n1 = entrada.nextDouble();
14         System.out.println("Digite sua segunda nota:");
15         n2 = entrada.nextDouble();
16         System.out.println("Digite sua terceira nota:");
17         n3 = entrada.nextDouble();
18         System.out.println("Digite sua quarta nota:");
19         n4 = entrada.nextDouble();
20
21         media = (n1 + n2 + n3 + n4)/4;
22
23         System.out.println("Media = " + media);
24
25     }
26 }
27
28

```

Fonte: Do próprio autor, 2023.

O exemplo acima exibe um código básico Java onde o programa calculará a média de quatro notas, que serão ditadas pelo usuário. A explicação a seguir detalha o significado e a interpretação de cada linha.

Linha 1: corresponde ao package (pacote), pacote esse que serve para organizar e agrupar classes relacionadas em um projeto Java.

Linha 3: está importando a classe 'Scanner' da biblioteca "padrão" Java. A classe 'Scanner' é responsável pela coleta de dados inseridos por um usuário.

Linha 5: indica a abertura da classe, que nomeamos como 'ExemploTCC', classe que contém o método 'main' (linha 7) que será o pontapé da nossa aplicação.

Linha 8: declaração das variáveis 'n1', 'n2', 'n3', 'n4' e 'media' do tipo 'double' (permite a inserção de números decimais). A escolha da letra 'n' é para remeter a palavra 'nota' sem utilizá-la por inteiro.

Linha 10: cria a instância da classe 'Scanner', que nomeamos de 'entrada'.

Linha 12: o programa irá imprimir a mensagem e aguardar um retorno do usuário. Esse processo se repete nas linhas 14, 16 e 18.

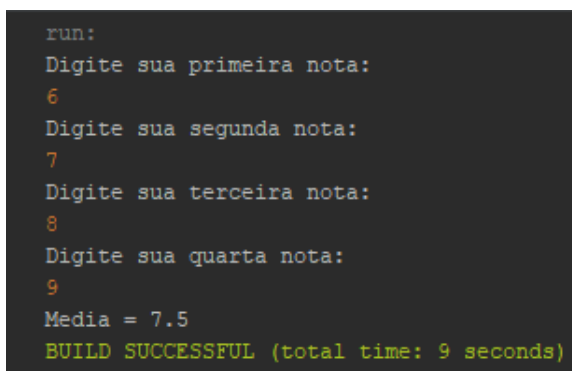
Linha 13: o programa irá armazenar o número digitado na variável 'n1'. Isso só é possível graças ao método 'nextDouble()' do objeto 'entrada'. Esse processo se repete com as outras variáveis nas linhas 15, 17 e 19.

Linha 21: responsável pelo cálculo da média, onde o programa irá primeiro somar a quatros notas e logo após dividir por 4 (quantidade de notas atribuídas).

Linha 23: será responsável por imprimir para o usuário a média calculada.

Linha 26 e 28: fechamento do método 'main' e da classe 'ExemploTCC', respectivamente.

Figura 2 - Resultado do código Java



```
run:
Digite sua primeira nota:
6
Digite sua segunda nota:
7
Digite sua terceira nota:
8
Digite sua quarta nota:
9
Media = 7.5
BUILD SUCCESSFUL (total time: 9 seconds)
```

Fonte: Do próprio autor, 2023.

2.3.3 Python

Python é uma linguagem de programação de alto nível e código aberto, criada pelo holandês Guido Van Rossum, em 1991. Ela possui uma estrutura de dados de alto nível, e é considerada como uma forma simples e eficiente de abordar programação orientada a objetos.

Sendo uma das linguagens mais utilizadas no mercado, pois com sua elegância de sintaxe, tipagem dinâmica e natureza interpretativa, ela se torna a preferida para desenvolvimento de aplicações rápidas em diversas áreas (PYTHON, 2023).

Segundo dados da pesquisa Stack Overflow Developer Survey (2022), Python aparece entre as 5 linguagens de programação mais utilizadas do mundo nos últimos 4 anos. Esse destaque recente mostra como essa linguagem de programação vem ganhando cada vez mais espaço e sendo explorada por aqueles que estão envolvidos na área de desenvolvimento de aplicações.

Figura 3 - As tecnologias mais populares de 2022



Fonte: (Stack Overflow Developer Survey, 2022)

O exemplo abaixo mostra uma aplicação feita para calcular a média aritmética de um aluno.

Figura 4 - Exemplo de código Python

```
1  def calculo_media(nota1, nota2, nota3, nota4):
2      soma = nota1 + nota2 + nota3 + nota4
3      media = (soma) / 4
4      return media
5
6  n1 = float(input("Insira a sua 1ª nota: "))
7  n2 = float(input("Insira a sua 2ª nota: "))
8  n3 = float(input("Insira a sua 3ª nota: "))
9  n4 = float(input("Insira a sua 4ª nota: "))
10
11 resultado = calculo_media(nota1=n1, nota2=n2, nota3=n3, nota4=n4)
12 print("Sua média é de:", resultado)
```

Fonte: Do próprio autor, 2023.

Na primeira linha, é declarada a função “calculo_media” por meio da palavra-chave “def”, essa função exige a passagem dos parâmetros “nota1”, “nota2”, “nota3” e “nota4”, que serão utilizados para a realização dos cálculos.

Segundo Borges (2010), uma função em Python é um conjunto de dados que executa uma determinada tarefa. Em um programa, é possível haver a composição de diversas funções, sendo então uma forma de dividi-lo em módulos, com finalidades específicas. Ela é responsável por evitar que um código seja escrito de forma repetida, já que, uma vez definida, a função poderá ser chamada diversas vezes, dependendo unicamente da necessidade do sistema em utilizá-la. Além disso, uma função poderá conter parâmetros para a sua execução ou retornar um valor, porém isso é opcional.

Nas linhas de 2 a 4, são inseridas as instruções para a realização do cálculo da média aritmética, e neste caso, é atribuído um valor de retorno da função por meio da palavra-chave “return”. Em seguida, nas linhas de 6 a 9 são solicitados e obtidos os valores das quatro notas do aluno, sendo atribuídos a n1, n2, n3 e n4, respectivamente.

Para conseguirmos inserir e obter os valores das 4 notas, utilizamos as funções “float” e “input”. De acordo com Saade (2019), Python possui uma biblioteca de funções predefinidas, que podem ser utilizadas para diversas finalidades. Nesse caso, a função “input” converte o que foi previamente inserido para um valor do tipo string, que, ao utilizar a função “float”, esse valor é convertido para um número decimal.

Na linha 11, é atribuído o valor de retorno da função “calculo_media” ao atributo “resultado”, utilizando os valores inseridos previamente (na linha de 6 a 9) como parâmetros para essa função. Já na última linha, é apresentado o valor da média do aluno. Vejamos um exemplo da execução deste programa, com dados aleatoriamente inseridos:

Figura 5 - Inserindo os dados após a execução do código

```
Insira a sua 1ª nota: 10
Insira a sua 2ª nota: 3
Insira a sua 3ª nota: 
```

Fonte: Do próprio autor, 2023.

A imagem a seguir expõe o resultado do cálculo, após realizar a inserção de dados.

Figura 6 - Resultado do código Python após a inserção de dados

```
Insira a sua 1ª nota: 10
Insira a sua 2ª nota: 3
Insira a sua 3ª nota: 1.5
Insira a sua 4ª nota: 5.5
Sua média é de: 5.0
```

Fonte: Do próprio autor, 2023.

2.3.4 OpenCV

O OpenCV (Open Source Computer Vision Library) é uma biblioteca de código aberto amplamente utilizada para processamento de imagens e visão computacional.

Segundo Antonello (2017), ela fornece um conjunto abrangente de funções e algoritmos que permitem aos desenvolvedores trabalharem com detecção de objetos, reconhecimento facial, entre outros.

A biblioteca OpenCV possui uma ampla gama de recursos e módulos. Segundo Marengoni (2014), alguns dos principais são o processamento de imagens, detecção e rastreamento de objetos, reconhecimento facial, visão estéreo e calibração de câmera, e aprendizado de máquina.

A imagem abaixo apresenta um exemplo básico de uso do OpenCV ao carregar e exibir uma imagem em Python.

Figura 7 - Exemplo da implementação do OpenCV

```
import cv2

# Carrega a imagem
image = cv2.imread('imagem.jpg')

# Exibe a imagem em uma janela
cv2.imshow('Imagem', image)

# Aguarda até que uma tecla seja pressionada para fechar a janela
cv2.waitKey(0)

# Fecha todas as janelas abertas
cv2.destroyAllWindows()
```

Fonte: Do próprio autor, 2023.

Esse exemplo ilustra o uso básico da OpenCV, nele estamos carregando uma imagem e a exibindo em uma janela.

Em suma, a OpenCV é uma biblioteca poderosa para processamento de imagens e visão computacional, oferecendo uma ampla gama de recursos e algoritmos para diversas aplicações.

2.3.5 Banco de Dados

Um banco de dados é uma ferramenta essencial no mundo da tecnologia, permitindo o armazenamento e gerenciamento eficiente de grandes volumes de dados. De acordo com Date (2004), um banco de dados pode ser definido como uma coleção de dados persistentes, usados pelas aplicações de um sistema de informação para suportar seu funcionamento.

Com o avanço das tecnologias, a utilização do banco de dados tem se tornado cada vez maior, segundo Elmasri e Navathe (2011), o banco de dados é de extrema importância para o mundo tecnológico, desempenhando um papel crítico em quase todas as áreas em que os computadores são utilizados, como na medicina.

Para garantir um bom desempenho e a integridade dos dados, é importante contar com um Sistema de Gerenciamento de Banco de Dados (SGBD). Como afirma Ramakrishnan e Gehrke (2007), um SGBD é um software que auxilia no

gerenciamento e na manutenção de um vasto conjunto de dados. Ele fornece recursos para armazenar, recuperar, atualizar e excluir informações, além de garantir a segurança e a consistência dos dados armazenados.

Segundo Elmasri e Navathe (2011), os SGBDs relacionais (SGBDR) são os mais utilizados na indústria atualmente. Esses sistemas seguem o modelo relacional, em que os dados são organizados em tabelas relacionadas entre si por meio de chaves primárias e chaves estrangeiras, conforme descrito por Date (2004).

2.3.5.1 SQL

A SQL (Structured Query Language, Linguagem de Consulta Estruturada) é a linguagem de banco de dados relacional comercial mais amplamente utilizada para gerenciar e consultar dados, como afirma Elmasri e Navathe (2011). Ela oferece uma sintaxe declarativa que permite aos usuários descreverem as informações que desejam obter, sem se preocupar com a implementação detalhada do processo de recuperação dos dados.

Com a SQL, é possível realizar uma variedade de operações em bancos de dados relacionais, desde consultas básicas até manipulações complexas de dados. Os comandos como SELECT, INSERT, UPDATE e DELETE permitem buscar informações específicas, adicionar registros, atualizar dados existentes e excluir informações indesejadas, conforme descrito por Ramakrishnan e Gehrke (2007). Segundo Elmasri e Navathe (2011), os dados manuseados por esses comandos, seguem uma estrutura baseada em tabelas, linhas e colunas ou conforme dita o modelo relacional formal: relação, tupla e atributos.

Além disso, a SQL oferece recursos avançados para a definição e manipulação de estruturas de banco de dados.

De acordo com Silberschatz, Korth e Sudarshan (2012), isso inclui a criação de tabelas, definição de chaves primárias e estrangeiras, criação de índices e outras operações no esquema, para que os usuários gerenciem seus bancos de dados

2.3.5.2 MySQL

Segundo Milani (2007, p.22), o MySQL é o Sistema de Gerenciamento de Banco de Dados Relacional (SGBDR) Open Source (código aberto) mais utilizado do mundo. Projetado inicialmente para suprir a necessidade de aplicações de pequeno e médio porte. Hoje, o MySQL pode ser utilizado em grandes aplicações e com mais vantagens que suas concorrentes.

De acordo Carvalho (2015), o MySQL, como qualquer banco de dados, é constituído por: campos, colunas, linhas ou tuplas e tabelas. Os campos são utilizados para a inserção de um dado específico; as colunas armazenam os registros de um determinado campo; as tuplas são as linhas de registro de um conjunto de campos; e as tabelas são os conjuntos de linhas, campos e colunas. Para uma melhor visualização, segue abaixo um exemplo de como esses componentes constituem um banco de dados.

Figura 8 - Composição de uma tabela de banco de dados

O diagrama mostra uma tabela com 5 linhas e 4 colunas. As colunas são: ID, NOME, DATA_NASCIMENTO e ID_ESTADO. As linhas representam registros de pessoas. Há três legendas: uma verde no topo direito apontando para o cabeçalho das colunas ('Nome do Campo ou Nome do Atributo'), uma amarela à esquerda apontando para uma linha ('Linha ou Tupla') e uma vermelha no fundo apontando para uma coluna ('Coluna ou Atributo').

ID	NOME	DATA_NASCIMENTO	ID_ESTADO
1	DANIEL	2014-03-29	3
2	BRUNA	2004-06-04	1
3	VINICIUS S	2004-02-12	1
4	VITÓRIA	2004-12-17	5
5	THAIS	1987-08-27	1

Fonte: (MySQL, 2015, p.5)

2.3.5.3 Diagrama Entidade-Relacionamento

O Diagrama Entidade-Relacionamento (DER) é uma ferramenta fundamental na modelagem de banco de dados, permitindo a representação gráfica das relações entre entidades de um sistema. Conforme destacado por Elmasri e Navathe (2011), o DER

é uma técnica que oferece uma visão visual e abstrata das entidades, seus atributos e os relacionamentos que existem entre elas.

2.3.5.4 Modelo Entidade-Relacionamento

O modelo entidade-relacionamento é construído para que todos os tipos de leitores consigam entender o relacionamento entre as entidades e seus atributos em um banco de dados relacional (HEUSER, 2009).

Além de ser muito utilizado e popular no processo de desenvolvimento conceitual de projetos, pois ele é considerado como uma representação formal e precisa do banco de dados de uma aplicação (ELMASRI, NAVATHE, 2011).

2.3.6 IoT

A Internet das Coisas (IoT) é a interconexão de dispositivos físicos por meio da internet, segundo Magrani (2018), ela possui uma ampla gama de aplicações, desde casas inteligentes até otimização industrial e cidades inteligentes, conectando dispositivos como sensores, câmeras e eletrodomésticos à internet.

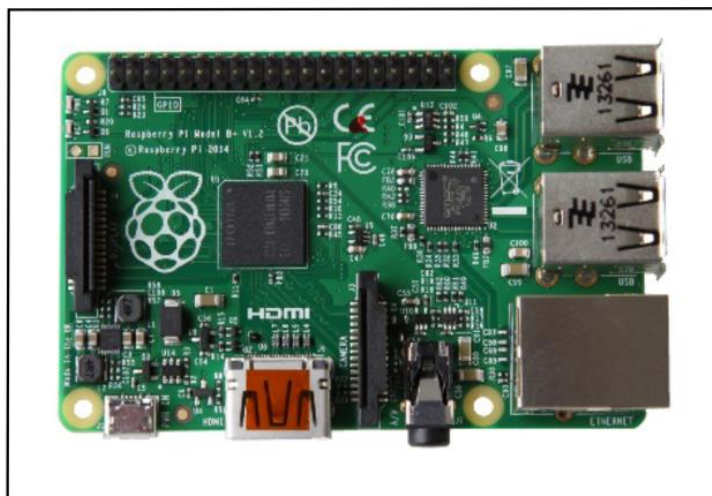
De acordo com Oliveira (2017), a IoT coleta dados valiosos sobre comportamentos e padrões de uso, possibilitando automação de tarefas e oferecendo insights baseados em dados assim a IoT tem o potencial de revolucionar diversos setores.

2.3.7 Raspberry Pi

O Raspberry Pi é um computador de placa única desenvolvido pela Fundação Raspberry Pi cujo um dos fundadores foi o renomado Eben Upton.

Segundo Upton (2017), cada versão do Raspberry possui suas próprias especificações e recursos, isso vai desde o processador, até a quantidade de memória RAM e as interfaces de comunicação que podem variar entre os modelos. Contudo, segundo Ebermam (2017), ele ainda vem a oferecer suporte a diferentes sistemas operacionais, sendo o Raspbian (uma distribuição do Linux) o mais comumente utilizado. A imagem abaixo apresenta um exemplo de modelo do Raspberry Pi.

Figura 9 - Exemplo de uma placa Raspberry Pi B+



Fonte: (Raspberry Pi, 2023)

Segundo Upton (2017), o Raspberry possui um slot para cartão de memória Micro SD, onde o sistema operacional e os dados são armazenados, com capacidade de até 64 GB.

2.3.8 Figma

O Figma é uma ferramenta de design colaborativo em ascensão, que permite a edição de vetores e prototipagem tanto em navegadores quanto em aplicativos desktop (FIGMA, 2023). Sua maior vantagem é possibilitar a colaboração em tempo real de equipes de design, independentemente da localização. A plataforma combina a acessibilidade da web com as funcionalidades avançadas de aplicativos nativos, sendo amplamente adotada por profissionais para criar designs interativos e colaborativos em projetos digitais.

2.3.8.1 Wireframe

Wireframe é uma representação visual de um modelo de design que pode ser simples ou até mesmo detalhado, e que muitas vezes vai vir a ser o seu design para o modelo final (TEIXEIRA, 2014). Também sendo considerada uma das ferramentas mais simples que você pode utilizar para transmitir uma grande ideia, assim facilitando o entendimento de todos aqueles que vão interagir com ela. Existem vários tipos de

wireframes, mas no SR's foram utilizados apenas os de baixa fidelidade e alta fidelidade.

3 DESENVOLVIMENTO

Neste capítulo, vamos explorar o processo de criação do SR's. O desenvolvimento da aplicação é registrado por meio da utilização de métodos de estudo UML, ilustrações de códigos e representações visuais das telas do software.

3.1 Documentação do Caso de Uso

A documentação de um caso de uso costuma descrever, de forma simples, informações como a função do caso de uso, quais atores interagem com ele, quais etapas devem ser realizadas pelo ator e pelo sistema para que o caso de uso cumpra sua finalidade, quais informações devem ser fornecidas e quais condições e verificações o caso de uso deve possuir.

3.1.1 Requisitos funcionais

Requisitos funcionais são especificações das funcionalidades propostas pelo sistema, definindo assim as ações que o sistema deve ser capaz de realizar. Esses requisitos visam satisfazer as necessidades dos usuários de um sistema.

Requisitos funcionais do Porteiro:

- RF01 – O sistema deverá permitir que o porteiro efetue login
- RF02 – O sistema deverá permitir que o porteiro cadastre os moradores
- RF03 – O sistema deverá permitir que o porteiro possa gerenciar os dados dos moradores
- RF04 – O sistema deverá permitir que o porteiro cadastre os veículos
- RF05 – O sistema deverá permitir que o porteiro possa gerenciar os dados dos veículos
- RF06 – O sistema deverá permitir que o porteiro possa acessar os registros de entrada dos veículos

Requisitos funcionais do Administrador:

- RF01 – O sistema deverá permitir que o administrador efetue login
- RF02 – O sistema deverá permitir que o administrador cadastre os porteiros
- RF03 – O sistema deverá permitir que o administrador possa gerenciar os dados dos porteiros
- RF04 – O sistema deverá permitir que o administrador cadastre os condomínios
- RF05 – O sistema deverá permitir que o administrador possa gerenciar os dados dos condomínios

Requisitos Funcionais do SRs:

- RF01 – O dispositivo deverá autenticar a placa do veículo, por meio da digitalização de imagem
- RF02 – O dispositivo deverá realizar a abertura automática do portão caso o veículo seja daquele condomínio.

3.1.2 Requisitos não-funcionais

Também conhecidos como RNF, são critérios que descrevem os atributos de qualidade de um sistema, além de suas funcionalidades principais. Eles especificam os critérios pelos quais o desempenho de um sistema pode ser avaliado, em vez de se referirem diretamente às funcionalidades específicas que o sistema deve realizar.

Requisitos não-funcionais da aplicação geral:

- RNF01 – O cadastro de porteiro não poderá ser realizado caso já exista algum registro com as mesmas informações no banco de dados.
- RNF02 – O porteiro terá acesso ao histórico de entrada dos veículos no estacionamento e ao gerenciamento de dados dos moradores.
- RNF03 – O tempo de resposta das interações entre o dispositivo e o sistema deverá ser curta
- RNF04 – O administrador terá acesso aos dados dos condomínios registrados e de seus respectivos porteiros

- RNF05 – O sistema deverá possuir uma interface amigável e de fácil entendimento

3.1.3 Regras de negócio

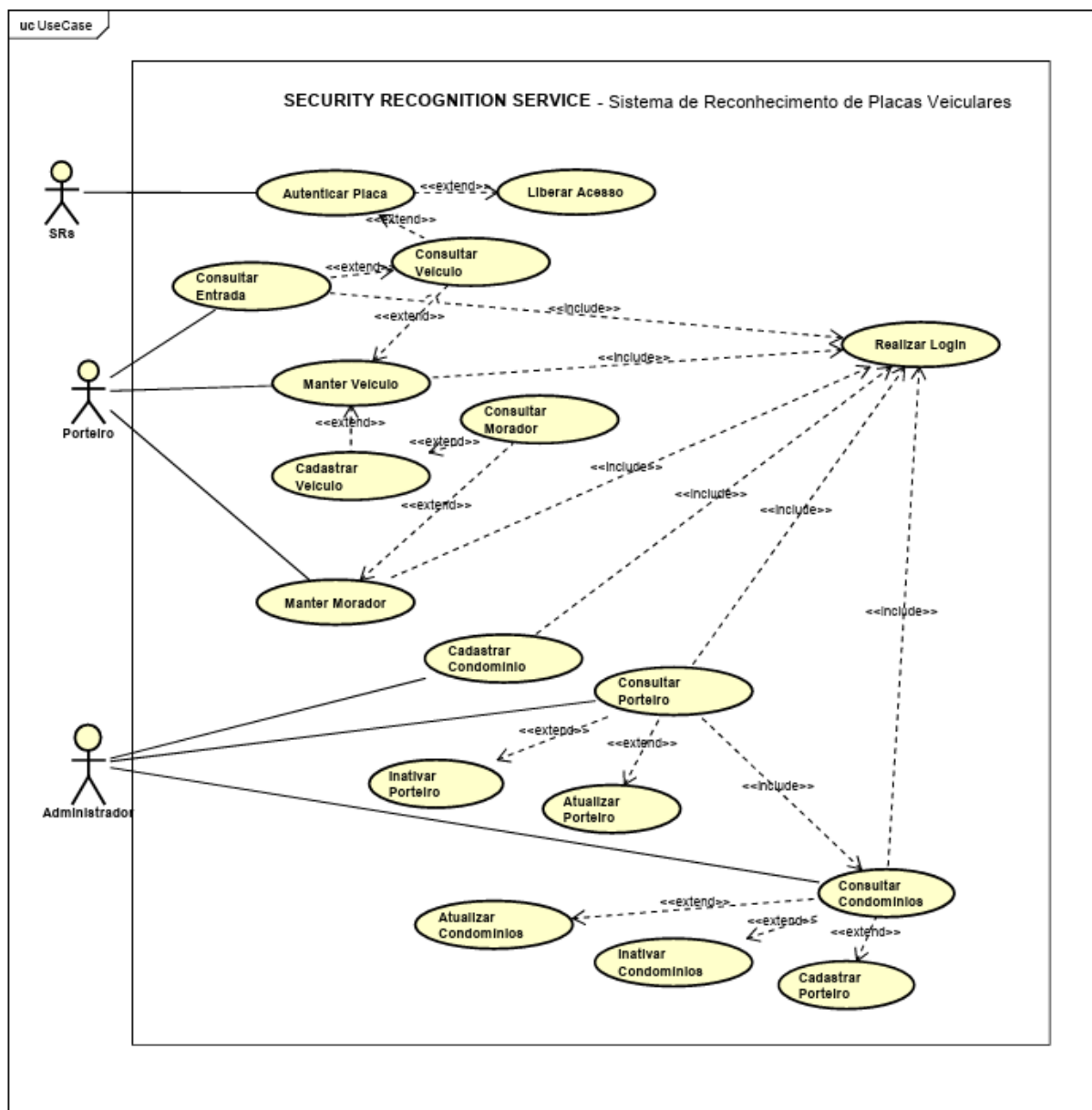
As regras de negócios são políticas, normas e condições estabelecidas pela empresa que devem ser seguidas na execução de alguma funcionalidade do sistema. Alguns requisitos não funcionais podem ser identificados como regras de negócios.

- RN01 – A aplicação será desenvolvida utilizando as linguagens Java e Python.
- RN02 – Os registros dos administradores serão feitos internamente, a própria empresa estará responsável por criar uma conta de administrador.
- RN03 – o campo de senha de login do administrador/porteiro deverá conter no mínimo 8 dígitos, necessariamente possuindo caracteres e números.
- RN04 – Os registros dos porteiros serão feitos pelo administrador, ou seja, ele estará responsável por criar a conta dos porteiros.

3.1.4 Diagrama de Caso de Uso

Através da figura a seguir, é possível observar o caso de uso do sistema, demonstrando a interação entre o porteiro, o SR's e o administrador no sistema, bem como as diferentes ações que cada um desses atores pode realizar.

Figura 10 - Diagrama de Caso de Uso geral da aplicação



Fonte: Do próprio autor, 2023.

3.1.4.1 Quadros de descrição dos casos de uso

Através dos quadros abaixo, é possível observar as descrições do caso de uso do sistema, demonstrando as informações de cada função do caso de uso, quais atores participam de cada interação e suas condições para funcionamento. Além de proporcionar uma perspectiva geral das ações de cada ator dentro do sistema.

Quadro 1 - Descrição do Caso de Uso geral a partir da perspectiva do administrador

Nome do Caso de Uso	SRs - Sistema de Reconhecimento de Placas Veiculares
Ator Principal	Administrador
Ator Secundário	Porteiro
Resumo	Este caso de uso descreve o fluxo de ações realizadas pelo administrador. Esta descrição é o cenário ideal em relação às ações do administrador.
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Cadastrar Condomínio	
	3. Listar Condomínios
4. Atualizar Condomínio	
5. Inativar Condomínio	
6. Cadastrar Porteiro	
	7. Listar Porteiros
8. Atualizar Porteiro	
9. Inativar Porteiro	

Fonte: Do próprio autor, 2023.

Quadro 2 - Descrição do Caso de Uso Geral da aplicação a partir da perspectiva do porteiro

Nome do Caso de Uso	SRs - Sistema de Reconhecimento de Placas Veiculares
Ator Principal	Porteiro
Ator Secundário	SRs
Resumo	Este caso de uso descreve o fluxo de ações realizadas pelo porteiro. Esta descrição é o cenário ideal em relação às ações do porteiro.
Ações do Ator	Ações do Sistema
1. Realizar Login	
	2. Listar Entrada
3. Cadastrar Morador	
	4. Listar Morador
5. Atualizar Morador	
6. Excluir Morador	
7. Cadastrar Morador	
	8. Listar Morador
9. Atualizar Morador	
10. Excluir Morador	

Fonte: Do próprio autor, 2023.

Quadro 3 – Descrição do Caso de Uso Geral da aplicação a partir da perspectiva do SRs

Nome do Caso de Uso	SRs – Sistema de Reconhecimento de Placas Veiculares
Ator Principal	SRs
Ator Secundário	Porteiro
Resumo	Este caso de uso descreve o fluxo de ações realizadas pelo SRs (dispositivo). Esta descrição é o cenário ideal em relação às ações do SRs.
Ações do Ator	Ações do Sistema
	1. Autenticar Placa
2. Liberar Acesso	

Fonte: Do próprio autor, 2023.

Quadro 4 – Descrição do Caso de Uso “Realizar Login” (administrador)

Nome do Caso de Uso	Realizar Login
Ator Principal	Administrador
Resumo	Este caso de uso descreve os passos para realização de login do administrador
Ações do Ator	Ações do Sistema
	1. Solicitar número de acesso e senha
2. Preencher os dados solicitados	
	3. Verificar dados
	4. Realizar Login

Fonte: Do próprio autor, 2023.

Quadro 5 – Descrição do Caso de Uso “Realizar Login” (porteiro)

Nome do Caso de Uso	Realizar Login
Ator Principal	Porteiro
Resumo	Este caso de uso descreve os passos para realização de login do porteiro
Ações do Ator	Ações do Sistema
	1. Solicitar número de acesso e senha
2. Preencher os dados solicitados	
	3. Verificar dados
	4. Realizar Login

Fonte: Do próprio autor, 2023.

Quadro 6 – Descrição do Caso de Uso “Listar Entrada”

Nome do Caso de Uso	Listar Entrada
Ator Principal	Porteiro
Resumo	Este caso de uso descreve os passos para que o porteiro liste o registro de entradas dos veículos no condomínio
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de listar a entrada dos veículos	
	3. Listar entrada

Fonte: Do próprio autor, 2023.

Quadro 7 – Descrição do Caso de Uso “Cadastrar Morador”

Nome do Caso de Uso	Cadastrar Morador
Ator Principal	Porteiro
Resumo	Este caso de uso descreve os passos para que o porteiro cadastre o morador
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de cadastrar morador	
	3. Solicitar nome e número do apartamento
4. Preencher campos solicitados	
	5. Solicitar confirmação do cadastro
6. Confirmar cadastro	
	7. Gravar no banco e concluir cadastro

Fonte: Do próprio autor, 2023.

Quadro 8 – Descrição do Caso de Uso “Listar Morador”

Nome do Caso de Uso	Listar Morador
Ator Principal	Porteiro
Resumo	Este caso de uso descreve os passos para que o porteiro liste os moradores cadastrados
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de listar os moradores cadastrados	
	3. Listar moradores

Fonte: Do próprio autor, 2023.

Quadro 9 – Descrição do Caso de Uso “Atualizar Morador”

Nome do Caso de Uso	Atualizar Morador
Ator Principal	Porteiro
Resumo	Este caso de uso descreve os passos para que o porteiro atualize o cadastro do morador
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de listar os moradores cadastrados	
	3. Listar moradores
4. Selecionar morador	
5. Editar informações de cadastro do morador	
	6. Solicitar confirmação para atualizar informações
7. Confirmar atualização	
	8. Gravar no banco e concluir atualização

Fonte: Do próprio autor, 2023.

Quadro 10 – Descrição do Caso de Uso “Excluir Morador”

Nome do Caso de Uso	Excluir Morador
Ator Principal	Porteiro
Resumo	Este caso de uso descreve os passos para que o porteiro exclua um morador cadastrado
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de listar os moradores cadastrados	
	3. Listar moradores
4. Selecionar morador	
5. Excluir morador	
	6. Solicitar confirmação para excluir informações
7. Confirmar exclusão	
	8. Remover registro do banco e concluir exclusão

Fonte: Do próprio autor, 2023.

Quadro 11 – Descrição do Caso de Uso “Cadastrar Veículo”

Nome do Caso de Uso	Cadastrar Veículo
Ator Principal	Porteiro
Resumo	Este caso de uso descreve os passos para que o porteiro cadastre o veículo
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de cadastrar veículo	
	3. Apresentar tela de cadastro de veículo
	4. Solicitar número da placa e o morador que utiliza o veículo
5. Preencher campos solicitados	
	6. Solicitar confirmação de cadastro
7. Confirmar cadastro	
	8. Gravar no banco e concluir cadastro

Fonte: Do próprio autor, 2023.

Quadro 12 – Descrição do Caso de Uso “Listar Veículo”

Nome do caso de uso	Listar Veículo
Ator Principal	Porteiro
Resumo	Este caso de uso descreve os passos para que o porteiro liste os veículos cadastrados
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de listar os veículos cadastrados	
	3. Listar veículos

Fonte: Do próprio autor, 2023.

Quadro 13 – Descrição do Caso de Uso “Atualizar Veículo”

Nome do caso de uso	Atualizar Veículo
Ator Principal	Porteiro
Resumo	Este caso de uso descreve os passos para que o porteiro atualize o cadastro dos veículos
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de listar os veículos cadastrados	
	3. Listar veículos
4. Selecionar veículo	
5. Editar informações de cadastro do veículo	
	6. Solicitar confirmação para atualizar informações
7. Confirmar atualização	
	8. Gravar no banco e concluir atualização

Fonte: Do próprio autor, 2023.

Quadro 14 - Descrição do Caso de Uso “Excluir Veículo”

Nome do caso de uso	Excluir Veículo
Ator Principal	Porteiro
Resumo	Este caso de uso descreve os passos para que o porteiro exclua o veículo
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de listar os veículos cadastrados	
	3. Listar veículos
4. Selecionar veículo	
5. Excluir veículo	
	6. Solicitar confirmação para excluir informações
7. Confirmar exclusão	
	8. Remover registro do banco e concluir exclusão

Fonte: Do próprio autor, 2023.

Quadro 15 - Descrição do Caso de Uso “Autenticar Placa”

Nome do caso de uso	Autenticar Placa
Ator Principal	SRs
Resumo	Este caso de uso descreve os passos para que o dispositivo verifique se a placa de um veículo está ou não registrada no banco de dados
Ações do Ator	Ações do Sistema
1. Capturar imagem da placa	
	2. Digitalizar imagem
	3. Converter imagem para um texto
	4. Listar veículos
	5. Verificar se há algum registro com essa placa no banco de dados

Fonte: Do próprio autor, 2023.

Quadro 16 - Descrição do Caso de Uso “Liberar Acesso”

Nome do Caso de Uso	Liberar Acesso
Ator Principal	SRs
Resumo	Este caso de uso descreve os passos do dispositivo para liberar a entrada de um veículo no estacionamento
Ações do Ator	Ações do Sistema
	1. Autenticar Placa
2. Realizar abertura do portão	
	3. Gravar no banco e liberar acesso
4. Verificar se o veículo já entrou no local	
5. Realizar fechamento do portão	

Fonte: Do próprio autor, 2023.

Quadro 17 - Descrição do Caso de Uso “Cadastrar Condomínio”

Nome do Caso de Uso	Cadastrar Condomínio
Ator Principal	Administrador
Resumo	Este caso de uso descreve os passos para que o administrador cadastre o condomínio
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de cadastrar condomínio	
	3. Solicitar nome, CNPJ, CEP, endereço, bairro, cidade e UF
4. Preencher campos solicitados	
	5. Verificar Dados
	6. Solicitar confirmação do cadastro
7. Confirmar cadastro	
	8. Gravar no banco e concluir cadastro

Fonte: Do próprio autor, 2023.

Quadro 18 - Descrição do Caso de Uso “Listar Condomínio”

Nome do Caso de Uso	Listar Condomínio
Ator Principal	Administrador
Resumo	Este caso de uso descreve os passos para que o administrador liste os condomínios cadastrados
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de listar os condomínios cadastrados	
	3. Listar condomínios

Fonte: Do próprio autor, 2023.

Quadro 19 - Descrição do Caso de Uso “Atualizar Condomínio”

Nome do Caso de Uso	Atualizar Condomínio
Ator Principal	Administrador
Resumo	Este caso de uso descreve os passos para que o administrador atualize o cadastro do condomínio
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de listar os condomínios cadastrados	
	3. Listar condomínios
4. Selecionar condomínio	
5. Editar informações de cadastro do condomínio	
	6. Verificar Dados
	7. Solicitar confirmação para atualizar informações
8. Confirmar atualização	
	9. Gravar no banco e concluir atualização

Fonte: Do próprio autor, 2023.

Quadro 20 - Descrição do Caso de Uso “Inativar Condomínio”

Nome do Caso de Uso	Inativar Condomínio
Ator Principal	Administrador
Resumo	Este caso de uso descreve os passos para que o administrador inative o condomínio
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de listar os condomínios cadastrados	
	3. Listar condomínios
4. Selecionar condomínio	
5. Inativar condomínio	
	6. Solicitar confirmação para inativar condomínio
7. Confirmar inativação	
	8. Gravar no banco e concluir atualização

Fonte: Do próprio autor, 2023.

Quadro 21 - Descrição do Caso de Uso “Cadastrar Porteiro”

Nome do Caso de Uso	Cadastrar Porteiro
Ator Principal	Administrador
Resumo	Este caso de uso descreve os passos para que o administrador cadastre o porteiro
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de cadastrar porteiro	
	3. Solicitar seleção do condomínio em que o porteiro trabalha
4. Realizar a seleção solicitada	
	5. Apresentar tela de cadastro de porteiros
	6. Solicitar nome, nome de acesso, senha de acesso e turno
7. Preencher campos solicitados	
	8. Verificar segurança da senha
	9. Solicitar confirmação de cadastro
10. Confirmar cadastro	
	11. Gravar no banco e concluir cadastro

Fonte: Do próprio autor, 2023.

Quadro 22 - Descrição do Caso de Uso “Listar Porteiro”

Nome do Caso de Uso	Listar Porteiro
Ator Principal	Administrador
Resumo	Este caso de uso descreve os passos para que o administrador liste os porteiros cadastrados
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de listar porteiro	
	3. Solicitar seleção do condomínio em que o porteiro trabalha
4. Realizar a seleção solicitada	
	5. Listar porteiros

Fonte: Do próprio autor, 2023.

Quadro 23 - Descrição do Caso de Uso “Atualizar Porteiro”

Nome do Caso de Uso	Atualizar Porteiro
Ator Principal	Administrador
Resumo	Este caso de uso descreve os passos para que o administrador atualize o cadastro do porteiro
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de listar porteiro	
	3. Solicitar seleção do condomínio em que o porteiro trabalha
4. Realizar a seleção solicitada	
	5. Listar porteiros
6. Selecionar porteiro	
7. Editar informações de cadastro do porteiro	
	8. Solicitar confirmação para atualizar informações
9. Confirmar atualização	
	10. Gravar no banco e concluir atualização

Fonte: Do próprio autor, 2023.

Quadro 24 - Descrição do Caso de Uso “Inativar Porteiro”

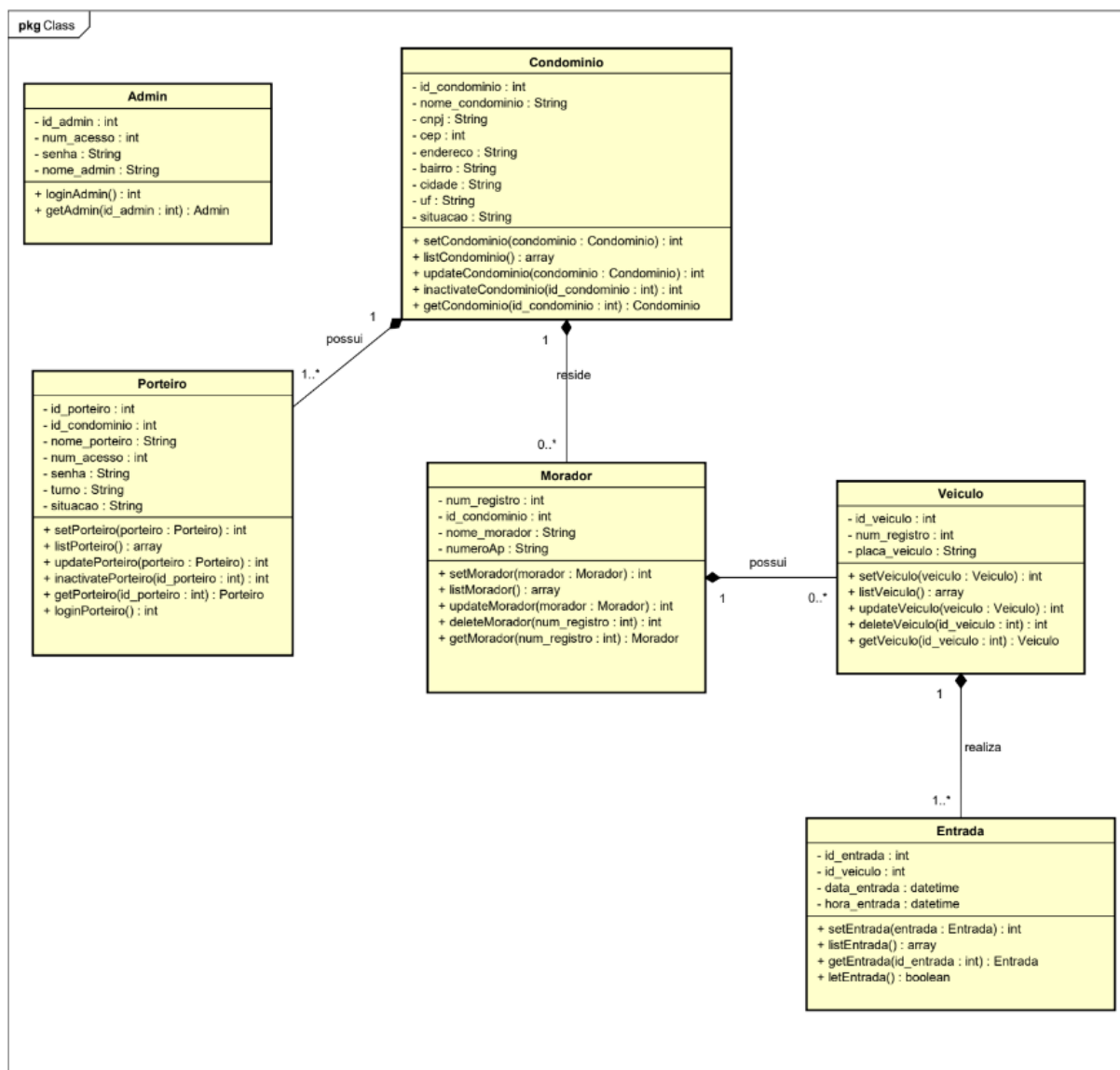
Nome do Caso de Uso	Inativar Porteiro
Ator Principal	Administrador
Resumo	Este caso de uso descreve os passos para que o administrador inative o porteiro
Ações do Ator	Ações do Sistema
1. Realizar Login	
2. Selecionar a opção de listar porteiro	
	3. Solicitar seleção do condomínio em que o porteiro trabalha
4. Realizar a seleção solicitada	
	5. Listar porteiros
6. Selecionar porteiro	
7. Inativar porteiro	
	8. Solicitar confirmação para inativar porteiro
9. Confirmar inativação	
	10. Gravar no banco e concluir atualização

Fonte: Do próprio autor, 2023.

3.2 Diagrama de Classe

Com a figura a seguir, é possível observar o diagrama de classes do sistema, que representa as estruturas e interações entre as entidades do sistema.

Figura 11 - Diagrama de classe da aplicação

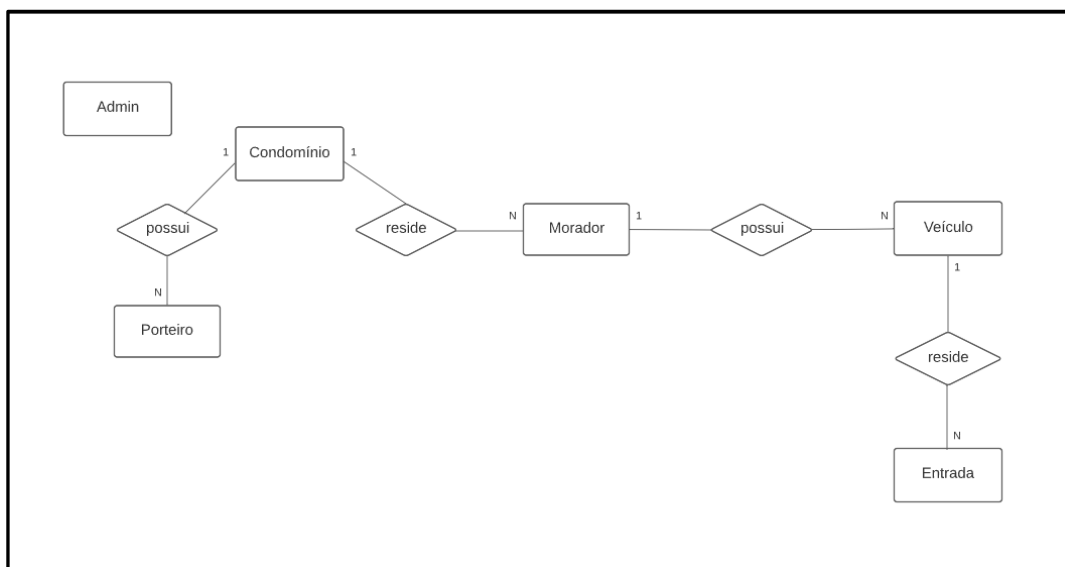


Fonte: Do próprio autor, 2023.

3.3 Diagrama Entidade-Relacionamento da Aplicação

Através da figura abaixo, podemos visualizar o Diagrama Entidade-Relacionamento (DER) que representa a estrutura conceitual do banco de dados. No diagrama, são apresentadas as entidades envolvidas, como elas se relacionam e suas especificações de relacionamento.

Figura 12 - DER da aplicação

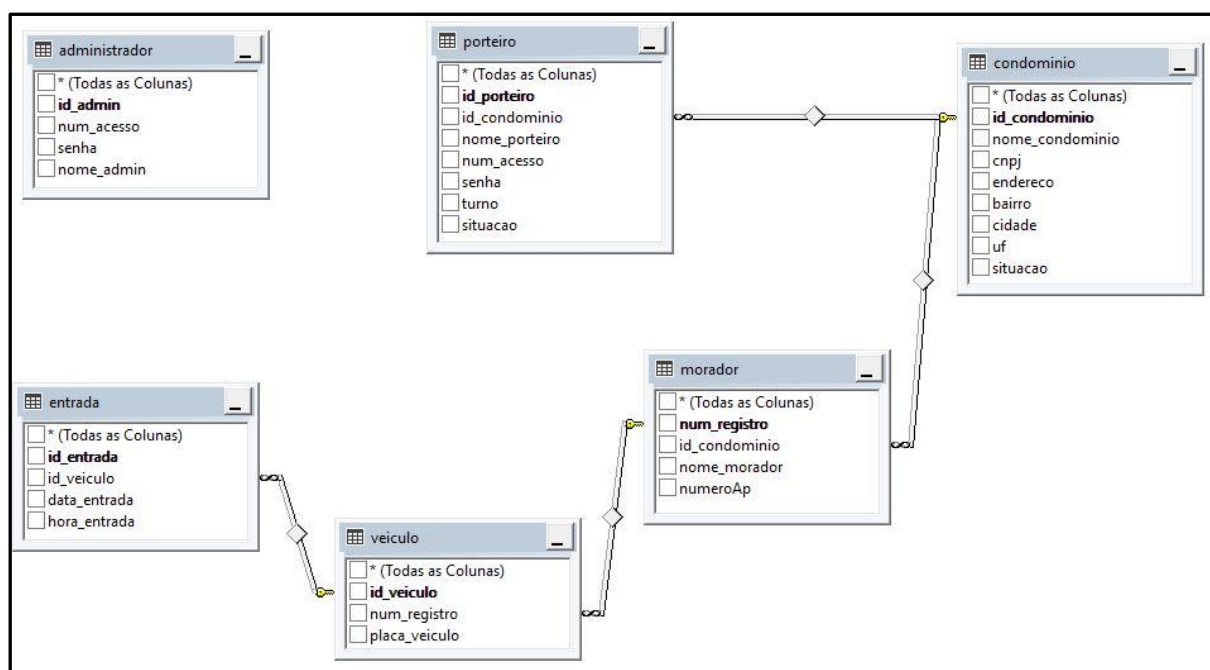


Fonte: Do próprio autor, 2023.

3.4 Modelo Entidade-Relacionamento da Aplicação

Através da figura abaixo, podemos visualizar o Modelo Entidade-Relacionamento (MER) que representa a estrutura do banco de dados. No modelo, são apresentadas as entidades envolvidas, seus atributos e como elas se relacionam mutualmente.

Figura 13 - MER da aplicação

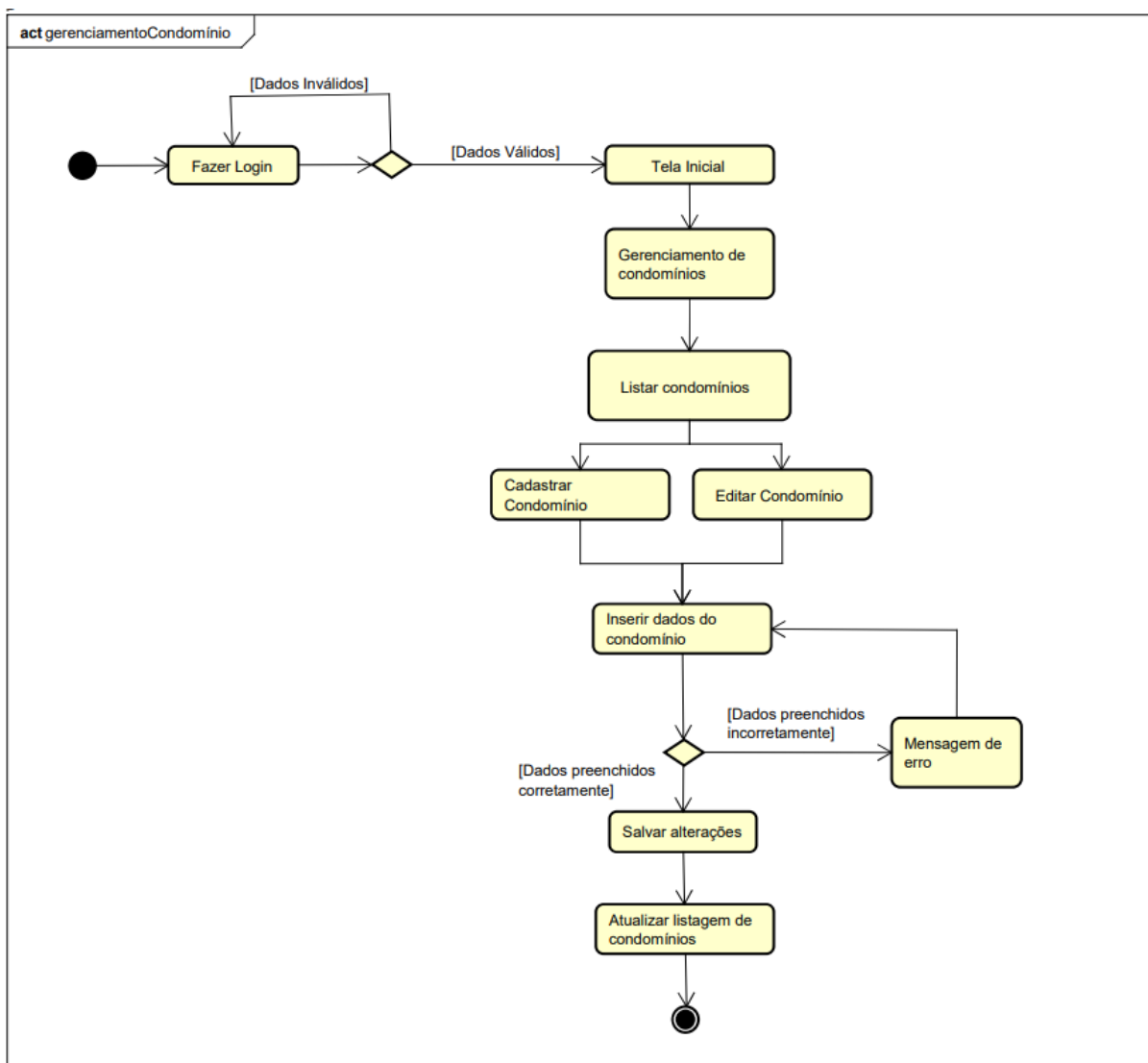


Fonte: Do próprio autor, 2023.

3.5 Diagramas de Atividade

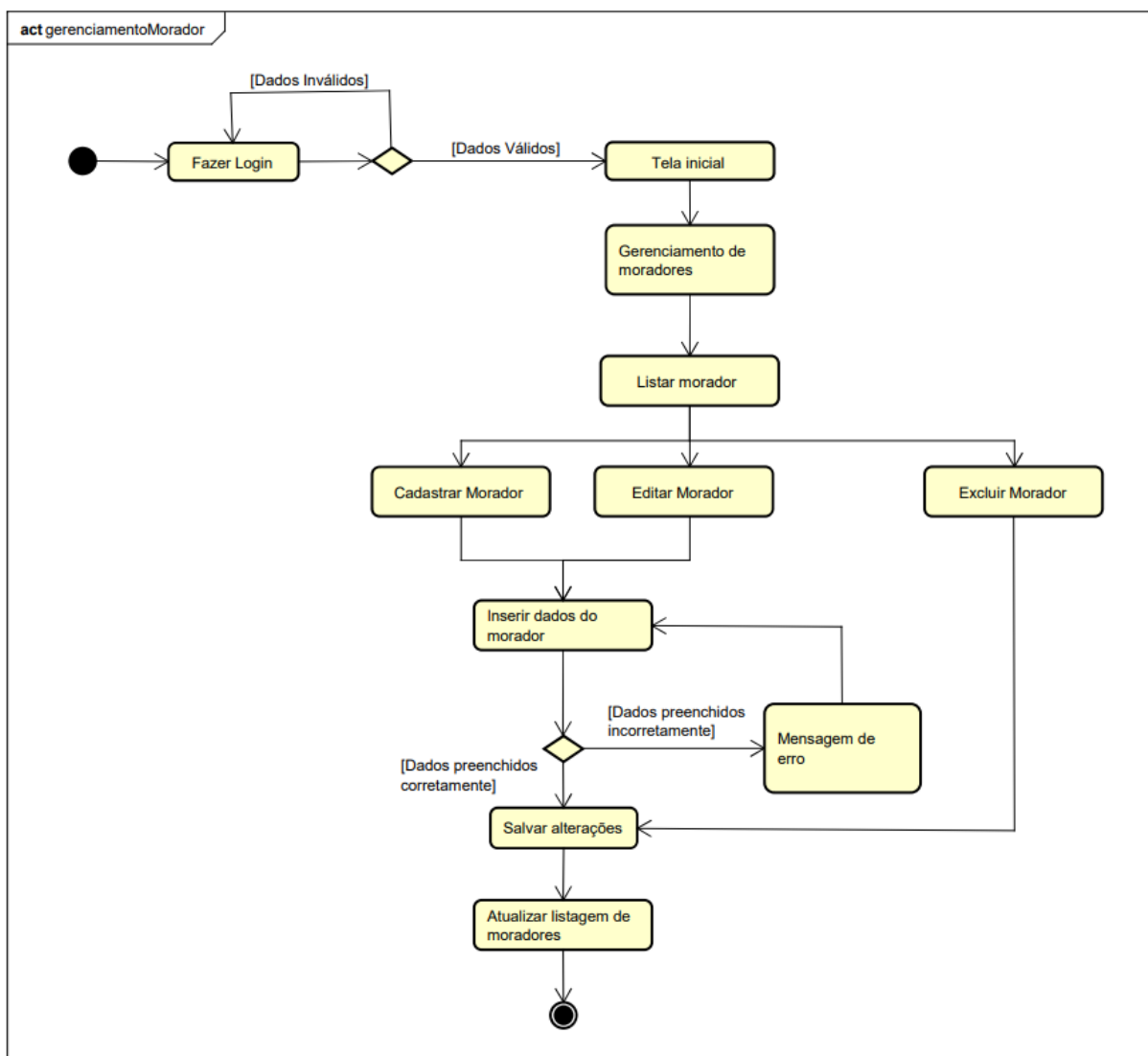
Ao examinar os diagramas a seguir, é viável analisar de forma sequencial o fluxo de controle de cada funcionalidade presente no SR's.

Figura 14 - Diagrama de atividade do gerenciamento de condomínio



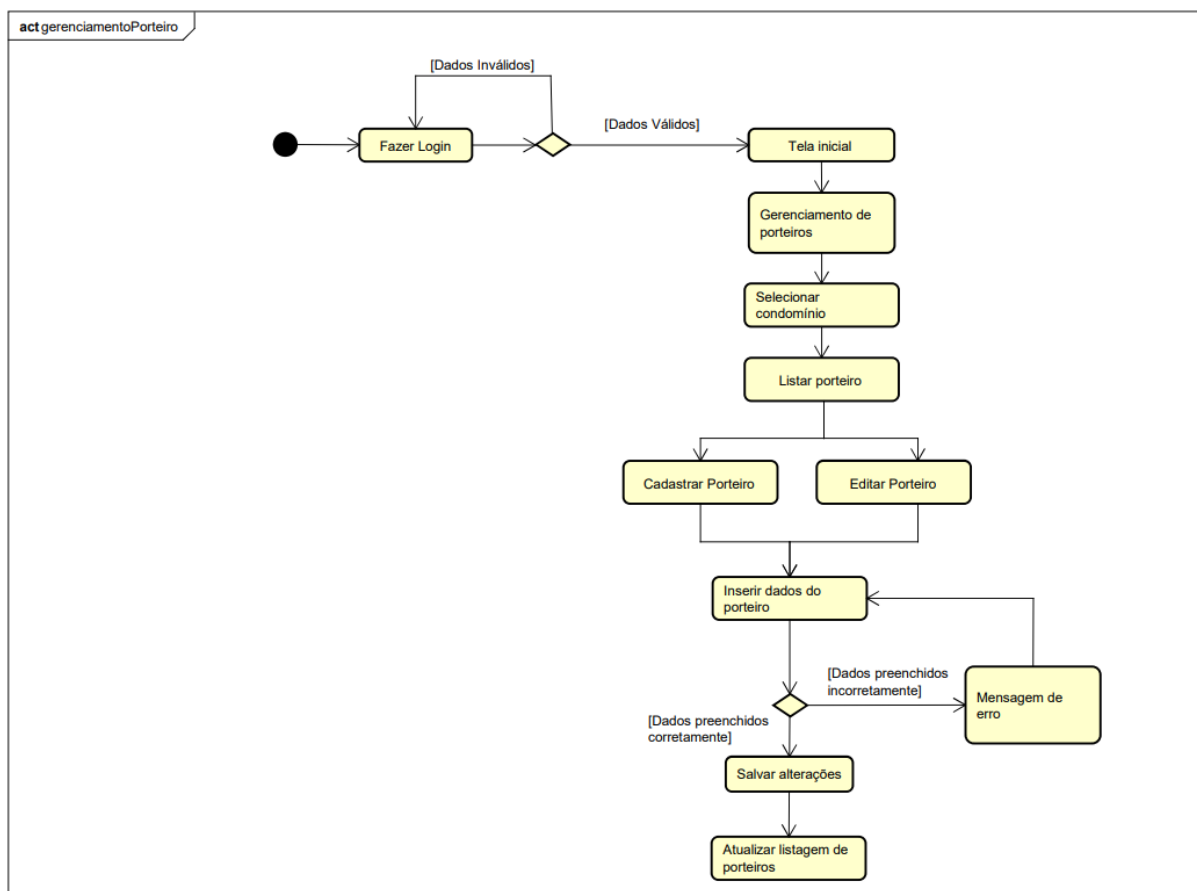
Fonte: Do próprio autor, 2023.

Figura 15 - Diagrama de atividade do gerenciamento de morador



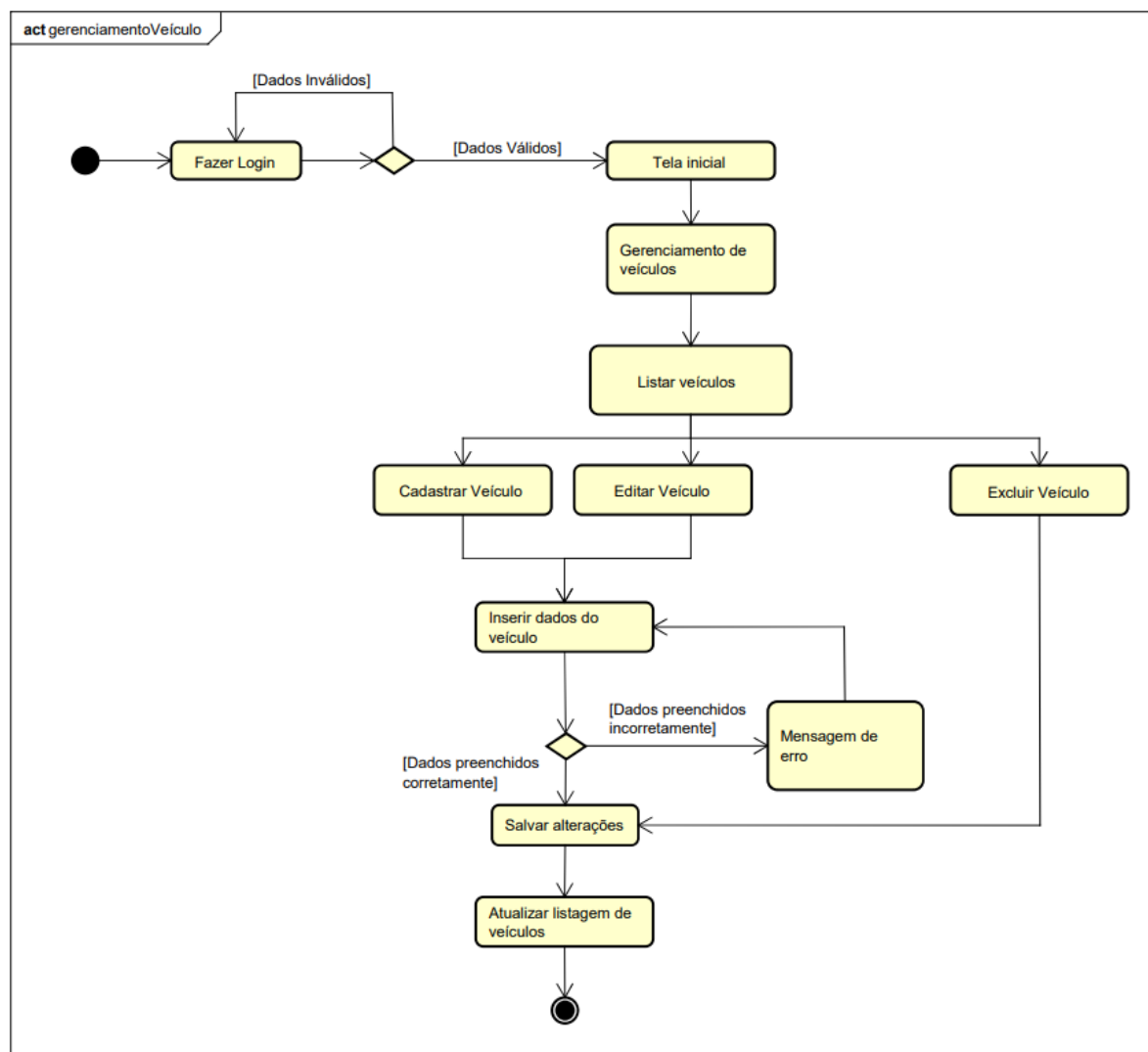
Fonte: Do próprio autor, 2023.

Figura 16 - Diagrama de atividade do gerenciamento de porteiro



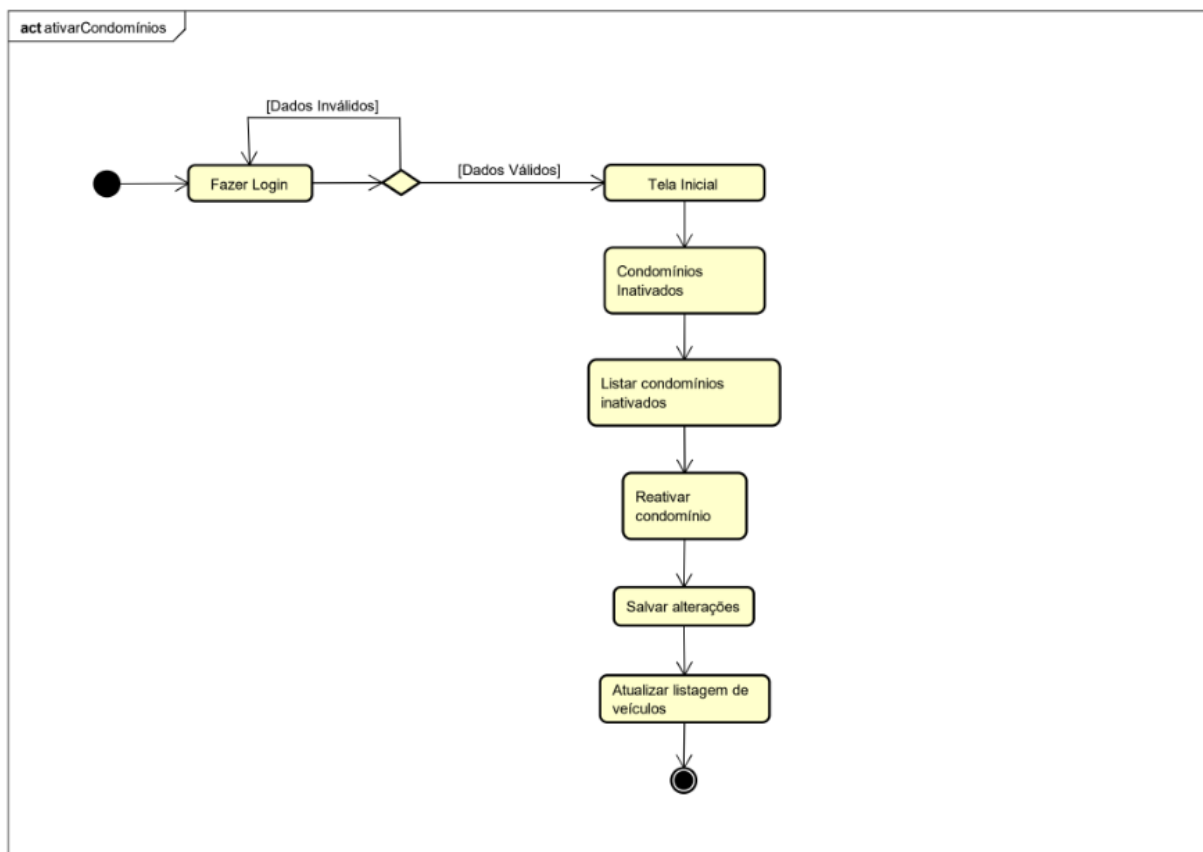
Fonte: Do próprio autor, 2023.

Figura 17 - Diagrama de atividade do gerenciamento de veículos



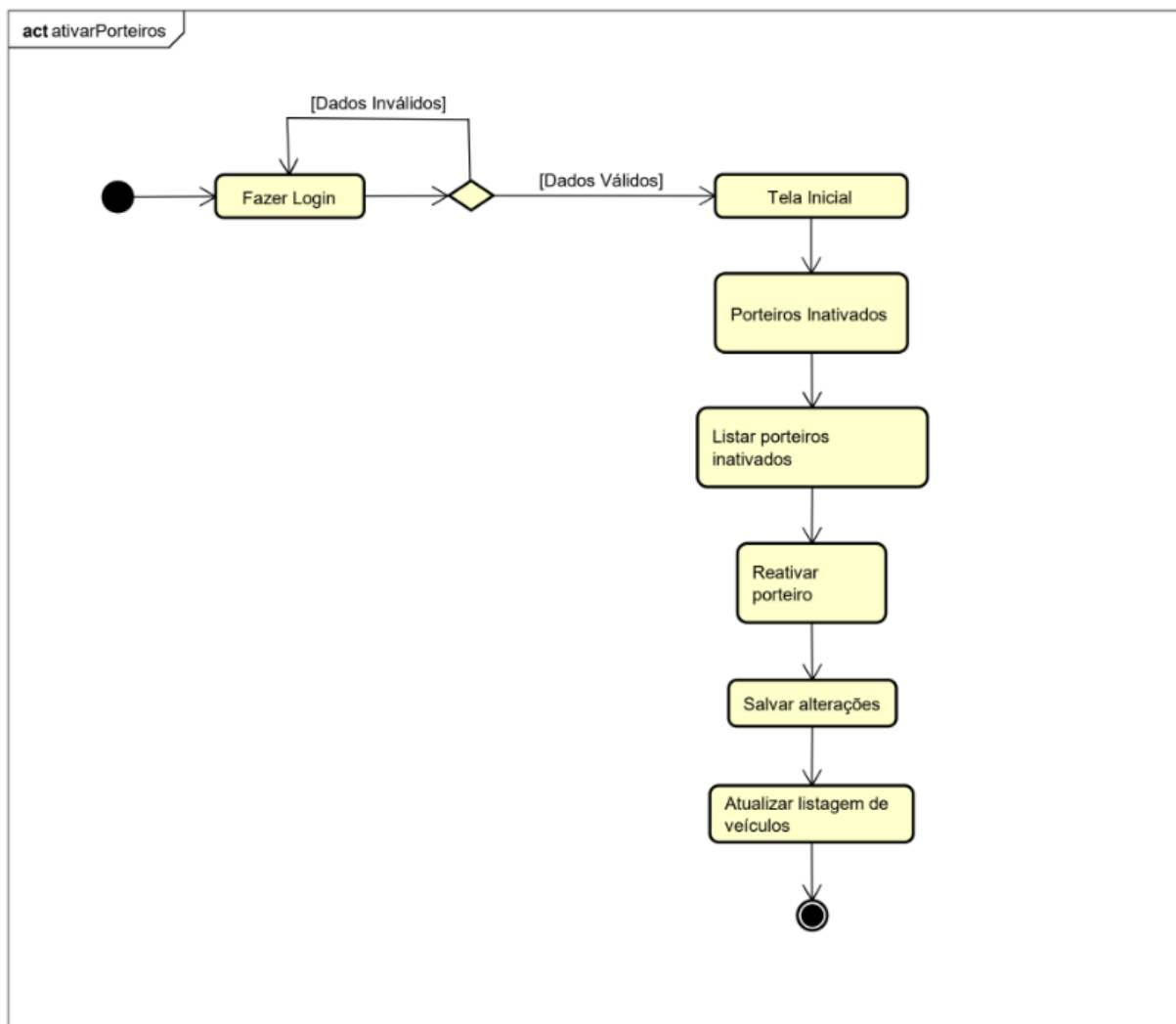
Fonte: Do próprio autor, 2023.

Figura 18 - Diagrama de atividade para ativação de condomínios



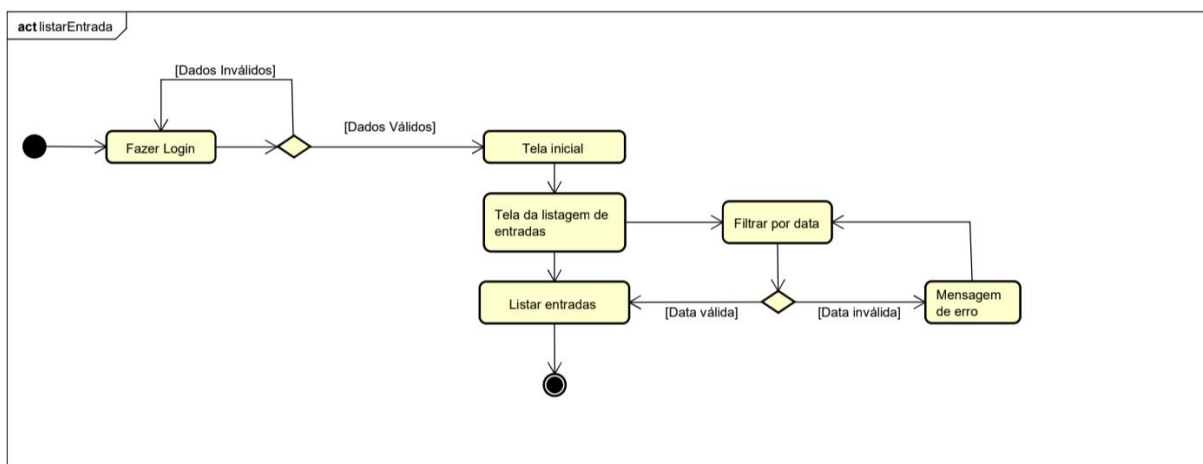
Fonte: Do próprio autor, 2023.

Figura 19 - Diagrama de atividades para ativação de porteiros



Fonte: Do próprio autor, 2023.

Figura 20 - Diagrama de atividade para listar entrada

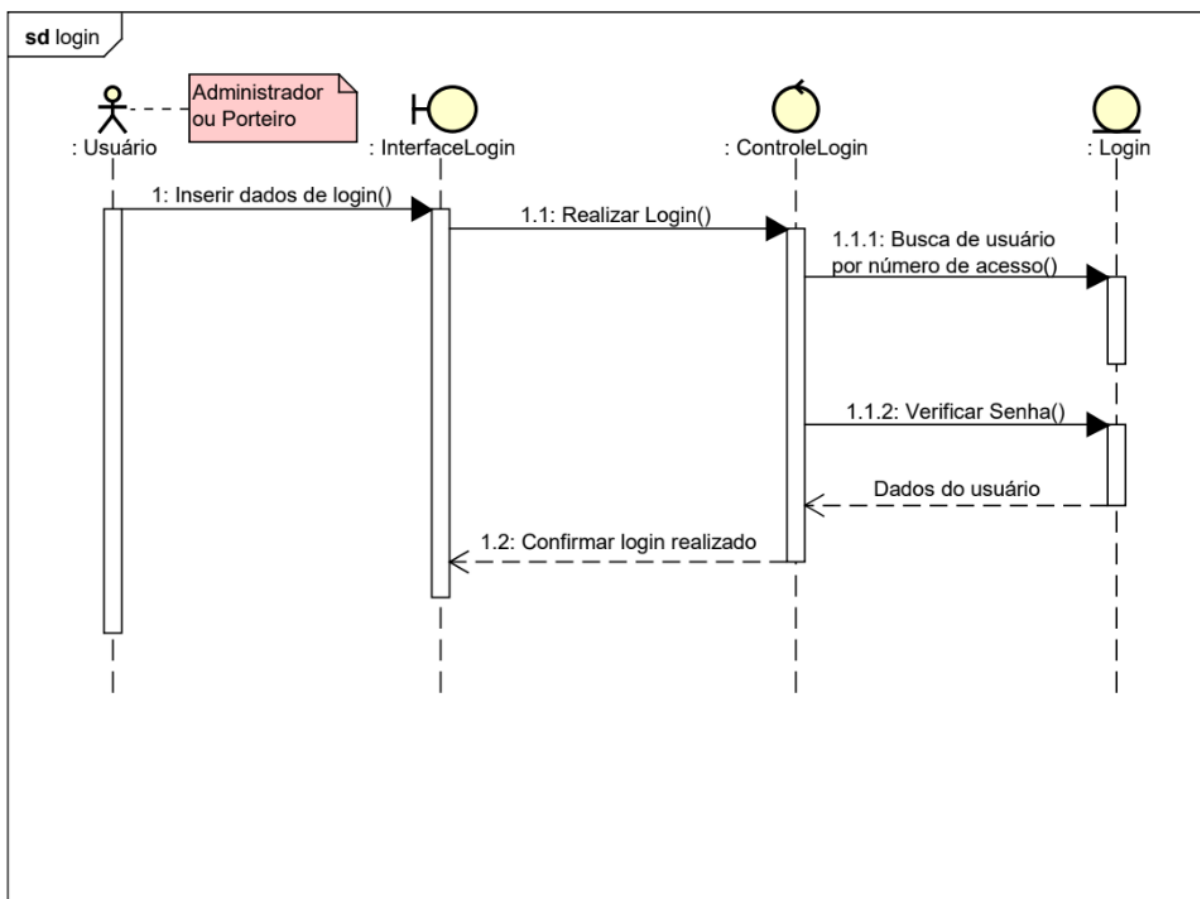


Fonte: Do próprio autor, 2023.

3.6 Diagramas de Sequência

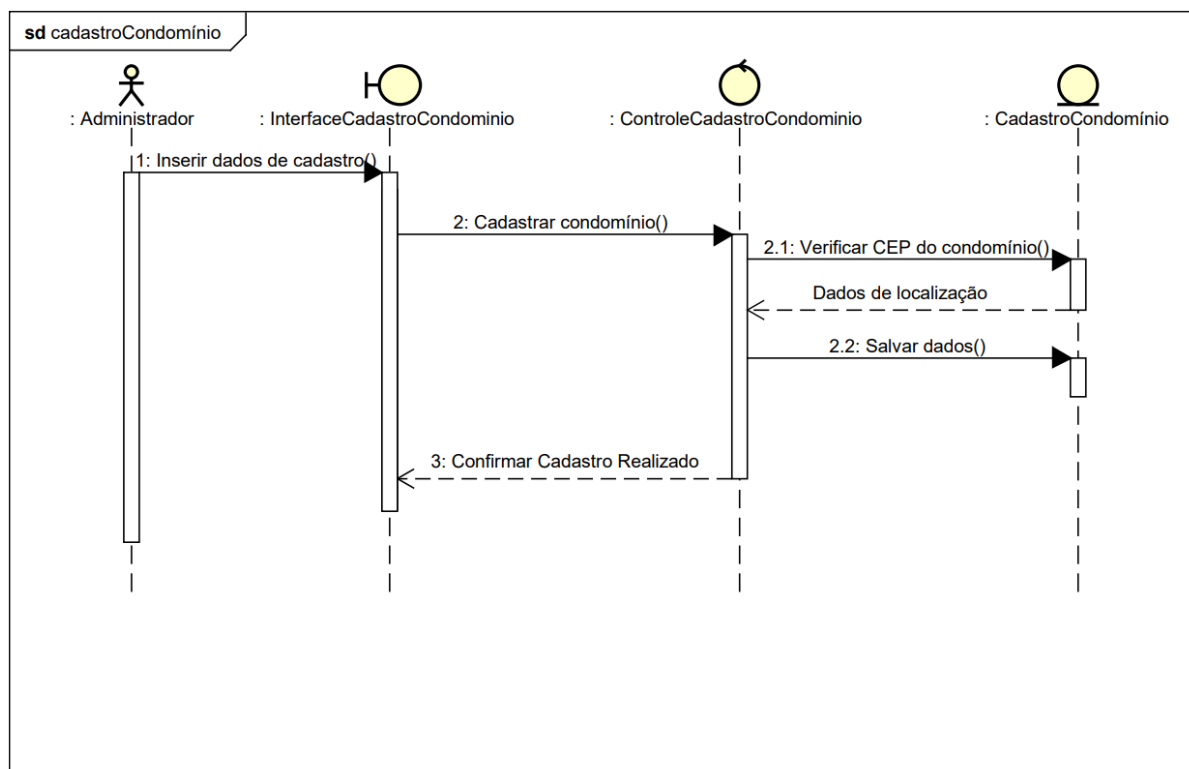
Com os diagramas a seguir é possível analisar a organização dos componentes em ordem temporal e como eles interagem entre si durante o decorrer de determinado processo.

Figura 21 - Diagrama de sequência do login do administrador e do porteiro



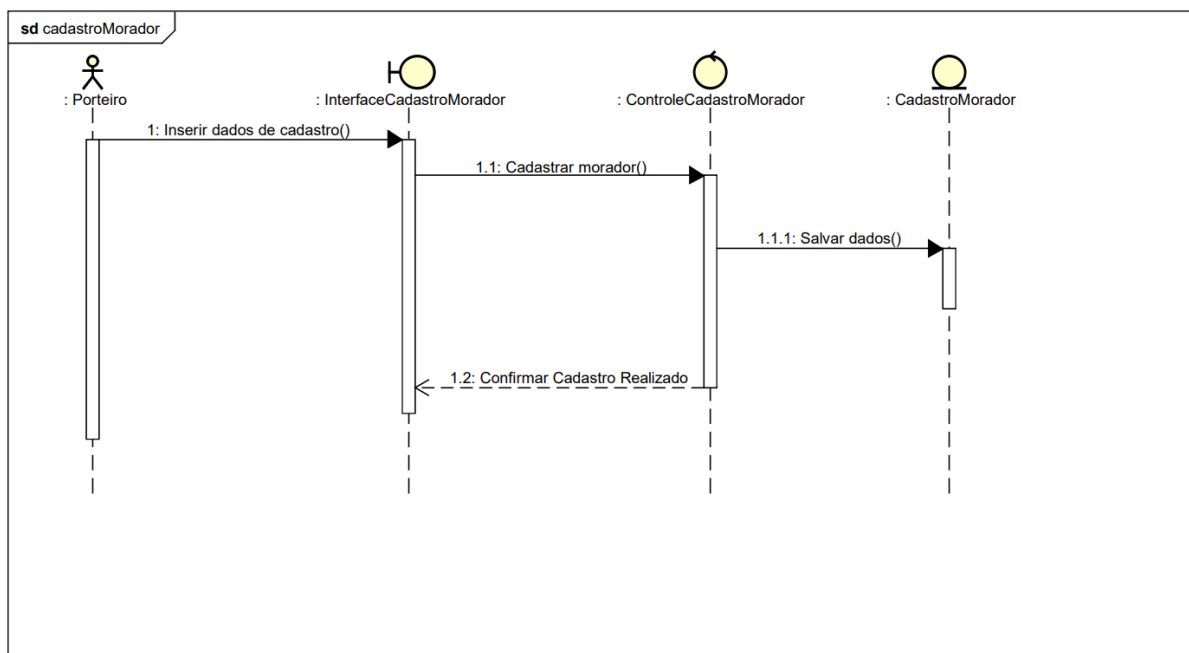
Fonte: Do próprio autor, 2023.

Figura 22 - Diagrama de sequência do cadastro de condomínio



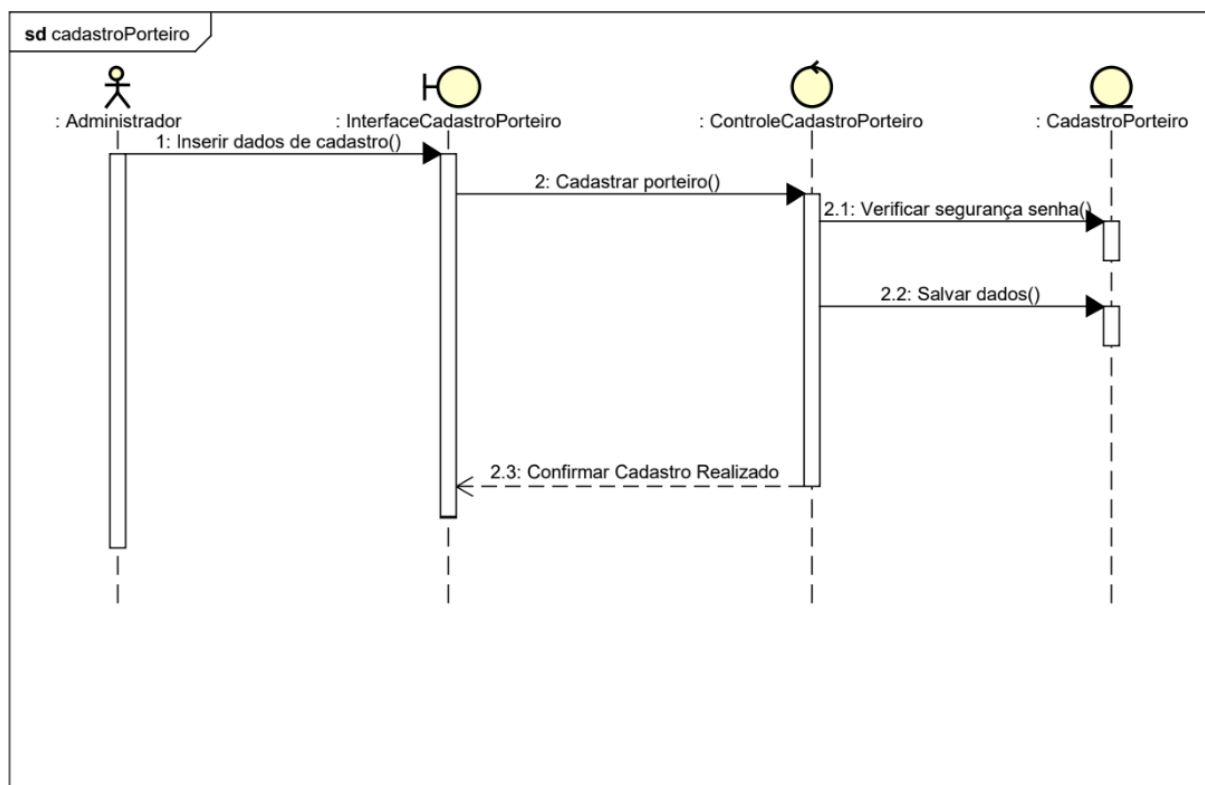
Fonte: Do próprio autor, 2023.

Figura 23 - Diagrama de sequência do cadastro de morador



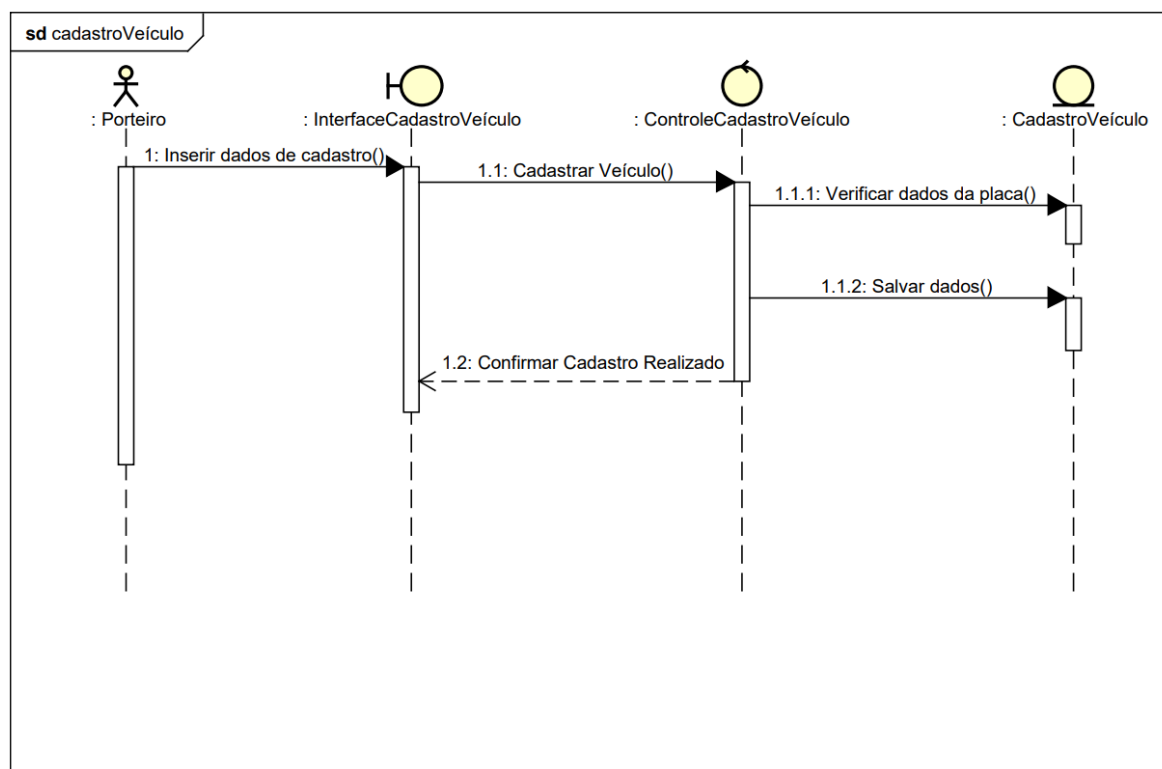
Fonte: Do próprio autor, 2023.

Figura 24 - Diagrama de sequência do cadastro de porteiro



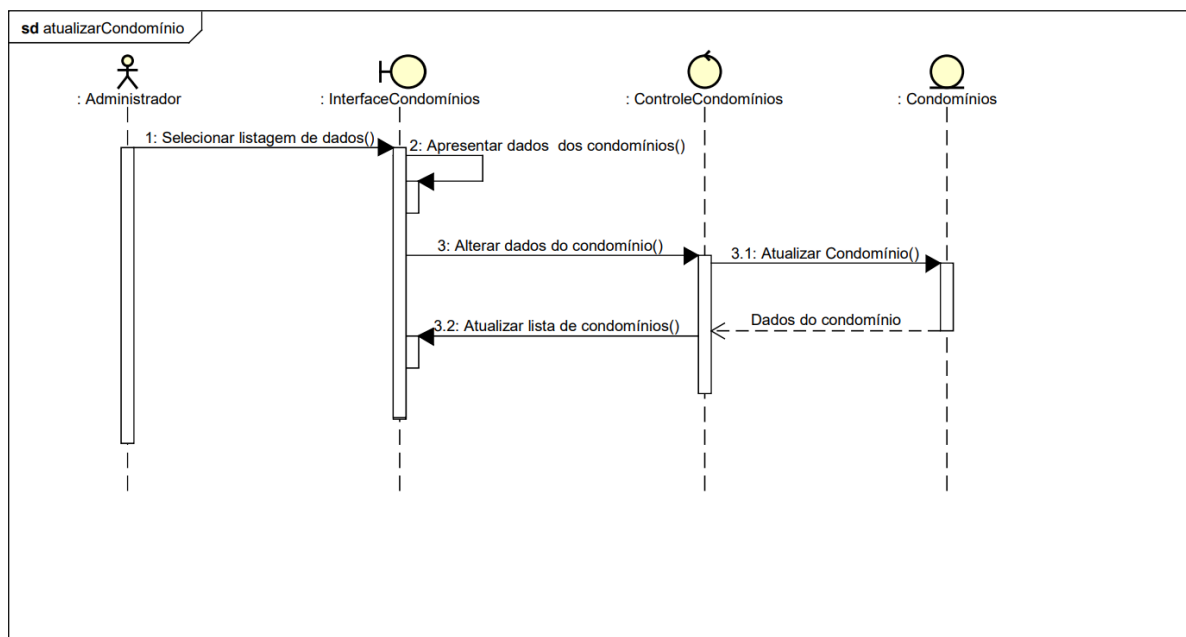
Fonte: Do próprio autor, 2023.

Figura 25 - Diagrama de sequência do cadastro de veículo



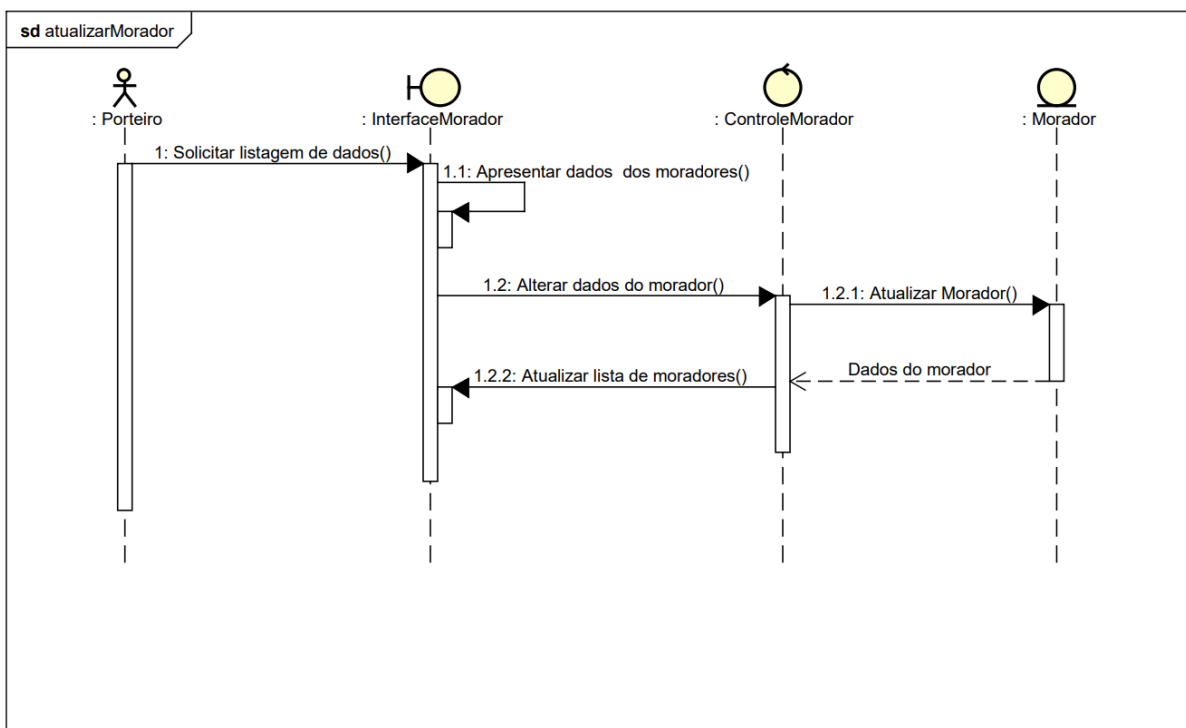
Fonte: Do próprio autor, 2023.

Figura 26 - Diagrama de sequência da atualização de condomínio



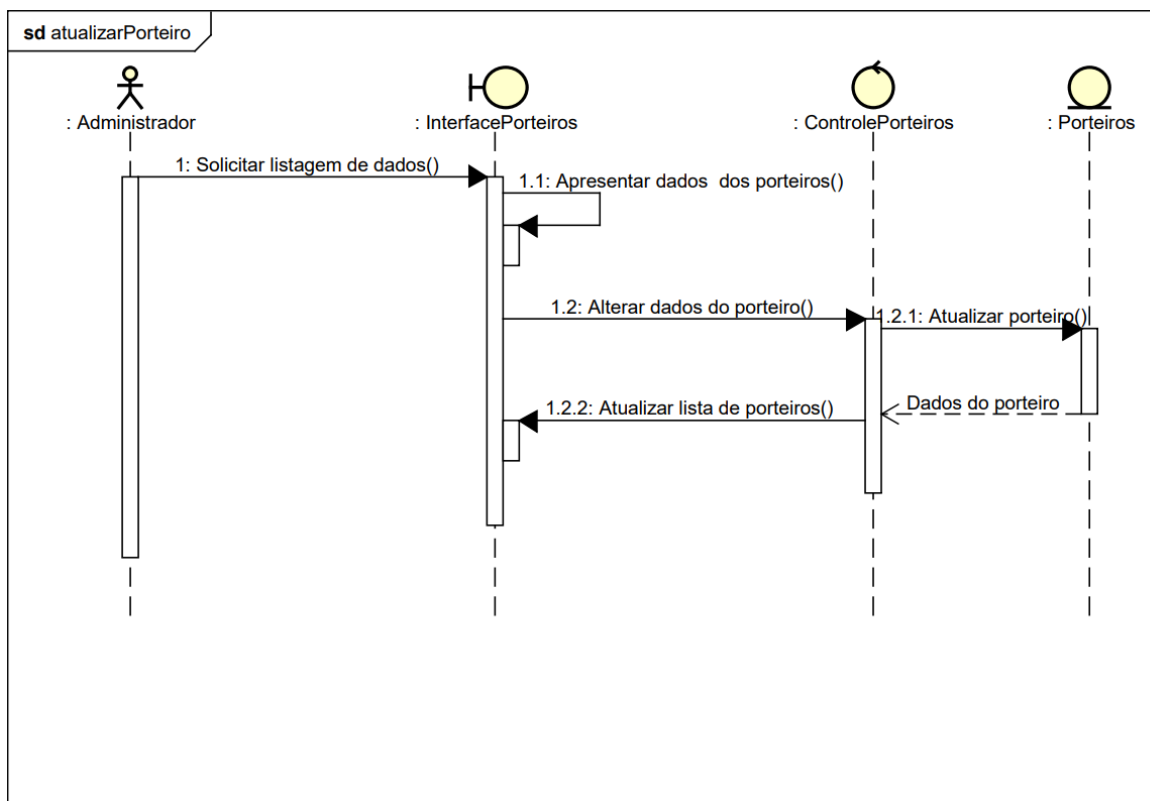
Fonte: Do próprio autor, 2023.

Figura 27 - Diagrama de sequência da atualização de morador



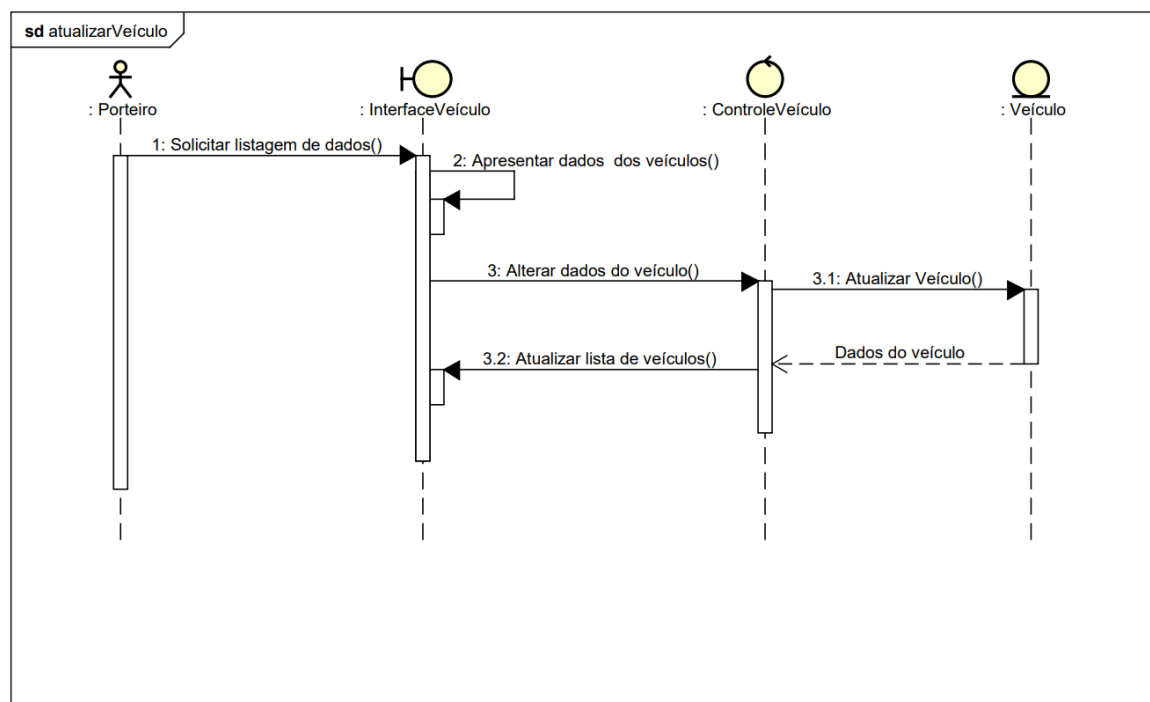
Fonte: Do próprio autor, 2023.

Figura 28 - Diagrama de sequência da atualização de porteiro



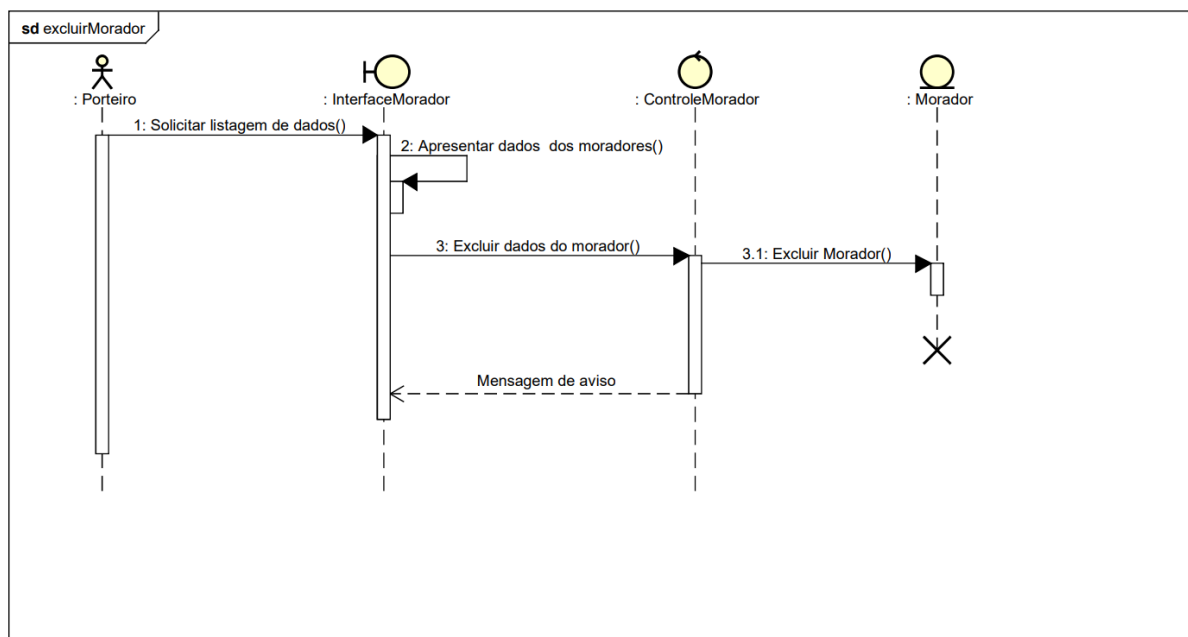
Fonte: Do próprio autor, 2023.

Figura 29 - Diagrama de sequência da atualização de veículo



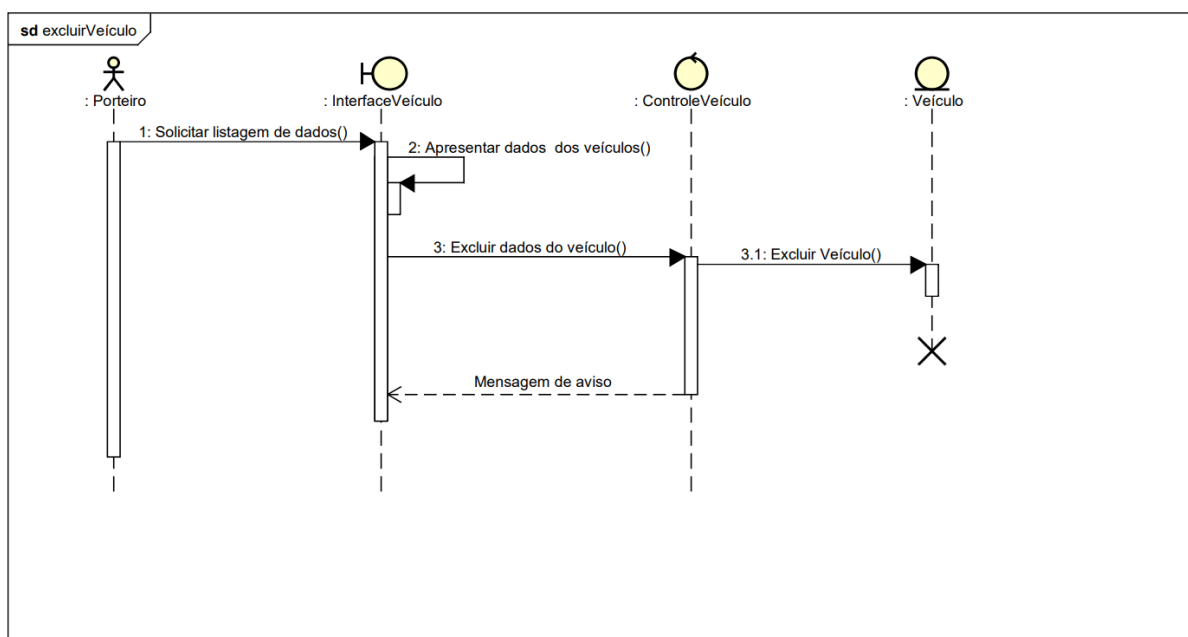
Fonte: Do próprio autor, 2023.

Figura 30 - Diagrama de sequência da exclusão do cadastro de morador



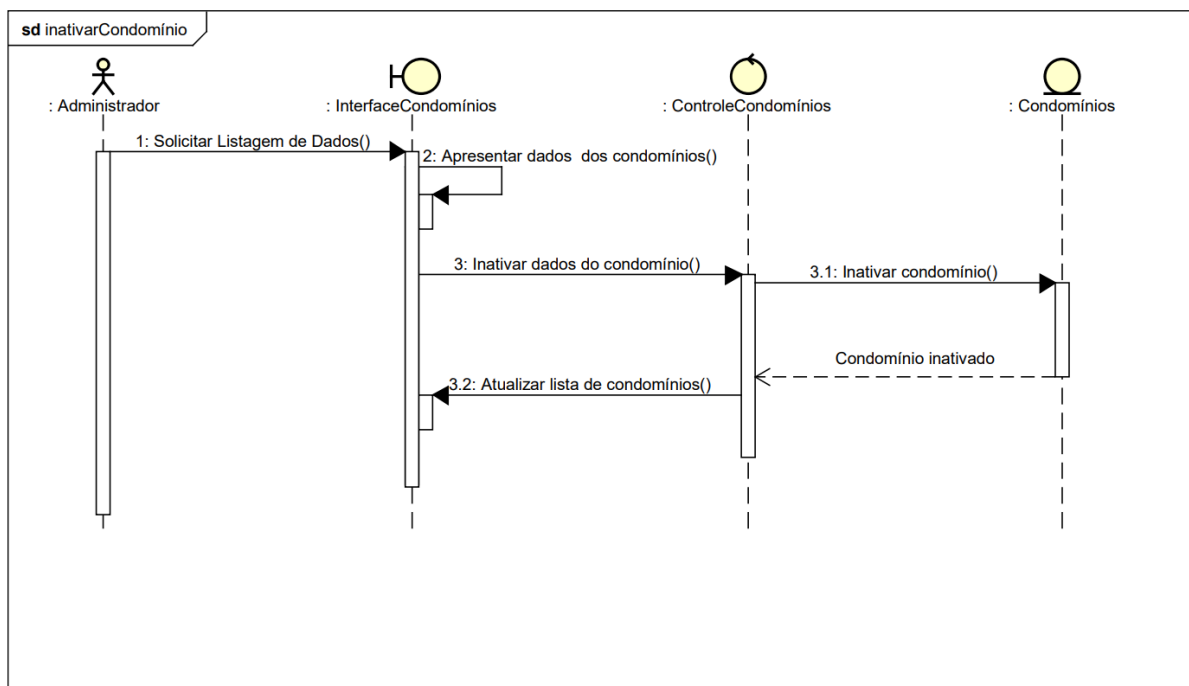
Fonte: Do próprio autor, 2023.

Figura 31 - Diagrama de sequência da exclusão do cadastro de veículo



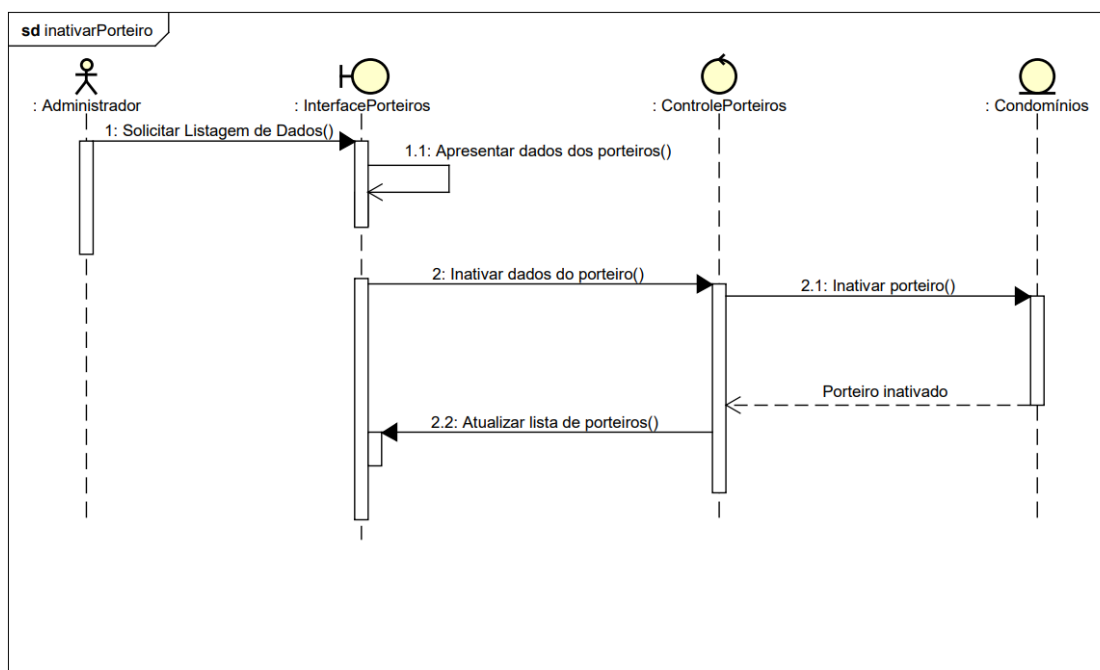
Fonte: Do próprio autor, 2023.

Figura 32 - Diagrama de sequência para a inativação de condomínio



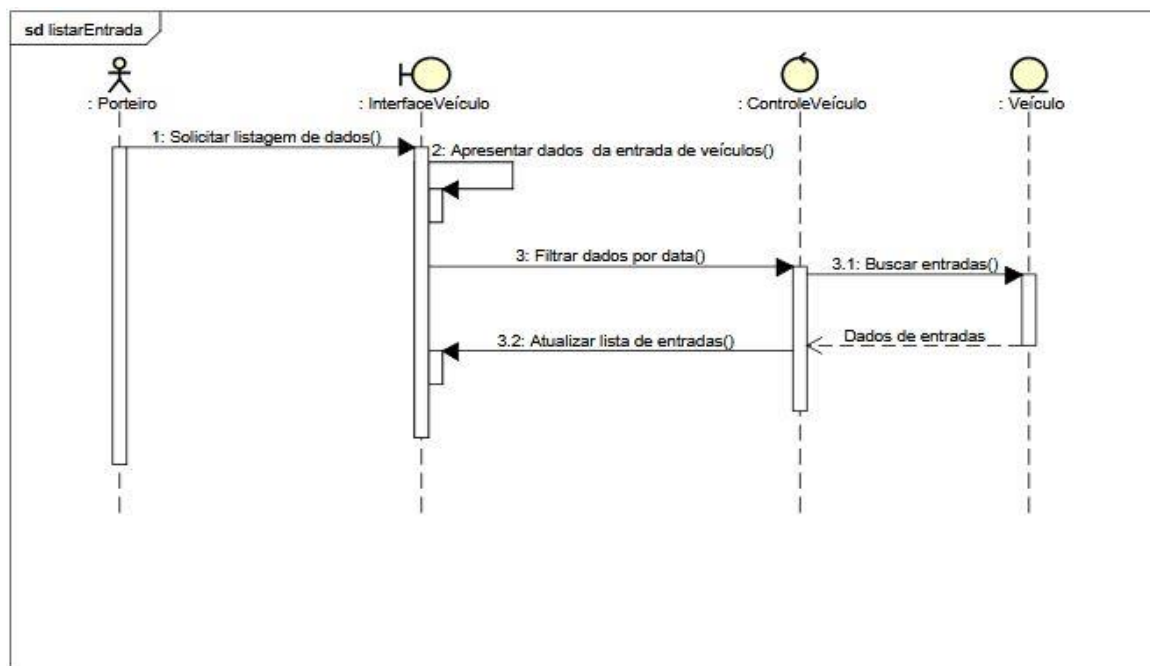
Fonte: Do próprio autor, 2023.

Figura 33 - Diagrama de sequência para a inativação de porteiro



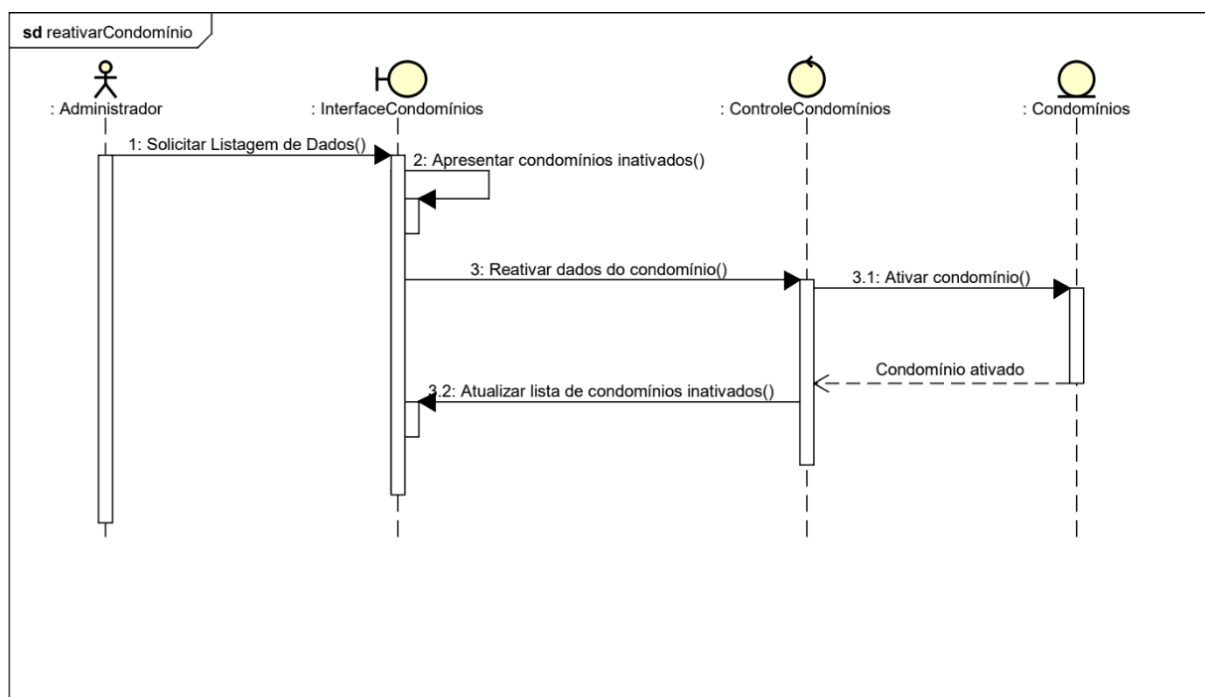
Fonte: Do próprio autor, 2023.

Figura 34 - Diagrama de sequência para a listagem de entrada



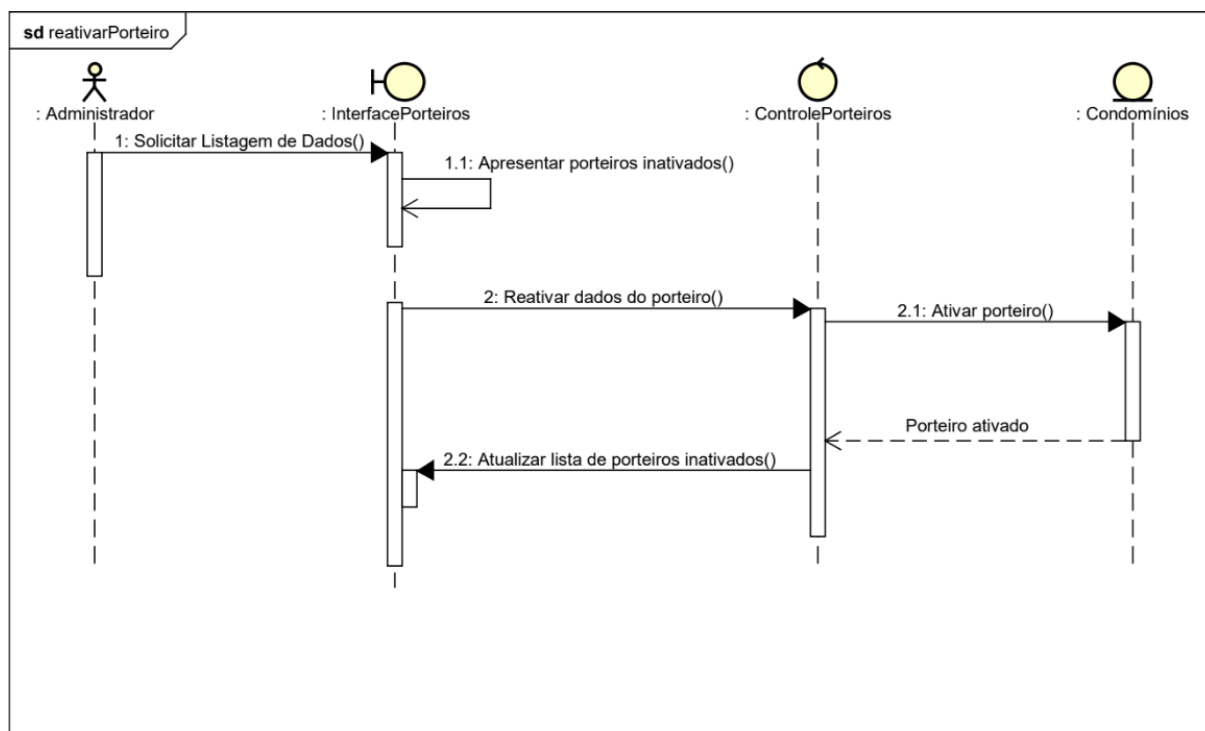
Fonte: Do próprio autor, 2023.

Figura 35 - Diagrama de sequência para ativação de condomínios



Fonte: Do próprio autor, 2023.

Figura 36 - Diagrama de sequência para ativação de porteiros



Fonte: Do próprio autor, 2023.

3.7 Representação Gráfica das Interfaces

Durante o processo de desenvolvimento visual da aplicação, foram utilizadas algumas ferramentas que auxiliaram em sua construção e que facilitaram no processo de organização e disposição dos elementos presentes na interface gráfica. A seguir serão abordadas as etapas desse processo.

3.7.1 Wireframe de Baixa Fidelidade

Os wireframes de baixa fidelidade são esquemas de página bastante simples, onde são utilizadas cores neutras, mas possuindo variações entre suas tonalidades.

Nos wireframes de baixa fidelidade desse projeto, foi utilizada a variação monocromática das cores cinzas, os layouts foram criados para ser de fácil clareza, prezando a eficiência ao uso do mesmo.

Com os wireframes a seguir é possível analisar as características visuais citadas anteriormente, assim como alguns outros elementos no layout em si:

Figura 37 - Wireframe de baixa fidelidade da interface de “Login”



Fonte: Do próprio autor, 2023.

A figura acima representa a interface de login de usuário em seu escopo, sendo posteriormente utilizada para a criação de sua versão de alta fidelidade.

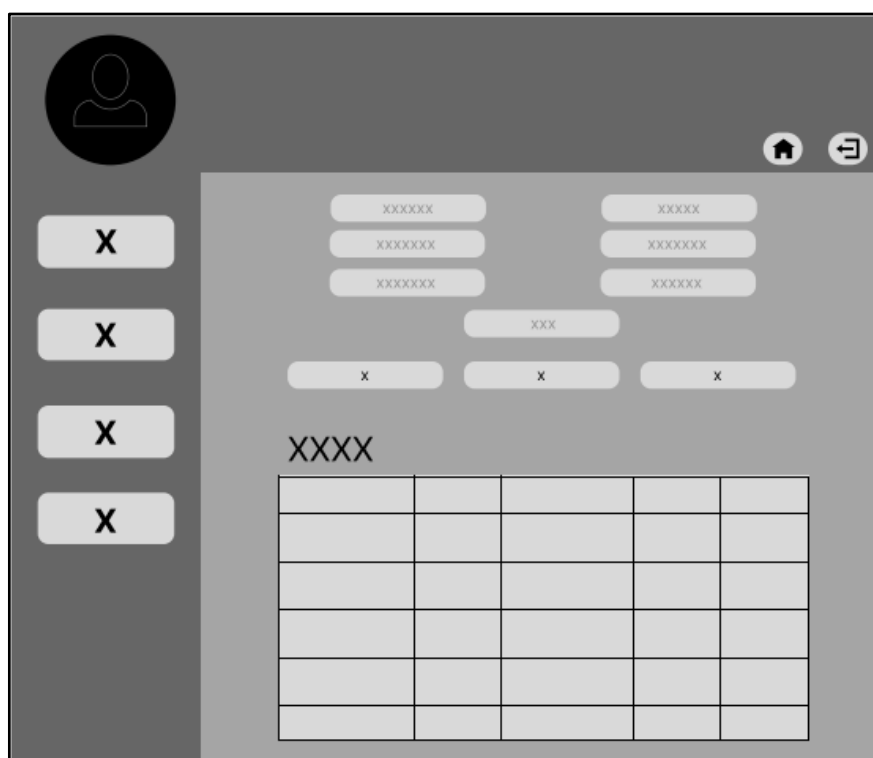
Figura 38 - Wireframe de baixa fidelidade da interface “Menu Administrador”



Fonte: Do próprio autor, 2023.

A figura acima representa o menu do administrador em seu escopo, sendo posteriormente utilizada para a criação de sua versão de alta fidelidade.

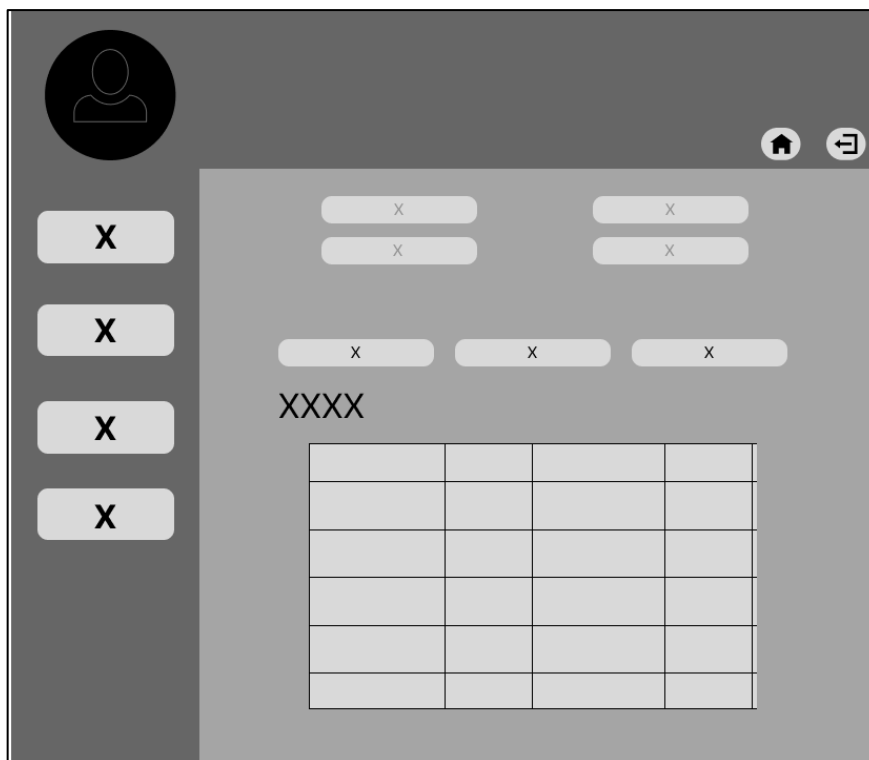
Figura 39 - Wireframe de baixa fidelidade da interface “Gerenciamento de Condomínios”



Fonte: Do próprio autor, 2023.

A figura acima representa a interface do gerenciamento de condomínios em seu escopo, sendo posteriormente utilizada para a criação de sua versão de alta fidelidade.

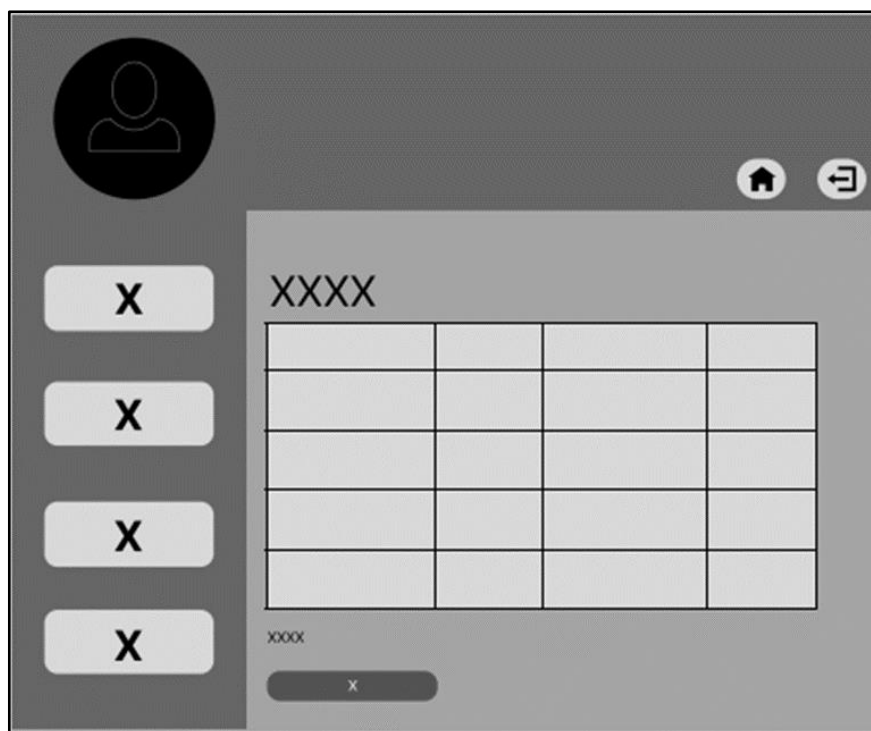
Figura 40 - Wireframe de baixa fidelidade da interface “Gerenciamento de Porteiros”



Fonte: Do próprio autor, 2023.

A figura acima representa a interface do gerenciamento de porteiros em seu escopo, sendo posteriormente utilizada para a criação de sua versão de alta fidelidade.

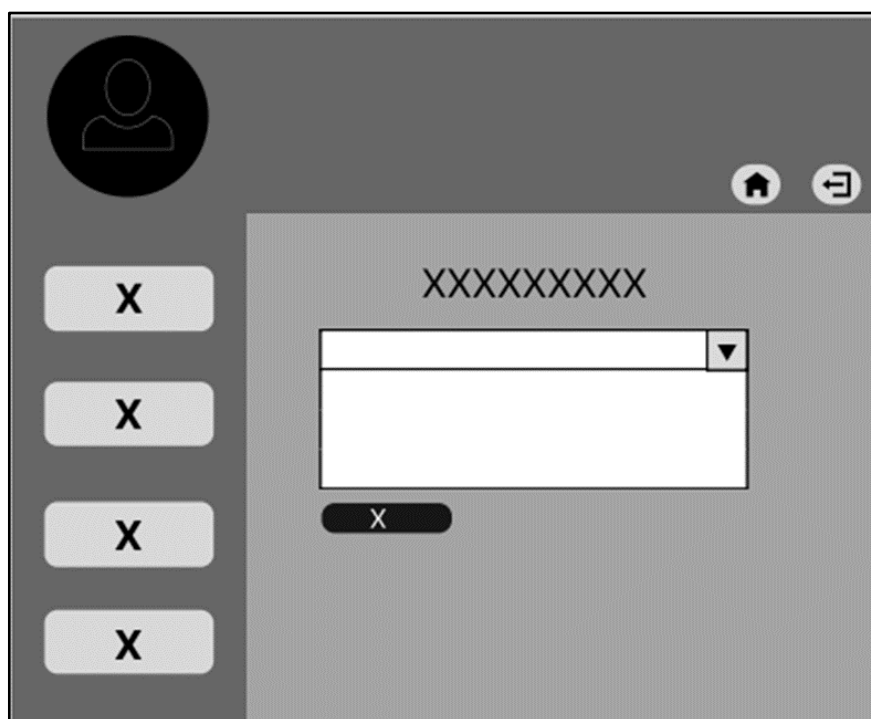
Figura 41 - Wireframe de baixa fidelidade da interface “Condomínios Inativados”



Fonte: Do próprio autor, 2023.

A figura acima representa a interface dos condomínios inativados em seu escopo, sendo posteriormente utilizada para a criação de sua versão de alta fidelidade.

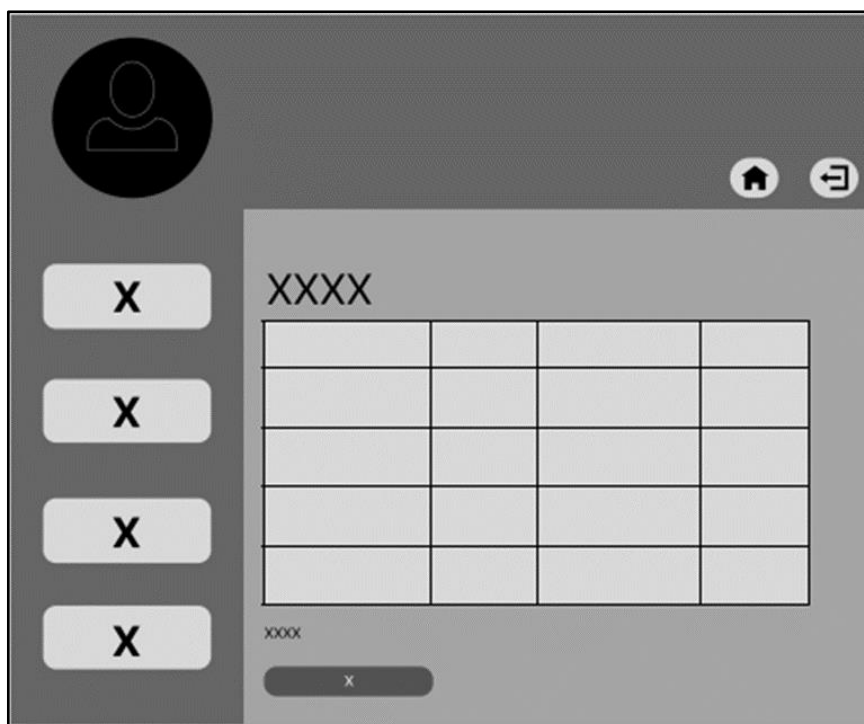
Figura 42 - Wireframe de baixa fidelidade da interface "Pré-Gerenciamento de Dados do Porteiro"



Fonte: Do próprio autor, 2023.

A figura acima representa a interface do pré-gerenciamento de dados dos porteiros inativados em seu escopo, sendo posteriormente utilizada para a criação de sua versão de alta fidelidade.

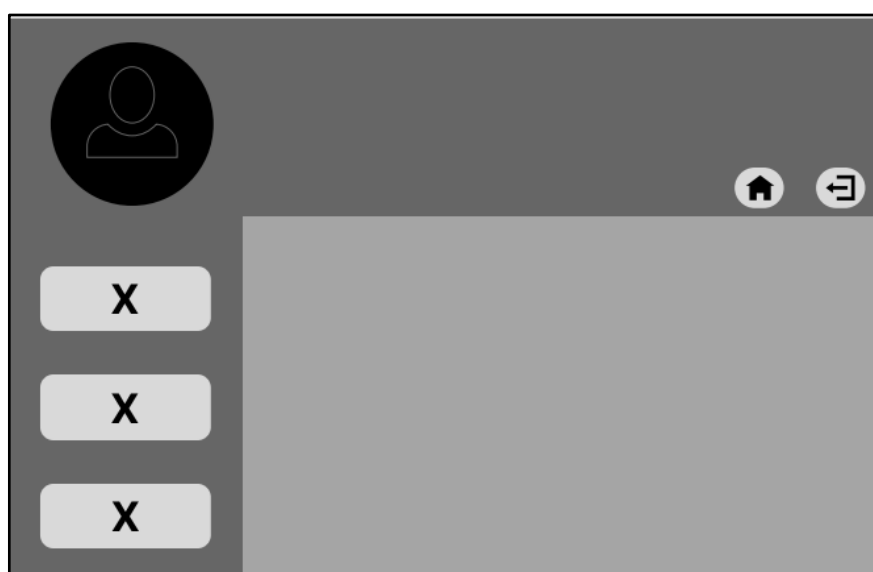
Figura 43 - Wireframe de baixa fidelidade da interface "Porteiros Inativados"



Fonte: Do próprio autor, 2023.

A figura acima representa a interface dos porteiros inativados em seu escopo, sendo posteriormente utilizada para a criação de sua versão de alta fidelidade.

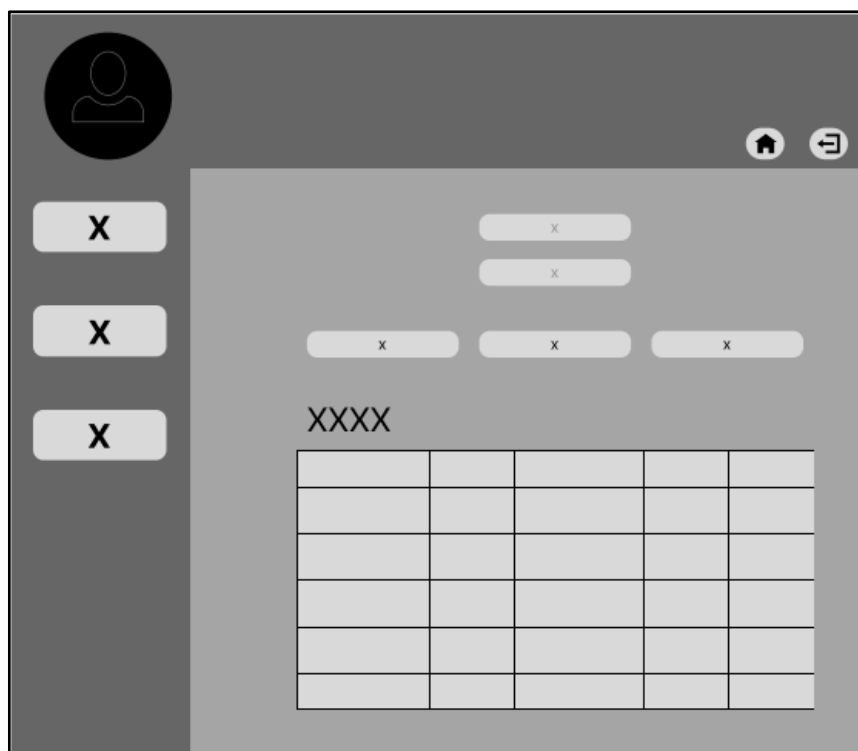
Figura 44 - Wireframe de baixa fidelidade da interface "Menu Porteiro"



Fonte: Do próprio autor, 2023.

A figura acima representa o menu do porteiro em seu escopo, sendo posteriormente utilizada para a criação de sua versão de alta fidelidade.

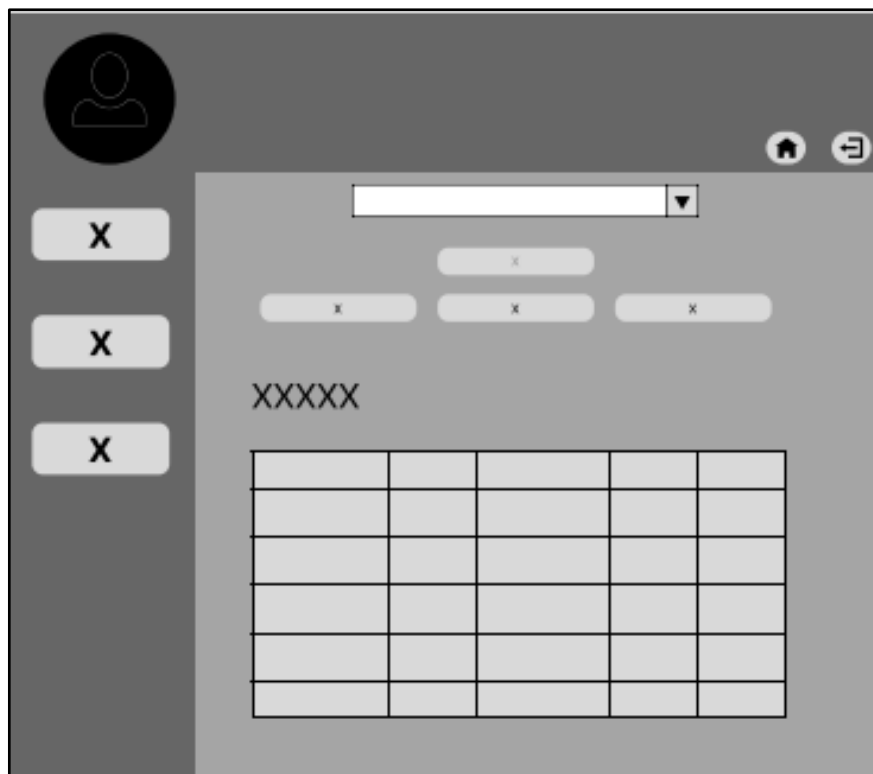
Figura 45 - Wireframe de baixa fidelidade da interface “Gerenciamento de Moradores”



Fonte: Do próprio autor, 2023.

A figura acima representa a interface do gerenciamento de moradores em seu escopo, sendo posteriormente utilizada para a criação de sua versão de alta fidelidade.

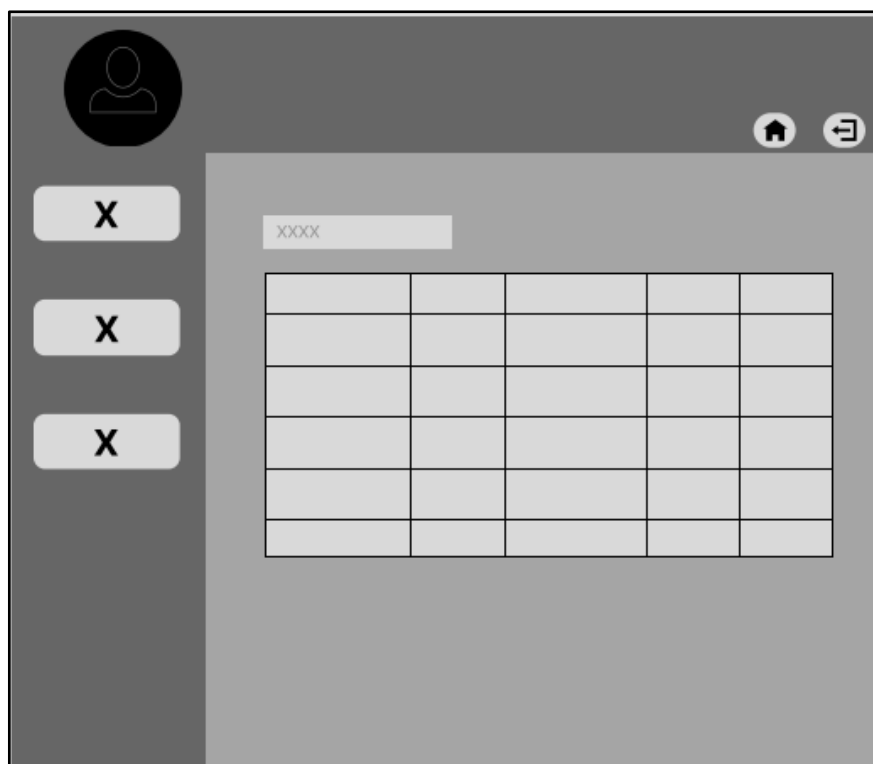
Figura 46 - Wireframe de baixa fidelidade da interface "Gerenciamento de Veículos"



Fonte: Do próprio autor, 2023.

A figura acima representa a interface do gerenciamento de veículos em seu escopo, sendo posteriormente utilizada para a criação de sua versão de alta fidelidade.

Figura 47 - Wireframe de baixa fidelidade da interface "Listar Entradas"



Fonte: Do próprio autor, 2023.

A figura acima representa a interface da listagem dos registros de entrada no condomínio em seu escopo, sendo posteriormente utilizada para a criação de sua versão de alta fidelidade.

3.7.2 Wireframe de Alta fidelidade

Após o desenvolvimento dos wireframes de baixa fidelidade, foram produzidos os wireframes de alta fidelidade. Eles incluem marcadores visuais, cores, estilo de fonte, além dos elementos da interface do usuário parecerem mais realistas, fazendo então a sua ideia ser mais visualmente entendível.

Com os wireframes a seguir é possível analisar a presença de alguns novos elementos visuais, assim como a alteração do esquema de cores, possuindo então uma proximidade maior das interfaces gráficas que serão desenvolvidas posteriormente:

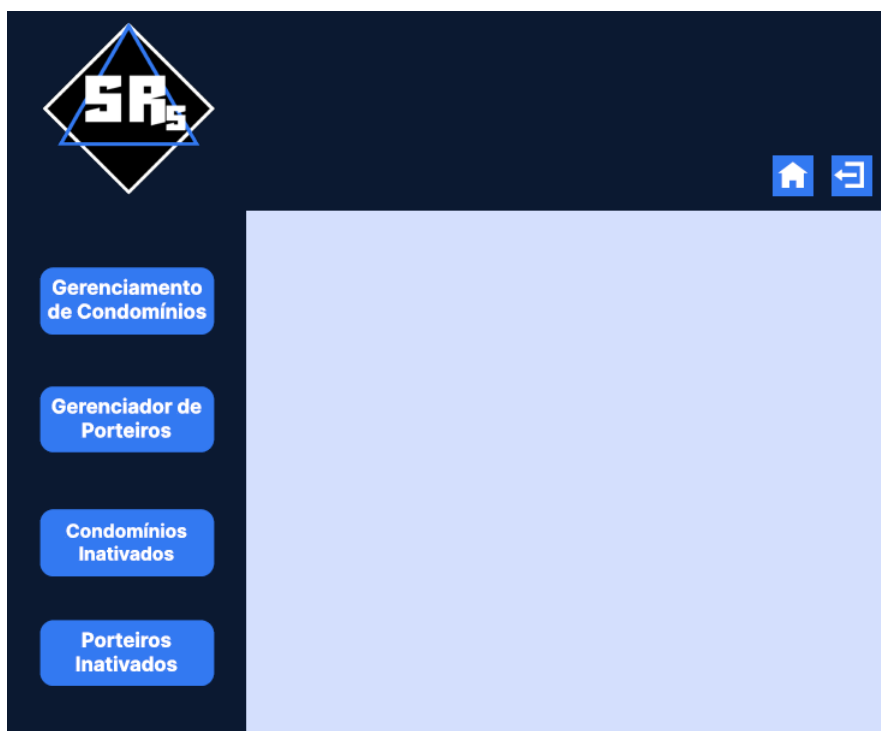
Figura 48 - Wireframe de alta fidelidade da interface "Login"



Fonte: Do próprio autor, 2023.

A figura acima representa a interface de login de usuário em sua versão de alta fidelidade, sendo posteriormente utilizada como meio para criação da interface gráfica.

Figura 49 - Wireframe de alta fidelidade da interface "Menu Administrador"



Fonte: Do próprio autor, 2023.

A figura acima representa o menu do administrador em sua versão de alta fidelidade, sendo posteriormente utilizada como meio para criação da interface gráfica.

Figura 50 - Wireframe de alta fidelidade da interface "Gerenciamento de Condomínios"

O wireframe apresenta uma interface de usuário para o gerenciamento de condomínios. No topo, há um cabeçalho escuro com o logo 'SRs' à esquerda e ícones de home e menu à direita. Um menu lateral à esquerda contém cinco itens: 'Gerenciamento de Condomínios', 'Gerenciamento de Porteiros', 'Condomínios Inativados' e 'Porteiros Inativados'. A área principal, com fundo claro, contém um formulário de criação/edição de condomínio. Este formulário possui campos de entrada para: Nome, CNPJ, CEP, Endereço, Bairro, Cidade, UF e Estado. Abaixo dos campos, há três botões: 'Criar', 'Alterar' e 'Inativar'. Segue-se o título 'CONDOMÍNIOS ATIVOS' e uma tabela com uma única linha visível, representando a lista de condomínios ativos.

Fonte: Do próprio autor, 2023.

A figura acima representa o gerenciamento de condomínios em sua versão de alta fidelidade, sendo posteriormente utilizada como meio para criação da interface gráfica.

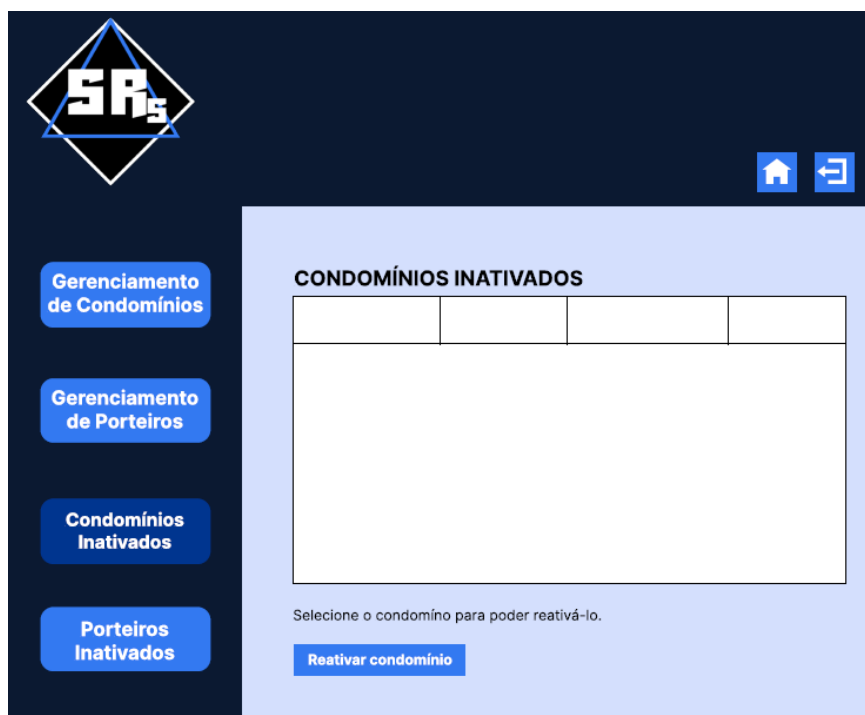
Figura 51 - Wireframe de alta fidelidade da interface "Gerenciamento de Porteiros"

Wireframe de alta fidelidade da interface "Gerenciamento de Porteiros". A interface possui um cabeçalho escuro com o logo "SR" no canto superior esquerdo e ícones de home e menu no canto superior direito. O menu lateral, também em escuro, contém quatro itens: "Gerenciamento de condomínios", "Gerenciador de porteiros", "Condomínios inativados" e "Porteiros Inativados". O conteúdo principal, em um fundo claro, apresenta quatro campos de entrada para login: "Digite o Nome..", "Digite o Número..", "Digite o Senha.." e "Digite o Turno..". Abaixo desses campos, há três botões azuis: "Criar", "Alterar" e "Inativar". O título "PORTEIROS ATIVOS" precede uma tabela com uma única linha visível, composta por quatro células vazias.

Fonte: Do próprio autor, 2023.

A figura acima representa a interface do gerenciamento de porteiros em sua versão de alta fidelidade, sendo posteriormente utilizada como meio para criação da interface gráfica.

Figura 52 - Wireframe de alta fidelidade da interface "Condomínios Inativados"



Fonte: Do próprio autor, 2023.

A figura acima representa a interface dos condomínios inativados em sua versão de alta fidelidade, sendo posteriormente utilizada como meio para criação da interface gráfica.

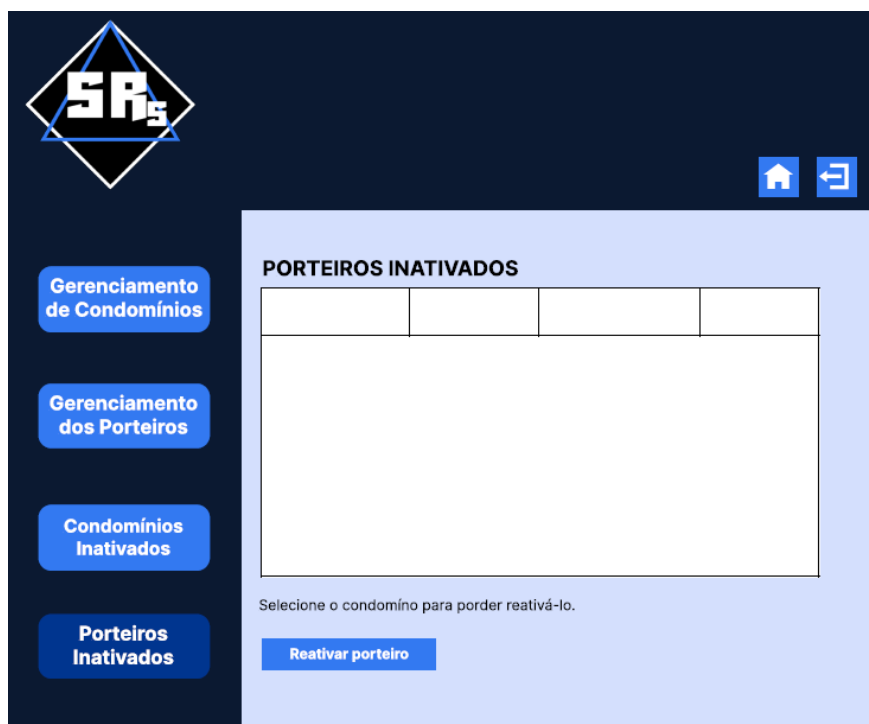
Figura 53 - Wireframe de alta fidelidade da interface "Pré-Gerenciamento de Dados do Porteiro"



Fonte: Do próprio autor, 2023.

A figura acima representa a interface do pré-gerenciamento de dados dos porteiros inativados em sua versão de alta fidelidade, sendo posteriormente utilizada como meio para criação da interface gráfica.

Figura 54 - Wireframe de alta fidelidade da interface "Porteiros Inativados"



Fonte: Do próprio autor, 2023.

A figura acima representa a interface dos porteiros inativados em sua versão de alta fidelidade, sendo posteriormente utilizada como meio para criação da interface gráfica.

Figura 55 - Wireframe de alta fidelidade da interface "Menu Porteiro"



Fonte: Do próprio autor, 2023.

A figura acima representa o menu do porteiro em sua versão de alta fidelidade, sendo posteriormente utilizada como meio para criação da interface gráfica.

Figura 56 - Wireframe de alta fidelidade da interface "Gerenciamento de Moradores"



Fonte: Do próprio autor, 2023.

A figura acima representa a interface do gerenciamento de moradores em sua versão de alta fidelidade, sendo posteriormente utilizada como meio para criação da interface gráfica.

Figura 57 - Wireframe de alta fidelidade da interface "Gerenciamento de Veículos"



Fonte: Do próprio autor, 2023.

A figura acima representa a interface do gerenciamento de veículos em sua versão de alta fidelidade, sendo posteriormente utilizada como meio para criação da interface gráfica.

Figura 58 - Wireframe de alta fidelidade da interface "Listar Entradas"



Fonte: Do próprio autor, 2023.

A figura acima representa a interface da listagem dos registros de entrada no condomínio em sua versão de alta fidelidade, sendo posteriormente utilizada como meio para criação da interface gráfica.

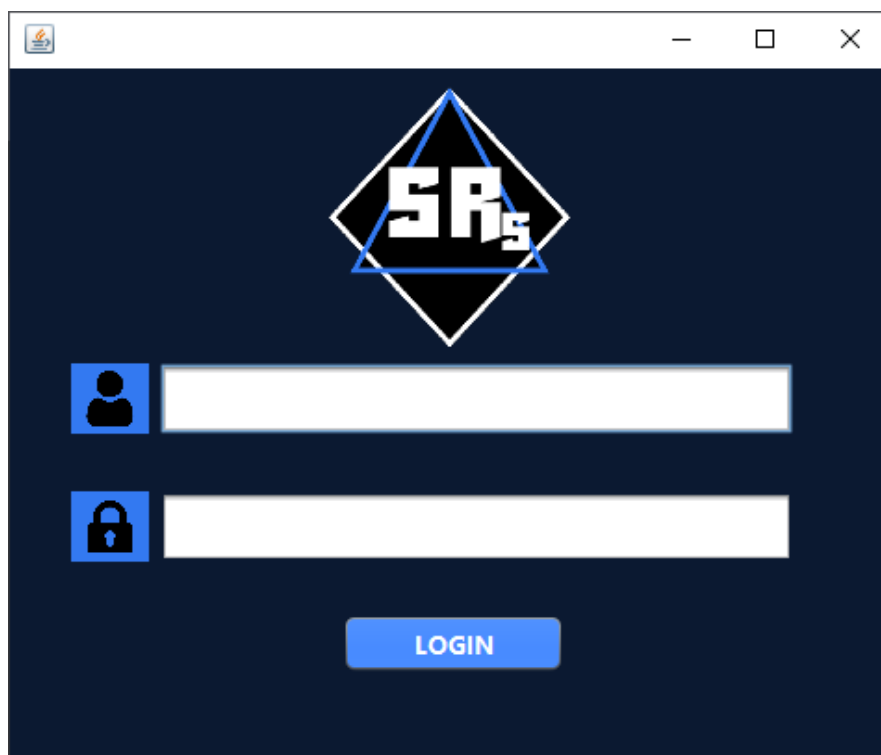
3.7.3 Aplicação no Java

Após o desenvolvimento dos wireframes, foram desenvolvidas as prototipações das interfaces gráficas. Nessa etapa, foi priorizada a utilização de cores que estimulassem sensações relacionáveis aos objetivos do projeto SR's.

A cor azul remete segurança, estabilidade e confiança, mas também pode estar relacionada a inovação e tecnologia. Ou seja, encaixando-se perfeitamente com os objetivos do SR's. Consequentemente, essa cor foi utilizada, onde foi feita uma variante entre análoga e monocromática das cores azuis, que estão presentes nas interfaces.

As figuras e descrições a seguir ilustram e descrevem o funcionamento de cada interface da aplicação:

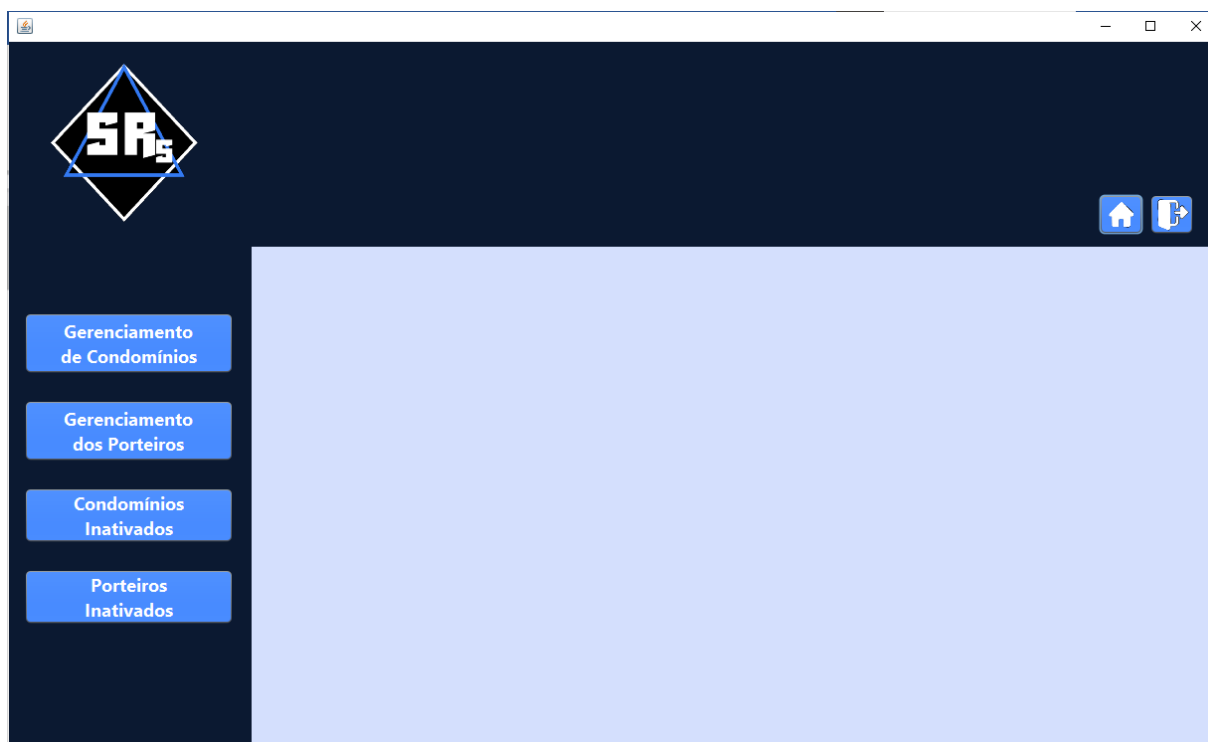
Figura 59 - Interface "Login"



Fonte: Do próprio autor, 2023.

A figura acima representa a interface de login de usuário, onde o administrador e o porteiro podem inserir seus dados de login (número de acesso e senha) para prosseguir dentro da aplicação.

Figura 60 - Interface "Menu Administrador"



Fonte: Do próprio autor, 2023.

A figura acima representa o menu do administrador, onde o mesmo terá acesso a todas as suas principais funções disponíveis dentro da aplicação. Sendo elas: o gerenciamento de condomínios, gerenciamento de porteiros, o acesso aos condomínios inativados, e o acesso aos porteiros inativados.

Figura 61 - Interface "Gerenciamento de Condomínios"

Gerenciamento de Condomínios

Gerenciamento dos Porteiros

Condomínios Inativados

Porteiros Inativados

Criar Alterar Inativar

CONDOMÍNIOS ATIVOS

Title 1	Title 2	Title 3	Title 4

Fonte: Do próprio autor, 2023.

A figura acima representa o gerenciamento dos condomínios ativos, onde o administrador poderá adicionar um novo registro ao preencher os campos solicitados. Além de conseguir visualizar, alterar ou inativar os registros dos condomínios já cadastrados.

Figura 62 - Interface "Gerenciamento de Porteiros"

Gerenciamento de Condomínios

Gerenciamento dos Porteiros

Condomínios Inativados

Porteiros Inativados

Alterar Inativar Criar

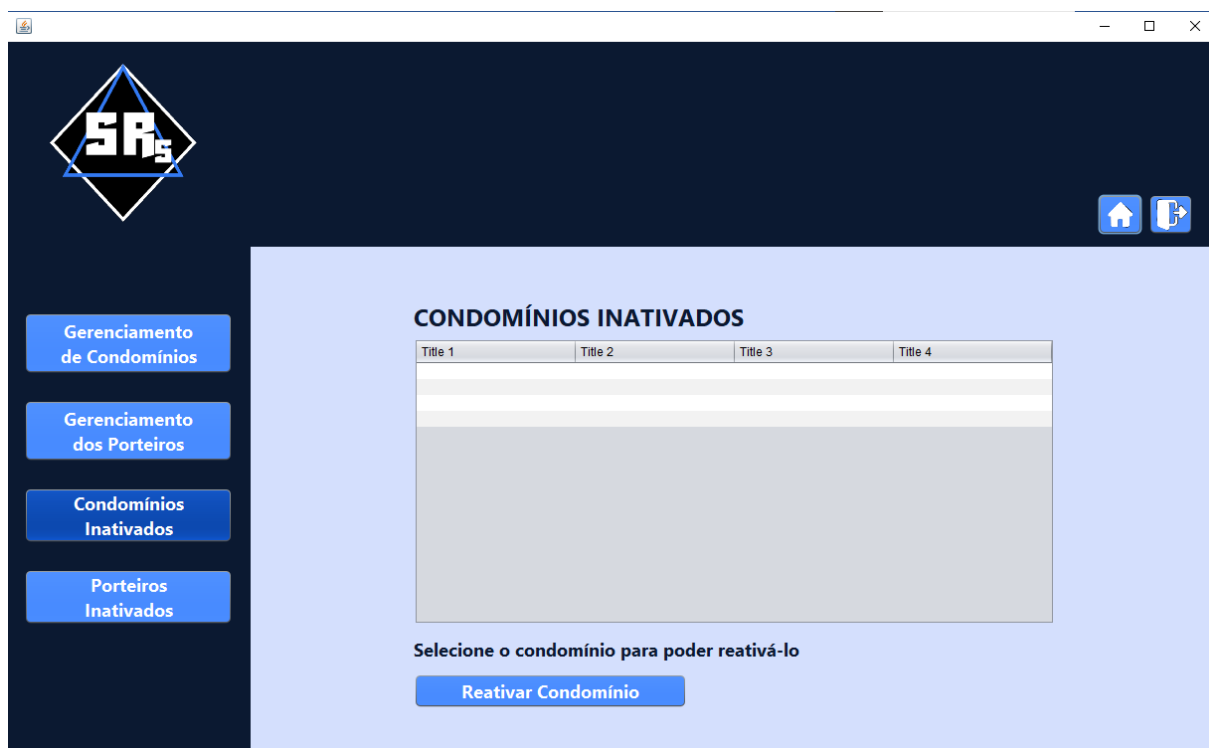
PORTEIROS ATIVOS

Title 1	Title 2	Title 3	Title 4

Fonte: Do próprio autor, 2023.

A figura acima representa o gerenciamento dos porteiros ativos, onde o administrador poderá adicionar um novo registro ao preencher os campos solicitados. Além de conseguir visualizar, alterar ou inativar os registros dos porteiros já cadastrados.

Figura 63 - Interface "Condomínios Inativados"



Fonte: Do próprio autor, 2023.

A figura acima representa a listagem de condomínios inativados, onde o administrador poderá reativar os condomínios que foram previamente inativados.

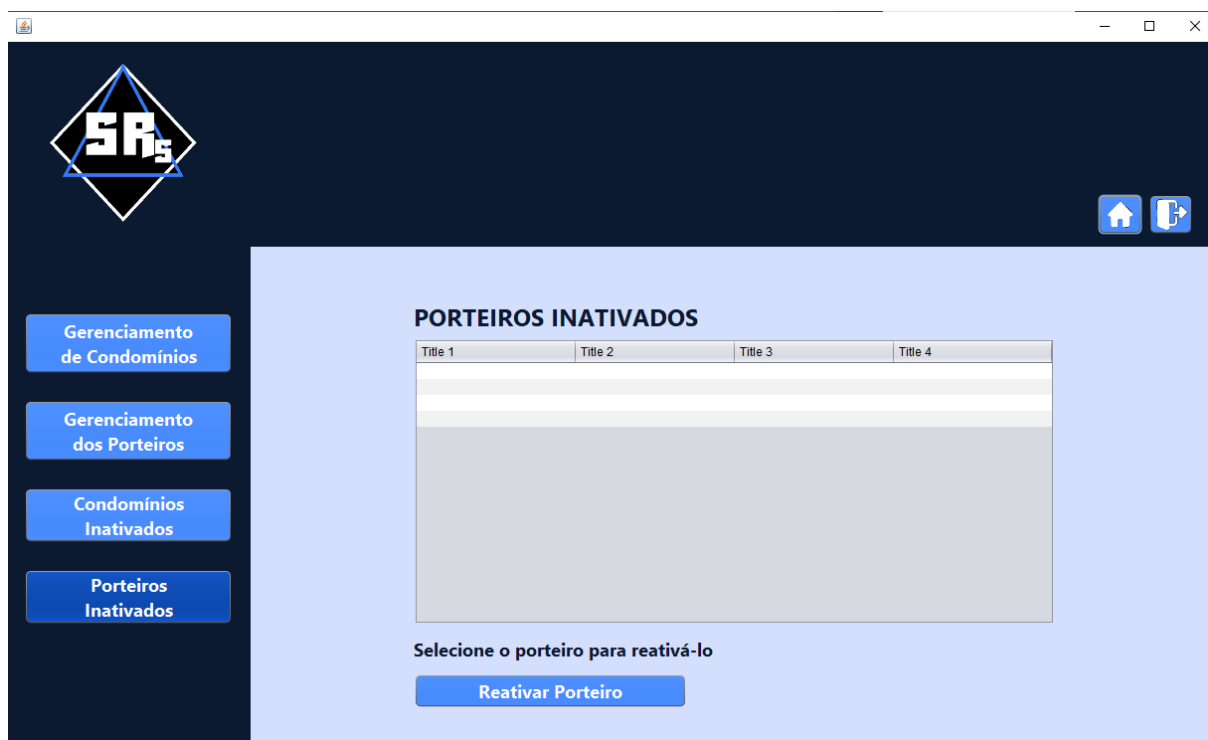
Figura 64 - Interface "Pré-Gerenciamento de Dados dos Porteiros"

The interface is a web application window with a dark blue header. On the left side of the header is a logo featuring the letters 'SR' inside a diamond shape. On the right side of the header are two small icons: a home icon and a user profile icon. Below the header, on the left, is a vertical sidebar with four blue buttons with white text: 'Gerenciamento de Condomínios', 'Gerenciamento dos Porteiros', 'Condomínios Inativados', and 'Porteiros Inativados'. The main content area has a light blue background. In the center of this area is a white rectangular box. At the top of this box is the title 'SELECIONE UM CONDOMÍNIO'. Below the title is a dropdown menu with the text 'Item 1' and a small downward arrow on the right. Below the dropdown menu is a blue button with the white text 'Avançar'.

Fonte: Do próprio autor, 2023.

A figura acima representa o pré-gerenciamento de dados dos porteiros inativados, onde administrador irá selecionar qual é condomínio dos porteiros que ele deseja acessar.

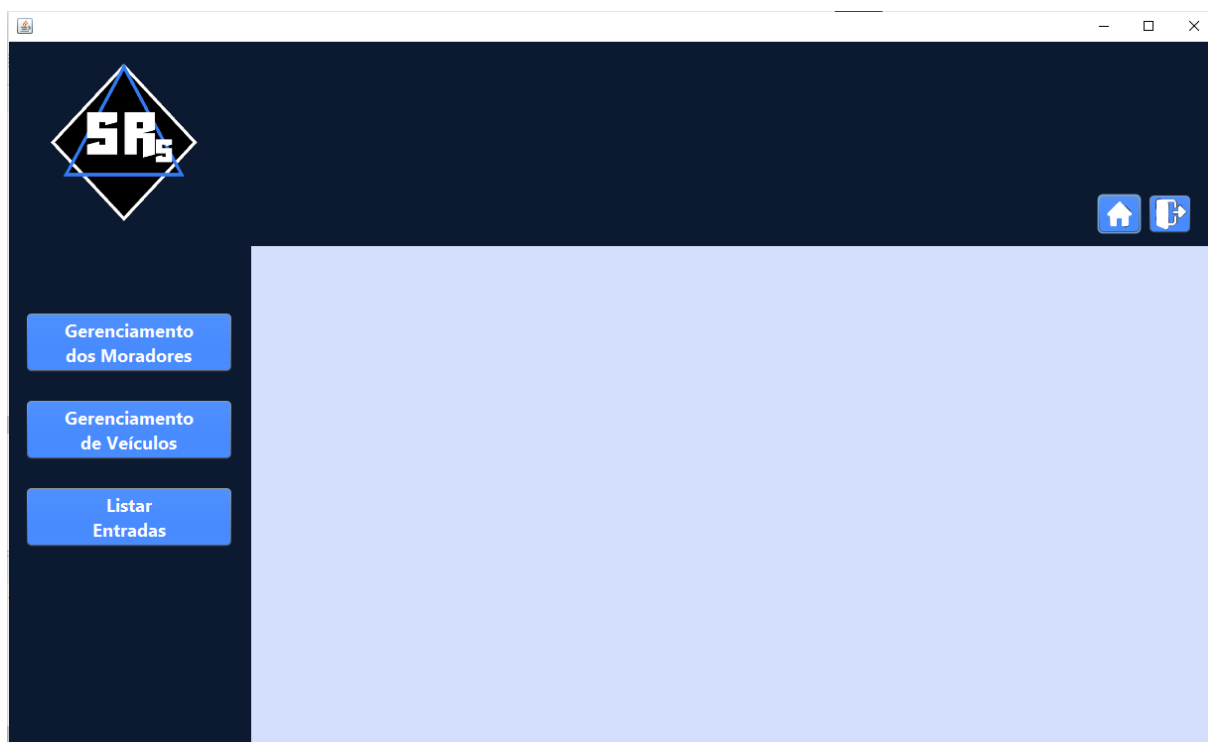
Figura 65 - Interface "Porteiros Inativados"



Fonte: Do próprio autor, 2023.

A figura acima representa a listagem de condomínios inativados, onde o administrador poderá reativar os porteiros que foram previamente inativados.

Figura 66 - Interface "Menu Porteiro"



Fonte: Do próprio autor, 2023.

A figura acima representa o menu do porteiro, onde o mesmo terá acesso a todas as suas principais funções disponíveis dentro da aplicação. Sendo elas: o gerenciamento de moradores, gerenciamento de veículos, e o acesso aos registros de entrada dos veículos no condomínio.

Figura 67 - Interface "Gerenciamento de Moradores"

Gerenciamento dos Moradores

Gerenciamento de Veículos

Listar Entradas

Excluir Alterar Criar

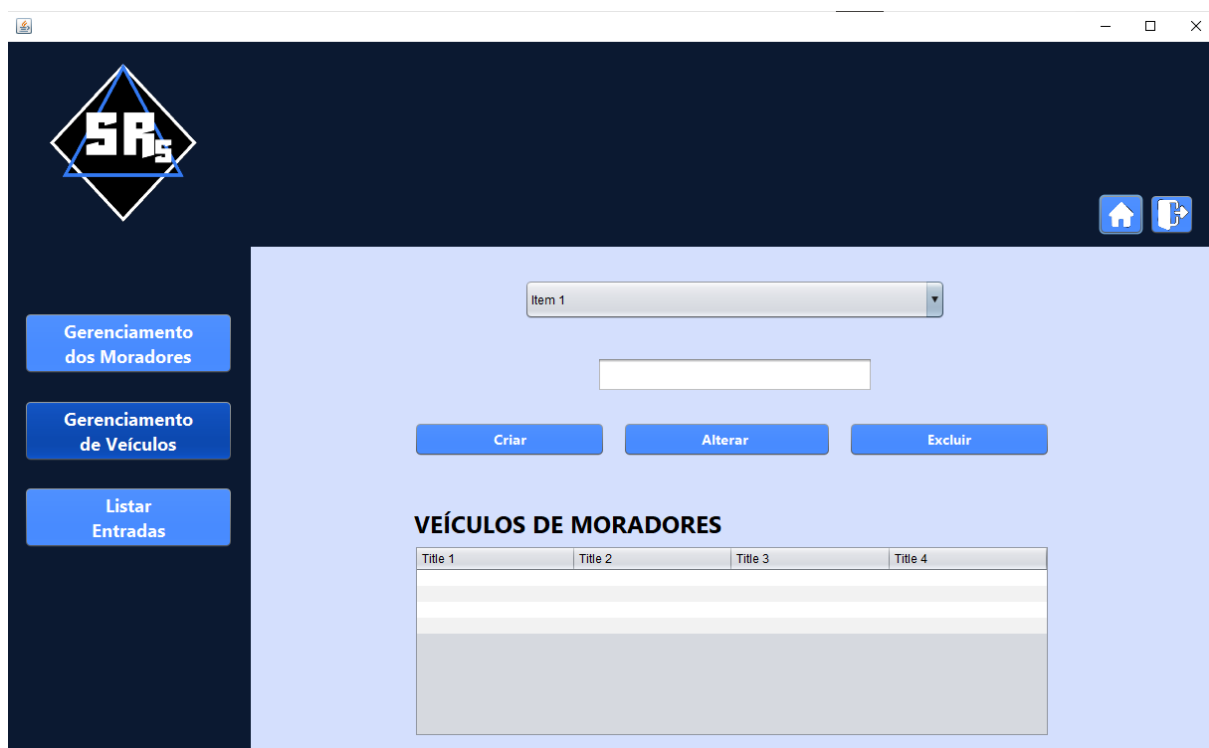
MORADORES DO CONDOMÍNIO

Title 1	Title 2	Title 3	Title 4

Fonte: Do próprio autor, 2023.

A figura acima representa o gerenciamento dos moradores, onde o porteiro poderá adicionar um novo registro ao preencher os campos solicitados. Além de conseguir visualizar, alterar ou excluir os registros dos moradores já cadastrados.

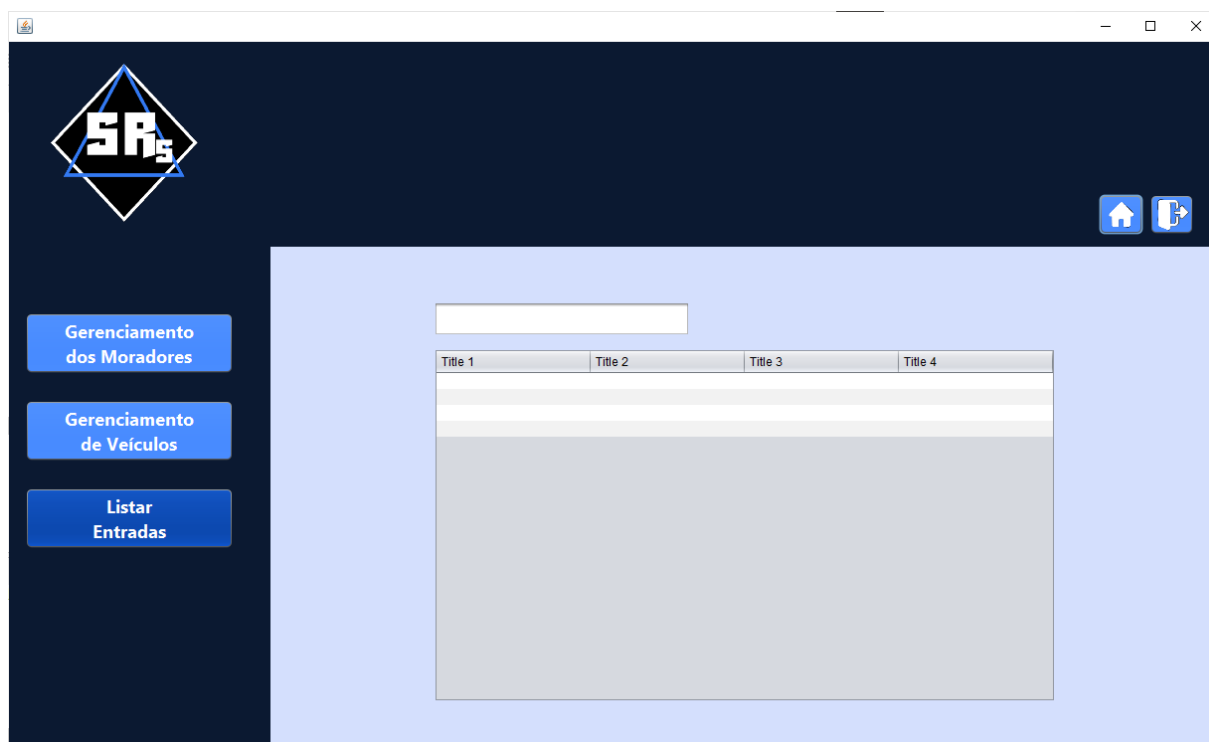
Figura 68 - Interface "Gerenciamento de Veículos"



Fonte: Do próprio autor, 2023.

A figura acima representa o gerenciamento dos moradores, onde o porteiro poderá adicionar um novo registro ao selecionar o morador que possui o veículo, e inserir a sua placa no campo seguinte. Além de conseguir visualizar, alterar ou excluir os registros dos moradores já cadastrados.

Figura 69 - Interface "Listar Entradas"



Fonte: Do próprio autor, 2023.

A figura acima representa a interface da listagem de entrada de veículos no condomínio, onde o porteiro poderá visualizar todos os registros de entrada cadastrados. Além de também conseguir filtrar esses registros por data.

4 CONSIDERAÇÕES FINAIS

Este projeto foi desenvolvido com a finalidade de automatizar os processos de controle de acesso em condomínios, utilizando de conceitos importantes e modernos, conforme mencionado em outros tópicos, para a construção da aplicação e do dispositivo. O resultado alcançado foi aplicável, funcional e satisfatório em diversos contextos, apesar de ter sido direcionado principalmente aos condomínios.

Embora seja reconhecido que os desafios relacionados à gestão de tráfego sejam complexos e possuem uma difícil solução, acreditamos que tenhamos apresentado uma proposta interessante para auxiliar nesse desafio. A validação da hipótese proposta, no entanto, depende das apresentações de argumentos favoráveis ou contrários à abordagem que foi adotada, e a taxa de aceitação da proposta é algo que está além do nosso controle.

O SR's apresenta um funcionamento teoricamente ideal, mas sua eficácia ainda não foi comprovada. Portanto, para determinar sua taxa de utilização e nível de utilidade, seria necessário implantá-lo em condomínios de uma determinada região, especialmente aqueles com alto fluxo de pessoas. Apesar dessa limitação, consideramos o resultado geral do trabalho como satisfatório.

REFERÊNCIAS

ANTONELLO, Ricardo (ed.). **Introdução a Visão Computacional com Python e OpenCV**. [S. l.: s. n.], 2017.

BARELLI, Felipe. **Introdução à Visão Computacional**: Uma abordagem prática com Python e OpenCV. 1. ed. São Paulo: Casa do Código, 2018. 256 p.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML**: Guia do Usuário. 2. ed. Rio de Janeiro: Elsevier, 2012.

BORGES, Luiz E. **Python para desenvolvedores**. 2. ed. Rio de Janeiro: Novatec, 2010. 360 p.

CARVALHO, Vinícius **MySQL**: Comece com o principal banco de dados open source do mercado. 5. ed. São Paulo: Casa do Código, 2015. 158 p.

DATE, C. J. **Introdução a Sistemas de Banco de Dados**. 8. ed. rev. [S. l.]: Gen LTC., 2004. 896 p.

EBERMAM, Eivelto et al. **Programação para leigos com Raspberry Pi**. Paraíba: Edifes, 2017. 285 p.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Pearson Education, 2011. 788 p.

FIGMA. **What is Figma?** [S.l.]. Figma, 2023. Disponível em: <https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma->. Acesso em: 14 ago. 2023.

FOWLER, Martin. **UML essencial**. 3. ed. São Paulo: Bookman, 2005. 160 p.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento Digital de Imagens**. 3. ed. São Paulo: Pearson Education, 2009. 624 p.

GUEDES, Gilleanes T. A. **UML 2 - Uma Abordagem Prática**. 3. ed. São Paulo: Novatec, 2018. 496 p.

HEUSER, Carlos A. **Projeto de Banco de Dados**. 6. ed. Rio Grande do Sul: Bookman, 2008. 282 p.

JAVA e Orientação a Objetos: Curso FJ-11. 6. ed. [S. l.]: Caelum, 2003. 295 p.

JUNIOR, Peter Jandl. **Java - Guia do Programador**: Atualizado Para Java 16. 4. ed. São Paulo: Novatec, 2021. 552 p. ISBN 6586057574.

MAGRANI, Eduardo. **A internet das coisas**. 1. ed. Rio de Janeiro: FGV, 2018. 192 p.

MARENGONI, Maurício. Visão Computacional e OpenCV. **Tutorial: Introdução à Visão Computacional usando OpenCV**, [s. l.], 2014.

MILANI, André. **MySQL - Guia do Programador**. 1. ed. São Paulo: Novatec, 2007. 400 p.

OLIVEIRA, Sérgio. **Internet das Coisas**: com ESP8266, Arduino e Raspberry Pi. 1. ed. São Paulo: Novatec, 2017. 240 p.

PYTHON. **Documentação Python 3.11.4**. [S. l.]. Python, 2023. Disponível em: <https://docs.python.org/pt-br/3/>. Acesso em: 09 jun. 2023.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. **Sistemas de Gerenciamento de Banco de Dados**. 3. ed. São Paulo: McGraw-Hill, 2011. 884 p.

RASPBERRY. **Raspberry Pi Documentation**. [S. l.]. Rasberry, 2023. Disponível em: <https://www.raspberrypi.com/documentation/>. Acesso em: 10 jun. 2023.

SAADE, Joel. **Python sem Mistérios**. 1. ed. São Paulo: Novatec, 2019. 432 p. ISBN 978-8575228166.

SCHILDT, Hebert. **Java para Iniciantes**. 6. ed. [S. l.]: Bookman, 2015. 684 p.

SENATRAN **Frota de Veículos - 2022**. [S. l.], 2022. Disponível em: <https://www.gov.br/transportes/pt-br/assuntos/transito/conteudo-Senatran/frota-de-veiculos-2022>. Acesso em: 19 jun. 2023.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de Banco de Dados**. 5. ed. [S. l.]: Elsevier, 2006. 781 p.

STACK OVERFLOW. **Developer Survey**. [S.l.]. Stack Overflow, 2022. Disponível em: <https://survey.stackoverflow.co/2022/>. Acesso em: 08 jun. 2023

TEIXEIRA, Fabricio. **Introdução e boas práticas em UX Design**. 1. ed. São Paulo: Casa do Código, 2014. 217 p.

UPTON, Eben. **Raspberry Pi Guia do Usuário**. 4. ed. Rio de Janeiro: Alta Books, 2017. 288 p.