

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIAS E TECNOLOGIA DE  
RONDÔNIA – CAMPUS JI-PARANÁ**

**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS**

**SISTEMA DE REGISTRO AUTOMATIZADO DA ENTRADA E SAÍDA DE  
VEÍCULOS COM USO DE INTELIGÊNCIA ARTIFICIAL**

**Wígnny Bora de Almeida**

Ji-Paraná – RO  
2023

Wigny Bora de Almeida

**SISTEMA DE REGISTRO AUTOMATIZADO DA ENTRADA E SAÍDA DE  
VEÍCULOS COM USO DE INTELIGÊNCIA ARTIFICIAL**

Trabalho de Conclusão de Curso apresentado ao curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, como parte dos requisitos necessários à obtenção do título de tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Me. Jackson Henrique da Silva Bezerra

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO,  
com dados informados pelo(a) próprio(a) autor(a).

Almeida, Wígnny Bora de.

Sistema de registro automatizado da entrada e saída de veículos com uso  
de inteligência artificial / Wígnny Bora de Almeida, Ji-Paraná-RO, 2023.  
70 f. : il.

Orientador(a): Prof. Me. Jackson Henrique da Silva Bezerra.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Análise e  
Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e  
Tecnologia de Rondônia - IFRO, Ji-Paraná-RO, 2023.

1. Aplicação. 2. Desenvolvimento. 3. Rede neural. 4. Web. 5. Servidor. I.  
Bezerra, Jackson Henrique da Silva (orient.). II. Instituto Federal de  
Educação, Ciência e Tecnologia de Rondônia - IFRO. III. Título.

**Bibliotecário(a) Responsável:** Cleuza Diogo Antunes, CRB-11/864 (Campus Ji-Paraná)

**BANCA AVALIADORA**

Me. João Eujácio Teixeira Junior

---

Membro da Comissão de TCC  
(Professor(a) Avaliador(a))

Me. Danilo Pereira Escudero

---

Membro da Comissão de TCC  
(Professor(a) Avaliador(a))

Me. Jackson Henrique da Silva Bezerra

---

Prof. Orientador

**Ji-Paraná - RO, 07 de Dezembro de 2023**

## RESUMO

O controle de acesso em estacionamentos é um processo comum e grandemente utilizado com o objetivo de garantir a segurança de pessoas e patrimônios. Este controle pode ser aplicado de diferentes formas, sendo algumas das mais comuns através de recurso humano (ex. vigilantes) ou implantação de barreiras físicas (ex. uma guarita). Embora estes tipos de controle de acesso funcionem no âmbito da restrição de acesso, juntamente com o aumento do fluxo de veículos em um estacionamento é trazido também uma ineficiência em relação ao registro histórico dessas movimentações. Dessa forma o objetivo dessa pesquisa é concepção de uma solução de software de fácil implantação e utilização que permita eficazmente registrar os eventos de entrada e saída de veículos de um estacionamento através de nenhuma interação humana, coletando estes dados de veículos conhecidos para posterior visualização, consulta e auditoria. Para a elaboração desta pesquisa foi utilizada a metodologia estudo de caso, já para o desenvolvimento do produto final foi utilizado o framework Kanban. Desta forma, o presente trabalho resultou no desenvolvimento de uma aplicação web utilizando um modelo de rede neural previamente definido que permite o reconhecimento de placas veiculares obtidos de imagens de câmeras de vigilância em tempo real, realizando o registro de parqueamentos para os veículos cadastrados, além de entregar uma interface web que oferece acesso através de qualquer navegador web para os usuários do sistema.

**Palavras-chave:** Aplicação. Desenvolvimento. Rede neural. Web. Servidor.

## **ABSTRACT**

Access control in parking lots is a common and widely used process with the aim of ensuring the safety of people and property. This control can be applied in different ways, some of the most common being through human resources (e.g. security guards) or the implementation of physical barriers (e.g. a guardhouse). Although these types of access control work within the scope of access restriction, with the increase in the flow of vehicles in a parking lot, there is also an inefficiency in relation to the historical record of these activities. Therefore, the objective of this research is to design an easy-to-implement and easy-to-use software solution that allows to effectively record the events of entry and exit of vehicles from a parking lot through no human interaction, collecting this data from known vehicles for later visualization, consultation and audit. For the elaboration of this research, the case study methodology was used, while the Kanban framework was used for the development of the final product. In this way, the present work resulted in the development of a web application using a previously defined neural network model that allows the recognition of vehicle license plates obtained from images from surveillance cameras in real time, recording parking for registered vehicles, in addition to providing a web interface that offers access through any web browser to system users.

**Keywords:** Application. Development. Neural network. Web. Server.

## LISTA DE FIGURAS

Figura 1 - Tipos de redes neurais .....	12
Figura 2 - Etapas do sistema de reconhecimento .....	14
Figura 3 - Comparativo entre Phoenix LiveView e frameworks Javascript.....	20
Figura 4 - Tela inicial .....	22
Figura 5 - Tela de login .....	23
Figura 6 - Página principal.....	24
Figura 7 - Acesso restrito .....	25
Figura 8 - Listagem de motoristas .....	25
Figura 9 - Criação de motorista .....	26
Figura 10 - Validação dos dados do motorista.....	27
Figura 11 - Detalhes do motorista .....	28
Figura 12 - Listagem de marcas de veículos .....	29
Figura 13 - Criação de marca de veículo.....	29
Figura 14 - Listagem de modelos de veículos.....	30
Figura 15 - Criação de modelo de veículo .....	30
Figura 16 - Listagem de veículos.....	31
Figura 17 - Cadastro de veículo .....	32
Figura 18 - Validação dos dados do veículo .....	32
Figura 19 - Listagem de associações entre veículos e motoristas .....	33
Figura 20 - Associação entre veículo e motorista.....	33
Figura 21 - Listagem de câmeras .....	34
Figura 22 – Posicionamento de câmeras em guarita.....	35
Figura 23 – Configuração de rede.....	36
Figura 24 - Cadastro de câmera.....	36
Figura 25 - Edição de câmera.....	37
Figura 26 - Reconhecimento automático da entrada do veículo .....	38
Figura 27 – Registro de estacionamento por diferentes câmeras .....	38
Figura 28 - Reconhecimento automático da saída do veículo.....	39
Figura 29 - Listagem de estacionamentos .....	40

## LISTA DE ABREVIATURAS E SIGLAS

ALPR	Automatic License Plate Recognition
CSS	Cascading Style Sheets
DNN	Deep Neural Network
IA	Inteligência Artificial
HTML	Hypertext Markup Language
IFRO	Instituto Federal de Rondônia
MVC	Model View Controller
OCR	Optical Character Recognition
RNA	Rede Neural Artificial
SGBD	Sistema de Gerenciamento de Banco de Dados
YOLO	You Only Look Once

## SUMÁRIO

1. INTRODUÇÃO.....	9
2. REFERENCIAL TEÓRICO.....	11
2.1. SISTEMAS INTELIGENTES.....	11
2.2. REDES NEURAIIS PARA VISÃO COMPUTACIONAL.....	11
2.2.1. Modelo de rede neural ALPR.....	13
2.3. RECONHECIMENTO DE PLACAS VEICULARES .....	13
2.4. CONTROLE DE ACESSO .....	14
3. METODOLOGIA.....	16
3.1. METODOLOGIA DA PESQUISA.....	16
3.2. METODOLOGIA DO DESENVOLVIMENTO DE SOFTWARE .....	16
3.2.1. Ciclo de vida do produto .....	16
3.2.2. Tecnologias utilizadas no desenvolvimento do sistema.....	18
3.2.2.1. Banco de dados.....	18
3.2.2.2. PostgreSQL.....	19
3.2.2.3. Linguagem de programação .....	19
3.2.2.4. Linguagem de programação Elixir .....	19
3.2.2.5. Framework Phoenix.....	20
3.2.2.6. OpenCV .....	21
3.2.2.7. Docker .....	21
4. RESULTADOS E DISCUSSÕES.....	22
5. CONCLUSÃO.....	41

## 1. INTRODUÇÃO

O avanço da tecnologia atualmente tem revolucionado diferentes áreas da sociedade com notáveis avanços em relação a existentes processos. Dentro do contexto de controle de tráfego de veículos, a automação tecnológica tem desempenhado um papel crucial na otimização desse processo. Nesse cenário, redes neurais de reconhecimento de objetos emergem como ferramentas poderosas a serem utilizadas no âmbito do reconhecimento de placas veiculares, viabilizando uma abordagem mais eficiente e precisa na autenticação e registro de eventos.

Inicialmente, o objetivo do projeto surgiu a partir da necessidade de solucionar desafios identificados no atual cenário do Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO) - Campus Ji-Paraná, onde o controle de entrada e saída de veículos acontece de forma manual, sendo passível de falhas e carecendo da possibilidade de auditabilidade (não é possível posterior consulta de registros de parqueamentos para dado momento).

Dessa forma, a implantação do projeto que propunha uma melhoria substancial do processo aquele conhecido como controle de acesso manual - conforme Brasiliano (2003, apud Silva, 2018, p. 22) - que é atualmente empregado no instituto necessitou um re-escopo dado à impedimentos burocráticos por parte da instituição. Neste escopo inicial, planejávamos desenhar e desenvolver uma solução que atendesse as necessidades da instituição, seguida de sua implantação e testes de eficácia. Este planejamento não pôde ser seguido dado que questões administrativas relacionadas a captura de imagens nos portões da instituição não puderam ser solucionadas durante o tempo de execução definido para esta pesquisa. Dessa forma a automatização de autenticação e do registro de eventos de parqueamentos inicialmente planejado a ser implantado e testado no Campus Ji-Paraná do IFRO não pode ser realizado.

Apesar disso, a hipótese estabelecida para esta pesquisa tem como base a premissa de que a implementação e utilização de uma aplicação web de alta disponibilidade capaz de realizar reconhecimento de veículos de maneira automatizada alimentada por redes neurais resultará na melhoria do processo de controle de acesso de toda e qualquer instituição que a utilize. Assim é proposto que o desenvolvimento de tal aplicação possa ser implantada em instituições além do IFRO.

Dessa forma esta pesquisa se propõe a sistematizar e aprimorar o processo de controle de acesso manual, implementar uma solução baseada em redes neurais para o reconhecimento automatizado de placas veiculares, reduzir falhas na autenticação e autorização de veículos, além de viabilizar a realização de auditorias nos eventos do controle de acesso.

O desenvolvimento do trabalho será apresentado em uma estrutura organizada em capítulos. No Capítulo 2, será abordado o referencial teórico, que fornecerá as bases conceituais necessárias, explorando tópicos relevantes para a compreensão do projeto. Já no Capítulo 3, serão detalhadas as metodologias empregadas no desenvolvimento da pesquisa, incluindo o estudo de caso. O Capítulo 4 apresentará a descrição da implementação da aplicação, abordando aspectos técnicos e funcionais. E as considerações finais e possíveis direções futuras serão apresentadas no Capítulo 5.

## 2. REFERENCIAL TEÓRICO

### 2.1. SISTEMAS INTELIGENTES

Os sistemas inteligentes e o aperfeiçoamento da Inteligência Artificial (IA) tem sido um objetivo buscado desde os primórdios da computação, através da tentativa por replicar a inteligência humana a fim da sua aplicação em tarefas complexas. Como destacado por Osório e Bittencourt (2000, p. 3), a implementação de sistemas inteligentes pressupõe a automação de processos e comportamentos através da "cópia" da inteligência humana.

Segundo Silva e Bastos (2023) sistemas inteligentes têm por objetivo serem “sistemas computacionais e algoritmos que tentam simular alguns mecanismos inteligentes encontrados na natureza. Na maioria das vezes os sistemas computacionais inteligentes são inspirados no funcionamento do comportamento humano e englobam diversas abordagens e técnicas (também chamadas de subáreas)”.

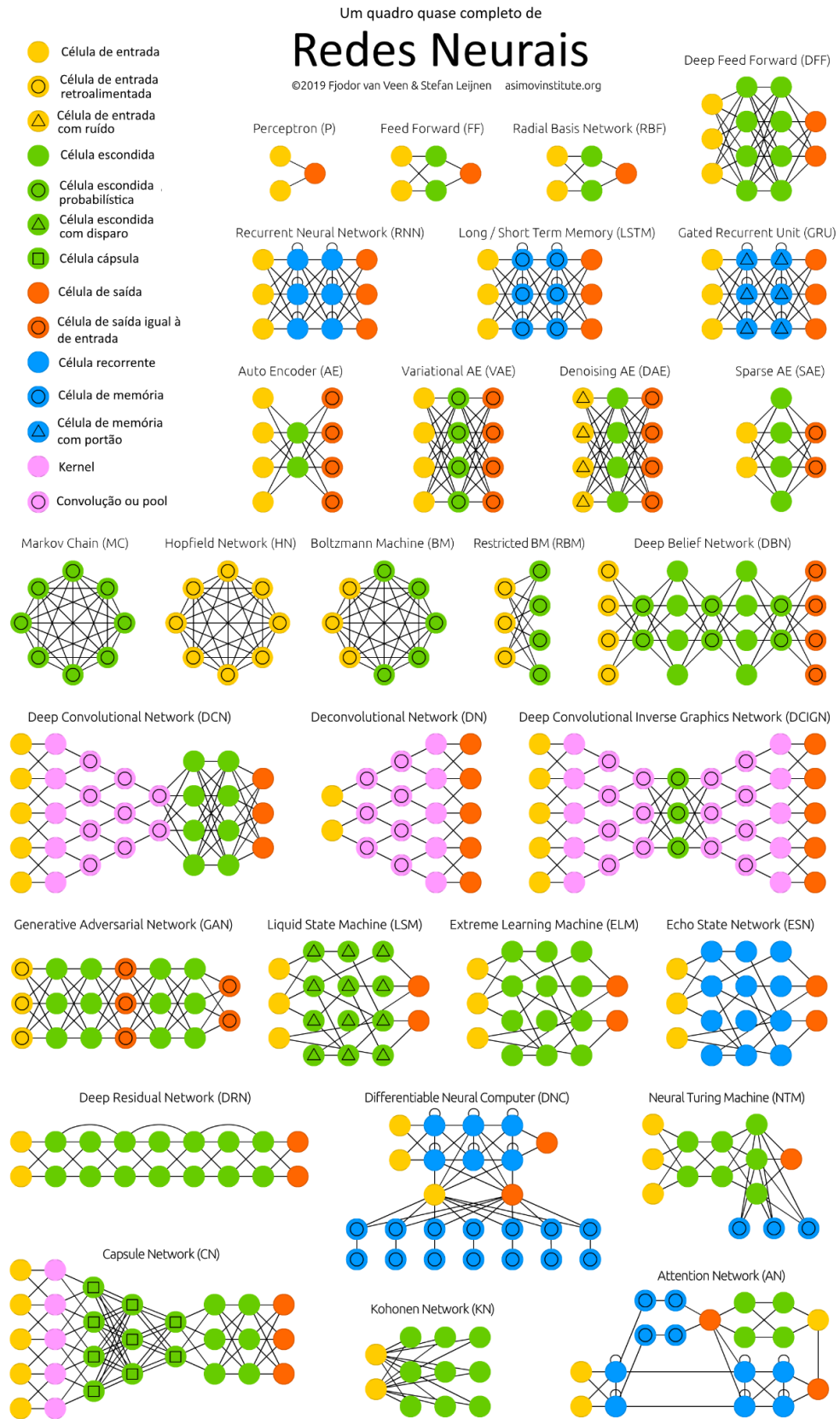
Sistemas inteligentes poder ser desenvolvidos através de diferente técnicas-chave que oferecem diferentes graus de habilidade para representar o conhecimento humano, sendo as principais dessas técnicas a Aquisição de Conhecimento, Aprendizado de Máquina, Redes Neurais, Logica *Fuzzy* etc. (Rezende, 2003). Dessa forma, uma das abordagens utilizadas para atingir o objetivo de replicar inteligência artificialmente é através de Redes Neurais Artificiais (RNAs). As RNAs, inspiradas nos estudos da maneira como se organiza e como funciona o cérebro humano, são ferramentas da IA que demonstram a capacidade única de adaptação e aprendizado a partir de exemplos fornecidos (Osório; Bittencourt, 2000).

### 2.2. REDES NEURAIS PARA VISÃO COMPUTACIONAL

Redes Neurais Artificiais (RNAs) são técnicas computacionais - derivadas dos sistemas inteligentes - que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. Elas são compostas por várias unidades de processamento, que geralmente são conectadas por canais de comunicação que estão associados a determinado peso (Carvalho, 2009).

As RNAs podem ser classificadas em diferentes tipos (arquiteturas), as quais são usadas para diferentes propósitos. Sendo segundo Fleck et al. (2016) a escolha da melhor arquitetura de rede neural uma das maiores dificuldades em seu uso, uma vez que esse processo é experimental e demanda um grande tempo de execução.

**Figura 1 - Tipos de redes neurais**



Por conta desta variedade de arquiteturas e diferentes propósitos, as RNAs podem ser aplicadas à diferentes tipos de tarefas, tais como: o reconhecimento de padrões (e.g. reconhecimento de faces humanas), classificação (ex. reconhecimento de caracteres - OCR), transformação de dados (ex. compressão de informações), a predição (ex. previsão de séries temporais), o controle de processos e a aproximação de funções (ex. aplicações na área da robótica) sendo um de seus usos mais intensivos o processamento de imagens, que em seu eixo de atuação possui o reconhecimento de padrões, definido como o processamento de dada imagem original que permite a análise e identificação de seus componentes. Dessa forma é possível a extração de uma descrição de alto nível de seus componentes ou a classificação de seus componentes em grupos pré-definidos (classes de objetos) (Osório; Bittencourt, 2000, p. 19).

### **2.2.1. Modelo de rede neural ALPR**

O modelo de rede neural utilizado para o reconhecimento de caracteres de placas veiculares é um modelo baseado no sistema de detecção You only look once (YOLO), que contém uma abordagem unificada para detecção e classificação de leiautes de placas veiculares para melhorar os resultados de reconhecimento usando regras de pós-processamento (Redmon, 2016, tradução nossa).

Enquanto frameworks de detecção de objetos anteriores ao YOLO percorriam diferentes partes de uma imagem várias vezes em diferentes escalas utilizando técnicas de classificação na tarefa de detecção de objetos em imagens, o que torna a detecção lenta e custosa, o YOLO simplesmente considera a detecção de objetos como um único problema de regressão, onde uma única rede convolucional infere simultaneamente várias molduras e suas prováveis classes de uma só vez, permitindo que a detecção de objetos seja extremamente rápida (Redmon, 2016, tradução nossa).

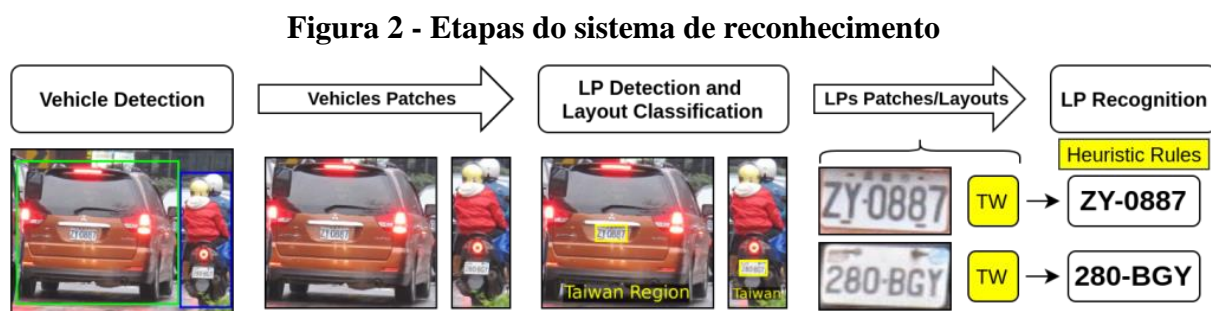
## **2.3. RECONHECIMENTO DE PLACAS VEICULARES**

De acordo com Du et al. (2012, apud Silva, 2022), um sistema de reconhecimento automático de placas de licença automotiva, do inglês *Automatic License Plate Recognition* (ALPR), é aquele que “utiliza uma imagem ou uma sequência delas para extrair informações da placa de um veículo.” O reconhecimento da placa “é atingido por meio da combinação de

diversas técnicas de processamento de imagens, detecção de objetos e reconhecimento de padrões.”.

Dessa forma utilizaremos o sistema de ALPR descrito anteriormente que conforme Larota et al. (2021) - seus autores - propõe uma abordagem unificada no processo de detecção de placas veiculares e classificação de leiautes e tem por objetivo alcançar a melhor relação velocidade/precisão em cada etapa de detecção.

O modelo propõe primeiramente localizar veículos na imagem de entrada e depois, a partir das detecções realizadas, detectar suas respectivas placas. Logo em seguida a rede neural é alimentada com cada placa identificada, detectando e reconhecendo todos os caracteres simultaneamente, evitando assim a necessidade de lidar com a tarefa de segmentação de caracteres (Larota et al, 2021, tradução nossa).



Fonte: Laroca, 2021

## 2.4. CONTROLE DE ACESSO

Controle de acesso é entendido como a atividade que resulta no controle de circulação de veículos ou pessoas em determinado espaço, através de obstáculos que dificultam, retardam e controlam toda a movimentação. “É definido como meios que estabelecem restrições à circulação e/ou acesso à determinada área e são constituídas, em conjunto com a ação humana, de barreiras físicas como cancelas, catracas, portas, portões e torniquetes” (Silva, 2018, p. 21).

Segundo Brasiliano (2003, apud Silva, 2018, p. 22) os sistemas de controle de acesso podem ser divididos em três categorias: manuais, semiautomáticos e automáticos. Desses três grupos, é definido como sistemas semiautomáticos aqueles que “integram o recurso humano com a tecnologia, geralmente esses sistemas selecionam o acesso por meio de um interfone e/ou porteiro eletrônico, [...] sendo a autorização liberada pelo agente de segurança se as condições de acesso forem preenchidas” (Brasiliano, 2003, apud Silva, 2018, p. 22).

Dessa forma, um controle de acesso semiautomático pode ser implementado utilizando tecnologias de inteligência artificial que empreguem o papel de autenticação de atores. Figueira (2023) em seu trabalho sugere a utilização de redes neurais para a identificação de características biométricas, permitindo a autenticação de indivíduos através do reconhecimento facial e impressão digital. Ele afirma que “entre as vantagens destes sistemas estão a garantia de que o usuário identificado é a pessoa devidamente autorizada, a facilidade de automatização de acesso, além de permissão de controle e monitoramento remotos” (Figueira 2023, p. 15).

### **3. METODOLOGIA**

#### **3.1. METODOLOGIA DA PESQUISA**

Para o desenvolvimento da pesquisa, foi utilizado a metodologia de Estudo de Caso, que, segundo Yin (2001), “apenas uma das muitas maneiras de se fazer pesquisa [...]” e os “representam a estratégia preferida quando se colocam questões do tipo ‘como’ e ‘por que’”.

Assim, Yin (1994, apud Araújo *et al.* 2008) define “estudo de caso” com base nas características do fenômeno em estudo e com base num conjunto de características associadas ao processo de recolha de dados e às estratégias de análise dos mesmos.

Desta forma, a realização desta pesquisa deve se consistir primeiramente na compreensão do controle de acesso manual, de forma a entender as informações necessárias para sua realização e sua eficácia para que, por fim, seja realizado uma análise dos dados obtidos através da implantação do produto, iniciando assim uma discussão e comparação dos resultados obtidos.

#### **3.2. METODOLOGIA DO DESENVOLVIMENTO DE SOFTWARE**

Para o desenvolvimento do sistema, foi optado a utilização do framework Kanban com práticas ágeis como metodologia de desenvolvimento, dado à sua abordagem visual e flexível para o gerenciamento do fluxo de trabalho.

O método Kanban – que em japonês significa sinal visual - surgiu no Japão com o Toyota Production System com o objetivo de controlar a fabricação de automóveis, é entre as metodologias para desenvolvimento de software a menos prescritiva, sendo suas prescrições segundo Kniberg (2009, apud Silva; Santos; Santos Neto, 2012) somente visualize o fluxo de trabalho atual; limite o fluxo de trabalho; e acompanhe e gerencie o fluxo de trabalho.

A implantação do Kanban se dá através da representação das tarefas (ou itens de trabalho) por meio de cartões fixados em um quadro. Enquanto este quadro, por sua vez, é dividido em colunas que representam as fases do fluxo de trabalho (Silva; Santos; Santos Neto, 2012, p. 6).

##### **3.2.1. Ciclo de vida do produto**

Neste tópico é apresentado as iterações de desenvolvimento realizadas até a conclusão do produto.

<b>Nº da tarefa</b>	<b>Tarefa</b>	<b>Descrição</b>	<b>Estimativa de esforço</b>
1	Configuração inicial do projeto	Gerar um novo projeto Phoenix LiveView, configurando as ferramentas de desenvolvimento e criando o commit GIT inicial.	4h
2	Desenvolvimento do registro de usuários	Gerar os módulos responsáveis por lidar com os dados de usuários do sistema.	8h
3	Desenvolvimento do processo de autenticação de usuários	Permitir que usuários realizem login através de uma interface no sistema.	8h
4	Desenvolvimento das páginas de motoristas	Desenvolver as telas de listagem, detalhes, adição e edição dos dados de motoristas.	8h
5	Desenvolvimento das páginas de veículos	Desenvolver as telas de listagem, detalhes, adição e edição dos dados de veículos.	8h
6	Desenvolvimentos das páginas de veículo & motorista	Desenvolver as telas de listagem, detalhes, adição e edição dos dados de relacionamento entre veículos motoristas.	8h
7	Desenvolvimento do módulo de captura de frames da câmera	Desenvolver o módulo para captura de frames para cada câmera.	40h
8	Desenvolvimento do módulo de reconhecimento de placas veiculares	Desenvolver o módulo para o reconhecimento de placas veiculares para dada imagem utilizando a rede neural definida.	40h
9	Desenvolvimento do registro automático de parqueamentos	Desenvolver o serviço que encontrará cada placa veicular de cada frame recuperado, registrando o parqueamento para cada veículo detectado.	20h
10	Desenvolvimento da página de parqueamentos	Desenvolver as telas de listagem e detalhes de parqueamentos.	8h

11	Desenvolvimento das telas de câmeras	Desenvolver as telas de listagem, detalhes, edição e adição de câmeras para detecção.	8h
12	Desenvolvimento das regras de autorização e permissão de usuários	Implementar regras de permissão de usuários para páginas baseada em papéis de usuários.	8h

Os artefatos do projeto podem ser encontrados no apêndice deste trabalho por nome de Projeto de Software.

### 3.2.2. Tecnologias utilizadas no desenvolvimento do sistema

Nos próximos tópicos, são apresentadas as tecnologias utilizadas para o desenvolvimento do sistema, incluindo o banco de dados, a linguagem de programação e os frameworks utilizados. Cada uma das ferramentas utilizadas desempenha um papel fundamental na arquitetura e no processo de desenvolvimento do sistema, tendo sido elas escolhidas com base em suas funcionalidades, especificidades e requisitos do projeto.

Juntamente com a linguagem de programação escolhida, frameworks e bibliotecas específicas foram utilizados na criação do sistema - definindo sua estrutura funcional – em conjunto com um banco de dados utilizado para persistência dos mesmos.

#### 3.2.2.1. Banco de dados

Desde o princípio, a humanidade sempre precisou armazenar informações que julgavam importantes para sua vida para que, quando necessário, pudesse realizar posterior consultas. Dessa necessidade surgiram diversas técnicas para registro de informações, no qual a partir do surgimento da era computacional, a demanda de uma maneira de armazenamento e recuperação de informações de forma prática, eficiente e confiável levaram ao desenvolvimento dos bancos de dados computadorizados. (Alves, 2009).

Dado seu surgimento e problemas que procura resolver, podemos dizer que um banco de dados é um conjunto de dados devidamente relacionados e armazenados segundo uma determinada lógica que permita serem recuperados quando necessários (Carneiro, 2004;

Machado, 2020). Para Date (2004, p. 3), um banco de dados é “um repositório ou recipiente para uma coleção de arquivos de dados computadorizados”, sendo um sistema de banco de dados “basicamente apenas um sistema computadorizado de manutenção de registros”.

#### 3.2.2.2. PostgreSQL

O PostgreSQL é um poderoso Sistema de Gerenciamento de Banco de Dados (SGBD) de código aberto (Carvalho, 2017). De acordo com Caldeira (2015), ele é considerado o SGBD relacional *open source* mais avançado do mundo e possui vantagens como ser amigável e de fácil administração, utilizar SQL como linguagem e ser *multi-threaded*.

#### 3.2.2.3. Linguagem de programação

Sabe-se que uma linguagem de programação é uma maneira padronizada usada para expressar as instruções de um programa a um computador programável (Gotardo, 2015). Dessa forma, a linguagem de programação é uma ferramenta essencial para a execução do desenvolvimento de um *software*, já que “qualquer tipo de informação que deva ser transferida, processada ou armazenada deve estar na forma de uma linguagem” (Ferrari e Cechinel, 2008, p. 18). Gotardo (2015, p. 17) enfatiza dizendo:

Através da especificação de uma linguagem de programação você pode especificar quais dados um computador vai usar; como estes dados serão tratados, armazenados, transmitidos; quais ações devem ser tomadas em determinadas circunstâncias.

#### 3.2.2.4. Linguagem de programação Elixir

A linguagem de programação Elixir é uma linguagem criada pelo brasileiro José Valim em 2012, sendo uma linguagem que segue o paradigma funcional de programação e auxilia na resolução de problemas de escalabilidade, tolerância a falhas e alta concorrência (Bordignon, 2020; Davi, 2017).

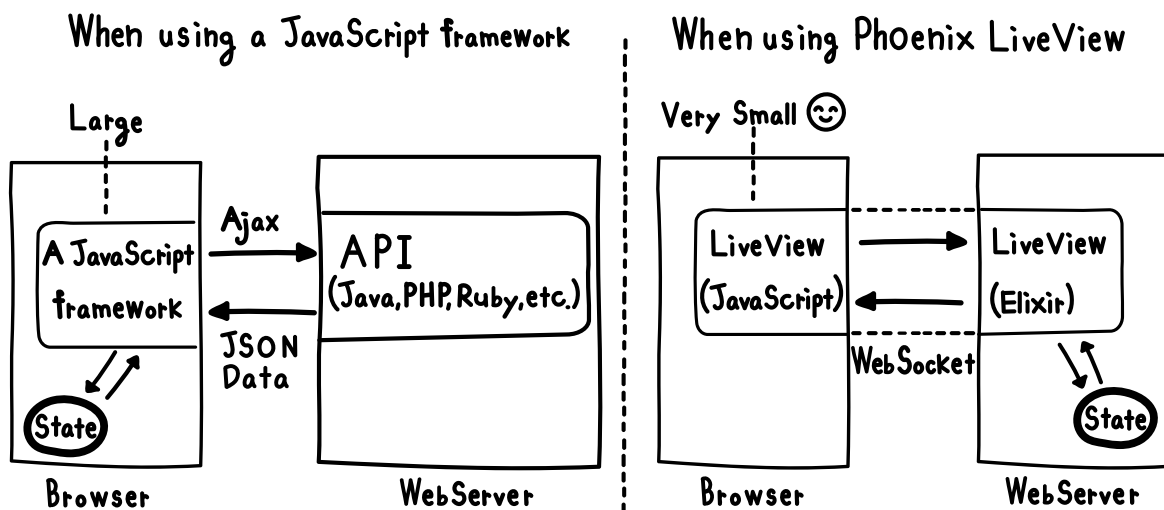
Por ser uma linguagem funcional, ela engloba vários benefícios desse paradigma, como a imutabilidade de dados, que associada à capacidade da linguagem de criar *threads* leves faz

com que sistemas altamente concorrentes e escaláveis possam ser desenvolvidos (Borges, 2016).

### 3.2.2.5. Framework Phoenix

O framework Phoenix é um framework para desenvolvimento web escrito em Elixir que implementa o padrão *Model View Controller* (MVC) no *server-side*. Ele oferece alta produtividade para o desenvolvedor e alto desempenho para a aplicação, além de funcionalidades que permitem implementar recursos de tempo real e fornecer altíssima velocidade (Phoenix Framework, 2022, tradução nossa).

**Figura 3 - Comparativo entre Phoenix LiveView e frameworks Javascript**



Fonte: TeamGenik, 2023

Phoenix LiveView é uma extensão do framework Phoenix que provê uma alternativa os existentes frameworks para a construção de aplicações web. Ele é um framework centrado no servidor que reativamente atualiza a página web renderizada no *front-end* baseado na mudança de estados mantida no *back-end*, eliminando camadas de abstração dado que soluciona cliente e servidor em uma única abstração, evitando a necessidade de lidar com camadas de comunicação (Phoenix Framework, 2022; McCord, 2021, tradução nossa).

#### 3.2.2.6. OpenCV

OpenCV é uma biblioteca popular de software de visão computacional e aprendizado de máquina de código aberto, que foi criada para fornecer uma infraestrutura comum para aplicações de visão computacional e para acelerar o uso da percepção de máquina em produtos comerciais (OpenCV Team, 2023, tradução nossa).

Dentre as várias funcionalidades oferecidas pelo OpenCV, o interesse do projeto é voltado para o módulo DNN (Deep Neural Network), que oferece a capacidade de executar inferências em modelos de aprendizado profundo pré-treinados dentro do próprio OpenCV (Ponnusamy, 2018, tradução nossa).

#### 3.2.2.7. Docker

Docker é uma ferramenta utilizada no desenvolvimento, implantação e execução de aplicações, permitindo separar aplicações de infraestrutura, executando-as em ambientes isolados. Utilizando o Docker é possível empacotar uma aplicação em containers que são isolados a nível de disco, memória, processamento e rede da máquina *host* (Gomes, 2019; Docker, 2023, tradução nossa).

A utilização de containers Docker permite a implantação simples e rápida de aplicações, visto que containers podem ser montados como ambientes específicos prontos que são facilmente compartilháveis e executáveis (Gomes, 2019).

## 4. RESULTADOS E DISCUSSÕES

Neste capítulo são apresentadas as funcionalidades resultantes do desenvolvimento do produto, apresentando a interface do usuário e interações com o sistema.

Para executar a aplicação localmente ou até mesmo em um servidor, Docker deve ser utilizado para a execução do container da aplicação. Este container já possui todas as dependências necessárias para a execução da aplicação configuradas, restando somente que o desenvolvedor responsável pelo deploy configure as variáveis de ambiente. Desta forma, para configurar e iniciar a aplicação em uma máquina Linux, siga os passos descritos no repositório do projeto: <[https://github.com/wigny/parking\\_lot.git](https://github.com/wigny/parking_lot.git)>.

Após toda a configuração realizada, a aplicação deverá estar disponível no endereço definido. Ao acessar o endereço da aplicação pelo navegador, o usuário quando ainda não autenticado é direcionado a tela inicial.

**Figura 4 - Tela inicial**

# Parking Lot

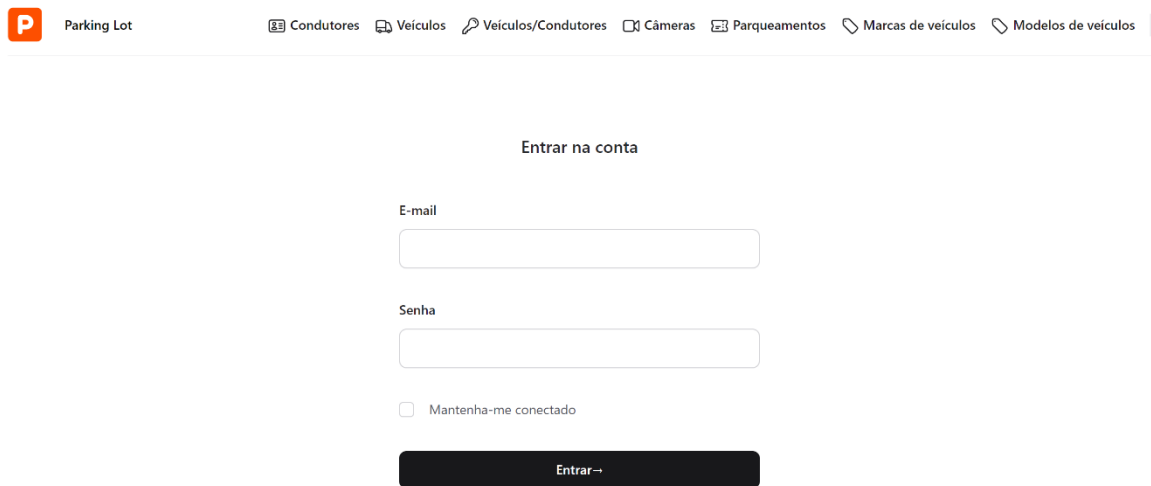
Gerencie motoristas e veículos autorizados do seu estacionamento juntamente com registro automático de parqueamentos alimentado por Reconhecimento Automático de Placas Veiculares (ALPR).

Entrar agora →

Fonte: Autor

Nesta tela o usuário tem acesso ao botão de login, que o redirecionará a fazer o login social com sua conta previamente criada.

## Figura 5 - Tela de login



The image shows a web application interface for a parking system. At the top left, there is a logo with the letter 'P' in a red square followed by the text 'Parking Lot'. To the right of the logo is a horizontal navigation menu with several items: 'Condutores', 'Veiculos', 'Veiculos/Condutores', 'Câmeras', 'Parqueamentos', 'Marcas de veiculos', and 'Modelos de veiculos'. Each item has a small icon next to it. Below the navigation menu, the page is centered and contains the following elements: the text 'Entrar na conta', a label 'E-mail' above a text input field, a label 'Senha' above another text input field, a checkbox labeled 'Mantenha-me conectado', and a dark button with the text 'Entrar→'.

Fonte: Autor

Após ter realizado o login com sua conta, o usuário é redirecionado para a página principal da aplicação.

Esta tela de login foi desenvolvida levando em consideração que este sistema deve permanecer como um sistema interno e não estará disponível para todos. Por isso não foi considerado a criação de um registro de usuários pela interface, visto que o administrador do sistema tem o poder de criar usuários através de um terminal remoto.

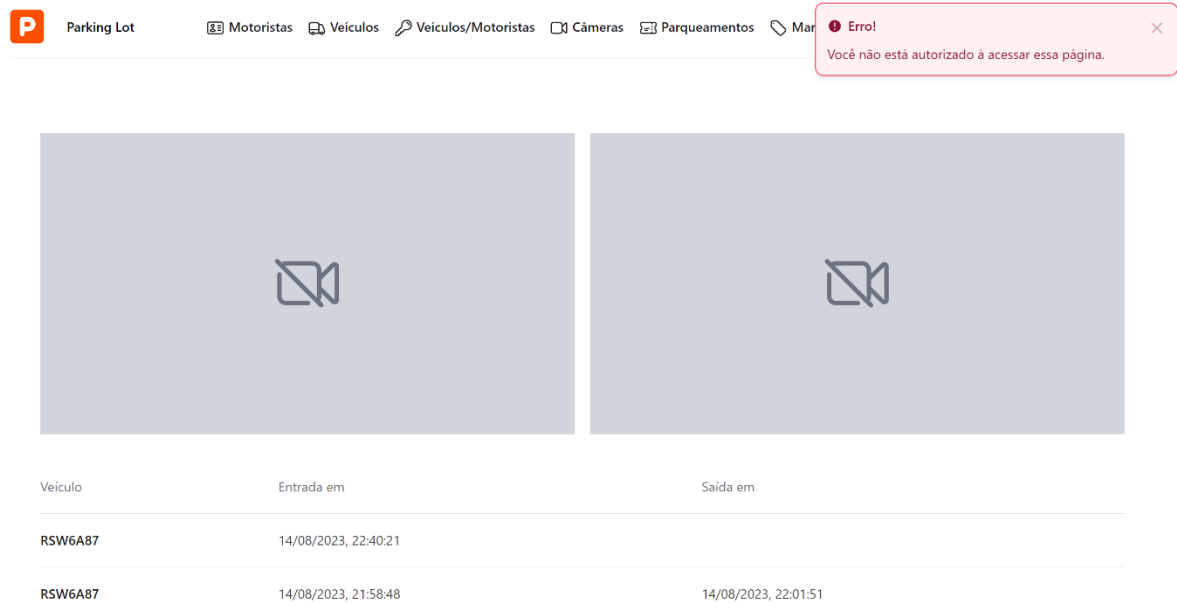
**Figura 6 - Página principal**

Veículo	Entrada em	Saída em
RSW6A87	14/08/2023, 22:40:21	
RSW6A87	14/08/2023, 21:58:48	14/08/2023, 22:01:51

Fonte: Autor

Nesta página o usuário encontra todos os menus de acesso para as outras páginas como também a visualização em tempo real do reconhecimento automático de veículos. Como a aplicação fornece diferentes níveis de acesso para usuários com diferentes cargos, as páginas de cadastro e edição de dados requerem que o usuário logado na aplicação possua o cargo elevado de administrador, enquanto páginas de informações não possuem essa restrição.

### Figura 7 - Acesso restrito



Fonte: Autor

Ao clicar no menu “Motoristas”, o usuário com devida permissão é levado à página de listagem, cadastro e edição de motoristas.

### Figura 8 - Listagem de motoristas



Fonte: Autor

Nesta página o usuário poderá adicionar as informações de motoristas que mais tarde poderão ser associados aos seus respectivos veículos. Ao clicar no botão “Novo motorista”, o modal de criação é aberto na página, no qual o usuário deverá preencher todas as informações relevantes de um motorista corretamente, como mostrado nas figuras abaixo:

**Figura 9 - Criação de motorista**

The figure consists of three screenshots from a web application. The first screenshot on the left shows a page titled "Parking Lot" with a list of drivers. The list has columns for "Nome" and "CPF", with entries for "Wigny" (045.434.132-61) and "Michele" (037.381.012-17). The middle screenshot shows a form titled "Novo motorista" with input fields for "Nome", "CPF", "CNH", "E-mail", and "Telefone". There is a checked checkbox for "Ativo" and a "Salvar motorista" button. The third screenshot on the right shows a modal titled "Modelos de veículos" with a "Novo motorista" button and a table with "Ativo" status and "Editar" and "Excluir" actions.

Nome	CPF
Wigny	045.434.132-61
Michele	037.381.012-17

Novo motorista

Nome

CPF

CNH

E-mail

Telefone

Ativo

Salvar motorista

Modelos de veículos Sair

Novo motorista

Ativo

true	Editar	Excluir
true	Editar	Excluir

Fonte: Autor

**Figura 10 - Validação dos dados do motorista**

The image shows a mobile application interface for driver registration. On the left, there is a sidebar menu with a 'P' icon and 'Parking Lot' text, and a 'Lista de motoristas' section containing a table with columns 'Nome' and 'CPF'. The table lists two drivers: 'Wigny' with CPF '045.434.132-61' and 'Michele' with CPF '037.381.012-17'. The main area displays a form with the following fields and validation messages:

- Nome:** Empty field with error message: "não pode estar em branco".
- CPF:** Field containing '12345678912' with error message: "possui dígito de verificação inválido".
- CNH:** Field containing '0000000000' with error message: "possui dígitos inválidos".
- E-mail:** Empty field with error message: "não pode estar em branco".
- Telefone:** Field containing '00000000' with error message: "deve ter 11 caracteres".

On the right side of the form, there is a 'Novo motorista' button and a table with columns 'Ativo', 'sim', 'Editar', and 'Excluir'. The 'sim' row has 'Editar' and 'Excluir' buttons, and the 'não' row has 'Editar' and 'Excluir' buttons. At the top right of the right panel, there are links for 'Modelos de veículos' and 'Sair'.

Fonte: Autor

Depois do motorista criado, o usuário poderá excluir, editar ou detalhar as informações do motorista a partir da listagem de motoristas. Ao clicar em uma das entradas de lista, os detalhes de um motorista são exibidos:

## Figura 11 - Detalhes do motorista

Parking Lot

Condutores Veículos Veículos/Condutores Câmeras Parqueamentos Marcas de veículos Modelos de veículos Sair

Conductor 2 Editar condutor

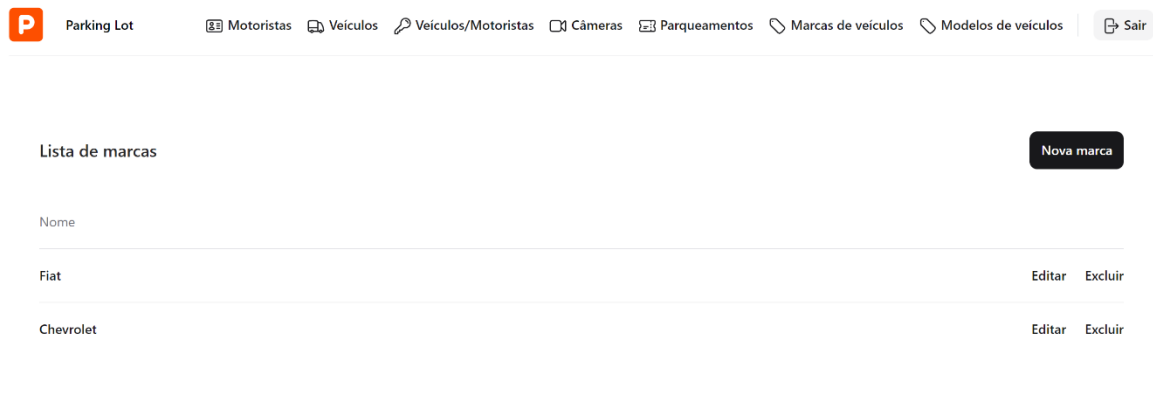
Nome	João Silva
CPF	123.456.789-09
CNH	12345678900
E-mail	joao@email.com
Telefone	+55 99 98765-4321
Ativo	sim

[← Voltar para lista de condutores](#)

Fonte: Autor

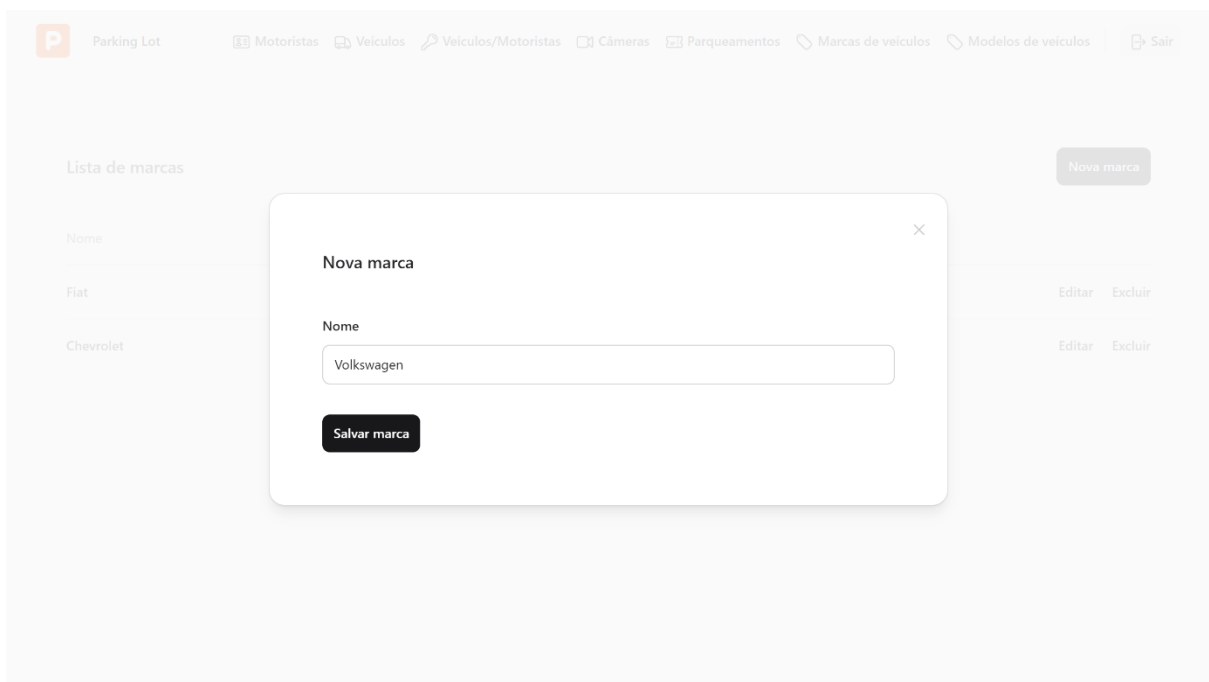
Para a criação de veículos, primeiro é necessário que o usuário popule a aplicação com informações de marcas e modelos de veículos que serão posteriormente utilizados para ajudar a identificar um veículo. Para isso o usuário poderá utilizar as telas “Marcas de veículos” e “Modelos de veículos” para realizar a adição e edição das informações.

**Figura 12 - Listagem de marcas de veículos**



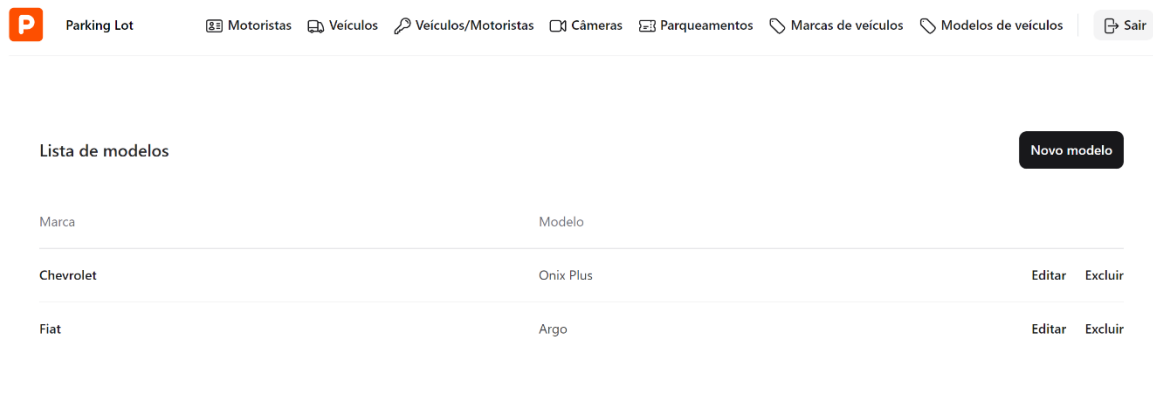
Fonte: Autor

**Figura 13 - Criação de marca de veículo**



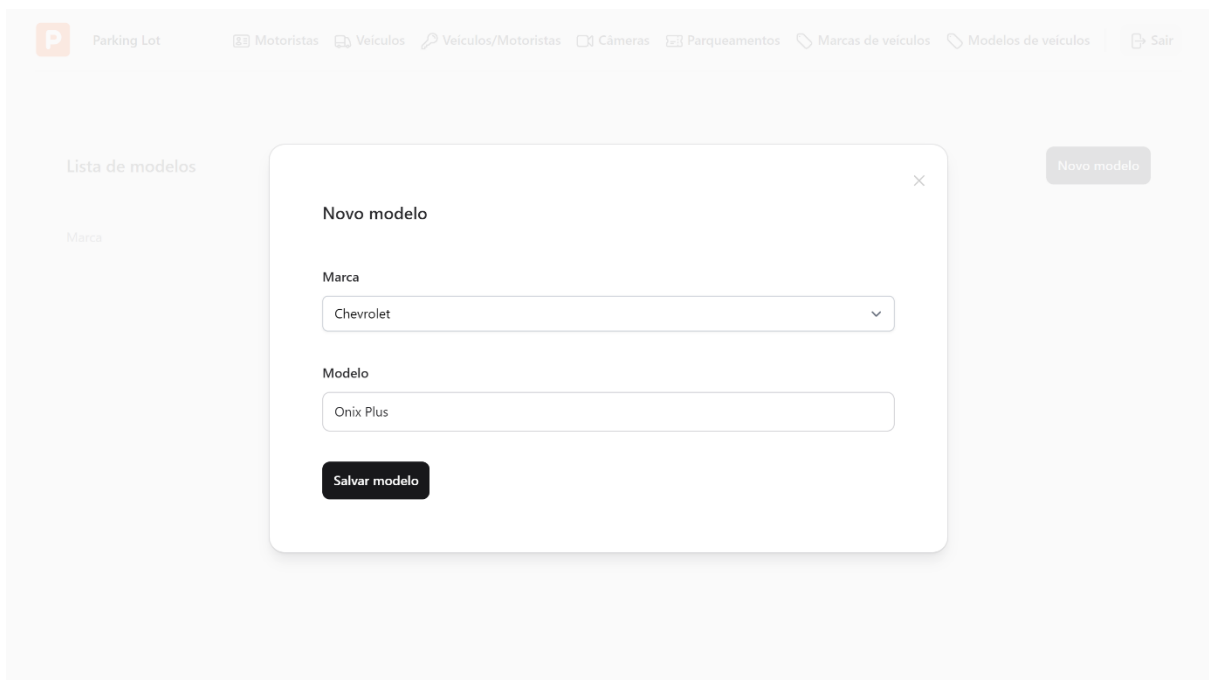
Fonte: Autor

**Figura 14 - Listagem de modelos de veículos**



Fonte: Autor

**Figura 15 - Criação de modelo de veículo**



Fonte: Autor

Logo após, o usuário poderá realizar a adição dos veículos conhecidos de seu estacionamento ao acessar a página pelo menu “Veículos”.

## Figura 16 - Listagem de veículos

Placa	Tipo	Marca	Modelo	Cor	Ativo	
RSW6A87	Automóvel	Chevrolet	Onix Plus	Cinza	sim	Editar Excluir

Fonte: Autor

Assim como para a página “Motoristas”, nesta página é possível a realização de cadastro, alteração e exclusão dos dados de um veículo. A partir das informações cadastradas nesta página a aplicação executará o serviço de reconhecimento dos veículos automaticamente, utilizando sua placa para realizar a identificação de cada veículo permitido.

**Figura 17 - Cadastro de veículo**

Novo veículo

Placa

Tipo

Modelo

Cor

Ativo

Salvar veículo

Fonte: Autor

**Figura 18 - Validação dos dados do veículo**

Novo veículo

Placa

Tipo

Modelo

Cor

Ativo

Salvar veículo

possui um formato inválido

Fonte: Autor

Para a realização do cadastro/edição do veículo, é necessário que os dados estejam corretos.

Após veículo e motorista já cadastrados, é possível realizar a associação entre eles a partir da página “Veículos/Motorista”.

**Figura 19 - Listagem de associações entre veículos e motoristas**



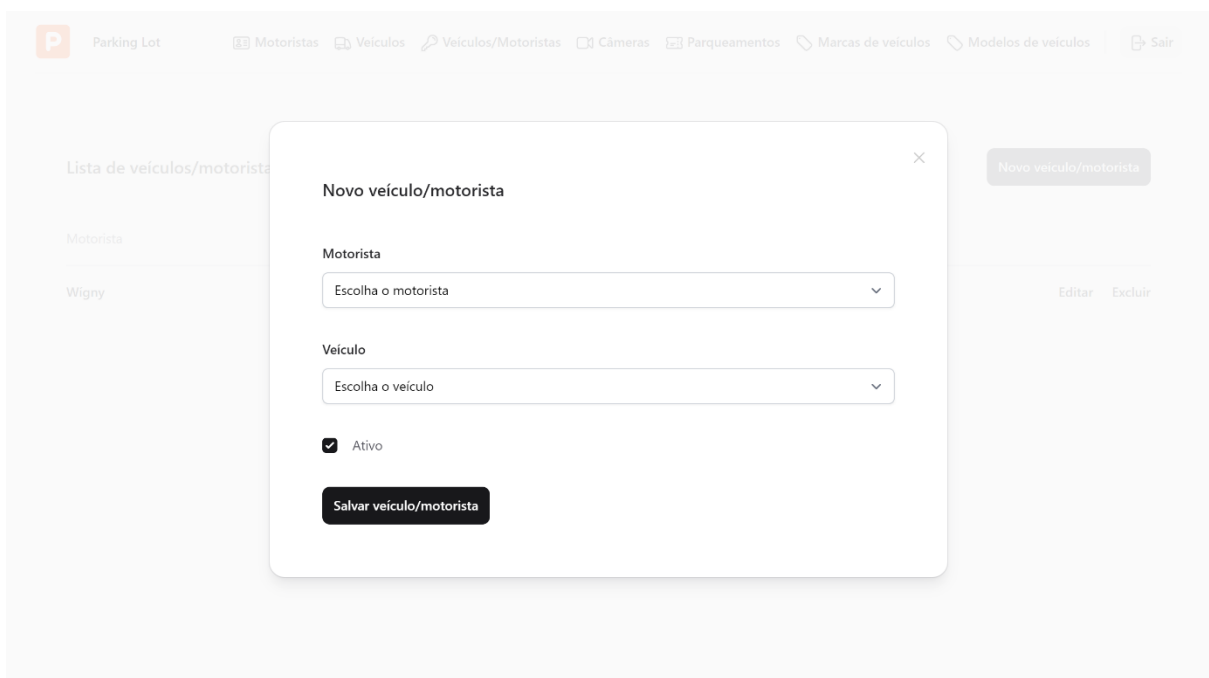
The screenshot shows a web interface for 'Parking Lot' with a navigation menu. The main content area is titled 'Lista de veículos/condutores' and features a table with the following data:

Condutor	Veículo	Ativo	
João Silva	ABC1234	sim	<a href="#">Editar</a> <a href="#">Excluir</a>

A 'Novo veículo/conductor' button is located in the top right corner of the table area.

Fonte: Autor

**Figura 20 - Associação entre veículo e motorista**



The screenshot shows a modal form titled 'Novo veículo/motorista' with the following fields and controls:

- Motorista:** A dropdown menu with the placeholder text 'Escolha o motorista'.
- Veículo:** A dropdown menu with the placeholder text 'Escolha o veículo'.
- Ativo:** A checked checkbox.
- Salvar veículo/motorista:** A black button at the bottom of the modal.

The background shows a blurred view of the 'Lista de veículos/motorista' page with a 'Novo veículo/motorista' button.

Fonte: Autor

Com as informações dos veículos e seus motoristas registrados, o usuário poderá, a partir do menu “Câmeras”, acessar a tela de configuração do endereço de transmissão das câmeras que serão utilizadas no reconhecimento automático de veículos.

**Figura 21 - Listagem de câmeras**



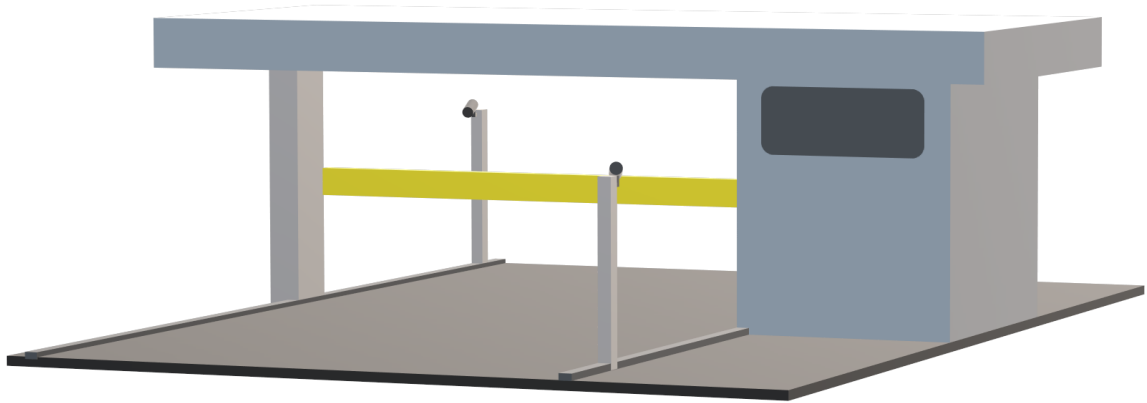
Tipo	URI	Ativo	Editar	Excluir
Entrada	rtsp://172.28.6.108:8554/video	não	Editar	Excluir
Saída	rtsp://172.28.6.108:8554/video2	não	Editar	Excluir

Fonte: Autor

A partir desta tela o usuário poderá definir qual câmera será utilizada para o registro de entrada ou saída do veículo, seu endereço de *stream* e se esta câmera se encontra ativa para o reconhecimento automático.

Duas câmeras deverão ser utilizadas para a captura da entrada e saída de veículos, respectivamente. Desta forma será possível distinguir as ações de entrada ou saída baseado no posicionamento das câmeras em relação aos portões do estacionamento. Podemos considerar o seguinte posicionamento de câmeras, sendo a mais externa responsável pela captura da entrada de veículos, enquanto a mais interna deverá capturar sua saída.

**Figura 22 – Posicionamento de câmeras em guarita**

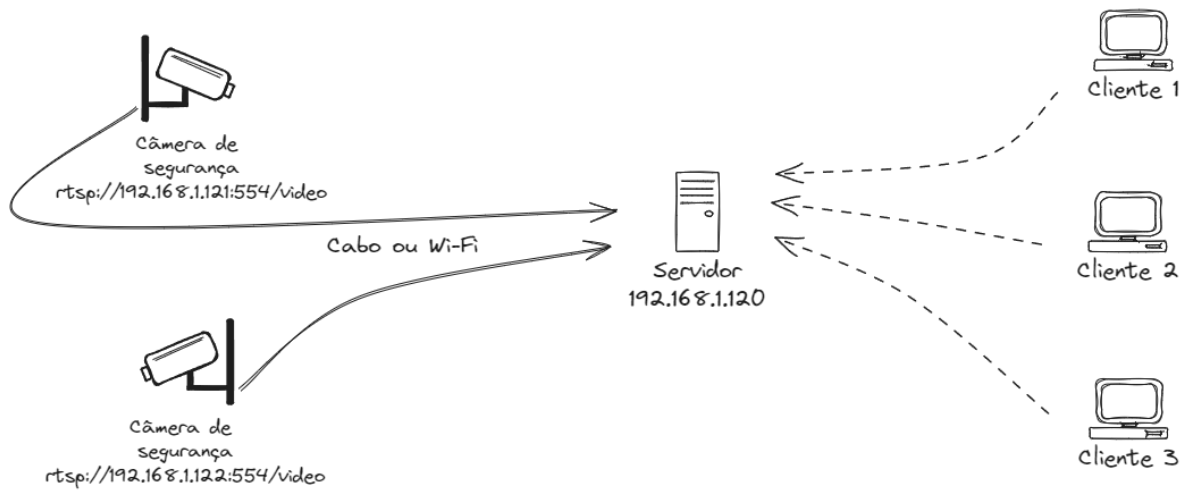


Fonte: Autor

É importante salientar que o posicionamento correto das câmeras de vigilância é de importantíssima necessidade, de forma a evitar o registro de veículos que não estão de fato adentrando/deixando a instituição ou o registro duplo de entrada e saída em um único fluxo. Dessa forma é desejado que as câmeras estejam diretamente alinhadas para capturar a placa veicular de cada veículo.

Uma vez que as câmeras estejam posicionadas, para que seja possível a conexão do servidor da aplicação com o stream das câmeras, é necessário que estes dispositivos estejam disponíveis na rede utilizada pelo servidor. Isto poderá ser atingido de diferentes formas, sendo a maneira mais simples utilizar a mesma rede Ethernet entre o servidor e as câmeras:

**Figura 23 – Configuração de rede**



Fonte: Autor

Assim o usuário poderá seguir com o registro das câmeras na aplicação acessando a tela exibida abaixo, preenchendo o tipo desta câmera (de entrada ou saída) e sua URL de transmissão de vídeo:

**Figura 24 - Cadastro de câmera**

A imagem mostra a interface de usuário de uma aplicação web. No topo, há uma barra de navegação com o ícone "P" e o texto "Parking Lot", além de links para "Motoristas", "Veículos", "Veículos/Motoristas", "Câmeras", "Parqueamentos", "Marcas de veículos", "Modelos de veículos" e "Sair". Abaixo, há uma seção "Lista de câmeras" com uma tabela contendo colunas para "Tipo", "Entrada" e "Saída". No lado direito, há botões "Editar" e "Excluir" para cada linha da tabela. Sobreposta ao centro da tela, há uma janela modal intitulada "Nova câmera". Esta janela contém um formulário com os seguintes campos: "Tipo" (menu suspenso com o texto "Escolha um tipo"), "URI" (campo de texto), e uma caixa de seleção "Ativo". No canto inferior esquerdo da janela, há um botão "Salvar câmera".

Fonte: Autor

Ao alterar o estado atual da câmera utilizando o campo "Ativo", o serviço de reconhecimento de veículos é iniciado ou parado com base no valor atual deste campo.

**Figura 25 - Edição de câmera**

The image shows a web interface for editing a camera. The main content area is titled "Editar câmera" and contains the following fields and controls:

- Tipo:** A dropdown menu with "Entrada" selected.
- URI:** A text input field containing "rtsp://172.28.6.108:8554/video".
- Ativo:** A checkbox that is currently unchecked.
- Salvar câmera:** A black button to save the changes.

The interface also features a sidebar on the left with a "Lista de câmeras" section and a top navigation bar with various menu items.

Fonte: Autor

Desta forma, após o usuário ativar a câmera de captura, ao retornar na tela principal verá o registro automático dos veículos conhecidos, justamente com a visualização da câmera utilizada no reconhecimento.

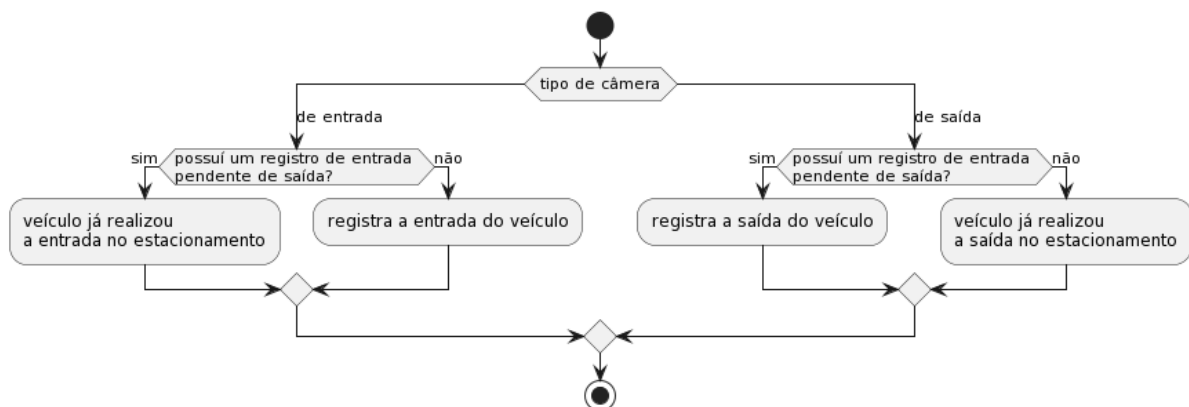
**Figura 26 - Reconhecimento automático da entrada do veículo**



Fonte: Autor

Dado que diferentes câmeras serão utilizadas para o registro de entrada ou saída de um veículo, o sistema deliberadamente salvará um único registro de entrada ou de saída para dado veículo independente do tempo que ele permaneça em foco nas câmeras. Isso evitará que vários registros sejam criados para o mesmo veículo durante a mesma ação (entrada ou saída). Esta regra pode ser representada pelo seguinte fluxo:

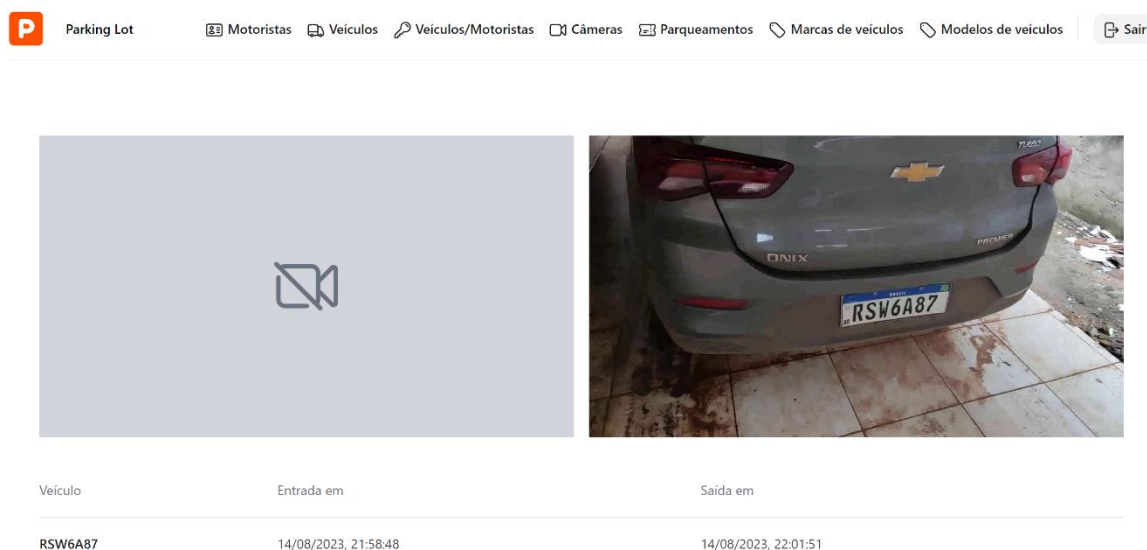
**Figura 27 – Registro de estacionamento por diferentes câmeras**



Fonte: Autor

Dessa forma, na tela principal da aplicação é mostrado os últimos parqueamentos registrados, contendo a placa do veículo, data e hora de entrada e saída do estacionamento.

**Figura 28 - Reconhecimento automático da saída do veículo**



Fonte: Autor

Para encontrar o detalhamento do parqueamento, o usuário poderá através do menu “Parqueamentos” checar a lista de parqueamentos registrados juntamente com seus detalhes.

## Figura 29 - Listagem de parqueamentos



Veiculo	Entrada em	Saida em	
RSW6A87	14/08/2023, 21:58:48	14/08/2023, 22:01:51	<a href="#">Detalhar</a>
RSW6A87	14/08/2023, 22:40:21		<a href="#">Detalhar</a>

Fonte: Autor

Enquanto o desenvolvimento das páginas apresentadas aconteceu de forma simples (dado que estas páginas oferecidas aos usuários do sistema são de fato simples), utilizando as tecnologias bases (HTML e CSS) para construção de páginas web nos *templates* Phoenix, os maiores desafios relacionados à implementação deste projeto não podem ser exibidos em telas, dado que todo o processamento das imagens das câmeras de captura, como o uso da rede neural para o reconhecimento das placas veiculares são realizados em um serviço da aplicação que deve estar em execução a todo o momento.

Dentre os desafios encontrados na implementação desses serviços é possível listarmos o tempo gasto no desenvolvimento exigido ao utilizar o OpenCV para a captura de frames em vídeo, que requer o *build* local da biblioteca FFmpeg, levando tempo e processamento computacional em cada pequena mudança.

Enquanto este desafio pode somente ser encontrado na etapa de desenvolvimento, outro desafio importante foi observado na construção dos serviços, que é o consumo de memória do sistema. Dado que estes serviços executam em um loop indeterminado de tempo, o consumo de memória RAM ser tornou um desafio visto que o processamento de imagens é uma ação custosa executada várias vezes seguidas. A solução pensada para este problema foi evitar o processamento desnecessário de frames antigos e inúteis dos dispositivos de captura.

## 5. CONCLUSÃO

Este trabalho de conclusão de curso foi produzido com o objetivo de automatizar, agilizar e persistir registros de parqueamentos, visando solucionar problemas identificados no processo de controle de acesso manual através do desenvolvimento de uma aplicação que ofereça essas funcionalidades em conjunto com uma interface de gerenciamento de veículos e condutores, oferecendo uma solução eficaz, prática e rápida.

Analisando os resultados obtidos desta pesquisa, é possível afirmar que os resultados foram alcançados. O trabalho realizado resultou em uma aplicação web baseada em um sistema existente de ALPR e desenvolvida usando uma linguagem de alta concorrência e disponibilidade capaz do registro automático de parqueamentos através do reconhecimento de placas veiculares por meio de um modelo de rede neural. O desenvolvimento deste sistema demonstrou a possibilidade da utilização de um modelo de rede neural em uma aplicação web *server-side* que, diferentemente de outras abordagens, permite seu *deploy* em um servidor remoto que não necessariamente exige de proximidade com os dispositivos de captura ou com o usuário.

O desenvolvimento do projeto não resultou somente no produto de software em si, mas também em lições importantes sobre a automação de processos utilizando redes neurais, a implementação dessas redes neurais utilizando tecnologias não convencionais para isso, como também a aquisição de expertise em relação à escopo de funcionalidades de um sistema.

Apesar do objetivo de desenvolvimento da aplicação alcançado, a implantação e testes inicialmente planejados a serem realizados no Campus Ji-Paraná do IFRO não puderam ser realizados devido a burocracias impostas pela instituição, obrigando uma mudança de planejamento para a pesquisa.

Desta forma é planejado para trabalhos futuros a implantação e utilização do sistema desenvolvido em cenários reais, possibilitando uma avaliação do sistema com dados não-fictícios e permitindo a real solução de problemas. Além disso, é planejado a implementação e o aprimoramento das funcionalidades do sistema, disponibilizando uma melhor interface do usuário que permita uma pesquisa e filtro da lista de parqueamentos ou veículos e condutores. O aperfeiçoamento no consumo de memória na captura de imagens também é desejado em uma próxima versão do sistema. Por fim, será pensado a possibilidade da expansão da utilização desta aplicação em outros estacionamentos além da instituição (IFRO, Campus Ji-Paraná).

## REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, William Pereira. **Banco de dados: teoria e desenvolvimento**. Saraiva Educação SA, 2009.

ARAÚJO, Cidália et al. **Estudo de Caso**. 2008. Disponível em: <<http://nelsonreyes.com.br/Estudo%20de%20Caso%20-%20Doutora%20Clara%20Pereira%20Coutinho.pdf>>. Acesso em: 07 nov. 2023.

BORDIGNON, Matheus Deon; SILVA, Rodolfo Adamshuk. **Taxonomia de Falhas em Programas Concorrentes em Elixir**. Anais do Computer on the Beach, v. 11, n. 1, p. 245-252, 2020. Disponível em <<https://periodicos.univali.br/index.php/acotb/article/view/16775/0>> Acesso em: 15 jul. 2022.

BORGES, Mateus Moury Fernandes da Rosa. **Uso de Elixir para a construção de um middleware baseado em RPC**. 2016. Tese de Doutorado. Universidade Federal de Pernambuco. Disponível em <<http://www.cin.ufpe.br/~tg/2016-1/mmfrb.pdf>> Acesso em: 16 nov. 2022.

CALDEIRA, Carlos. **PostgreSQL: Guia Fundamental**. 2015. Disponível em <<http://hdl.handle.net/10174/13327>> Acesso em: 16 nov. 2022.

CARNEIRO, José Luís. **Introdução a Banco de Dados**. 2004.

CARVALHO, André. **Redes Neurais Artificiais**. 2009. Disponível em: <<https://sites.icmc.usp.br/andre/research/neural/index.htm>>. Acesso em: 27 nov. 2023.

CARVALHO, Vinícius. **PostgreSQL: Banco de dados para aplicações web modernas**. Editora Casa do Código, 2017.

CECCON, Denny. Os tipos de redes neurais. **Expert Academy**, 2020. Disponível em: <<https://iaexpert.academy/2020/06/08/os-tipos-de-redes-neurais/>>. Acesso em: 27 nov. 2023.

DATE, Christopher J. **Introdução a sistemas de bancos de dados**. Elsevier Brasil, 2004.

DAVI, Tiago. **Elixir: Do zero à concorrência**. Casa do Código, 2017.

DOCKER. **Docker overview**. Disponível em: <<https://docs.docker.com/get-started/overview>>. Acesso em: 10 dez. 2023.

FERRARI, Fabricio; CECHINEL, Cristian. **Introdução a algoritmos e programação**. Bagé: Universidade Federal do Pampa, 2008.

FIGUEIRA, David Laureano dos Santos. **Sistema de Controle e Gerenciamento de Acesso**. 2023. Monografia - Curso de Faculdade de Tecnologia, Departamento de Eletrônica e Computação, Universidade Federal do Amazonas, Manaus, 2023.

FLECK, Leandro et al. **Redes Neurais Artificiais: Princípios Básicos**. Revista Eletrônica Científica Inovação e Tecnologia, Medianeira, v. 1, n. 13, p. 47-57, jan. 2016.

GOMES, Rafael. **Docker para desenvolvedores**. Leanpub. 2019.

GOTARDO, Reginaldo. **Linguagem de programação**. Rio de Janeiro: Sesus, p. 34, 2015.

LAROCA, Rayson, et al. **An Efficient and Layout-Independent Automatic License Plate Recognition System Based on the YOLO Detector**. IET Intelligent Transport Systems, 15, p. 483-503, 2021.

MACHADO, Felipe Nery Rodrigues. **Banco de Dados–Projeto e Implementação**. Saraiva Educação SA, 2020.

MCCORD, Chris. **How We Got to LiveView**. Fly.io, 2021. Disponível em: <<https://fly.io/blog/how-we-got-to-liveview>>. Acesso em: 07 nov. 2023.

OPENCV TEAM. **About**. Disponível em: <<https://opencv.org/about/>>. Acesso em: 07 nov. 2023.

OSÓRIO, Fernando; BITTENCOURT, João Ricardo. **Sistemas inteligentes baseados em redes neurais artificiais aplicados ao processamento de imagens**. Workshop de inteligência artificial, 2000. Disponível em <<https://osorio.wait4.org/oldsite/wia-unisc/wia2000-full.pdf>> Acesso em: 18 ago. 2023.

PHOENIX FRAMEWORK. **Overview**. HexDocs. Disponível em: <<https://hexdocs.pm/phoenix/overview.html>>. Acesso em: 16 nov. 2022.

PONNUSAMY, Arun. YOLO Object Detection with OpenCV and Python. **Towards Data Science**, 22, ago. 2018. Disponível em: <<https://towardsdatascience.com/yolo-object-detection-with-opencv-and-python-21e50ac599e9>>. Acesso em: 09 nov. 2023.

REDMON, Joseph. **Darknet: Open Source Neural Networks in C**. 2016. Disponível em: <<http://pjreddie.com/darknet/>>. Acesso em: 04 ago 2023.

REDMON, Joseph. **You Only Look Once: Unified, Real-Time Object Detection**. 2016. Disponível em: <<https://arxiv.org/abs/1506.02640>>. Acesso em: 04 ago. 2023.

REZENDE, Solange Oliveira. **Sistemas inteligentes: fundamentos e aplicações**. Editora Manole Ltda, 2003.

SILVA, Diogo Vinícius de S.; SANTOS, F. Alan de O.; SANTOS NETO, Pedro. **Os benefícios do uso de Kanban na gerência de projetos de manutenção de software**. 2012. Disponível em: <<https://sol.sbc.org.br/index.php/sbsi/article/view/14454>>. Acesso em: 07 nov. 2023.

SILVA, Fernando Léo Bueno de Oliveira. **Sistema de controle de acesso para estacionamento**. 2018. Disponível em: <<http://repositorio.unitau.br/jspui/handle/20.500.11874/4918>>. Acesso em: 15 jul. 2022.

SILVA, Geizane Lima da; BASTOS, Ana Clara Leal. **Introdução a Sistemas Inteligentes: Inteligência Artificial, Redes Neurais, Lógica Fuzzy e Aplicações**. In: XVI SEMANA DE MATEMÁTICA DA UESC, 2023, Ilhéus. SEMAT UESC, 2023.

SILVA, José Vitor Santos. **AtalaIA: Uso de Deep Learning para reconhecimento automático de placas de licença automotiva em dispositivos com recursos limitados**. 2022. Disponível em <<http://ri.ufs.br/jspui/handle/riufs/16257>>. Acesso em: 17 nov. 2022.

TEAMGENIK, **Teamgenik is made with Elixir and Phoenix LiveView**. c2023. Disponível em: <<https://teamgenik.com/docs/tech>>. Acesso em: 07 nov. de 2023.

YIN, Robert K. **Estudo de caso: planejamento e métodos**. 2. ed. Porto Alegre: Bookman, 2001. Disponível em: <<https://edisciplinas.usp.br/mod/resource/view.php?id=3878524>>. Acesso em: 07 nov. 2023.

## **APENDICE I – PROJETO DE SOFTWARE**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
RONDÔNIA – CAMPUS JI-PARANÁ**  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS

**PROJETO DO SISTEMA PARKING LOT**

Equipe: Wígnny Bora de Almeida

Ji-Paraná, 2023

Wígnny Bora de Almeida

## **PROJETO DO SISTEMA PARKING LOT 1.0**

Projeto de Software do sistema Parking Lot desenvolvido para obtenção de nota no TCC I e II do curso de Análise e Desenvolvimento de Sistema do IFRO Campus Ji-Paraná, sob a orientação do professor **Me. Jackson Henrique da Silva Bezerra.**

Ji-Paraná, 2023

## SUMÁRIO

1. DOCUMENTO DE ESPECIFICAÇÃO DO PROJETO.....	4
1.1. SOBRE O PRODUTO DE SOFTWARE.....	4
1.2. ASPECTOS TÉCNICOS DO PRODUTO .....	4
1.2.1. Tecnologias utilizadas .....	4
1.2.2. Ferramentas de desenvolvimento .....	4
1.2.3. Ferramentas de análise.....	4
1.3. EQUIPE DE DESENVOLVIMENTO .....	5
2. DOCUMENTO DE REQUISITOS .....	6
3. MODELAGEM DE CASOS DE USOS.....	13
3.1. DIAGRAMA DE CASO DE USO .....	13
3.2.1. Caso de Uso – Gerenciar condutores.....	14
3.2.2. Caso de Uso – Gerenciar veículos .....	15
3.2.3. Caso de Uso – Gerenciar associação entre veículos e condutores.....	16
3.2.4. Caso de Uso – Gerenciar dispositivos de captura.....	17
3.2.5. Caso de Uso – Consultar parqueamentos .....	18
3.2.6. Caso de Uso – Gerenciar marcas de veículos.....	19
3.2.7. Caso de Uso – Gerenciar modelos de veículos.....	20
4. PROJETO DE BANCO DE DADOS .....	22
4.1. MODELO LÓGICO .....	22
5. DIAGRAMA DE ATIVIDADES .....	23

## 1. DOCUMENTO DE ESPECIFICAÇÃO DO PROJETO

### 1.1. SOBRE O PRODUTO DE SOFTWARE

Será desenvolvido uma aplicação web para o registro automatizado do fluxo de veículos para estacionamentos. A aplicação poderá ser implantada por qualquer empresa/pessoa administradora de um estacionamento, embora tenha sido desenvolvido para atender as lacunas do Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO) - Campus Ji-Paraná em relação ao controle de acesso manual de veículos.

Na primeira iteração da aplicação, será apresentada com nome de **Parking Lot**, com o objetivo de realizar o registro automatizado de eventos de parqueamentos para todos os veículos conhecidos (previamente cadastrados), provendo uma interface simples para a checagem desses registros e cadastro de informações.

### 1.2. ASPECTOS TÉCNICOS DO PRODUTO

#### 1.2.1. Tecnologias utilizadas

O sistema será desenvolvido para plataforma web, de forma a ser executado em um servidor remoto, sendo sua interface do usuário acessível por qualquer navegador da web. Isso será atingido através do uso da linguagem de programação Elixir juntamente com o framework web Phoenix LiveView, permitindo o desenvolvimento de uma aplicação server-side.

Para persistência de dados, um banco de dados PostgreSQL será criado para a aplicação, que a partir de então realizará todas as operações de DDL através de *migrations*.

#### 1.2.2. Ferramentas de desenvolvimento

A construção da aplicação requererá nenhuma ferramenta específica para seu desenvolvimento quanto em relação à código, precisando somente de um editor de código, dessa forma, o Visual Studio Code foi a opção selecionada.

Já que a aplicação será executada em um servidor remoto, seu desenvolvimento deve levar em conta a plataforma de destino desde o início. Por isso, o Docker será utilizado, inserindo a aplicação em um container que será executado em uma máquina Linux no deploy.

#### 1.2.3. Ferramentas de análise

Durante a fase de análise da aplicação algumas ferramentas específicas serão utilizadas. Para a modelagem de diagramas (UML) a ferramenta open-source PlantUML será utilizada durante a elaboração dos diagramas de caso de usos e atividade, enquanto a ferramenta brModelo será usada para a elaboração dos diagramas de entidade-relacionamento (ERD) e modelo lógico de dados. Assim que definido os requisitos, mockups da interface serão criados utilizando a ferramenta proprietária Adobe XD.

### 1.3. EQUIPE DE DESENVOLVIMENTO

A equipe de desenvolvimento será composta por apenas um único desenvolvedor, Wígnny Almeida, que será encarregado do desenvolvimento do projeto de software, dos diagramas de UML, design da estrutura de tabelas do banco de dados e desenvolvimento da aplicação em si.

## 2. DOCUMENTO DE REQUISITOS

Durante a primeira etapa do projeto, é realizado o levantamento dos requisitos necessários para o desenvolvimento da aplicação. Esta etapa foi composta por capturar os requisitos funcionais e não-funcionais e representá-los de uma forma clara e simples através de dados classificados em tabelas. Para a aplicação em questão a ser desenvolvida, os seguintes requisitos foram encontrados:

**Quadro 1 - Requisito funcional - Cadastrar veículo**

<b>RF 1 Cadastrar veículo</b>			
<b>Descrição:</b> Permitir que o administrador cadastre um novo veículo.			
<b>Requisitos não funcionais</b>			
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Prioridade</b>
RNF 1.1 Cadastro de informações	Requerer ao usuário o preenchimento do formulário contendo as informações do veículo.	Usabilidade	Essencial
RNF 1.2 Validação das informações	Somente submeter o formulário se os dígitos da placa do veículo forem válidos.	Segurança	Essencial
RNF 1.3 Validação das informações	Somente cadastrar o veículo se o mesmo não estiver cadastrado anteriormente, impedindo duplicações.	Segurança	Essencial

**Fonte:** autoria própria

**Quadro 2 - Requisito funcional - Cadastrar condutor**

<b>RF 2 Cadastrar condutor</b>
--------------------------------

<b>Descrição:</b> Permitir que o administrador cadastre um novo condutor.			
<b>Requisitos não funcionais</b>			
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Prioridade</b>
RNF 2.1 Cadastro de informações	Requerer ao usuário o preenchimento do formulário contendo o nome, CPF e CNH.	Usabilidade	Essencial
RNF 2.2 Validação das informações	Somente submeter o formulário se os dígitos do CPF e CNH forem válidos.	Segurança	Essencial
RNF 2.3 Validação das informações	Somente cadastrar o condutor se o mesmo não estiver cadastrado anteriormente, impedindo duplicações.	Segurança	Essencial

**Fonte:** autoria própria

### **Quadro 3 - Requisito funcional - Associar veículo a condutor**

<b>RF 3 Associar veículo a condutor</b>			
<b>Descrição:</b> Permitir que o administrador associe um veículo a um condutor.			
<b>Requisitos não funcionais</b>			
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Prioridade</b>
RNF 3.1 Cadastro das informações	Requerer ao usuário selecionar um veículo e condutor existente para associá-los.	Usabilidade	Essencial
RNF 3.2 Validação das informações	Somente cadastrar a associação se não houver uma associação ativa existente, impedindo duplicações.	Segurança	Essencial

**Fonte:** autoria própria

#### Quadro 4 - Requisito funcional - Inabilitar veículo

RF 4 Desassociar veículo a condutor			
<b>Descrição:</b> Permitir que o administrador inabilite um veículo.			
Requisitos não funcionais			
Nome	Restrição	Categoria	Prioridade
RNF 4.1 Cadastro das informações	Requerer ao usuário selecionar um veículo para inabilitá-lo.	Usabilidade	Essencial
RNF 4.2 Validação das informações	Somente inabilitar um veículo atualmente ativo.	Segurança	Essencial
RNF 4.3 Validação das informações	Desabilitar um veículo através de <i>soft delete</i> , garantindo a possibilidade de consulta de associações anteriores.	Segurança	Essencial
RNF 4.4 Restrições de relacionamentos	Desabilitar a associação com o condutor ao desabilitar o veículo.	Segurança	Essencial

**Fonte:** autoria própria

#### Quadro 5 - Requisito funcional - Inabilitar condutor

RF 5 Desassociar veículo a condutor			
<b>Descrição:</b> Permitir que o administrador inabilite um condutor.			
Requisitos não funcionais			
Nome	Restrição	Categoria	Prioridade
RNF 5.1 Cadastro	Requerer ao usuário	Usabilidade	Essencial

das informações	selecionar um condutor para inabilitá-lo.		
RNF 5.2 Validação das informações	Somente inabilitar um condutor atualmente ativo.	Segurança	Essencial
RNF 5.3 Validação das informações	Desabilitar um condutor através de <i>soft delete</i> , garantindo a possibilidade de consulta de associações anteriores.	Segurança	Essencial
RNF 5.4 Restrições de relacionamentos	Desabilitar a associação com o veículo ao desabilitar o condutor.	Segurança	Essencial

**Fonte:** autoria própria

#### Quadro 6 - Requisito funcional - Desassociar veículo a condutor

RF 6 Desassociar veículo a condutor			
<b>Descrição:</b> Permitir que o administrador desassocie um veículo a um condutor.			
Requisitos não funcionais			
Nome	Restrição	Categoria	Prioridade
RNF 6.1 Cadastro das informações	Requerer ao usuário selecionar um veículo e condutor existente para desassociá-los.	Usabilidade	Essencial
RNF 6.2 Validação das informações	Somente remover uma associação já existente.	Segurança	Essencial
RNF 6.3 Validação das informações	Remover a associação através de <i>soft delete</i> , garantindo a	Segurança	Essencial

	possibilidade de consulta de associações anteriores.		
--	--	--	--

**Fonte:** autoria própria

### Quadro 7 - Requisito funcional – Cadastrar câmera

<b>RF 7 Cadastrar câmera</b>			
<b>Descrição:</b> Permitir que o administrador cadastre uma câmera.			
<b>Requisitos não funcionais</b>			
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Prioridade</b>
RNF 7.1 Cadastro das informações	Requerer ao usuário selecionar o tipo de câmera (de entrada ou saída), e seu link de transmissão.	Usabilidade	Essencial

### Quadro 8 - Requisito funcional - Registrar estacionamento

<b>RF 8 Registrar estacionamento</b>			
<b>Descrição:</b> Registrar automaticamente um estacionamento.			
<b>Requisitos não funcionais</b>			
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Prioridade</b>
RNF 8.1 Captura das informações	Processar imagem periodicamente obtida por um dispositivo de captura.	Usabilidade	Essencial
RNF 8.2 Captura das informações	Automaticamente obter dígitos da placa do veículo em imagem processada.	Usabilidade	Essencial

RNF 8.3 Validação das informações	Buscar veículo cadastrado usando dígitos da placa extraído da imagem.	Usabilidade	Essencial
RNF 8.4 Validação das informações	Somente registrar estacionamento se houver um veículo cadastrado para dada placa identificada.	Segurança	Essencial
RNF 8.5 Apresentação das informações	Notificar o usuário sobre veículo não identificado/cadastrado para dada placa.	Usabilidade	Desejável
RNF 8.6 Apresentação das informações	Notificar o usuário sobre o registro de estacionamento para dado veículo identificado.	Usabilidade	Essencial

**Fonte:** autoria própria

### Quadro 9 - Requisito funcional - Consultar parqueamentos

<b>RF 9 Registrar parqueamento</b>			
<b>Descrição:</b> Permitir que o usuário liste os veículos estacionados por filtro de data e hora.			
<b>Requisitos não funcionais</b>			
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Prioridade</b>
RNF 9.1 Filtragem das informações	Requisitar filtro de data e hora inicial final ao usuário para a busca dos dados.	Usabilidade	Desejável
RNF 9.2 Apresentação das informações	Listar veículos apresentando sua placa, nome do condutor, data de entrada e saída do veículo no estacionamento.	Usabilidade	Essencial
RNF 9.3 Apresentação das informações	Permitir impressão das informações apresentadas na	Usabilidade	Desejável

	listagem.		
--	-----------	--	--

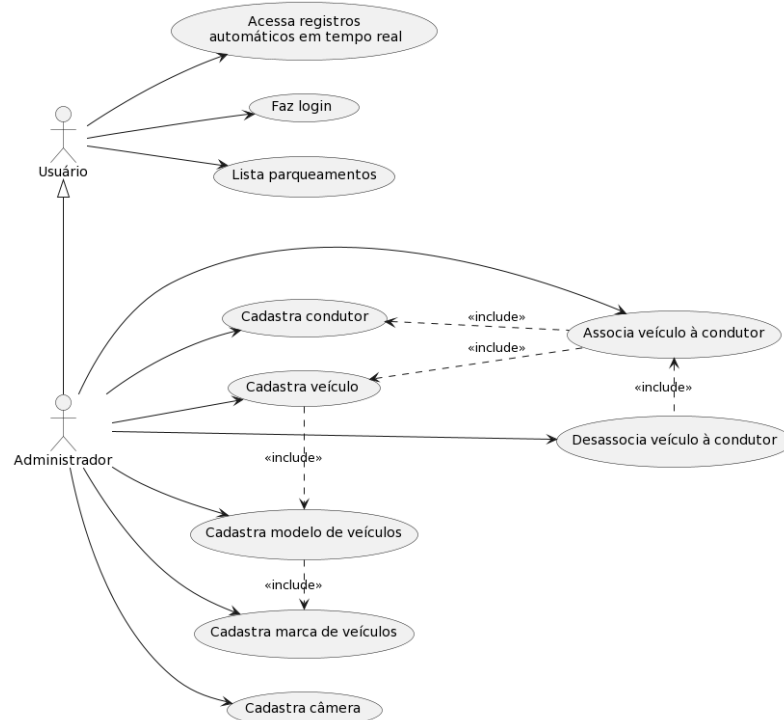
**Fonte:** autoria própria

### 3. MODELAGEM DE CASOS DE USOS

#### 3.1. DIAGRAMA DE CASO DE USO

Sendo o diagrama de caso de uso a parte da UML que documenta o que o sistema faz do ponto de vista do usuário, descrevendo as funcionalidades do sistema e a interação dessas funcionalidades com os usuários, as seguintes funcionalidades foram capturadas para o sistema a ser desenvolvido:

**Figura 1 - Diagrama de caso de uso**



**Fonte:** autoria própria

## 3.2. CASO DE USO EXPANDIDO

### 3.2.1. Caso de Uso – Gerenciar condutores

**Caso de Uso:** Consultar condutores

**Descrição:** O administrador pode consultar os condutores cadastrados

**Ator Primário:** Administrador

**Precondições:** O administrador deve estar logado no sistema

**Fluxo Principal:**

1. O administrador clica no menu “Motoristas”
2. A aplicação redireciona para uma página que lista os dados de todos os motoristas já cadastrados, incluindo ações de “Editar” e “Excluir” para cada motorista na listagem, como também apresenta o botão “Novo motorista”.

**Fluxo alternativo 1: Excluir condutor**

1. Após o passo 2 o administrador pode desejar excluir os dados de um motorista do sistema.
2. O administrador identifica o motorista a ser excluído.
3. O administrador clica no botão “Excluir”.
4. A aplicação solicita confirmação da exclusão.
5. O usuário clica em “Sim” confirmando a exclusão.

**Fluxo alternativo 2: Editar condutor**

1. Após o passo 2 o administrador pode desejar editar os dados de um motorista no sistema.
2. O administrador identifica o motorista a ser editado.
3. O administrador clica no botão “Editar”.
4. A aplicação redireciona para uma tela contendo campos de edição preenchidos com os dados atuais do motorista.
5. O usuário edita os campos desejados.
6. O usuário clica em “Salvar motorista”, confirmando a edição.
7. A aplicação redireciona para uma tela contendo os novos dados do usuário e uma mensagem da confirmação da edição.

**Fluxo alternativo 3: Criar condutor**

1. Após o passo 2 o administrador pode desejar inserir dados de um novo motorista no sistema.

2. O administrador clica no botão “Novo motorista”.
3. A aplicação redireciona para uma tela contendo campos de edição a serem preenchidos com os dados do motorista.
4. O usuário preenche os campos.
5. O usuário clica em “Salvar motorista”, confirmando a adição.
6. A aplicação redireciona para uma tela contendo os dados do usuário e uma mensagem da confirmação da adição.

### **3.2.2. Caso de Uso – Gerenciar veículos**

**Caso de Uso:** Consultar Veículos

**Descrição:** O administrador pode consultar os veículos cadastrados

**Ator Primário:** Administrador

**Precondições:** O administrador deve estar logado no sistema

**Fluxo Principal:**

1. O administrador clica no menu “Veículos”
2. A aplicação redireciona para uma página que lista os dados de todos os veículos já cadastrados, incluindo ações de “Editar” e “Excluir” para cada veículo na listagem, como também apresenta o botão “Novo veículo”.

**Fluxo alternativo 1: Excluir veículo**

1. Após o passo 2 o administrador pode desejar excluir os dados de um veículo do sistema.
2. O administrador identifica o veículo a ser excluído.
3. O administrador clica no botão “Excluir”.
4. A aplicação solicita confirmação da exclusão.
5. O usuário clica em “Sim” confirmando a exclusão.

**Fluxo alternativo 2: Editar veículo**

1. Após o passo 2 o administrador pode desejar editar os dados de um veículo no sistema.
2. O administrador identifica o veículo a ser editado.
3. O administrador clica no botão “Editar”.
4. A aplicação redireciona para uma tela contendo campos de edição preenchidos com os dados atuais do veículo.
5. O usuário edita os campos desejados.
6. O usuário clica em “Salvar veículo”, confirmando a edição.

7. A aplicação redireciona para uma tela contendo os novos dados do veículo e uma mensagem da confirmação da edição.

### **Fluxo alternativo 3: Criar veículo**

1. Após o passo 2 o administrador pode desejar inserir dados de um novo veículo no sistema.
2. O administrador clica no botão “Novo veículo”.
3. A aplicação redireciona para uma tela contendo campos de edição a serem preenchidos com os dados do veículo.
4. O usuário preenche os campos.
5. O usuário clica em “Salvar veículo”, confirmando a adição.
6. A aplicação redireciona para uma tela contendo os dados do veículo e uma mensagem da confirmação da adição.

### **3.2.3. Caso de Uso – Gerenciar associação entre veículos e condutores**

**Caso de Uso:** Consultar associação entre veículos e condutores

**Descrição:** O administrador pode consultar a associação entre veículos e condutores cadastrados.

**Ator Primário:** Administrador

**Precondições:** O administrador deve estar logado no sistema

**Fluxo Principal:**

1. O administrador clica no menu “Veículos/Motoristas”
2. A aplicação redireciona para uma página que lista a associação entre todos os veículos e condutores já cadastrados, incluindo ações de “Editar” e “Excluir” para cada associação na listagem, como também apresenta o botão “Novo veículo/motorista”.

**Fluxo alternativo 1: Excluir associação**

1. Após o passo 2 o administrador pode desejar excluir a associação entre um veículo e um condutor do sistema.
2. O administrador identifica a associação a ser excluída.
3. O administrador clica no botão “Excluir”.
4. A aplicação solicita confirmação da exclusão.
5. O usuário clica em “Sim” confirmando a exclusão.

**Fluxo alternativo 2: Editar associação**

1. Após o passo 2 o administrador pode desejar editar a associação entre um veículo e um condutor no sistema.
2. O administrador identifica a associação a ser editada.
3. O administrador clica no botão “Editar”.
4. A aplicação redireciona para uma tela contendo campos de edição preenchidos com os dados atuais da associação.
5. O usuário edita os campos desejados.
6. O usuário clica em “Salvar veículo/motorista”, confirmando a edição.
7. A aplicação redireciona para uma tela contendo os novos dados da associação e uma mensagem da confirmação da edição.

### **Fluxo alternativo 3: Criar associação**

1. Após o passo 2 o administrador pode desejar criar uma associação entre um veículo e um condutor no sistema.
2. O administrador clica no botão “Novo veículo/motorista”.
3. A aplicação redireciona para uma tela contendo campos de edição a serem preenchidos com os dados da associação.
4. O usuário preenche os campos.
5. O usuário clica em “Salvar veículo/motorista”, confirmando a adição.
6. A aplicação redireciona para uma tela contendo os dados da associação e uma mensagem da confirmação da adição.

### **3.2.4. Caso de Uso – Gerenciar dispositivos de captura**

**Caso de Uso:** Consultar câmeras

**Descrição:** O administrador pode consultar câmeras cadastradas

**Ator Primário:** Administrador

**Precondições:** O administrador deve estar logado no sistema

**Fluxo Principal:**

1. O administrador clica no menu “Câmeras”
2. A aplicação redireciona para uma página que lista as câmeras cadastradas para captura da entrada e saída de veículos, incluindo ações de “Editar” e “Excluir” para cada câmera na listagem, como também apresenta o botão “Nova câmera”.

**Fluxo alternativo 1: Excluir câmera**

1. Após o passo 2 o administrador pode desejar excluir uma câmera do sistema.

2. O administrador identifica a câmera a ser excluída.
3. O administrador clica no botão “Excluir”.
4. A aplicação solicita confirmação da exclusão.
5. O usuário clica em “Sim” confirmando a exclusão.

#### **Fluxo alternativo 2: Editar câmera**

1. Após o passo 2 o administrador pode desejar editar os dados de uma câmera no sistema.
2. O administrador identifica a câmera a ser editada.
3. O administrador clica no botão “Editar”.
4. A aplicação redireciona para uma tela contendo campos de edição preenchidos com os dados atuais da câmera.
5. O usuário edita os campos desejados.
6. O usuário clica em “Salvar câmera”, confirmando a edição.
7. A aplicação redireciona para uma tela contendo os novos dados da câmera e uma mensagem da confirmação da edição.

#### **Fluxo alternativo 3: Criar câmera**

1. Após o passo 2 o administrador pode desejar inserir dados de uma nova câmera no sistema.
2. O administrador clica no botão “Nova câmera”.
3. A aplicação redireciona para uma tela contendo campos de edição a serem preenchidos com os dados da câmera.
4. O usuário preenche os campos.
5. O usuário clica em “Salvar câmera”, confirmando a adição.
6. A aplicação redireciona para uma tela contendo os dados da câmera e uma mensagem da confirmação da adição.

### **3.2.5. Caso de Uso – Consultar parqueamentos**

**Caso de Uso:** Consultar parqueamentos

**Descrição:** O usuário pode listar os parqueamentos automaticamente registrados

**Ator Primário:** Usuário

**Precondições:** O usuário deve estar logado no sistema

**Fluxo Principal:**

1. O usuário clica no menu “Parqueamentos”.

2. A aplicação carrega uma tela listando todos os parqueamentos automaticamente registrados.

#### **Fluxo alternativo 1: Detalhar parqueamento**

1. Após o passo 2 o usuário pode desejar detalhar os dados de um parqueamento no sistema.
2. O administrador identifica o parqueamento a ser detalhado.
3. O administrador clica no botão “Detalhar”.
4. A aplicação redireciona para uma tela contendo informações sobre o parqueamento automaticamente registrado.

### **3.2.6. Caso de Uso – Gerenciar marcas de veículos**

**Caso de Uso:** Consultar marcas de veículos

**Descrição:** O administrador pode consultar marcas de veículos cadastradas

**Ator Primário:** Administrador

**Precondições:** O administrador deve estar logado no sistema

**Fluxo Principal:**

1. O administrador clica no menu “Marcas de Veículos”.
2. A aplicação redireciona para uma página que lista as marcas de veículos cadastradas, incluindo ações de “Editar” e “Excluir” para cada marca de veículo na listagem, como também apresenta o botão “Nova marca”.

**Fluxo alternativo 1: Excluir marca de veículo**

1. Após o passo 2 o administrador pode desejar excluir uma marca de veículo do sistema.
2. O administrador identifica a marca de veículo a ser excluída.
3. O administrador clica no botão “Excluir”.
4. A aplicação solicita confirmação da exclusão.
5. O usuário clica em “Sim” confirmando a exclusão.

**Fluxo alternativo 2: Editar marca de veículo**

1. Após o passo 2 o administrador pode desejar editar os dados de uma marca de veículo no sistema.
2. O administrador identifica a marca de veículo a ser editada.
3. O administrador clica no botão “Editar”.
4. A aplicação redireciona para uma tela contendo campos de edição preenchidos com os dados atuais da marca de veículo.

5. O usuário edita os campos desejados.
6. O usuário clica em “Salvar marca”, confirmando a edição.
7. A aplicação redireciona para uma tela contendo os novos dados da marca e uma mensagem da confirmação da edição.

#### **Fluxo alternativo 3: Criar marca de veículo**

1. Após o passo 2 o administrador pode desejar inserir dados de uma nova marca de veículo no sistema.
2. O administrador clica no botão “Nova marca”.
3. A aplicação redireciona para uma tela contendo campos de edição a serem preenchidos com os dados da marca de veículo.
4. O usuário preenche os campos.
5. O usuário clica em “Salvar marca”, confirmando a adição.
6. A aplicação redireciona para uma tela contendo os dados da marca e uma mensagem da confirmação da adição.

### **3.2.7. Caso de Uso – Gerenciar modelos de veículos**

**Caso de Uso:** Consultar modelos de veículos

**Descrição:** O administrador pode consultar modelos de veículos cadastradas

**Ator Primário:** Administrador

**Precondições:** O administrador deve estar logado no sistema

**Fluxo Principal:**

3. O administrador clica no menu “Modelos de Veículos”.
4. A aplicação redireciona para uma página que lista os modelos de veículos cadastrados, incluindo ações de “Editar” e “Excluir” para cada modelo de veículo na listagem, como também apresenta o botão “Novo modelos”.

#### **Fluxo alternativo 1: Excluir marca de veículo**

6. Após o passo 2 o administrador pode desejar excluir um modelo de veículo do sistema.
7. O administrador identifica o modelo de veículo a ser excluído.
8. O administrador clica no botão “Excluir”.
9. A aplicação solicita confirmação da exclusão.
10. O usuário clica em “Sim” confirmando a exclusão.

#### **Fluxo alternativo 2: Editar modelo de veículo**

8. Após o passo 2 o administrador pode desejar editar os dados de um modelo de veículo no sistema.
9. O administrador identifica o modelo de veículo a ser editado.
10. O administrador clica no botão “Editar”.
11. A aplicação redireciona para uma tela contendo campos de edição preenchidos com os dados atuais do modelo de veículo.
12. O usuário edita os campos desejados.
13. O usuário clica em “Salvar modelo”, confirmando a edição.
14. A aplicação redireciona para uma tela contendo os novos dados do modelo e uma mensagem da confirmação da edição.

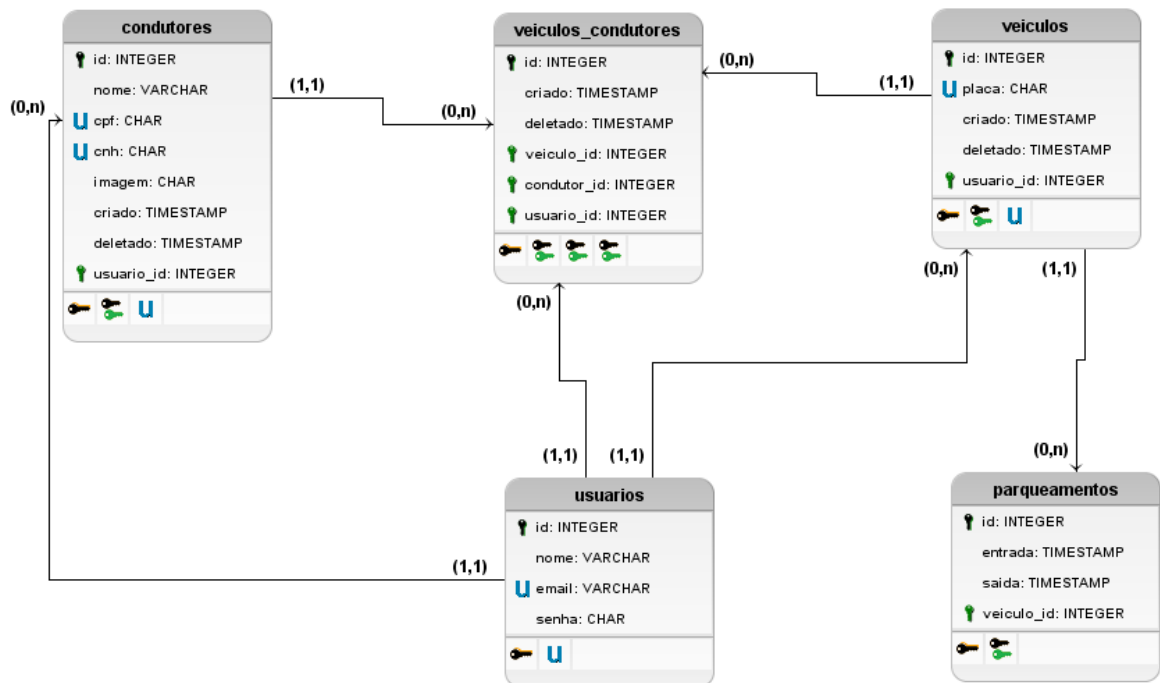
### **Fluxo alternativo 3: Criar modelo de veículo**

7. Após o passo 2 o administrador pode desejar inserir dados de um novo modelo de veículo no sistema.
8. O administrador clica no botão “Novo modelo”.
9. A aplicação redireciona para uma tela contendo campos de edição a serem preenchidos com os dados do modelo de veículo.
10. O usuário preenche os campos.
11. O usuário clica em “Salvar modelo”, confirmando a adição.
12. A aplicação redireciona para uma tela contendo os dados da marca e uma mensagem da confirmação da adição.

## 4. PROJETO DE BANCO DE DADOS

### 4.1. MODELO LÓGICO

Figura 2 - Diagrama do Modelo Lógico

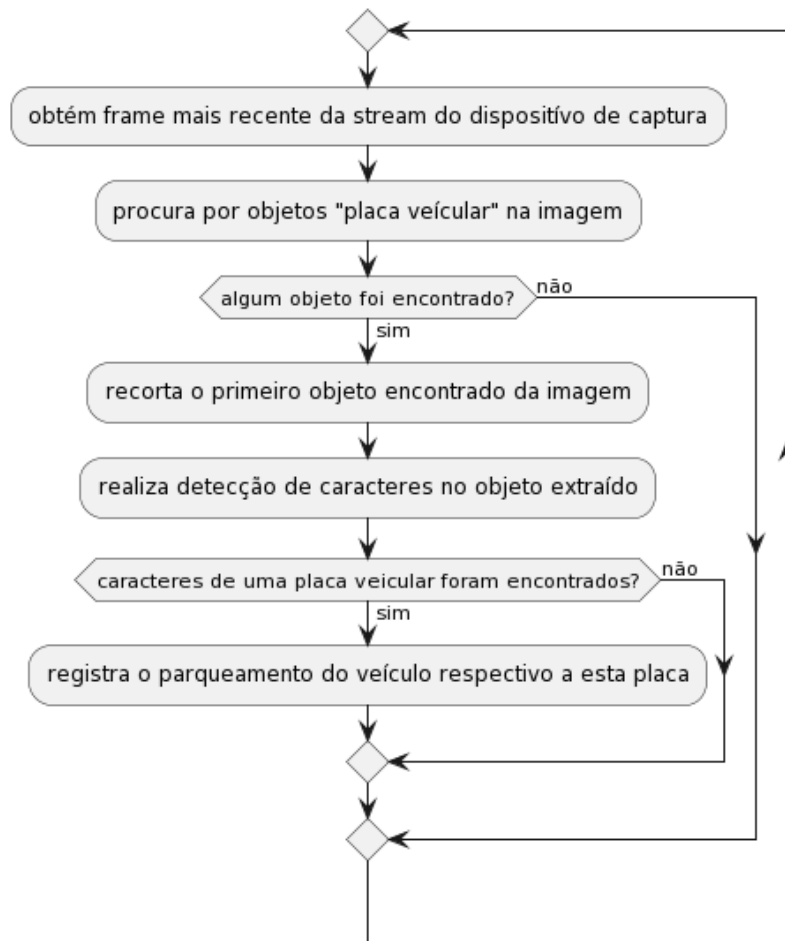


Fonte: autoria própria

## 5. DIAGRAMA DE ATIVIDADES

O diagrama de atividade é definido como parte da UML responsável pela modelagem de ações/atividades como também o fluxo de controle entre elas. Ele pode ser utilizado para descrever processos de negócio, fluxos de controle de casos de uso, como também a estrutura algorítmica de funções integrantes de atividades de um sistema. Para o sistema a ser desenvolvido neste trabalho, o seguinte diagrama de atividade foi desenhado para representar a lógica de ALPR utilizando os dispositivos de captura:

**Figura 3 - Diagrama de atividade: registro automático de estacionamento**

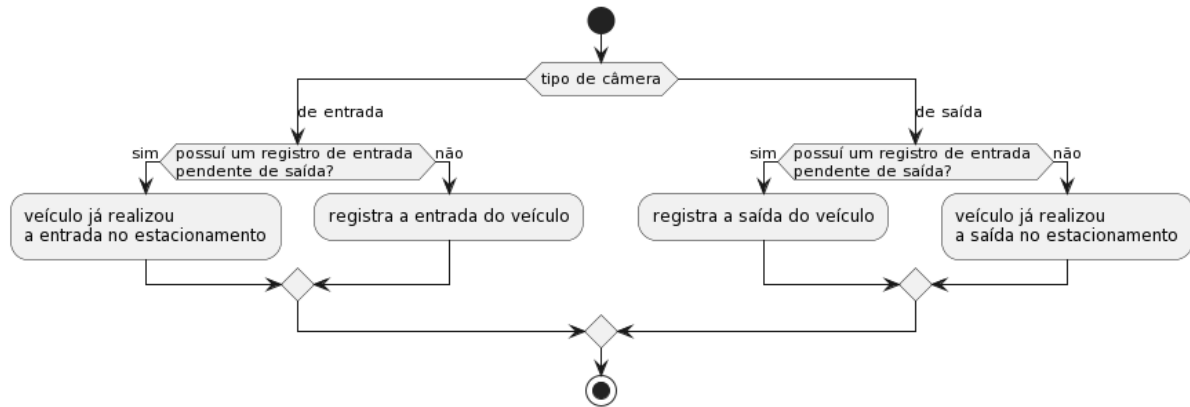


**Fonte:** autoria própria

Nesta lógica, o processo de captura, identificação da placa e registro da entrada ou saída do veículo estará em repetição por tempo indeterminado, permitindo assim a checagem intermitente de veículos nos portões do instituto a serem autenticados.

Dessa forma, para que seja possível a identificação e registro tanto da entrada como saída de veículos, será necessário a utilização de dois dispositivos de captura estrategicamente posicionados em diferentes direções, ambos utilizando a mesma lógica apresentada no diagrama de atividade, mas executando em diferentes processos (*threads*).

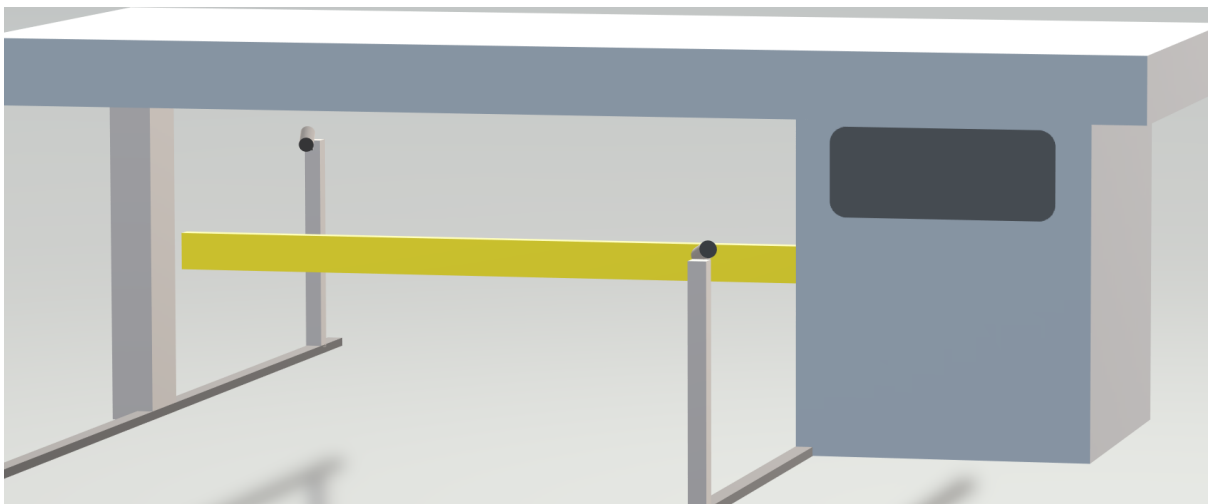
**Figura 4 – Lógica do registro de estacionamento para diferentes câmeras**



**Fonte:** autoria própria

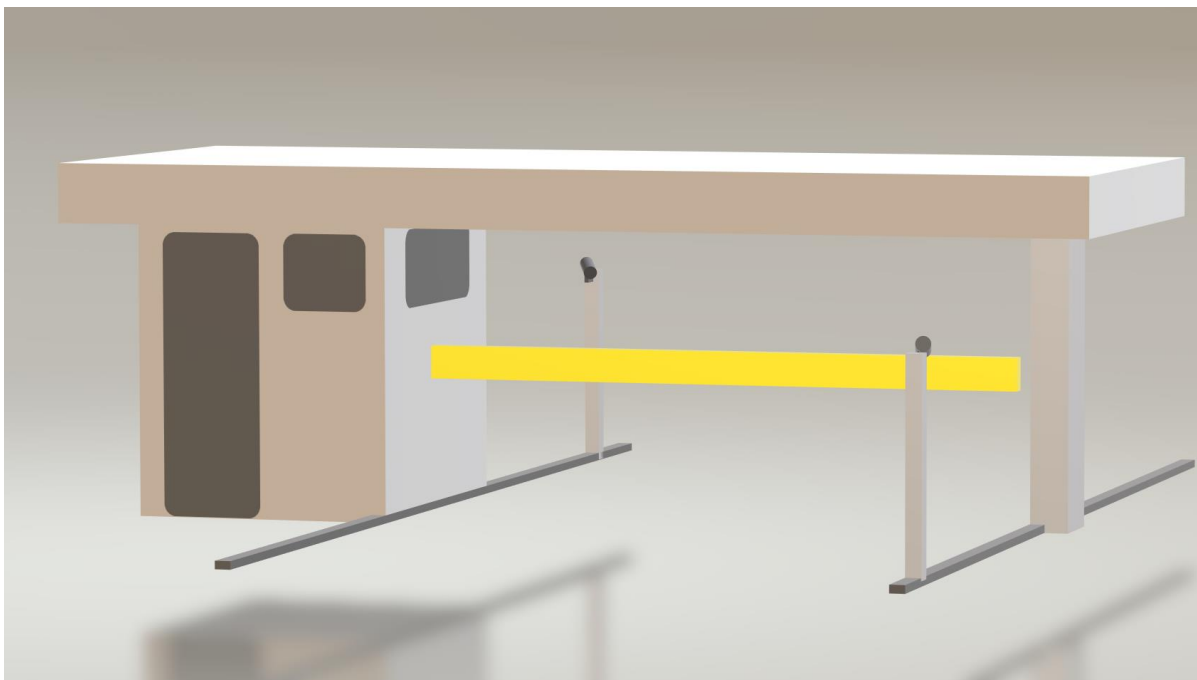
Utilizando esta estratégia, a instalação dos dispositivos ocorrerá de forma que o primeiro dispositivo deverá ser instalado voltado para o exterior do estacionamento, enquanto o segundo dispositivo será instalado voltado para o interior do estacionamento. As seguintes imagens exemplificam a instalação dos dispositivos:

**Figura 5 - Exemplo de guarita equipada com dispositivos de captura (exterior)**



**Fonte:** autoria própria

**Figura 6 -- Exemplo de guarita equipada com dispositivos de captura (interior)**



**Fonte:** autoria própria

Neste cenário, o dispositivo voltado para o exterior da guarita ficará responsável somente pelo registro de entrada de veículos, enquanto o dispositivo voltado para o interior, ficará responsável somente pelo registro de saída de veículos.