

**UNIVERSIDADE PAULISTA – UNIP**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**BRUNO DA SILVA PIMENTEL**

**GUILHERME VIEIRA ABBENANTE GOMES**

**HITALO CHAVES DOS SANTOS**

**JOÃO VICTOR CRISCI**

**DETECÇÃO AUTOMATIZADA DE INFRAÇÕES DE  
ESTACIONAMENTO UTILIZANDO VISÃO COMPUTACIONAL E  
INTELIGÊNCIA ARTIFICIAL**

**SÃO PAULO**

**2025**

**UNIVERSIDADE PAULISTA – UNIP**

**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**BRUNO DA SILVA PIMENTEL**

**GUILHERME VIEIRA ABBENANTE GOMES**

**HITALO CHAVES DOS SANTOS**

**JOÃO VICTOR CRISCI**

**DETECÇÃO AUTOMATIZADA DE INFRAÇÕES DE  
ESTACIONAMENTO UTILIZANDO VISÃO COMPUTACIONAL E  
INTELIGÊNCIA ARTIFICIAL**

Trabalho de conclusão de curso como  
requisito parcial para obtenção do grau de  
bacharel em Ciência da Computação  
apresentado à Universidade Paulista – UNIP.

Orientador: Prof. MSc. Álvaro A Colombero  
Prado.

**SÃO PAULO - SP**

**2025**

**BRUNO DA SILVA PIMENTEL**

**GUILHERME VIEIRA ABBENANTE GOMES**

**HITALO CHAVES DOS SANTOS**

**JOÃO VICTOR CRISCI**

**DETECÇÃO AUTOMATIZADA DE INFRAÇÕES DE  
ESTACIONAMENTO UTILIZANDO VISÃO COMPUTACIONAL E  
INTELIGÊNCIA ARTIFICIAL**

Trabalho de conclusão de curso como  
requisito parcial para obtenção do grau de  
bacharel em Ciência da Computação  
apresentado à Universidade Paulista – UNIP.

Aprovado em:

**BANCA EXAMINADORA**

\_\_\_\_\_  
/ / \_\_\_\_\_

Prof. Universidade Paulista – UNIP

\_\_\_\_\_  
/ / \_\_\_\_\_

Prof. Universidade Paulista – UNIP

\_\_\_\_\_  
/ / \_\_\_\_\_

Prof. Universidade Paulista – UNIP

# FICHA CATALOGRÁFICA

CIP - Catalogação na Publicação

DETECÇÃO AUTOMATIZADA DE INFRAÇÕES DE ESTACIONAMENTO  
UTILIZANDO VISÃO COMPUTACIONAL E  
INTELIGÊNCIA ARTIFICIAL / Bruno Pimentel...[et al.]. - 2025.  
124 f. : il. color

Trabalho de Conclusão de Curso (Graduação) apresentado ao Instituto  
de Ciência Exatas e Tecnologia da Universidade Paulista, São Paulo,  
2025.

Área de Concentração: Ciência da Computação  
Orientador: Prof. Me. Álvaro Prado.

1. Inteligência Artificial. 2. Infração. 3. Estacionamento irregular. 4.  
Visão Computacional. 5. Detecção Automatizada. I. Pimentel, Bruno. II.  
Prado, Álvaro (orientador).

## **RESUMO**

O estacionamento irregular é um problema crescente nos centros urbanos, impactando diretamente a mobilidade, a acessibilidade e a segurança viária. Veículos parados em locais proibidos prejudicam o fluxo do trânsito, aumentam o risco de acidentes e dificultam a locomoção de pedestres, especialmente pessoas com deficiência ou mobilidade reduzida. Embora as leis de trânsito estabeleçam regras claras para coibir essas infrações, a fiscalização tradicional enfrenta desafios como a limitação de recursos humanos, a cobertura ineficiente e a dificuldade de monitoramento contínuo, tornando o problema persistente e de difícil controle. Diante desse cenário, este trabalho propõe uma solução inovadora baseada em Visão Computacional e Inteligência Artificial para a detecção automatizada de infrações de estacionamento. O sistema desenvolvido utiliza técnicas avançadas de processamento de imagens e aprendizado de máquina para identificar veículos estacionados de forma irregular em áreas proibidas, como faixas de pedestres, calçadas (passeios) e em locais definidos pela sinalização. Além de automatizar o processo de fiscalização, a solução visa aumentar a eficiência na identificação de infrações, reduzir a recorrência de estacionamentos irregulares e auxiliar na formulação de políticas públicas voltadas para a mobilidade urbana. Para validar a abordagem proposta, serão conduzidos testes em diferentes cenários urbanos, analisando a precisão e a confiabilidade da detecção automatizada em comparação com os métodos tradicionais de fiscalização.

**Palavras-Chave:** Infração; Estacionamento irregular; Visão Computacional; Inteligência Artificial; Detecção Automatizada.

## **ABSTRACT**

Irregular parking is a growing problem in urban centers, directly impacting mobility, accessibility, and road safety. Vehicles parked in prohibited areas disrupt traffic flow, increase the risk of accidents, and impede pedestrian movement, especially for people with disabilities or reduced mobility. Although traffic laws establish clear rules to curb these violations, traditional enforcement faces challenges such as limited human resources, inefficient coverage, and difficulties in continuous monitoring, rendering the problem persistent and hard to control. In this context, this work proposes an innovative solution based on Computer Vision and Artificial Intelligence for the automated detection of parking infractions. The developed system employs advanced image-processing techniques and machine learning algorithms to identify vehicles parked irregularly in regulated and prohibited areas, such as crosswalks (pedestrian crossings), sidewalks, and locations defined by signage. Beyond automating the enforcement process, the solution aims to enhance infraction-detection efficiency, reduce the recurrence of irregular parking, and support the formulation of public policies focused on urban mobility. To validate the proposed approach, tests will be conducted across diverse urban scenarios to analyze the precision and reliability of the automated detection in comparison with traditional enforcement methods.

**Keywords:** Infraction; Irregular Parking; Computer Vision; Artificial Intelligence; Automated Detection.

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>11</b>
1.1 Contextualização .....	12
1.2 Problema .....	13
1.3 Objetivo .....	14
1.3.1 Objetivo Geral .....	14
1.3.2 Objetivos Específicos .....	14
1.4 Motivação .....	16
1.5 Organização do projeto .....	18
<b>2 REFERENCIAL TEÓRICO.....</b>	<b>20</b>
2.1 Inteligência Artificial.....	20
2.1.1 Redes Neurais Artificiais.....	21
2.2 Visão computacional .....	24
2.2.1 Captura de imagem.....	24
2.2.2 You Only Live Once (Yolo) .....	25
2.2.3 Detecção de Veículos e Áreas de Estacionamento .....	26
2.2.4 Automatic License Plate Recognition (ALPR) .....	27
2.2.5 Métricas de Avaliação e Conceitos de Treinamento .....	28
2.2.5.1 Métricas de Desempenho .....	28
2.2.5.2 Componentes de Detecção e Funções de Perda ( <i>Loss Functions</i> ) .....	29
2.2.5.3 Hiperparâmetros e Fenômenos de Treinamento .....	29
2.3 Regulamentação Brasileira.....	31
2.3.1 Código de Trânsito Brasileiro .....	31
2.3.1.1 Estacionamento .....	31
2.3.1.2 Infrações .....	32
2.3.1.3 Infrações por Estacionamento de Veículos .....	32
2.3.1.4 Sinalização.....	32
<b>3 TRABALHOS RELACIONADOS .....</b>	<b>34</b>
3.1 Convolutional Neural Network for Person and Car Detection using YOLO Framework.....	34
3.2 Computer vision in automated parking systems: Design, implementation and challenges ..	34
3.3 Real-Time Illegal Parking Detection Algorithm in Urban Environments .....	34
3.4 Vision-Based Parking-Slot Detection: A Benchmark and A Learning-Based Approach .....	35
3.5 Vacant Parking Slot Detection in the Around View Image Based on Deep Learning .....	35
3.6 Automatic detection of long-term parked cars.....	35

3.7 Automatic Detection of Parking Violation and Capture of License Plate .....	36
3.8 Detection of Illegal Parking Based on Deep Learning.....	36
3.9 Deep Learning-based ALPR system for vehicle access control trained on brazilian license plates.....	36
3.10 An Embedded Automatic License Plate Recognition System using Deep Learning .....	36
3.11 A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector ..	37
3.12 Considerações Finais .....	37
<b>4 PROJETO DE SOFTWARE.....</b>	<b>39</b>
4.1 Documento de Visão .....	40
4.1.1 Nome do projeto .....	40
4.1.2 Objetivo Geral .....	40
4.1.3 Justificativa .....	40
4.1.4 Objetivos Específicos .....	40
4.1.5 Escopo do Projeto .....	41
4.1.6 Stakeholders Acadêmicos .....	41
4.1.7 Resultados Esperados .....	41
4.1.8 Arquitetura do sistema e ferramentas utilizadas.....	42
4.1.9 Tecnologias Sugeridas.....	42
4.1.10 Restrições e Considerações .....	42
4.2 Documento de requisitos (funcionais e não funcionais).....	43
4.2.1 Requisitos Funcionais (Casos de uso) .....	43
4.2.1.1 [UC01] Cadastrar usuário .....	43
4.2.1.2 [UC02] Fazer login do usuário .....	43
4.2.1.3 [UC03] Enviar Arquivo .....	43
4.2.1.4 [UC04] Consultar infrações.....	43
4.2.1.5 [UC05] Detectar elementos desejados .....	43
4.2.1.6 [UC06] Validar infração de estacionamento .....	43
4.2.1.7 [UC07] Classificar o tipo e a gravidade da infração.....	43
4.2.1.8 [UC08] Extrair dados da infração .....	44
4.2.1.9 [UC09] Armazenar os dados.....	44
4.2.1.10 [UC10] Notificar o usuário que uma infração foi cometida .....	44
4.2.2 Requisitos Não Funcionais .....	45
4.2.2.1 Estabilidade Funcional .....	45
4.2.2.2 Eficiência de Desempenho .....	45
4.2.2.3 Compatibilidade.....	45
4.2.2.4 Usabilidade.....	45

4.2.2.5 Confiabilidade.....	46
4.2.2.6. Segurança.....	46
4.2.2.7 Manutenibilidade .....	46
4.2.2.8 Portabilidade.....	47
4.3 Arquitetura de Solução .....	48
4.4 Projeto lógico do banco de dados .....	49
4.5 Projeto físico do banco de dados .....	51
4.6 Modelagem do Software .....	53
4.6.1 Diagrama de Caso de Uso.....	54
4.6.2 Documentação de caso de uso.....	56
4.6.3 Diagrama de Classe .....	66
4.6.4 Diagramas de Sequência (DS).....	68
4.6.4.1 DS Cadastro .....	68
4.6.4.2 DS Login.....	70
4.6.4.3 DS Enviar Arquivo .....	72
4.6.4.4 DS Consultar Infrações .....	73
4.6.4.5 DS Fluxo completo de detecção.....	75
4.6.4.6 DS Notificação .....	77
<b>5. PROTÓTIPO DO PROJETO.....</b>	<b>78</b>
5.1 Tela inicial.....	78
5.2 Tela de login.....	78
5.3 Tela de cadastro.....	79
5.4 Tela inicial após autenticação (fluxo de processamento) .....	79
5.4.1 Tela inicial após autenticação (funcionalidades e métricas).....	80
5.5 Tela de consultar infrações (vazia) .....	80
5.5.1 Tela de consultar infrações onde placa não encontrada .....	81
5.5.2 Tela de consultar infrações com ocorrências encontradas .....	81
5.6 Tela com o modal da infração .....	82
5.7 Tela de validação de Infrações (vazia) .....	82
5.7.1 Tela de validação de Infrações (tutorial) .....	83
5.7.2 Tela de validação com uma imagem para processar.....	83
5.7.3 Tela de validação com uma imagem de infração processada .....	84
5.7.4 Tela de validação com uma imagem sem infrações detectadas .....	84
5.7.5 Tela de validação com uma imagem sem veículos detectados.....	85
5.8 Geolocalização das Infrações (mapa) .....	85

5.8.1 Geolocalização das Infrações (mapa com modais).....	86
5.9 Tela de Perfil .....	86
5.10 Tela de infrações enviadas.....	87
5.10.1 Tela de infrações enviadas (nenhuma encontrada) .....	87
<b>6. CRONOGRAMA PARA TC-II .....</b>	<b>88</b>
<b>7 TESTES .....</b>	<b>89</b>
7.1 Aquisição e Cura do Dataset.....	89
7.2 Estratégias de Aquisição de Imagens e Anotação .....	89
7.2.1 Detalhamento Operacional da Criação e Segmentação do Dataset .....	93
7.3 Processo de Treinamento do Modelo .....	95
7.4 Implementação e Validação do Modelo .....	96
7.5 Gráficos de métricas do treinamento do modelo .....	96
7.6 Testes e Implementação da validação de Infrações.....	104
7.7 Testes de Interface Gráfica do Usuário (GUI) e Usabilidade.....	105
<b>8 RESULTADOS.....</b>	<b>107</b>
8.1 Detecção em imagens de casos reais .....	107
8.2 Validação de infrações em imagens de casos reais .....	111
8.2.1 Veículo estacionado sobre faixa de pedestre.....	112
8.2.2 Veículo estacionado sobre calçada (passeio).....	113
8.2.3 Veículo em locais proibidos pela sinalização (placa - Proibido Estacionar) .....	114
8.2.4 Casos especiais (múltiplas infrações/múltiplos veículos infratores) .....	115
8.2.5 Veículo estacionado de forma regular .....	116
<b>9 CONCLUSÃO .....</b>	<b>117</b>
9.1 Retomada do Trabalho e Solução Proposta.....	117
9.2 Principais Resultados e Contribuições.....	117
9.3 Limitações e Desafios Enfrentados .....	118
9.4 Trabalhos Futuros .....	118
<b>10 REFERÊNCIAS BIBLIOGRAFICAS.....</b>	<b>119</b>

## 1 INTRODUÇÃO

O presente trabalho visa criar um sistema automático para monitorar estacionamentos irregulares, empregando visão computacional e Inteligência Artificial (IA). O objetivo é aumentar a eficácia no controle de transgressões na cidade, diminuir a necessidade de fiscalização humana e auxiliar numa gestão de trânsito mais inteligente e ecologicamente correta.

A solução planejada une tecnologias como YOLOv11 (*You Only Look Once*) para identificar veículos instantaneamente, técnicas com base no ALPR (*Automatic License Plate Recognition*) (Reconhecimento Automático de Placas) para ler placas e redes neurais complexas para classificar infrações segundo o Código de Trânsito Brasileiro (CTB). O sistema examina fotos de câmeras urbanas ou de usuários, detectando carros parados em locais proibidos (faixas de pedestres, calçadas, dentre outras), anotando suas placas e armazenando informações cruciais para futura análise (características dos veículos, data, hora e local da infração, etc.).

Para assegurar que o modelo funcione bem, serão usados algoritmos para preparar as imagens (como ajustar brilho e contraste) e técnicas de aprendizado de máquina supervisionado, ensinando a IA com várias situações reais. O sistema também possui uma interface fácil de usar, onde as pessoas podem enviar denúncias, verificar infrações por placa e ver mapa de ocorrências com endereços e horários, ajudando os órgãos de trânsito a decidirem.

Serão feitas análises com situações reais, simulando denúncias de usuários, passando pelo processamento de imagem (detecção de elementos e validação de infrações) que trazem como resultado métricas de confiabilidade do modelo de detecção, como também a visualização de dados conclusivos sobre a ocorrência capturada.

Sendo assim a aplicação pretende ser mais do que uma alternativa à fiscalização humana, mas uma ferramenta para diminuir a impunidade e criar cidades mais organizadas e seguras. Este trabalho ajudará a utilizar a IA no transporte urbano, atendendo às necessidades por tecnologias novas e sustentáveis na administração pública.

## 1.1 Contextualização

À medida que as cidades se desenvolvem, as rodovias se estendem e devido a melhora da qualidade dos veículos, carros e motos ganham preferência como transporte, criando problemas na mobilidade urbana. Um desses problemas é o aumento das infrações por estacionamento irregular, cada vez mais comuns em áreas urbanas com pouca fiscalização.

Segundo dados do Portal do Trânsito, em 2023, foram registradas 2,1 milhões de autuações por estacionamento irregular no Brasil, evidenciando um aumento expressivo dessa infração nos últimos anos.

Em território nacional, o Código de Trânsito Brasileiro (CTB) define um conjunto de regras para assegurar a organização e a proteção nas vias. Dentre essas regras, estão as que abordam o estacionamento indevido, que ocorre quando o condutor para o veículo em um local onde a lei ou a sinalização não permite.

A falta de espaço adequado para o estacionamento faz com que veículos sejam estacionados em locais proibidos e de maneiras incorretas, de acordo com o CTB, nos artigos 181 e 182 do capítulo XV, existem um total de 31 infrações relacionadas ao estacionamento de veículos, por exemplo: onde houver guia de calçada (meio-fio) rebaixada destinada à entrada ou saída de veículos; na área de cruzamento de vias, prejudicando a circulação de veículos e pedestres; nas vagas reservadas às pessoas com deficiência ou idosos, sem credencial que comprove tal condição; em local e horário proibidos especificamente pela sinalização (placa - Proibido Parar) dentre outras. Dessa forma comprometendo a fluidez do trânsito, gerando transtornos e colocando em risco a segurança e a acessibilidade da população.

Anteriormente, a supervisão de infrações era feita unicamente por um guarda de trânsito no local, o que exigia tempo e dinheiro, restringindo-se ao espaço e às horas de trabalho do agente.

Segundo a Prefeitura de Piracicaba (2022), "a prefeitura investiu R\$ 269.890,00 para capacitar 274 guardas civis municipais como agentes de fiscalização de trânsito.

Nesse contexto, o uso de novas tecnologias, como a visão computacional e a inteligência artificial, tem surgido como um forte apoio para apresentar formas automáticas de identificar infrações de estacionamento.

## 1.2 Problema

Garantir a mobilidade nas áreas urbanas é vital para o bem-estar diário, impactando diretamente a organização e a segurança do trânsito. Conforme apontado por VASCONCELLOS (2001), a adoção de uma mobilidade urbana sustentável desempenha papel crucial na melhoria da qualidade de vida nas cidades, ao minimizar os congestionamentos, diminuir a poluição, encurtar os tempos de deslocamento e favorecer a inclusão social.

No contexto contemporâneo, a supervisão de infrações ligadas ao estacionamento ilegal ainda se baseia majoritariamente no monitoramento manual. Isso demanda a presença constante de agentes fiscais e acarreta diversas restrições, tais como a vulnerabilidade a erros humanos, falhas na identificação e no registro das infrações, resultando em uma resposta tardia às irregularidades. Tais erros acabam permeando uma sensação de descaso com a fiscalização, que, por sua vez, influencia o aumento do estacionamento em locais inadequados.

Conforme observado por RUSSO (2012), a impunidade é um dos principais fatores que colabora e alimenta a violência e irregularidades no trânsito. Devido a ineficiência e dificuldade do Estado em penalizar os infratores, juntamente com a percepção desse problema pelos motoristas, estes tendem a desafiar e testar cada vez mais a eficiência da fiscalização, além de replicar comportamentos ilegais, em princípio quando veem outros motoristas se aproveitando dessas ausências.

Tal concepção é reforçada pela análise de que estacionar em locais proibidos está entre as infrações de trânsito mais comuns em áreas urbanas no Brasil, contexto agravado devido à dificuldade de sistemas de fiscalizações eficientes, juntamente com a percepção de baixos casos de punição entre os condutores. Assim, tais problemas contribuem para a manutenção destas práticas, ao invés do desencorajamento dos motoristas (MARTINS 2020, p. 112).

Nesse sentido, o DENATRAN (2019) destaca que “A fiscalização ostensiva e contínua é essencial para a mudança de comportamento no trânsito. Na ausência dela, os índices de infrações tendem a crescer, comprometendo a segurança viária.”

Considerando a situação atual, fica claro a necessidade de um método aprimorado, que consiga detectar, de maneira precisa e constante, as violações ligadas ao estacionamento irregular. Assim, seria viável acelerar a resposta das fiscalizações de trânsito e, consequentemente, desestimular os motoristas a cometerem esse tipo de infração.

### **1.3 Objetivo**

#### **1.3.1 Objetivo Geral**

Desenvolver um sistema automatizado baseado em inteligência artificial e visão computacional, para detecção e fiscalização de infrações de estacionamento em áreas urbanas. Otimizando a gestão do espaço urbano e contribuindo para a fluidez do trânsito ao reduzir a ocorrência de infrações relacionadas ao estacionamento por meio de uma atuação mais precisa e dinâmica.

#### **1.3.2 Objetivos Específicos**

- Implementar através de algoritmos de visão computacional (como YOLOv11) a habilidade de analisar, identificar e capturar automaticamente, através de imagens quando um veículo está cometendo uma infração de estacionamento;
- Implementar no sistema a capacidade de classificar o tipo de infração cometida pelo veículo, baseada em regras pré-definidas do algoritmo baseadas nos artigos do CTB;
- Implementar a identificação de características como numeração da placa e cor do veículo, quando possível;
- Implementar a identificação de características gerais como local, data e hora da infração;
- Armazenar os dados registrados em um banco de dados, para futura análise e consulta das infrações cometidas;
- Produzir uma interface, conectado ao sistema, que permita ao usuário:

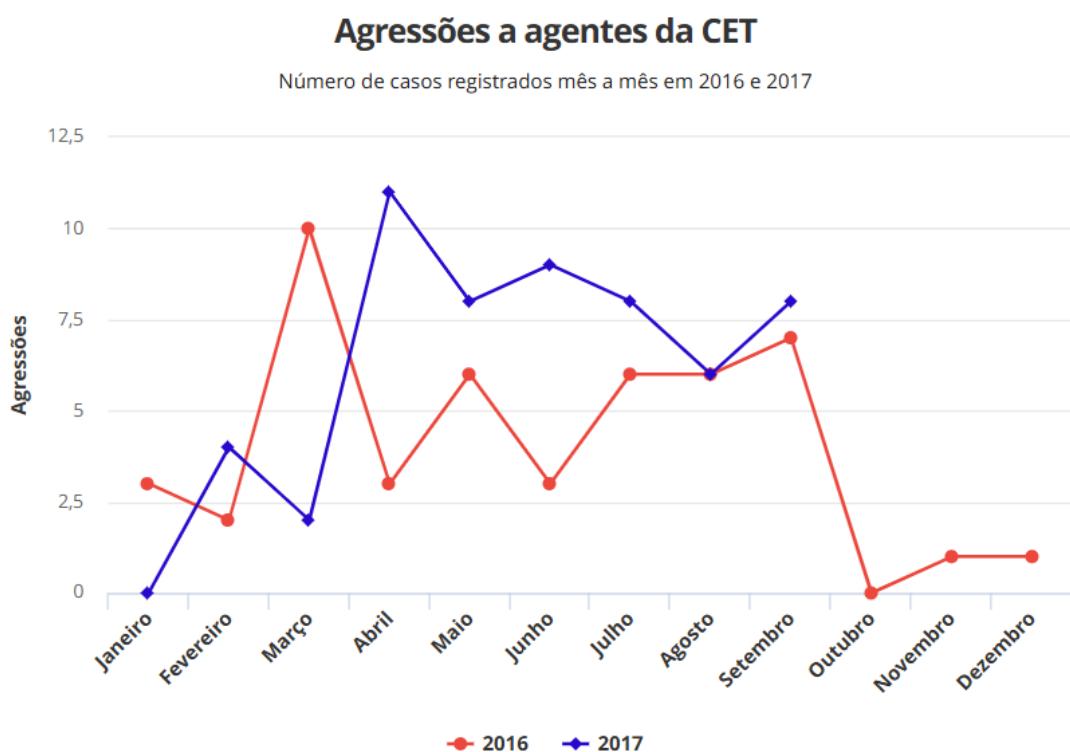
- A validação direta de infrações de estacionamento através do upload de foto;
- Consulta das infrações cometidas através da pesquisa por placa do veículo;
- Visualização de mapa que contém a análise de características gerais de infrações, como local, data, hora etc.

## 1.4 Motivação

A motivação por trás da criação de um sistema que identifica automaticamente irregularidades no estacionamento veio da constatação de uma dificuldade corriqueira, notada em quase todos os centros urbanos, veículos parados em locais proibidos, impedindo o uso de calçadas, áreas de travessia ou utilizando espaços destinados a outros fins. Essas infrações, mesmo que vista como pequenas por muitos, causam consequências alarmantes na organização do tráfego, na proteção dos pedestres e na velocidade com que os veículos se movem pelas ruas.

Além disso, pode-se perceber que a forma como a fiscalização é feita hoje, principalmente por indivíduos, assim tendo vários pontos fracos, como precisar de fiscais nas ruas, assim muitas vezes colocando-os em perigo devido a abordagem direta a motoristas infratores dessa forma gerando possíveis confrontos com alguns cidadãos que reagem de maneira agressiva diante da autuação. (Gráfico 1)

Gráfico 1 - Casos de agressões registrados pela Companhia de Engenharia de Tráfego em 2016 e 2017



Fonte: SOARES, 2017 apud COMPANHIA DE ENGENHARIA DE TRÁFEGO, 2017.

Casos de agressões a agentes de trânsito têm se tornado cada vez mais comuns e evidencia como o modelo atual de fiscalização não só é ineficiente em muitos aspectos, mas também expõe os profissionais a ameaças à sua integridade física, reforçando a necessidade de alternativas mais seguras e tecnológicas para a fiscalização do trânsito.

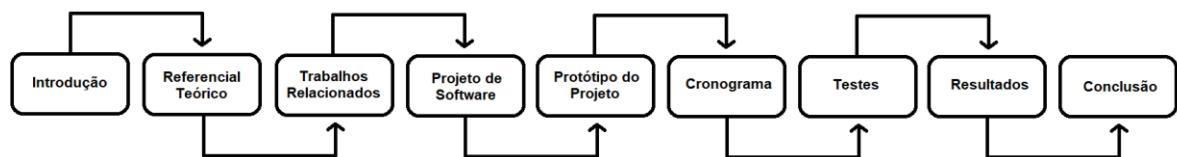
Outro ponto muito importante é o alto custo para manter agentes nas ruas 24h por dia e a dificuldade de manter uma vigilância que seja constante e justa. A Confederação Nacional do Transporte (CNT, 2015) destaca que, entre os mais de 5,5 mil municípios brasileiros, apenas cerca de 1,4 mil contam com órgãos municipais de trânsito, evidenciando que a cobertura institucional de fiscalização é limitada. Além disso, mesmo nas cidades onde esses órgãos estão presentes, a CNT aponta que o número de agentes ainda é insuficiente para atender à demanda. Isso demonstra a enorme falta de fiscalização em território brasileiro. Complementando esse cenário, o Departamento Nacional de Trânsito (DENATRAN, 2015) recomenda a existência de um agente fiscal para cada dois mil veículos, mas a realidade é bem diferente: na cidade de São Paulo, por exemplo, onde a frota ultrapassa 8 milhões de veículos, há apenas 1,8 mil agentes, o que equivale a aproximadamente um fiscal para cada 4,3 mil carros, o dobro do recomendado.

Este projeto se propõe a resolver essa situação com o apoio da visão computacional aliada à inteligência artificial que pode oferecer uma alternativa mais eficiente, precisa e escalável, dessa forma melhorando todo o sistema de fiscalização atual garantindo que a aplicação das infrações seja devidamente realizada.

## 1.5 Organização do projeto

Este trabalho está organizado em 9 capítulos (Figura 1), que abordam desde os fundamentos teóricos até os resultados obtidos na aplicação de visão computacional e inteligência artificial para detecção automatizada de infrações de estacionamento.

Figura 1 – Estrutura do projeto



Fonte: Autoria própria, 2025

- Capítulo 1 – Introdução: Apresenta o contexto do problema, os objetivos da pesquisa, a justificativa para o desenvolvimento do sistema e uma visão geral da metodologia adotada.
- Capítulo 2 – Fundamentação Teórica: Discute conceitos essenciais de visão computacional (como modelos YOLO e OpenCV), técnicas de *deep learning* aplicadas à detecção de objetos e a regulamentação sobre estacionamento urbano.
- Capítulo 3 – Trabalhos Relacionados: Apresenta pesquisas e projetos de diferentes autores com temas semelhantes, que foram utilizados como base para pesquisa e inspiração.
- Capítulo 4 – Projeto de software: Demonstra a modelagem técnica da solução proposta através de documentos, arquiteturas, modelos e diagramas seguindo os devidos procedimentos de engenharia de software.
- Capítulo 5 – Protótipo do projeto: Apresenta através de capturas de tela (*prints*), a interface (protótipo de alta fidelidade) desenvolvida para o programa.

- Capítulo 6 – Cronograma para TC-II: Exibe o cronograma planejado para o segundo semestre do desenvolvimento do TCC.
- Capítulo 7 – Testes: Detalha o processo metodológico adotado para o desenvolvimento, implementação e avaliação do projeto.
- Capítulo 8 – Resultados: Demonstração da capacidade de generalização do modelo de detecção e dos testes de eficácia do Módulo de Validação de Infrações em cenários de uso prático.
- Capítulo 9 – Considerações Finais: Sintetiza as conclusões, destaca as contribuições do projeto e sugere melhorias futuras, como a integração de outras infrações menores.

## 2 REFERENCIAL TEÓRICO

### 2.1 Inteligência Artificial

A Inteligência Artificial é um campo da tecnologia, que busca fazer com que os computadores ajam ou pensem de maneira inteligente (GOMES, 2010). Para isso, o principal objetivo da IA é gerar uma máquina que seja capaz de imitar o comportamento humano de forma inteligente, por meio de linguagens naturais, reconhecimento de padrões e aprender com base em experiências prévias de tomadas de decisões baseadas em dados (TAULLI, 2020).

Segundo TAULLI (2020), Inteligência Artificial pode ser classificada em dois tipos:

- IA Fraca: essa se limita a identificação de padrões (*pattern matching*), além de focar em tarefas específicas, como a assistente virtual Alexa, da Amazon, por exemplo.
- IA Forte: é a fortemente conhecida por filmes de ficção científica, no qual ela realmente possui compreensão da realidade inserida, podendo demonstrar criatividade e emoções.

Com base nisso, para TAULLI (2020), a Inteligência Artificial está localizada inicialmente na IA Fraca, e o objetivo dependendo do caso é alcançar o nível de IA Forte, o que pode levar anos e até décadas, apesar de que há pesquisadores que não acreditam nesta possibilidade.

Como comentado por GOMES (2010), Alan Turing, conhecido como o pai da computação, foi o pioneiro a ter uma visão abrangente da IA, como apresentado em seu artigo “*Computing Machinery and Intelligence*”, em 1950, no qual propôs o Teste de Turing. Esse teste consiste em avaliar a capacidade de um computador em exibir comportamento inteligente indistinguível ao de um ser humano, de modo que um interrogador, ao realizar perguntas por escrito, não seja capaz de determinar se as respostas foram enviadas por um humano ou por uma máquina.

Dentro da área de Inteligência Artificial, há o subcampo *Machine Learning* (ML), que foca em desenvolver algoritmos que são capazes de aprender padrões por meio de dados, sem que sejam explicitamente programados (MITCHELL, 1997). Segundo

MITCHELL (1997, p. 2), "um programa de computador é dito aprender com a experiência E em relação a alguma classe de tarefas T e medida de desempenho P, se seu desempenho em T, medido por P, melhora com a experiência E". Para ALPAYDIN (2020), o ML pode ser definido como uma área da ciência que busca capacitar os computadores a aprenderem com base em dados, aprimorando seu desempenho em atividades específicas por meio da experiência.

Dentro de Machine Learning, destacam-se duas categorias principais. O aprendizado supervisionado envolve modelos treinados com dados já rotulados, como classificação de imagens, enquanto o aprendizado não supervisionado identifica padrões em dados que não estão rotulados, como *clustering* (BISHOP, 2006). Segundo HASTIE et al. (2009), pode-se comparar um estudante que aprende com seu professor, onde os rótulos agem como feedback correto, como um exemplo análogo de aprendizado supervisionado.

Por fim, observa-se que a aplicação das técnicas e conceitos abordados é essencial, uma vez que o sistema desenvolvido lida com imagens de veículos e sinalizações de estacionamento que demandam reconhecimento de padrões. Dessa forma, torna-se possível a identificação automatizada das infrações de trânsito cometidas, a partir de modelos treinados, eliminando a necessidade de intervenção humana no processo de reconhecimento.

### 2.1.1 Redes Neurais Artificiais

A trajetória das redes neurais artificiais tem início com o modelo proposto por McCulloch e Pitts, em 1943, que buscava representar o funcionamento dos neurônios biológicos por meio de circuitos lógicos. Esse trabalho pioneiro lançou as bases teóricas para os avanços que viriam nas décadas seguintes, como o Perceptron desenvolvido por Rosenblatt em 1958, e, mais recentemente, as redes neurais profundas. De acordo com GOODFELLOW, BENGIO e COURVILLE (2016), um dos marcos mais importantes dessa evolução foi a popularização do algoritmo de retropropagação, na década de 1980, que possibilitou o treinamento eficaz de redes com múltiplas camadas. Esse avanço foi essencial para que as redes neurais passassem a ser aplicadas com sucesso em áreas como a visão computacional e o reconhecimento de padrões complexos, temas que também são abordados no capítulo 2.2.

As redes neurais artificiais (RNAs), são modelos de computadores que visam simular o sistema nervoso humano e biológico. HAYKIN (2001) define as redes neurais como processadores distribuídos e altamente paralelizados, constituídos por unidades de processamento básicas, que possuem uma capacidade inerente de armazenar conhecimento adquirido experimentalmente e disponibilizá-lo quando necessário. Tal característica encaixa perfeitamente para o sistema proposto neste trabalho, onde é preciso haver aprendizado adaptativo para que padrões de imagens de trânsito considerados complexos, possam ser identificados.

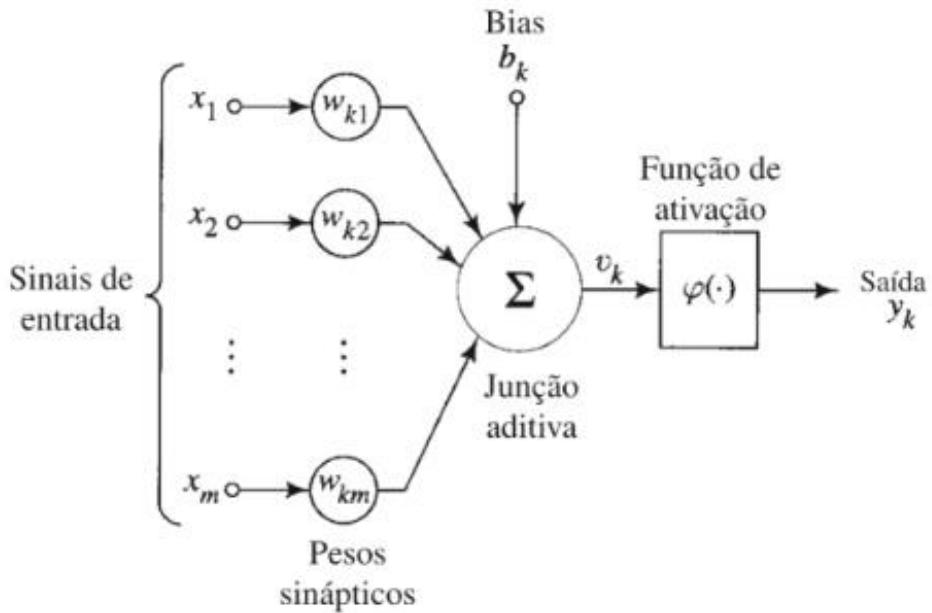
Uma rede neural é formada por três camadas essenciais: a de entrada (*input layer*), as intermediárias (*hidden layers*) e a de saída (*output layer*). A primeira recebe os dados iniciais, como os pixels de uma imagem, enquanto as intermediárias processam essas informações por meio de operações matemáticas para identificar padrões importantes. Por fim, a camada de saída gera o resultado, como a identificação de uma infração de trânsito (RUSSELL; NORVIG, 2016).

Segundo HAYKIN (2001, p. 36), o neurônio artificial representa a unidade básica de processamento de informação em uma rede neural, sendo estruturalmente composto por três elementos principais:

1. Conjunto de sinapses (conexão de entrada), cada uma com um peso individual.
2. Um somador, que combina os sinais de entrada.
3. Uma função de ativação, para limitar a amplitude de saída

A rede neural mais simples, chamada de Perceptron, contém apenas um neurônio. (Figura 2)

Figura 2 - Modelo de Neurônio Artificial



Fonte: HAYKIN (2001)

Para que uma rede seja desenvolvida com eficiência, é essencial que ela passe pela etapa de Machine Learning (ML), uma vez que esse processo de treinamento é fundamental para que o sistema compreenda e atue adequadamente em uma área específica (BIANCHI, 2004). Essa abordagem está alinhada com a visão de ALPAYDIN (2020), que define o ML como um subcampo da Inteligência Artificial (IA) voltado para capacitar computadores a aprender com dados, melhorando seu desempenho em tarefas específicas por meio da experiência.

Neste projeto, as redes neurais serão aplicadas em dois estágios principais: detecção de objetos, por meio da arquitetura YOLO (You Only Look Once), responsável pela identificação de veículos e elementos de sinalização em imagens capturadas por câmeras, de acordo com o capítulo 2.2.2, o YOLO é um modelo que combina precisão e velocidade de detecção em tempo real (REDMON et al., 2016). Como também, a classificação de infrações, em que a rede neural analisará as características extraídas das imagens, como a placa de regulamentação, o veículo e o local, assim categorizando a infração adequada conforme ao Código de Trânsito Brasileiro (CTB).

A Inteligência Artificial, particularmente com o uso de técnicas como Machine Learning e Redes Neurais Artificiais, tem se destacado como uma ferramenta eficiente na automação de tarefas complexas. Essas tecnologias são capazes, por exemplo, de identificar padrões em imagens e classificar infrações de trânsito com precisão, otimizando processos que demandariam grande esforço humano. Segundo RUSSELL; NORVIG (2016), o aprendizado de máquina possibilita que os sistemas aprimorem seu desempenho de forma contínua por meio da experiência, aumentando assim sua eficiência em aplicações específicas. Logo, a aplicação dessas tecnologias no reconhecimento automatizado de infrações viárias não só otimiza processos, mas também reforça a importância da IA como um campo em constante evolução, capaz de transformar a maneira como interagimos com sistemas computacionais.

## 2.2 Visão computacional

A visão computacional, é uma maneira de fazer com que computadores extraiam elementos e/ou características de vídeos e imagens, através de redes neurais e *Machine Learning* (Aprendizado de máquina) (IMB, 2019).

### 2.2.1 Captura de imagem

A captura de imagens pode ser feita através de dispositivos de entrada, como uma câmera tirar uma foto ou através de um upload direto da imagem digital. É importante lembrar que para funcionar corretamente com uma ferramenta de visão computacional, principalmente se precisar implementar de maneira em tempo real, o vídeo acaba sendo transferido como um compilado de frames para ser processado, então muitas vezes alguns dos frames são perdidos para a visualização do resultado final assim, tendo um resultado com menor frame rate (BERTOLETI, 2017). Porém para esse tipo de aplicação será melhor utilizado, devido aos parâmetros e requisições presentes no processamento da imagem, apenas a opção de envio de arquivo de imagem (jpg, png, etc.).

## 2.2.2 You Only Live Once (Yolo)

O modelo de detecção de objetos e segmentação de imagens em tempo real, YOLO, pela empresa Ultralytics, é conhecido pela sua rapidez e precisão. É capaz de identificar e localizar objetos em imagens e vídeos, é possível aplicá-lo na robótica, em veículos autônomos e em sistemas de vigilância (KUNDU, 2023). (Imagem 1)

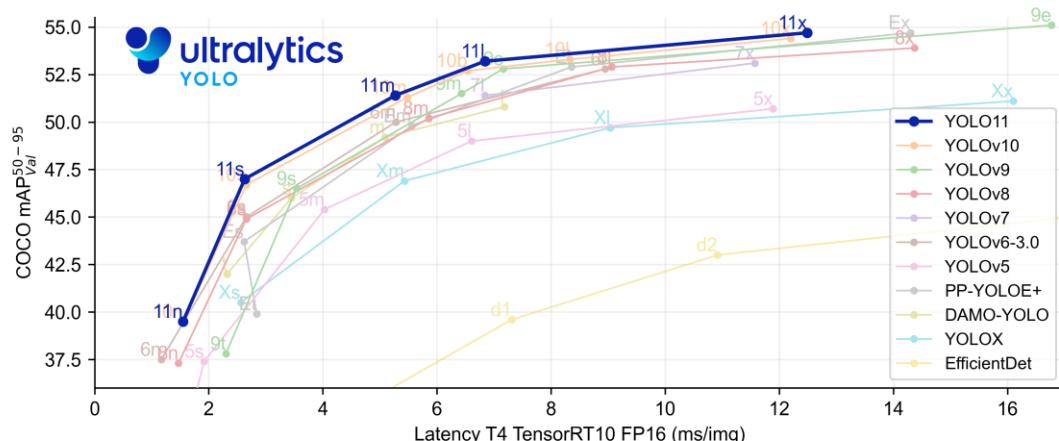
Imagen 1 – Gerenciamento de Estacionamento em cidades Inteligentes com YOLOv8



Fonte: VINA, 2025

A versão mais recente é o YOLOv11, oferecendo um desempenho bem maior que suas versões anteriores, com alta acurácia e baixa latência. É considerado um modelo SOTA (*State of the Art*), ou seja, de alta performance para realizar tarefas específicas de Inteligência artificial (SRIVASTAVA, 2025). Oferece também ajuda nas tarefas de estimativa de pose, rastreamento e classificação (ULTRALYTICS, 2025). (Gráfico 2)

Gráfico 2 – Comparação de Performance YOLOv11

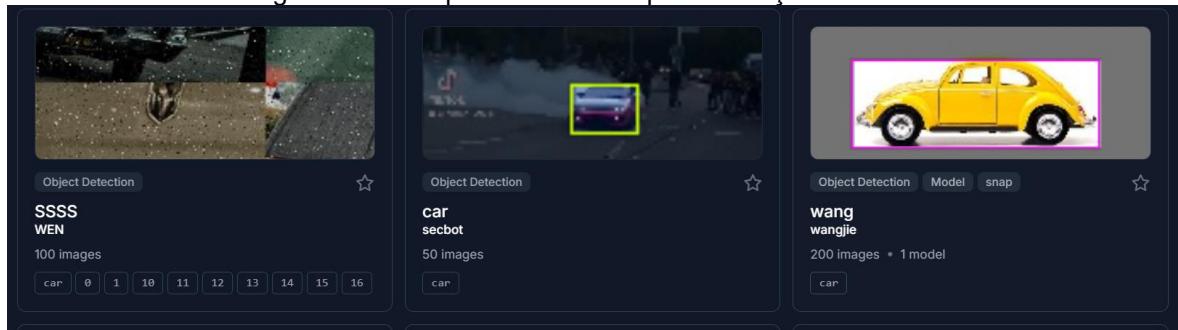


Fonte: Ultralytics, 2024

### 2.2.3 Detecção de Veículos e Áreas de Estacionamento

Também é possível treinar o YOLO, alimentando-o com um *dataset* (conjunto de dados, nesse caso imagens) customizado e ajustando os parâmetros, para assim poder identificar e localizar elementos específicos. É necessário que o conjunto de imagens esteja rotulado/classificado com as classes específicas, ou seja, quais são e onde estão localizados os elementos desejados. Para isso já existem fontes com *datasets* prontos, como o Roboflow, Ultralytics Hub e Google Open Images v7. Mas mesmo que precise rotular um dataset do zero, é possível utilizar ferramentas como o Label Studio para facilitar a classificação (ULTRALYTICS, 2025). (Imagem 2)

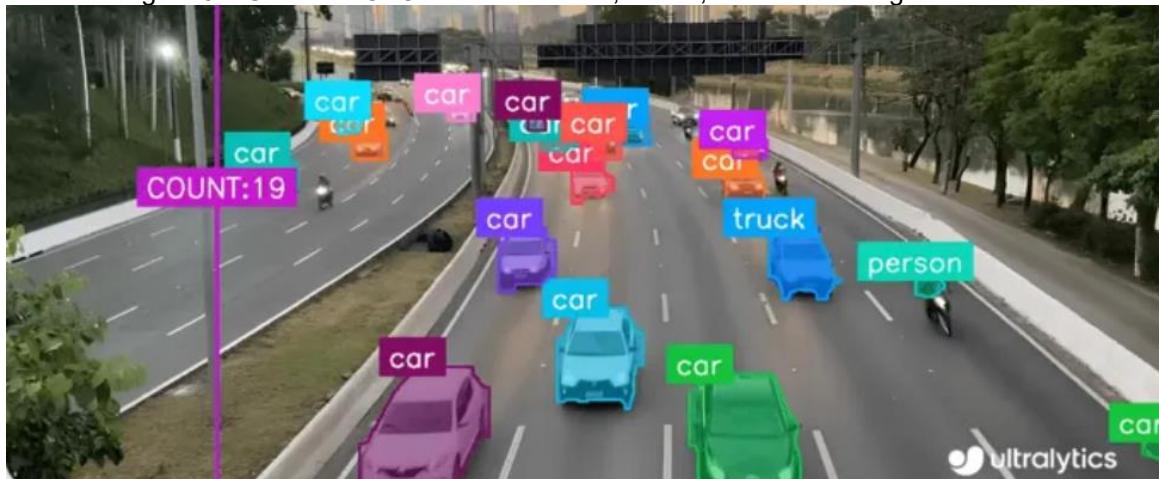
Imagen 2 – Exemplos de datasets para detecção de veículos



Fonte: Roboflow, 2025

Será utilizado uma mistura de *datasets* já existentes e fotos de fonte própria para treinar o modelo, com imagens de veículos, imagens de elementos de trânsito e de áreas de sinalização/estacionamento, como placas, faixas, guia rebaixada, calçada etc. Utilizando essas técnicas será possível ter um maior controle e clareza de dados para a detecção automatizada de infrações. (Imagen 3)

Imagen 3 – Usando YOLO11 Para detectar, contar, e rastrear tráfego de veículos

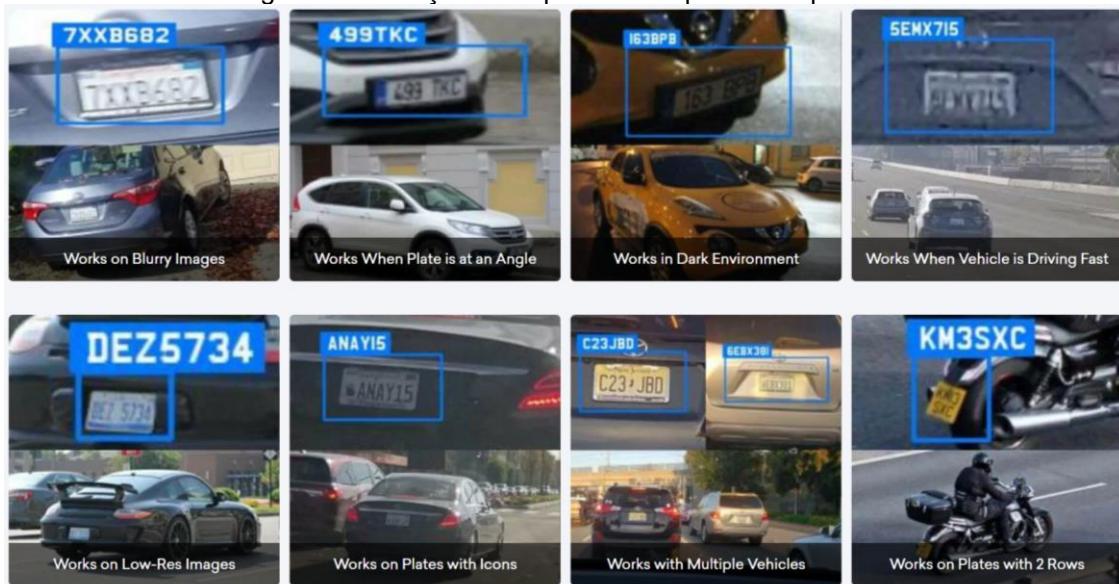


Fonte: VINA, 2025

## 2.2.4 Automatic License Plate Recognition (ALPR)

ALPR, do inglês “Reconhecimento automático de placas de veículo”, é uma tecnologia de Visão Computacional, onde utilizam-se técnicas para a extração da informação de uma placa de veículo de uma imagem ou uma sequência de imagens. Para aplicações em tempo real, é necessário que a ALPR processe de maneira rápida e precisa as placas de veículos em diferentes condições ambientais, como dia e noite, também com qualidade de câmera diferentes. Com ela também é possível processar placas de diferentes cidades, províncias ou nações, com letras ou palavras de idiomas diferentes, e até podem estar parcialmente cobertas com sujeira, partes do carro ou luzes fortes (DU, 2012). (Figura 3)

Figura 3 – Situações em que a ALPR pode ser aplicada



Fonte: Plate Recognizer, 2025

## 2.2.5 Métricas de Avaliação e Conceitos de Treinamento

Para validar a eficácia do sistema proposto, utilizam-se métricas quantitativas e conceitos técnicos padronizados pela documentação oficial da Ultralytics (2025), desenvolvedora da arquitetura YOLO. A avaliação do modelo baseia-se nos seguintes indicadores de desempenho e parâmetros de otimização:

### 2.2.5.1 Métricas de Desempenho

**Interseção sobre União (*IoU*):** É a métrica fundamental que quantifica a sobreposição espacial entre a caixa prevista pelo modelo (*Bounding Box*) e a caixa real anotada (*Ground Truth*). O IoU varia de 0 a 1, onde valores mais altos indicam que a localização prevista está mais próxima da realidade.

**Precisão (*Precision*):** Avalia a confiabilidade das previsões positivas. Uma alta precisão indica que, quando o modelo prevê a presença de um objeto (ex: um carro), ele está provavelmente correto, minimizando a ocorrência de falsos positivos.

**Revocação (*Recall*):** Mede a capacidade do modelo de encontrar todas as instâncias de objetos existentes na imagem. Um alto recall significa que o sistema consegue capturar a maioria dos objetos presentes na cena, minimizando os falsos negativos.

**mAP50 (*Mean Average Precision @ IoU 0.5*):** Representa a média da precisão média calculada considerando um limiar de IoU de 0,50. Esta métrica considera uma previsão correta se ela tiver pelo menos 50% de sobreposição com o objeto real, priorizando a detecção do objeto em detrimento do alinhamento perfeito.

**mAP50-95:** Considerada o padrão-ouro atual (popularizado pelo dataset COCO), é a média da precisão calculada em intervalos de IoU que variam de 0,50 a 0,95 (em passos de 0,05). Esta métrica premia modelos que não apenas localizam o objeto, mas o fazem com extrema precisão geométrica em nível de pixel.

**Matriz de Confusão (*Confusion Matrix*):** Uma ferramenta que visualiza a discrepância entre as etiquetas previstas e as reais, permitindo identificar exatamente onde o modelo está "confuso" ou cometendo erros sistemáticos entre classes específicas.

### 2.2.5.2 Componentes de Detecção e Funções de Perda (*Loss Functions*)

Durante o treinamento, o modelo busca minimizar funções de perda específicas para ajustar seus pesos:

**Boxes (Bounding Box):** Refere-se à região retangular (caixa) definida por coordenadas que isola um objeto específico na imagem. A `box_loss` mede o erro na localização dessas caixas, ajustando o modelo para que as coordenadas previstas se alinhem às reais.

**Masks (Máscaras):** Utilizadas na segmentação de instâncias, são máscaras que delineiam a forma exata do objeto com precisão de pixel, diferentemente da caixa que apenas o circunda. A `seg_loss` avalia a diferença entre a máscara prevista e a forma real do objeto.

**cls\_loss (Classification Loss):** Mede a penalidade quando o modelo classifica incorretamente um objeto (ex: confundir um carro com uma calçada). O objetivo é maximizar a probabilidade da classe correta.

**dfl\_loss (Distribution Focal Loss):** Uma função de perda projetada para refinar a regressão das caixas delimitadoras, focando em ajustar a distribuição de probabilidade dos valores das coordenadas, o que auxilia na precisão da localização em cenários difíceis.

### 2.2.5.3 Hiperparâmetros e Fenômenos de Treinamento

O comportamento do treinamento é regulado por configurações externas chamadas hiperparâmetros:

**Learning Rate (Taxa de Aprendizado):** Determina o tamanho do passo que o algoritmo de otimização dá a cada iteração em direção ao mínimo da função de perda. Uma taxa muito alta pode impedir a convergência, enquanto uma taxa muito baixa torna o treinamento excessivamente lento.

**Batch Size:** Define o número de amostras de treinamento utilizadas em uma única iteração para calcular o gradiente. O ajuste deste valor impacta a estabilidade da estimativa do gradiente e o uso de memória da GPU.

*Momentum*: Um fator que determina o quanto as atualizações passadas influenciam as atuais, ajudando a acelerar o treinamento e a superar mínimos locais na função de perda.

*Overfitting* (Sobreajuste): Ocorre quando o modelo aprende "demais" os detalhes e ruídos dos dados de treino, perdendo a capacidade de generalizar para novos dados. Geralmente acontece com excesso de épocas de treinamento.

*Underfitting* (Subajuste): Ocorre quando o modelo não consegue capturar a tendência subjacente dos dados, resultando em baixo desempenho tanto no treino quanto no teste, muitas vezes devido a uma taxa de aprendizado inadequada ou treinamento insuficiente.

## 2.3 Regulamentação Brasileira

O Código de Trânsito Brasileiro (CTB) é a legislação fundamental que regulamenta a circulação de veículos, pedestres e ciclistas em território nacional, estabelecendo normas, direitos e deveres. No contexto deste trabalho, o CTB assume especial relevância, pois define o que constitui uma infração, incluindo aquelas relacionadas ao estacionamento irregular, servindo como base para o desenvolvimento deste sistema automatizado de detecção de infrações de estacionamento.

Nesta seção, serão abordados os conceitos de estacionamento e infrações conforme o CTB, destacando sua importância para a fiscalização automatizada por meio de visão computacional e inteligência artificial.

### 2.3.1 Código de Trânsito Brasileiro

O Código de Trânsito Brasileiro (CTB) representa o principal instrumento que rege a circulação de veículos, condutores e pedestre. No contexto deste trabalho, o CTB é de suma importância, pois os fundamentos legais e o conceito de infrações de trânsito especialmente aquelas relacionadas ao estacionamento irregular rege este trabalho e serve como um dos pilares para o desenvolvimento do projeto.

#### 2.3.1.1 Estacionamento

O estacionamento de veículos é uma prática cotidiana nas vias urbanas, sendo regulamentada pelo código de Trânsito Brasileiro (CTB, 2021). O ato de estacionar segundo o CTB é a “imobilização de veículos por tempo superior ao necessário para embarque ou desembarque de passageiros”.

É preciso também diferenciar o ato de estacionar um veículo com o ato de fazer uma breve paragem para desembarque. Conforme o Código de Trânsito Brasileiro (CTB) o termo estacionar é definido como a imobilização de veículos por tempo superior ao necessário para embarque ou desembarque de passageiros. Essa definição diferencia-se da parada, que é descrita como a imobilização do veículo com a finalidade e pelo tempo estritamente necessário para embarque ou desembarque de passageiros.

A correta identificação do conceito de estacionar e a sua diferença entre a paragem de desembarque é fundamental para identificar e prevenir situações em que o veículo esteja infringindo normas de trânsito, assim como, para o contexto deste projeto de fiscalização automatizada.

#### 2.3.1.2 Infrações

No âmbito do Código de Trânsito Brasileiro (CTB), infração é caracterizada como toda ação ou omissão que contraria as normas estabelecidas na legislação de trânsito (BRASIL, 1997, art. 161). Essas condutas são classificadas conforme sua gravidade leves, médias, graves ou gravíssimas e acarretam penalidades específicas, como multas, pontos na Carteira Nacional de Habilitação (CNH), apreensão ou remoção do veículo, entre outras medidas administrativas.

Essas infrações atuam como uma medida para inibir o condutor a cometer atos inconsequentes, como é dito no Art. 256 do CTB (1997), as penalidades serão aplicadas com a devida fundamentação legal e observância do devido processo legal, com o objetivo de educar e prevenir a ocorrência de infrações, conforme o disposto neste Código.

Diante disso, evidencia-se a importância de mecanismos eficazes e automatizados para a detecção e punição de infrações, assegurando o cumprimento das normas de trânsito e contribuindo para a segurança e a organização do espaço urbano.

#### 2.3.1.3 Infrações por Estacionamento de Veículos

As infrações por estacionamento irregular se configuram como uma categoria específica dentro das normas de infrações previstas no Código de Trânsito Brasileiro. Segundo o Código de Trânsito Brasileiro (BRASIL, 1997), são consideradas infrações de estacionamento aquelas que envolvem a parada do veículo em locais proibidos pela sinalização ou em situações como sobre faixas de pedestres e calçadas (Art. 181).

#### 2.3.1.4 Sinalização

A sinalização de trânsito é um dos pilares fundamentais para a organização, segurança e fluidez nas vias públicas. Conforme estabelecido o art. 80 do Código de

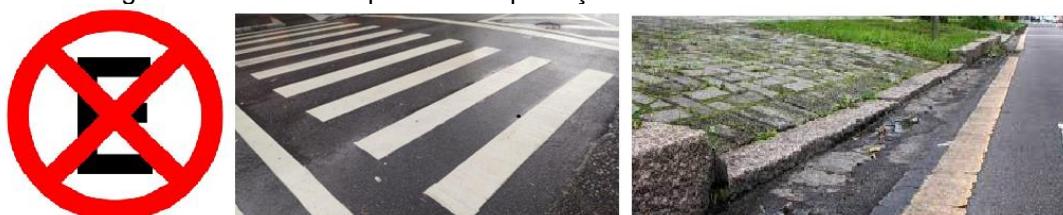
Trânsito Brasileiro (CTB), a sinalização deve ser utilizada sempre que necessária para garantir a segurança e a orientação dos usuários da via, devendo obedecer aos padrões e critérios estabelecidos pelo Conselho Nacional de Trânsito (CONTRAN).

A correta interpretação da sinalização de trânsito é um fator crucial para o funcionamento e eficácia deste trabalho, isso porque a existência ou não de uma sinalização proibitiva é um fator determinante para que este sistema consiga identificar a infração de forma válida. Para isso os reconhecimentos de placas como Proibido Estacionar (R-6a), guias (meio-fio), faixa de pedestre entre outros devem ser reconhecidas pelo sistema de forma devida.

Conforme é descrito no Art. 181 do CTB (2021), é definido mesmo na ausência de sinalização específica entre os locais onde o estacionamento é vedado, destacam-se: em esquinas ou a menos de cinco metros do encontro das vias; em guias de calçada rebaixadas destinadas à entrada ou saída de veículos; sobre faixas de pedestres; ou em locais que prejudiquem a movimentação de outros veículos.

Entre as situações que configuram infrações de estacionamento, é possível identificar elementos visuais padronizados que indicam a proibição da parada ou permanência de veículos em determinados locais. Esses elementos incluem sinalizações verticais, como a placa de Proibido Estacionar (R-6a), e características do ambiente urbano, como faixas de pedestres e passeios, que delimitam zonas de uso específico ou prioritário. (Imagem 4)

Imagen 4 – Elementos que indicam proibição de estacionamento de veículos



Fonte: Autoria própria, 2025

Conforme ilustrado nas imagens, esses padrões visuais são essenciais para que o sistema baseado em visão computacional reconheça áreas de estacionamento irregular de forma automatizada e assertiva.

### **3 TRABALHOS RELACIONADOS**

#### **3.1 Convolutional Neural Network for Person and Car Detection using YOLO Framework**

PUTRA et al. (2018) desenvolveram um sistema de detecção em tempo real de pessoas e veículos para aplicações em Sistemas Avançados de Assistência ao Condutor (ADAS). Utilizando uma versão reduzida do YOLO com apenas 7 camadas de rede neural convolucional, os autores avaliaram diferentes configurações de células de grade (*grid*) para melhorar a detecção de objetos pequenos. Os resultados mostraram que, mesmo com redução na precisão (mAP), o sistema mantém boa eficácia em cenários em tempo real, especialmente com grades de 11x11 células, tornando-o viável para ADAS devido ao equilíbrio entre velocidade e acurácia.

#### **3.2 Computer vision in automated parking systems: Design, implementation and challenges**

HEIMBERGER et al. (2017) descreveram o desenvolvimento de um sistema comercial de estacionamento automatizado, enfatizando como módulos de visão computacional, como reconstrução 3D, reconhecimento de marcações, detecção de área livre e de objetos, se integram para lidar com casos de uso diversos em cenários de manobra em baixa velocidade. O estudo discute os desafios de projetar uma solução de baixo custo com segurança funcional e como as câmeras complementam sensores ativos (ultrassom, radar) para aumentar a robustez do sistema, oferecendo a primeira visão integral dos componentes e fluxos de um sistema de estacionamento automatizado comercial.

#### **3.3 Real-Time Illegal Parking Detection Algorithm in Urban Environments**

PENG et al. (2022) identificaram a necessidade de reduzir custos humanos e de infraestrutura na detecção de infrações de estacionamento, propondo um algoritmo de votação baseado em redes profundas integrado a câmeras embarcadas. Construíram um novo conjunto de dados com mais de 10 000 imagens anotadas em sete categorias e introduziram a métrica “unidades mínimas ilegais” para correlacionar eficientemente veículos e tipos de infração, reduzindo tempo e custos de rotulagem. Testes em cenários urbanos reais e sob diferentes condições de iluminação

demonstraram a robustez do método a variações ambientais, estabelecendo um *benchmark* para estudos futuros em detecção móvel de estacionamento irregular.

### **3.4 Vision-Based Parking-Slot Detection: A Benchmark and A Learning-Based Approach**

ZHANG et al. (2018) preencheram duas lacunas no campo de detecção de vagas de estacionamento por visão computacional: criaram uma base de dados de 8 600 imagens de câmeras *surround-view*, com marcações e vagas cuidadosamente anotadas, e propuseram o método PSDL. Nesse método, pontos de marcação são detectados primeiro para, então, inferir as vagas válidas. Avaliações em sua própria base comprovaram a eficácia e a eficiência de PSDL, que assim passa a ser referência inicial para o desenvolvimento de técnicas mais avançadas de detecção de vagas.

### **3.5 Vacant Parking Slot Detection in the Around View Image Based on Deep Learning**

LI et al. (2020) abordaram as dificuldades de detecção de vagas livres em sistemas de assistência de estacionamento com visão ao redor do veículo (AVM), e apresentaram o VPS-Net. A abordagem divide a tarefa em duas etapas: detecção de estrutura de vaga, com um YOLOv3 ajustado para inferir pontos de marcação e geometria, e classificação de ocupação, por meio de uma rede customizada com tamanho de filtros e profundidade calibrados. Nos testes do ps2.0 e PSV *datasets*, VPS-Net alcançou precisão de 99,63 % e recall de 99,31 %, demonstrando elevada generalização e robustez.

### **3.6 Automatic detection of long-term parked cars**

CARLETTI et al. (2015) propuseram um método para detectar estacionamento irregular a partir de vídeos de câmeras de vigilância fixas, incorporando mapas de calor que acumulam informações espaço-temporais dos veículos parados e mantendo o *background* fixo baseado em todo o movimento registrado. Avaliações nos conjuntos de dados iLids e PETS 2000 mostraram que a técnica supera abordagens de atualização de *background* instantânea, confirmando sua eficácia na detecção automática de veículos em áreas proibidas.

### **3.7 Automatic Detection of Parking Violation and Capture of License Plate**

LIU et al. (2019) exploraram a limitação de ângulo de câmeras fixas para capturar todas as áreas de estacionamento irregular, empregando drones (UAVs) para coleta de imagens aéreas. Integraram segmentação semântica (DeepLabv3+) para delimitação de áreas e detecção de vagas, e um detector YOLOv5-like com módulo SPPF para maior eficiência computacional. O treinamento nesses dados proprietários mostrou desempenho significativo sem necessidade de pós-processamento, indicando viabilidade para monitoramento aéreo de infrações de estacionamento.

### **3.8 Detection of Illegal Parking Based on Deep Learning**

LI et al. (2023) investigaram técnicas de visão computacional e aprendizado profundo para detecção de infrações de estacionamento em cenários urbanos complexos. Embora mantenham o foco em segmentação e detecção de objetos, este trabalho introduz melhorias em modelos previamente treinados e *pipelines* de classificação temporal, aumentando a precisão em situações de oclusão e iluminação variável. Os resultados indicam avanços nos coeficientes de precisão e *recall* em comparação com métodos anteriores, contribuindo para o estado da arte em sistemas móveis de fiscalização automatizada de estacionamento.

### **3.9 Deep Learning-based ALPR system for vehicle access control trained on brazilian license plates**

LEFUNDES et al. (2021) propuseram o AccessALPR, um sistema voltado ao reconhecimento de placas veiculares brasileiras, combinando YOLOv3 para detecção e EfficientNet-B0 para leitura dos caracteres. O diferencial do trabalho está na dispensa da etapa de segmentação de caracteres e no uso de pré-processamentos como CLAHE e RETINEX para melhorar imagens capturadas em ambientes complexos. O sistema demonstrou precisão elevada e adaptação a diferentes cenários, sendo eficaz para controle automatizado de acesso e fiscalização veicular.

### **3.10 An Embedded Automatic License Plate Recognition System using Deep Learning**

IZIDIO et al. (2018) propuseram um sistema embarcado para detecção e reconhecimento de placas veiculares brasileiras utilizando redes neurais

convolucionais (CNNs). A solução foi implementada em um Raspberry Pi 3 com câmera Pi NoIR v2, utilizando a arquitetura Tiny YOLOv3 para detectar placas nas imagens capturadas, seguida por uma segunda rede treinada inicialmente com imagens sintéticas e refinada com dados reais para o reconhecimento dos caracteres. O sistema demonstrou robustez a variações de ângulo, iluminação e ruído, atingindo taxa de detecção de 99,37% e taxa geral de reconhecimento de 97,00% em imagens reais, com tempo médio de processamento de 2,70 segundos por imagem de resolução  $1024 \times 768$  contendo uma única placa. A proposta oferece uma alternativa de baixo custo, portátil e eficaz para aplicações em controle de tráfego e segurança.

### **3.11 A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector**

LAROCA et al. (2018) apresentaram um sistema robusto e eficiente de Reconhecimento Automático de Placas Veiculares (ALPR), baseado no detector de objetos YOLO e em redes convolucionais treinadas separadamente para cada etapa do processo, garantindo robustez a variações de câmera, iluminação e fundo. Para a segmentação e reconhecimento de caracteres, o sistema emprega uma abordagem em duas etapas com estratégias simples de aumento de dados, como inversão e espelhamento de placas. O método superou soluções comerciais como Sighthound e OpenALPR no conjunto de dados SSIG, alcançando taxa de reconhecimento de 93,53% a 47 FPS. Em um novo conjunto de dados realista (UFPR-ALPR), com veículos e câmeras em movimento, o sistema atingiu 78,33% de reconhecimento a 35 FPS, superando comercialmente sistemas que ficaram abaixo de 70%. Os resultados destacam a eficácia do sistema em cenários reais com diferentes tipos de veículos.

### **3.12 Considerações Finais**

Os estudos apresentados nas Seções 3.1 a 3.3 reforçam a necessidade de sistemas de detecção em tempo real para infrações de estacionamento. Putra et al. (Seção 3.1) demonstraram que arquiteturas YOLO reduzidas conseguem detectar pessoas e veículos com latência compatível a ADAS, ainda que com queda no mAP. Heimberger et al. (Seção 3.2) mostraram como sistemas comerciais de estacionamento automatizado combinam módulos de visão 3D e sensores ativos, mas esbarram em custos elevados e complexidade. Peng et al. (Seção 3.3), por sua vez, validaram algoritmos de votação em câmeras embarcadas para mobilidade urbana,

estabelecendo um *benchmark* robusto, porém ainda focado em cenários muito específicos. Esses trabalhos evidenciam que, embora viáveis, muitas soluções carecem de um equilíbrio ideal entre precisão, portabilidade e custo, lacuna que o sistema proposto busca preencher.

As Seções 3.4 a 3.6 concentram-se em detecção de vagas livres e ocupadas a partir de câmeras *surround-view* e AVM. Zhang et al. (Seção 3.4) introduziram um grande banco de imagens e o método PSDL, mas sem endereçar diretamente infrações; Li et al. (Seção 3.5) e Carletti et al. (Seção 3.6) propuseram VPS-Net e mapas de calor respectivamente, atingindo alta robustez, porém apenas em cenários estáticos de câmeras fixas. Entretanto, o presente estudo avança ao propor a unificação da detecção e a classificação imediata de diferentes infrações (irregularidades) em um único e coeso fluxo de execução, otimizando o *pipeline* de monitoramento. Isso permite uma resposta em tempo hábil e simplifica a arquitetura do sistema em comparação com as abordagens que tratam ocupação e infração em etapas separadas.

Para o reconhecimento de placas veiculares, as Seções 3.9 a 3.11 apresentam importantes contribuições: Lefundes et al. (Seção 3.9) e Izidio et al. (Seção 3.10) implementaram sistemas embarcados de ALPR capazes de operar em *hardware* restrito, enquanto Laroca et al. (Seção 3.11) alcançaram desempenho superior a soluções comerciais com estratégias de *data augmentation*. Esses trabalhos servem de base para o módulo de detecção de placas desenvolvido, o qual, além de reconhecer caracteres, integra identificação de cor, modelo e outras características do veículo quando possíveis de serem extraídas.

Portanto, o modelo proposto representa uma continuidade e um aprimoramento dos estudos apresentados neste capítulo. Ao combinar detecção em tempo real (YOLOv11), classificação de tipos de infração conforme o CTB, extração de metadados (local, data, hora) e um sistema de armazenamento e interface para validação e análise estatística, espera-se não apenas acelerar a resposta das fiscalizações de trânsito, mas também oferecer uma plataforma flexível para pesquisas futuras em mobilidade urbana e cidades inteligentes, como será explorado no Capítulo 5.

## 4 PROJETO DE SOFTWARE

O Capítulo 4 apresenta a modelagem técnica da solução proposta para detecção automatizada de infrações de estacionamento. São descritas as principais estruturas que compõem o sistema, iniciando-se pelo documento de visão, que descreve informações gerais sobre o projeto do sistema, agindo como um complemento da introdução.

Na sequência é apresentado o documento de requisitos, que se divide em dois: Requisitos Funcionais, que são as funções exatas que o sistema busca efetuar (também chamados de casos de uso), e os Requisitos Não Funcionais, que definem como as funcionalidades serão alcançadas, através de critérios como eficiência, usabilidade, segurança etc.

Logo em seguida vem a arquitetura da solução, a qual define a organização e a interação entre os componentes responsáveis pelo processamento de imagens, detecção de infrações, interface com o usuário e persistência de dados.

Posteriormente, é detalhado o projeto lógico e físico do banco de dados, contemplando a estrutura relacional utilizada para armazenamento das informações processadas, como dados de veículos, tipos de infrações, registros temporais e resultados das análises realizadas pelo sistema.

Este capítulo visa estabelecer uma visão clara, estruturada e fundamentada da solução proposta, o que permite compreender da melhor forma como o projeto de software está planejado e organizado.

## 4.1 Documento de Visão

### 4.1.1 Nome do projeto

IPD – Illegal Parking Detection

### 4.1.2 Objetivo Geral

Criar um sistema automático para monitorar estacionamentos irregulares, empregando visão computacional e Inteligência Artificial (IA). O objetivo é aumentar a eficácia no controle de transgressões na cidade, diminuir a necessidade de fiscalização humana e auxiliar numa gestão de trânsito mais inteligente e ecologicamente correta.

### 4.1.3 Justificativa

O aumento da frota de veículos nas cidades tem agravado problemas de mobilidade urbana, especialmente as infrações por estacionamento irregular. Em 2023, foram registradas 2,1 milhões de autuações no Brasil, reflexo da fiscalização limitada e ineficiente. A atuação manual dos agentes de trânsito é cara, restrita e sujeita a falhas, o que favorece a impunidade e a recorrência dessas infrações.

Nesse contexto, o uso de tecnologias como visão computacional e inteligência artificial oferece uma solução moderna e eficaz, permitindo a detecção automática e contínua de irregularidades. Além de otimizar a fiscalização, essas tecnologias contribuem para a organização do trânsito e a segurança nas vias.

Do ponto de vista acadêmico, o projeto possibilita a aplicação prática de técnicas de processamento de imagens e aprendizado de máquina em um problema urbano relevante e atual.

### 4.1.4 Objetivos Específicos

Estudar e comparar técnicas de visão computacional aplicadas à detecção de infrações de trânsito.

Implementar um protótipo funcional capaz de identificar automaticamente situações de estacionamento irregular.

Utilizar métricas adequadas para avaliar o desempenho do sistema com precisão.

Desenvolver uma interface simples para visualização dos resultados e interação com os usuários.

Documentar todas as etapas do desenvolvimento, incluindo decisões técnicas, testes e resultados obtidos.

#### 4.1.5 Escopo do Projeto

Sistema web funcional com banco de dados.

Cadastro de usuários.

Perfil do usuário, histórico de infrações e preferências.

Busca e visualização de veículos com infrações.

Banco de dados para treinamento da IA.

Sistema de validação e reconhecimento de infrações.

Sistema de extração de dados do veículo.

#### 4.1.6 Stakeholders Acadêmicos

Equipe de desenvolvimento: alunos participantes do projeto.

Professor/orientador: supervisiona o progresso técnico e metodológico.

Usuários *testers*: colegas ou voluntários que interagem com o sistema para fins de validação.

Instituição de Ensino: apoia o projeto com infraestrutura e recursos.

#### 4.1.7 Resultados Esperados

Protótipo funcional de um sistema automatizado para detecção de infrações de estacionamento irregular por meio de visão computacional.

Relatório técnico contendo:

Revisão bibliográfica sobre fiscalização de trânsito e tecnologias aplicadas.

#### 4.1.8 Arquitetura do sistema e ferramentas utilizadas.

Modelos de visão computacional e algoritmos aplicados, com justificativa de escolha.

Avaliação do desempenho com métricas quantitativas (ex: acurácia, precisão, recall).

Potencial de aplicação prática em cidades inteligentes e uso como base para pesquisas futuras.

#### 4.1.9 Tecnologias Sugeridas

*Backend:* Python

*Frontend:* ReactJS

Banco de Dados: MYSQL

Bibliotecas e ferramentas: YOLO (Ultralytics); OpenCV; NumPy

#### 4.1.10 Restrições e Considerações

O desenvolvimento do sistema está sujeito a algumas restrições técnicas e operacionais. A principal limitação está na disponibilidade e qualidade das imagens utilizadas, que podem variar conforme o ambiente, iluminação, ângulo da câmera e presença de obstáculos visuais. Essas variáveis podem impactar diretamente a acurácia do modelo de visão computacional.

Além disso, o treinamento de algoritmos de detecção exige uma base de dados robusta e representativa, o que demanda tempo e recursos para coleta e rotulagem adequada. Outra consideração relevante envolve questões legais e éticas, como o uso de imagens captadas em espaços públicos, respeitando a privacidade dos indivíduos e a legislação vigente.

Do ponto de vista técnico, o desempenho do sistema poderá variar conforme o poder computacional disponível, principalmente em aplicações em tempo real. Assim, será necessário buscar um equilíbrio entre eficiência, custo e precisão. Por fim, vale considerar que o sistema proposto não substituirá totalmente a atuação humana, mas atuará como ferramenta de apoio à fiscalização, ampliando sua efetividade e alcance.

## 4.2 Documento de requisitos (funcionais e não funcionais)

### 4.2.1 Requisitos Funcionais (Casos de uso)

#### 4.2.1.1 [UC01] Cadastrar usuário

O sistema deve permitir ao usuário realizar seu cadastro por meio do preenchimento de suas credenciais em uma etapa inicial.

#### 4.2.1.2 [UC02] Fazer login do usuário

O sistema deve permitir ao usuário acessar sua conta por meio do preenchimento de suas credenciais de login.

#### 4.2.1.3 [UC03] Enviar Arquivo

O sistema deve permitir ao usuário efetuar o envio de um arquivo de imagem, a fim de validar uma infração.

#### 4.2.1.4 [UC04] Consultar infrações

O sistema deve permitir ao usuário consultar infrações registradas no banco de dados.

#### 4.2.1.5 [UC05] Detectar elementos desejados

O sistema deve detectar passivamente os elementos presentes em imagens utilizadas como entrada.

#### 4.2.1.6 [UC06] Validar infração de estacionamento

O sistema deve identificar e processar automaticamente os frames em que uma infração de estacionamento estiver sendo cometida.

#### 4.2.1.7 [UC07] Classificar o tipo e a gravidade da infração

O sistema deve classificar automaticamente o tipo de infração de estacionamento detectada, bem como atribuir sua gravidade com base em regras pré-definidas.

#### 4.2.1.8 [UC08] Extrair dados da infração

O sistema deve extrair as características relevantes do momento da infração, utilizando-se de um modulo ALPR único para a identificação da placa e a extração dos caracteres e da sua cor, como também deve chamar uma API externa para extrair o horário, data e hora do ocorrido através dos metadados.

#### 4.2.1.9 [UC09] Armazenar os dados

O sistema deve armazenar no banco de dados todas as informações extraídas pela Inteligência Artificial, mantendo a integridade e rastreabilidade dos registros.

#### 4.2.1.10 [UC10] Notificar o usuário que uma infração foi cometida

O sistema deve exibir uma notificação ao usuário informando que uma infração de estacionamento foi identificada.

## 4.2.2 Requisitos Não Funcionais

### 4.2.2.1 Estabilidade Funcional

O sistema busca ser estável a partir do momento em que o software identifica um frame de uma entrada visual como válido, pois assim será possível a execução do algoritmo para obter algum tipo de resposta, mesmo se a resposta voltar como nula (nenhuma infração detectada).

### 4.2.2.2 Eficiência de Desempenho

O sistema desenvolvido deverá processar os dados e executar o algoritmo de uma só vez, ao buscar percorrer o fluxo completo do envio da imagem até o resultado esperado. Além de contar com uma *API Rest Full*, no qual suas requisições e respostas terão um tempo estimado de 15 segundos.

### 4.2.2.3 Compatibilidade

A presente aplicação apresenta compatibilidade pois integra de forma eficaz o *front-end* em React-JS com o *back-end* em Python, permitindo que ambos se comuniquem por meio de APIs padronizadas e formatadas em JSON, dessa forma garantindo que as informações transmitidas entre a interface do usuário e a IA de análise e validação de imagens, sejam interpretadas corretamente por cada componente, assim tornando o funcionamento do sistema coeso e sem conflitos ou necessidade de adaptações complexas, assegurando um fluxo contínuo de dados e uma operação estável.

### 4.2.2.4 Usabilidade

O sistema desenvolvido prioriza uma experiência intuitiva e eficiente, garantindo que o usuário consiga realizar todas as etapas, desde o envio da imagem até a visualização da infração identificada, de forma fluida e sem complicações. A interface foi projetada para minimizar cliques desnecessários, além de conter menus simplificados, linguagem direta e orientações contextualizadas, que complementam a usabilidade, tornando a fiscalização mais rápida e reduzindo a curva de aprendizado.

#### 4.2.2.5 Confiabilidade

O sistema deverá apresentar acurácia mínima de 85% na detecção de infrações, considerando que o modelo de visão computacional deverá ser treinado com base em um conjunto de dados diversificado, incluindo diferentes tipos modelos de carros, ângulos, perspectivas e posicionamento dos veículos, para garantir robustez.

Além de uma disponibilidade total de 24 horas por dia de operação, em caso de falha o sistema deverá ser capaz de tratar o erro e continuar suas operações sem comprometer a disponibilidade e a segurança. O sistema deverá ser capaz de detectar e descartar falsos positivos com base em uma política de verificação dupla ou análise por IA adicional, quando necessário.

#### 4.2.2.6. Segurança

O sistema fará uso de criptografia TLS/SSL de ponta a ponta, assegurando a proteção dos dados transmitidos entre o cliente e o servidor. Além disso, será implementado um mecanismo de autenticação de usuários, garantindo que cada indivíduo tenha acesso exclusivamente ao seu próprio perfil.

A gestão de permissões será aplicada de forma individualizada, permitindo que cada usuário realize apenas as operações de inserção, exclusão e atualização relativas aos seus próprios dados. Da mesma forma, o acesso às informações será estritamente restrito ao respectivo perfil, sendo vedado o acesso a contas de terceiros. Dessa forma, cada perfil permanece único, privado e intransferível.

#### 4.2.2.7 Manutenibilidade

O sistema busca precisar de poucas manutenções pois, para efetuar a testabilidade será necessário treinar o modelo de detecção com quantidades consideráveis de imagens (*datasets*), sendo elas de diferentes condições impostas. Além do algoritmo em si que, uma vez que estiver entregando algum resultado, significa que o fluxo inteiro foi percorrido com sucesso.

#### 4.2.2.8 Portabilidade

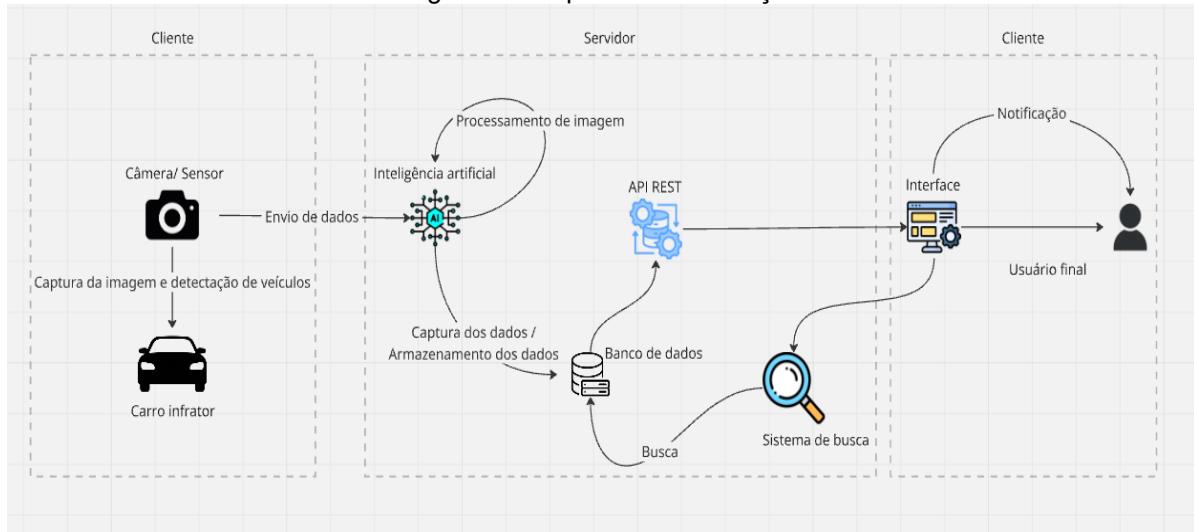
O sistema desenvolvido será compatível com qualquer eletrônico no qual esteja disponível um navegador web, seja ele PC ou mobile.

### 4.3 Arquitetura de Solução

A arquitetura de solução do sistema (Figura 4) é composta por diversos módulos integrados que viabilizam a detecção automatizada de infrações de estacionamento. O processo tem início com a captura de imagens por câmeras ou sensores, em seguida essas imagens são enviadas para um módulo de inteligência artificial, que realiza o processamento e identificação de veículos em situação irregular.

Os dados extraídos são capturados e armazenados em um banco de dados, sendo disponibilizados por meio de uma API REST para a interface do sistema que consome esses dados e permite que os usuários realizem buscas e visualizações de forma eficiente. Por fim, o sistema notifica o usuário final sempre que uma infração é identificada.

Figura 4 – Arquitetura de Solução

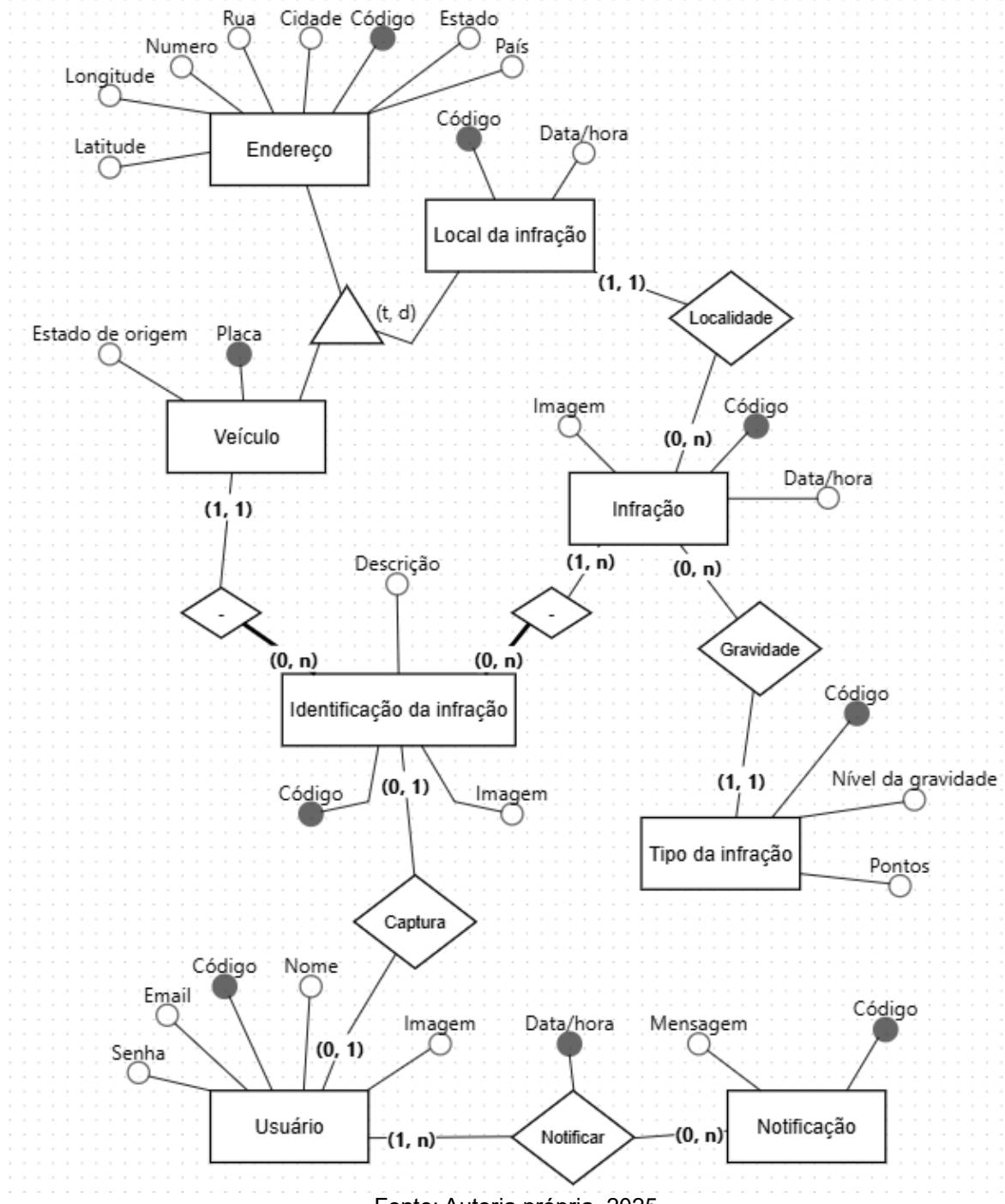


Fonte: Autoria própria, 2025

#### 4.4 Projeto lógico do banco de dados

O Modelo de Entidade Relacionamento (MER) presente (Figura 5) foi estruturado para refletir os principais elementos envolvidos no processo de identificação e registro de infrações, contemplando desde a captura das imagens até a notificação dos usuários.

Figura 5 – Modelo de Entidade Relacionamento (MER)



Fonte: Autoria própria, 2025

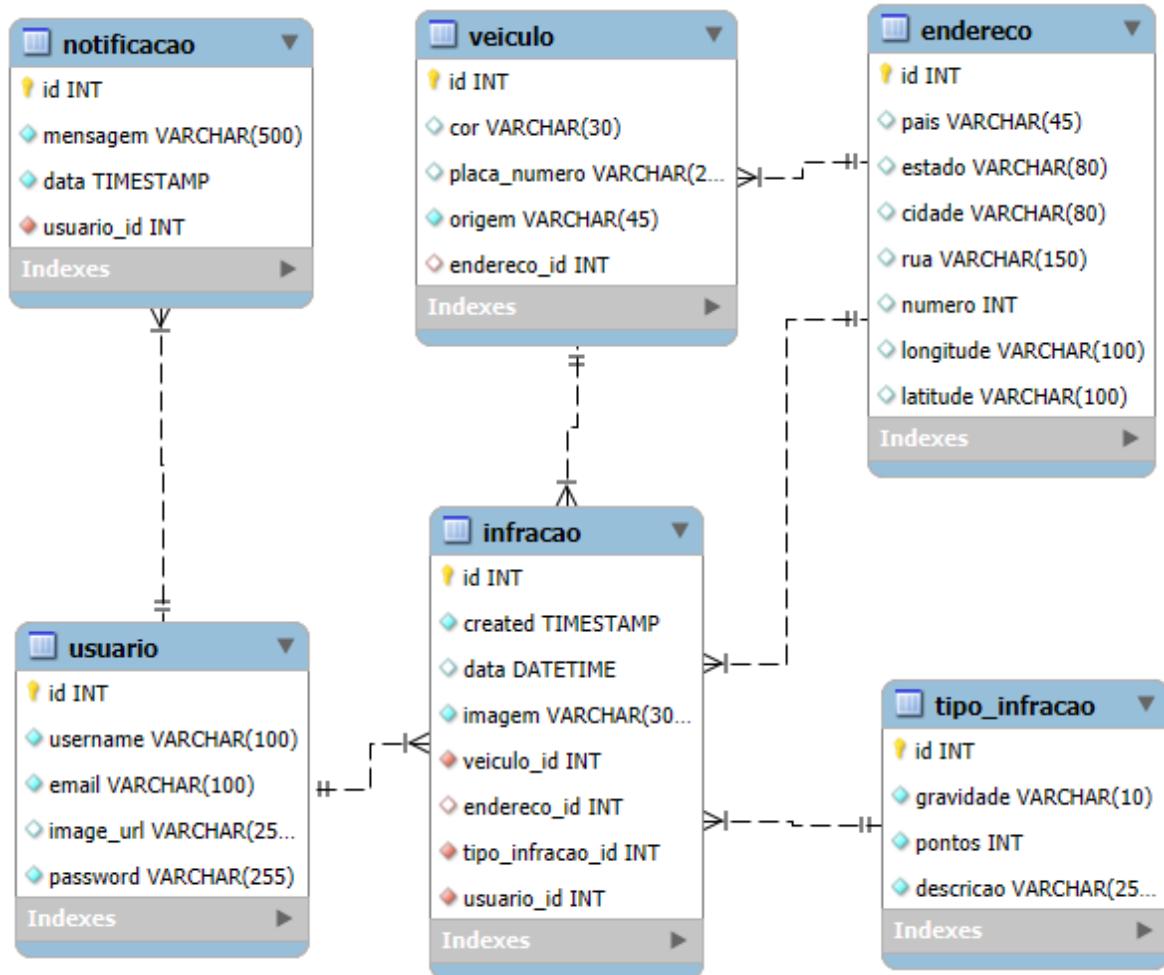
A seguir, uma breve descrição dos componentes principais:

- Usuário: Representa os cidadãos ou agentes que interagem com o sistema. Possui atributos como nome, e-mail, senha, imagem e uma preferência por notificações.
- Notificação: Armazena informações sobre mensagens enviadas aos usuários, com data e conteúdo.
- Veículo: Entidade identificada pela placa e estado de origem, essencial para o reconhecimento das infrações.
- Endereço e Local da infração: Detalham o local em que a infração ocorreu, com atributos com país, estado, cidade, rua, número, longitude, latitude e data/hora.
- Infração: Registra o evento infracional, com código, data/hora, imagem e relação com o tipo e gravidade da infração.
- Tipo da infração: Define a natureza da infração, nível de gravidade e os pontos atribuídos conforme o CTB (Código de Trânsito Brasileiro).
- Identificação da infração: Entidade que possui descrição e relaciona a imagem, o veículo e a infração capturada.

## 4.5 Projeto físico do banco de dados

O Diagrama Entidade Relacionamento (DER) apresentado (Figura 6) visa representar graficamente a estrutura lógica do banco de dados para o sistema de monitoramento de infrações de trânsito por meio de arquivos de imagem e inteligência artificial.

Figura 6 – Diagrama de Entidade Relacionamento (DER)



Fonte: Autoria própria, 2025

A seguir, uma breve descrição dos componentes principais:

Usuário:

- Responsável por acessar o sistema, consultar e enviar infrações.
- Atributos como username, e-mail, senha e imagem.
- Pode estar associado a notificações e infrações.

Veículo:

- Armazena informações como placa, cor e origem.
- Relacionado ao endereço e às infrações registradas.

Endereço:

- Representa a localização do evento, com país, estado, cidade, rua, número, longitude e latitude.
- Associado à entidade infração e veículo.

Infração:

- Entidade central do sistema.
- Contém dados como data da ocorrência, imagem registrada e “created” (data do envio).
- Relaciona-se com o veículo, endereço, tipo de infração e usuário.

Tipo\_Infração:

- Define a gravidade da infração, pontos e descrição.
- Associado diretamente à tabela infração.

Notificação:

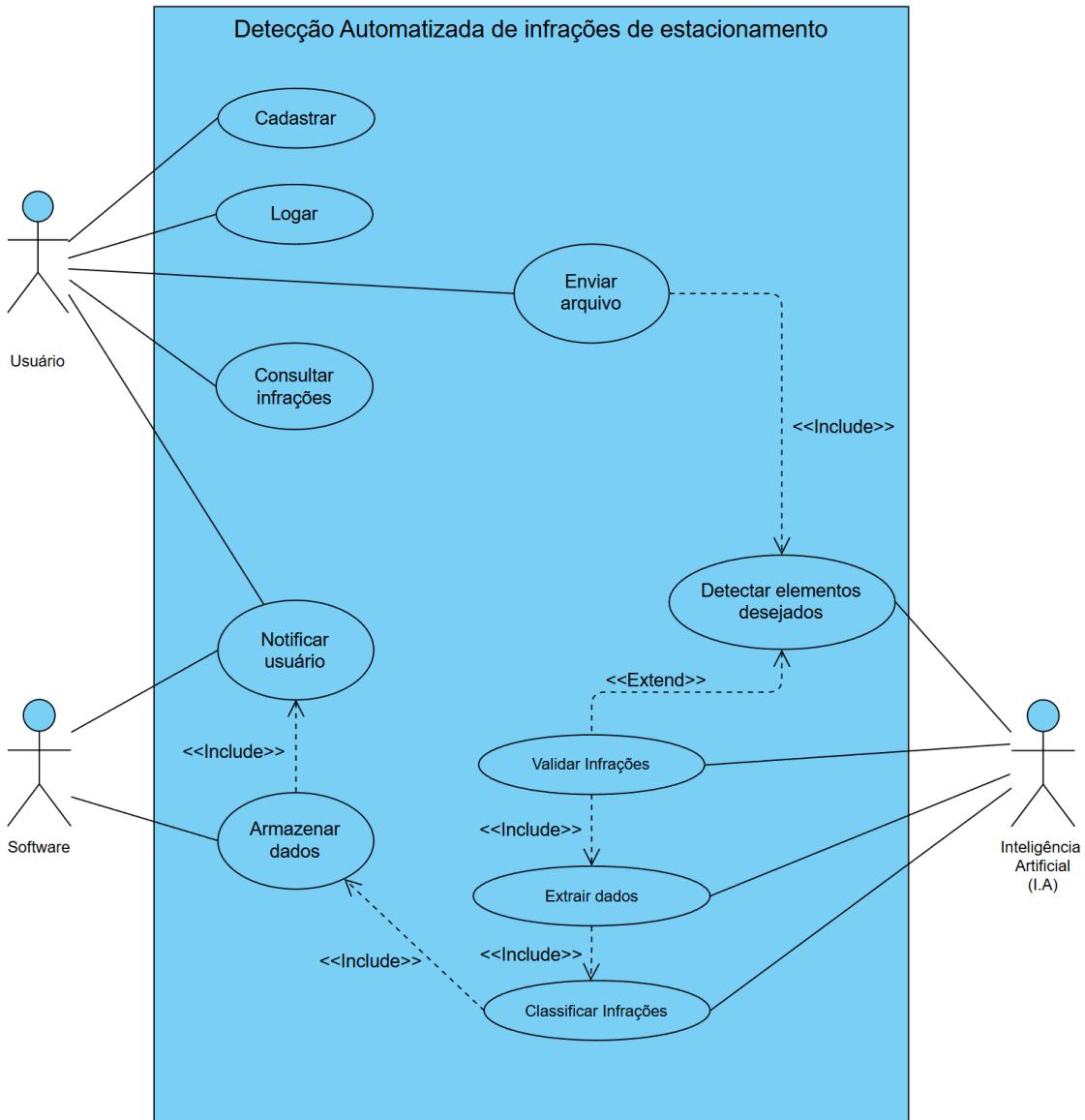
- Representa uma mensagem enviada ao usuário sobre uma infração.
- Contém texto da mensagem e data.
- Relaciona-se com a entidade usuário.

#### **4.6 Modelagem do Software**

A seção de modelagem do software apresenta os diagramas essenciais à representação funcional e estrutural do sistema. O diagrama de casos de uso expõe as funcionalidades disponibilizadas aos usuários e suas interações com o sistema. Em complemento, a documentação dos casos de uso formaliza os fluxos principais, as condições de execução e os tratamentos de exceções. O diagrama de classes modela as entidades e suas relações. Por fim, os diagramas de sequência ilustram os fluxos de mensagens entre os componentes do sistema em diferentes cenários operacionais.

#### 4.6.1 Diagrama de Caso de Uso

Figura 7 – Diagrama de Caso de Uso



Fonte: Autoria própria, 2025

O diagrama de caso de uso, representa todos os requisitos funcionais do projeto, em ação com os devidos atores estabelecidos. Conforme o diagrama (Figura 7), o ator Usuário tem como requisitos funcionais, cadastrar, logar, consultar infrações no sistema, como também enviar arquivos de diferentes formatos de imagem (png, jpg, etc.).

Por sequência, o ator Inteligência Artificial, é obrigado a detectar os elementos desejados nas imagens, tais como os veículos, sinalizações de estacionamento etc. Com base nessa detecção, pode-se validar se houve ou não uma ou mais infrações.

Em caso de ocorrência, o sistema de IA deve classificá-la conforme o tipo e o grau de gravidade, bem como extraír dados relevantes, tal como a cor do veículo, a placa de identificação, a data e o local.

Por último, o ator Software, é encarregado de armazenar os dados previamente extraídos, de mesmo modo que notifica aos usuários do software sempre que uma nova infração de trânsito for detectada e registrada.

#### 4.6.2 Documentação de caso de uso

Tabela 1 – Documentação de caso de uso 01 “Cadastrar”

Nome do caso de uso	UC01 – Cadastrar
Autor Principal	Usuário
Atores Secundários	
Resumo	Esse caso de uso descreve a etapa inicial em que o usuário realiza seu cadastro no sistema, por meio do preenchimento de suas credenciais
Pré-condições	O usuário não pode já estar logado para iniciar o cadastro
Pós-condições	
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Informar o nome	
	2. Verificar se o nome é válido
3. Informar o email	
	4. Verificar se o email é válido
5. Informar uma senha	
	6. Verificar se a senha é válida
	7. Armazenar os dados do usuário no banco de dados
Restrições/Validações	<ul style="list-style-type: none"> <li>1. O nome deve conter apenas caracteres válidos e não pode estar vazio.</li> <li>2. O email deve ter um formato válido e não estar cadastrado previamente.</li> <li>3. A senha deve atender aos critérios mínimos de segurança (ex.: 6 caracteres ou mais).</li> </ul>
Cenário de Exceção I – Email já cadastrado	
Ações do Ator	Ações do Sistema
	1. Informar que o email já está cadastrado
	2. Sugerir que o usuário faça login
Cenário de Exceção II – Nome, email e/ou senha Inválidos	
Ações do Ator	Ações do Sistema
	1. Comunicar ao usuário que existem campos inválidos
	2. Solicitar o preenchimento dos campos com dados válidos

Fonte: Autoria própria, 2025

Tabela 2 – Documentação de caso de uso 02 “Logar”

<b>Nome do caso de uso</b>		<b>UC02 – Logar</b>
<b>Autor Principal</b>		Usuário
<b>Atores Secundários</b>		
<b>Resumo</b>		Esse caso de uso descreve a etapa em que o usuário realiza seu login no sistema, através do preenchimento de suas credenciais.
<b>Pré-condições</b>		O usuário deve estar previamente cadastrado no sistema.
<b>Pós-condições</b>		O usuário será redirecionado à tela inicial após login bem-sucedido.
<b>Cenário Principal</b>		
<b>Ações do Ator</b>	<b>Ações do Sistema</b>	
1. Informar o email		
	2. Verificar se o email é válido	
3. Informar a senha		
	4. Buscar o email fornecido no banco de dados.	
	5. Verificar se a senha fornecida está de acordo com a do banco de dados.	
	6. Efetuar login do usuário.	
<b>Restrições/Validações</b>	1. Para logar, é necessário ter feito um cadastro com o mesmo email.	
<b>Cenário de Exceção I – Email não encontrado no banco</b>		
<b>Ações do Ator</b>	<b>Ações do Sistema</b>	
	1. Comunicar ao usuário que o email fornecido não foi encontrado	
	2. Solicitar o preenchimento dos campos com dados válidos	
	3. Sugerir ao usuário para fazer o cadastro, caso não tenha feito	
<b>Cenário de Exceção II – Dados Inválidos</b>		
<b>Ações do Ator</b>	<b>Ações do Sistema</b>	
	1. Sistema comunica ao usuário que um dos seus dados está inválido	
	2. Solicitar o preenchimento dos campos de texto com dados válidos	

Fonte: Autoria própria, 2025

Tabela 3 – Documentação de caso de uso 03 “Enviar Arquivo”

Nome do caso de uso	UC03 – Enviar Arquivo
Ator Principal	Usuário
Atores Secundários	
Resumo	Esse caso de uso descreve as possíveis etapas que o usuário deve percorrer para efetuar o envio de um arquivo no sistema a fim de validar uma infração.
Pré-condições	O usuário deve estar logado no sistema
Pós-condições	
Cenário Principal	
Ações do Ator	Ações do Sistema
	1. Abrir o explorador de arquivos
2. Selecionar o arquivo	
	3. Verificar se o tipo de arquivo é válido
	4. Verificar se o tamanho do arquivo é válido
	5. Mandar o arquivo para a IA processar
	6. Exibir um ícone de carregamento ao usuário
Restrições/Validações	1. O arquivo deve estar no formato correto 2. O arquivo não deve ultrapassar o limite do tamanho
Cenário de Exceção I – Imagem inválida	
Ações do Ator	Ações do Sistema
	1. Informar ao usuário que a imagem selecionada é inválida
	2. Informar que a imagem deve estar em png, jpeg ou jpg
	3. Informar que a imagem não deve passar do tamanho de 5mb

Fonte: Autoria própria, 2025

Tabela 4 – Documentação de caso de uso 04 “Consultar Infrações”

Nome do caso de uso	UC04 – Consultar infrações
Autor Principal	Usuário
Atores Secundários	
Resumo	Esse caso de uso descreve os passos necessários para o usuário realizar uma consulta das infrações registradas no banco de dados do sistema.
Pré-condições	O usuário deve estar logado no sistema
Pós-condições	
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Informar placa do veículo	
	2. Fazer pesquisa da placa do veículo no banco de dados
	3. Sintetizar dados de Infração para fácil visualização
	4. Exibir ao usuário uma lista de infrações do veículo
Restrições/Validações	1. A placa informada deve ser válida
Cenário de Exceção I – Placa de veículo inválida	
Ações do Ator	Ações do Sistema
	1. Comunicar ao usuário que a numeração da placa informada é inválida
	2. Solicitar o preenchimento dos campos com dados válidos
Cenário de Exceção II – Nenhuma Infração encontrada	
Ações do Ator	Ações do Sistema
	1. Comunicar ao usuário que não foi encontrado nenhuma infração para aquele veículo

Fonte: Autoria própria, 2025

Tabela 5 – Documentação de caso de uso 05 “Detectar elementos desejados”

Nome do caso de uso	UC05 – Detectar elementos desejados
Ator Principal	Inteligência Artificial
Atores Secundários	Software
Resumo	Esse caso de uso descreve a função da Inteligência Artificial de detectar passivamente os elementos presentes em imagens, utilizadas como entrada.
Pré-condições	Receber arquivos de imagem como entrada
Pós-condições	
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Processar o(s) frame(s) necessários no modelo de visão computacional	
	2. Salvar e enviar os frames rotulados para próxima etapa
Restrições/Validações	1. Ter pelo menos 1 frame com elementos desejados no arquivo
Cenário de Exceção I – Nenhum elemento detectado	
Ações do Ator	Ações do Sistema
	1. Comunicar ao usuário que nenhum elemento desejado foi detectado no arquivo
	2. Solicitar envio de outro arquivo

Fonte: Autoria própria, 2025

Tabela 6 – Documentação de caso de uso 06 “Validar Infrações”

Nome do caso de uso	UC06 – Validar infrações
Ator Principal	Inteligência Artificial
Atores Secundários	Software
Resumo	Esse caso de uso descreve a etapa em que a Inteligência Artificial deve identificar e processar automaticamente os frames em que uma infração de estacionamento estiver sendo cometida.
Pré-condições	Receber elementos desejados, já detectados
Pós-condições	
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Analisar elementos detectados	
	2. Processar os elementos com base nos padrões definidos
3. Identificar frames com possível infração	
	4. Validar se os elementos configuram uma infração real
5. Classificar os frames como válidos ou descartáveis	
	6. Salvar apenas os frames válidos rotulados para a próxima etapa
Restrições/Validações	1. Apenas serão considerados válidos os frames em que a posição do veículo infrator for consistente com as regras locais (ex.: em cima da faixa, em cima da calçada, etc.) 2. O sistema não deve validar múltiplas vezes o mesmo frame (evitar duplicidade)
Cenário alternativo – Nenhuma infração detectada	
Ações do Ator	Ações do Sistema
1. Analisar elementos detectados	
	2. Processar os elementos com base nos padrões definidos
	3. Comunicar ao usuário que a taxa de frames identificados como infração é mínima
	4. Aguardar o recebimento de novos frames para análise

Fonte: Autoria própria, 2025

Tabela 7 – Documentação de caso de uso 07 “Classificar tipo e gravidade da infração”

Nome do caso de uso	UC07 – Classificar tipo e gravidade da infração
Autor Principal	Inteligência Artificial
Atores Secundários	
Resumo	Esse caso de uso descreve a função da Inteligência Artificial de classificar, qual o tipo de infração de estacionamento cometida, e qual sua gravidade.
Pré-condições	Os frames recebidos devem estar rotulados como infração cometida
Pós-condições	
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Recebe os frames classificados como infração	
	2. Analisar os frames e atribuir um tipo de infração de acordo com as predefinições.
	3. Analisar o tipo da infração para atribuir gravidade de acordo com as predefinições
	4. Guardar dados em variáveis para serem usadas na etapa de armazenamento
Restrições/Validações	1. Os frames devem conter informações visuais e completas para classificação (ex.: todos elementos necessários rotulados). 2. Reconhecer o tipo de infração como válida
Cenário de Exceção I – Nenhum tipo de infração compatível	
Ações do Ator	Ações do Sistema
	1. Comunicar ao usuário que não foi possível atribuir a infração com um dos tipos existentes
	2. Aguardar o recebimento de novos frames para análise

Fonte: Autoria própria, 2025

Tabela 8 – Documentação de caso de uso 08 “Extrair dados”

Nome do caso de uso	UC08 – Extrair dados
Ator Principal	Inteligência Artificial
Atores Secundários	
Resumo	Esse caso de uso descreve a etapa em que a Inteligência Artificial realiza a extração de informações relevantes a partir dos frames previamente classificados como infrações.
Pré-condições	As infrações devem estar validadas e classificadas pela IA
	Cenário Principal
Ações do Ator	Ações do Sistema
1. Recebe os frames classificados como infração	
	2. Aplicar algoritmos para extrair dados como: placa, horário, tipo de infração, localização etc.
	3. Verificar a consistência e validade das informações extraídas
	4. Organizar os dados em formato estruturado (ex.: JSON ou tabelas)
	5. Enviar os dados para o módulo de armazenamento
Restrições/Validações	<ol style="list-style-type: none"> <li>1. Os frames devem conter informações visuais nítidas e completas para extração (ex.: placa visível, sinalização legível).</li> <li>2. A localização deve ser compatível com dados de georreferenciamento (se fornecido)</li> <li>3. O tipo de infração deve estar classificado corretamente para contextualização dos dados</li> <li>4. Os dados extraídos devem passar por uma checagem de integridade antes de serem armazenados</li> </ol>
	Cenário de Exceção I – Dados incompletos ou ilegíveis
Ações do Ator	Ações do Sistema
1. Tentar extrair os dados de um frame	
	2. Detectar ausência ou má qualidade dos elementos necessários (ex.: placa desfocada, sinalização ausente)
	3. Marcar o frame como “falha na extração”
	4. Registrar o erro para reprocessamento posterior

Fonte: Autoria própria, 2025

Tabela 9 – Documentação de caso de uso 09 “Armazenar dados”

Nome do caso de uso	UC09 – Armazenar dados
Autor Principal	Software
Resumo	Este caso de uso descreve o processo em que o sistema (Software) armazena, de forma estruturada e persistente, os dados extraídos e classificados pela Inteligência Artificial. As informações salvas são posteriormente disponibilizadas ao usuário por meio do site web do sistema.
Pré-condições	Os dados extraídos devem estar organizados, validados e classificados pela IA.
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Receber os dados estruturados da IA	
	2. Verificar se os dados estão completos e consistentes
	3. Estabelecer conexão com o banco de dados
	4. Inserir os dados em suas respectivas tabelas (ex.: placa, tipo da infração, horário, localização)
	5. Confirmar a persistência dos dados no banco de dados
	6. Disponibilizar os dados para o <i>front-end</i> (site web) para consulta dos usuários
	7. Exibir as informações no site web, de acordo com os critérios de busca
Restrições/Validações	1. Todos os campos obrigatórios devem estar preenchidos (ex.: placa, tipo da infração, data/hora). 2. O sistema deve impedir duplicação de registros idênticos para o mesmo evento.
Cenário Alternativo – Armazenamento parcial de dados	
Ações do Ator	Ações do Sistema
1. Receber dados estruturados da IA	
	2. Identificar falta de certos dados
	3. Rotular um ou mais campos como incompletos ou impossível de identificar.

Fonte: Autoria própria, 2025

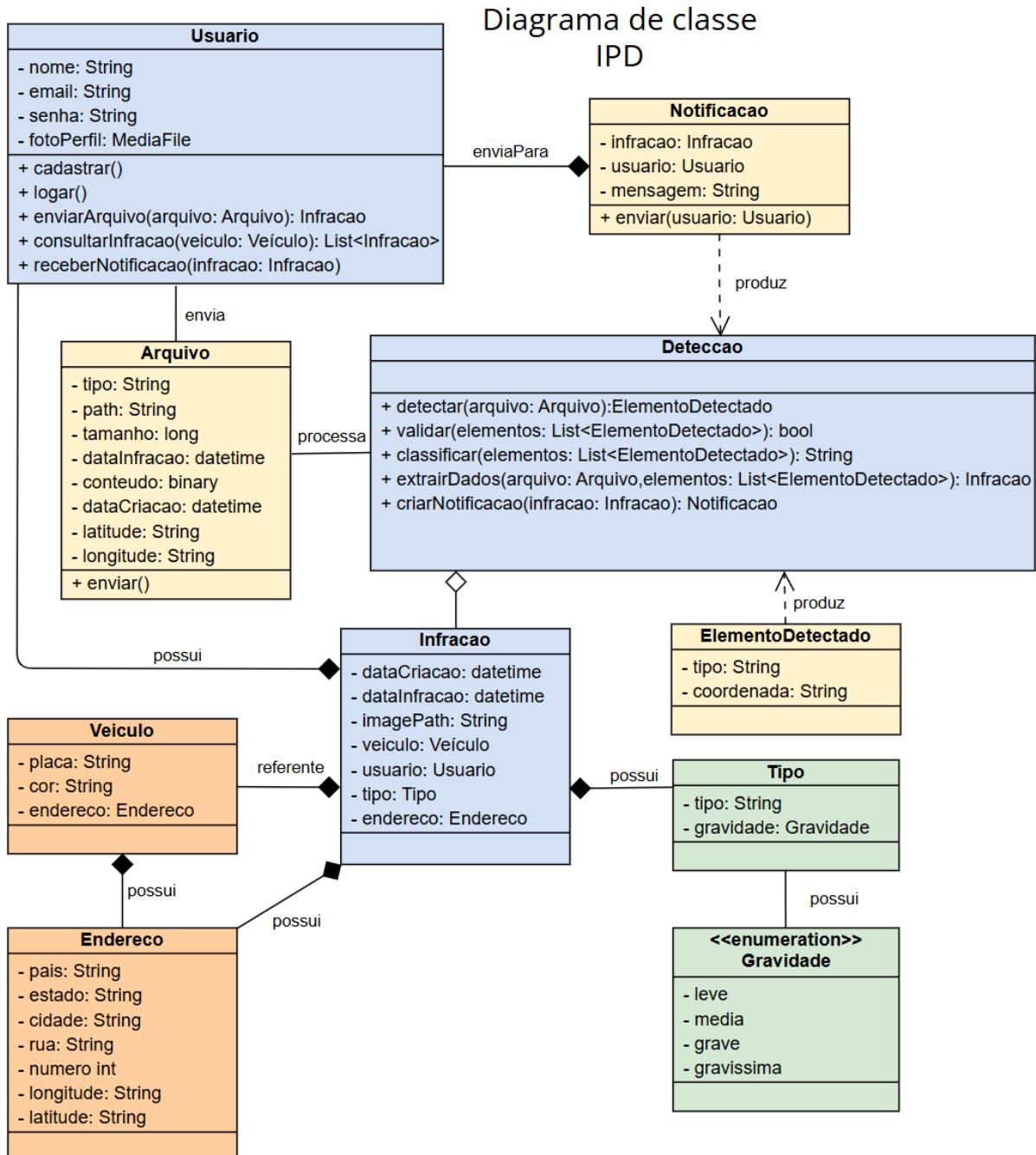
Tabela 10 – Documentação de caso de uso 10 “Notificação”

Nome do caso de uso	UC10 – Notificação
Ator Principal	Software
Atores Secundários	
Resumo	Esse caso de uso descreve a etapa em que o sistema (Software) envia para o usuário a notificação de que uma infração de estacionamento foi identificada.
Pré-condições	Verificar se algum dado contendo alguma infração foi guardada no banco de dados.
Pós-condições	
Cenário Principal	
Ações do Ator	Ações do Sistema
	1. Verificar se algum dado com infração foi inserido no banco de dados.
	2. Resgatar os dados da infração e as informações do usuário.
	3. Classificar os dados e configurar o tipo da mensagem.
	4. Enviar através de um middleware a notificação para o usuário de destino.
Restrições/Validações	1. Todas as informações devem estar preenchidas em suas tabelas correspondentes. 2. As informações devem conter uma semântica de valor coerente. 3. O endereço final da notificação deve ser um valor válido.
Cenário de Exceção I – Dados incompletos ou ilegíveis	
Ações do Ator	Ações do Sistema
	1. Verificar se existe algum valor como nulo ou como indefinido.
	2. Detectar se o middleware usado para o envio de mensagem obteve uma resposta de êxito ao enviar a mensagem.

Fonte: Autoria própria, 2025

#### 4.6.3 Diagrama de Classe

Figura 8 – Diagrama de Classes



Fonte: Autoria própria, 2025

O Diagrama de Classes (Figura 8) é composto por classes que simulam a detecção automática de infrações de estacionamento. A classe “Arquivo”, que contém dados como tipo, caminho e conteúdo da imagem, são enviados por um Usuário, que pode se cadastrar, logar, enviar arquivos e consultar infrações.

A classe “Detecção” processa os arquivos, detectando elementos como placas ou sinais, validando-os, classificando e extraíndo os dados para gerar uma “Infração”. Cada infração está associada a um “Veículo”, um “Endereço”, um “Tipo” e uma “Gravidade” (leve, média, grave ou gravíssima). Ao final, uma “Notificação” é criada e enviada ao usuário responsável.

O Diagrama de Classes reflete um fluxo automatizado de captura da imagem até o envio de uma notificação sobre a infração, integrando software (sistema de detecção) e interação com o usuário.

#### 4.6.4 Diagramas de Sequência (DS)

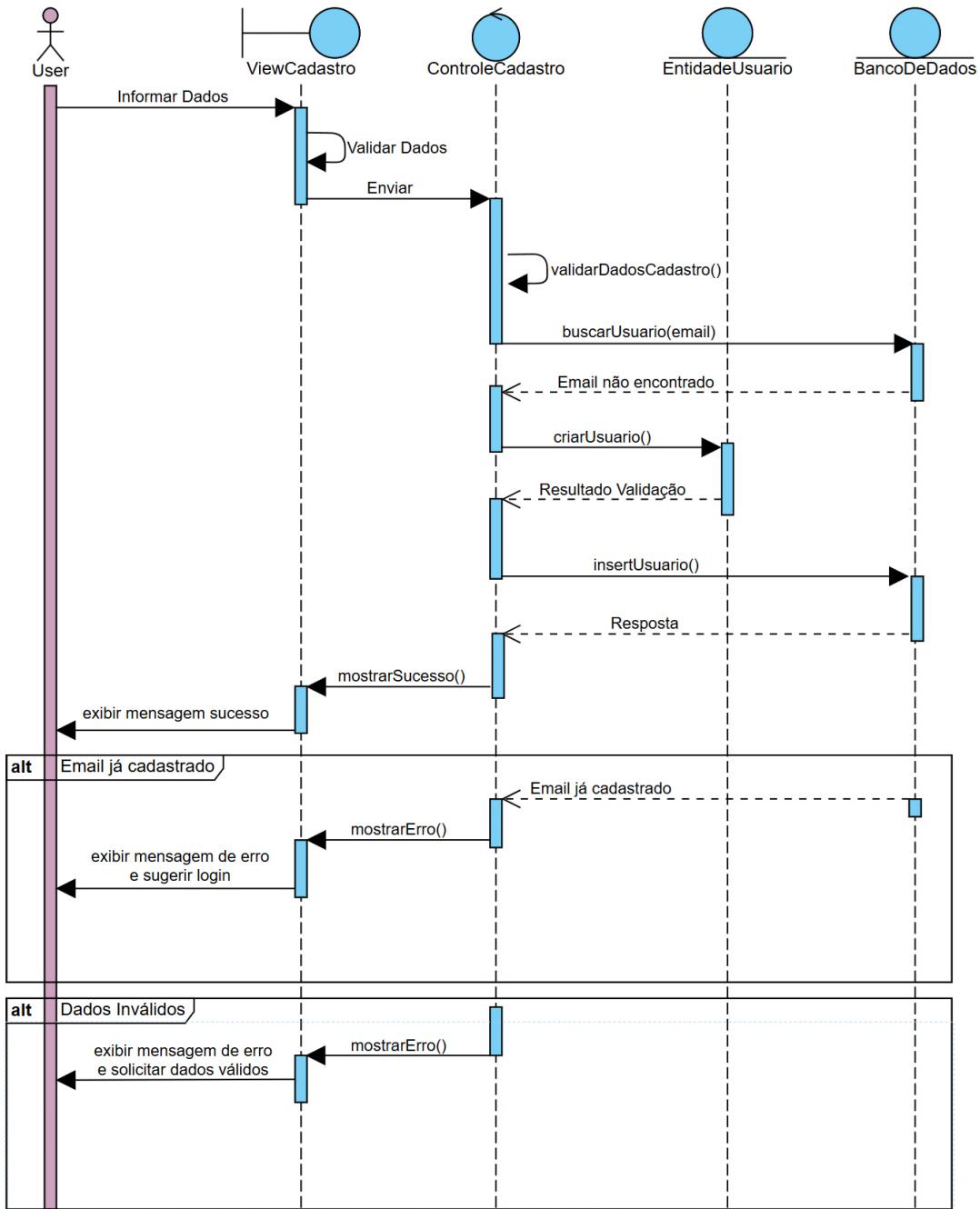
##### 4.6.4.1 DS Cadastro

O diagrama de sequência "Cadastro" (Figura 9) representa o fluxo em que o usuário fornece seus dados para realizar o registro no sistema. As informações são inseridas na interface e encaminhadas ao controlador, que realiza a validação dos dados. Estando os dados corretos, o sistema cria a instância do novo usuário e armazena suas informações no banco de dados. Ao final do processo, uma mensagem de confirmação é exibida ao usuário, indicando que o cadastro foi concluído com sucesso.

O diagrama também contempla dois cenários de exceção. O primeiro ocorre quando o e-mail informado já está cadastrado no sistema, impedindo a duplicidade e exigindo uma nova tentativa com outro endereço. O segundo cenário acontece quando algum dos dados fornecidos é inválido, levando o sistema a exibir uma mensagem de erro e solicitar que o usuário preencha novamente as informações de forma correta.

Figura 9 – Diagrama de Sequência “Cadastro”

sd cadastro



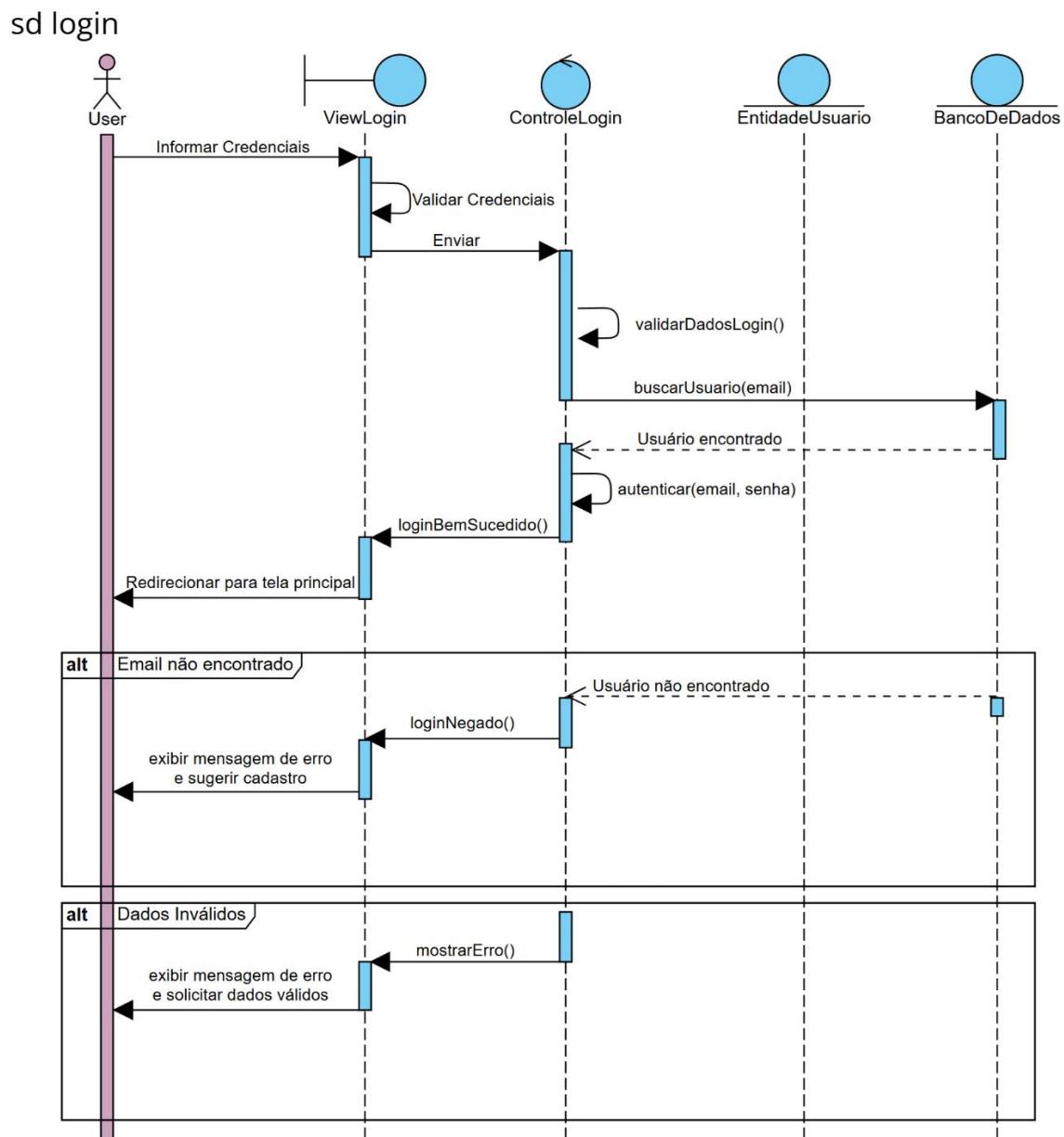
Fonte: Autoria própria, 2025

#### 4.6.4.2 DS Login

O diagrama de sequência "Login" (Figura 10) representa o fluxo em que o usuário informa suas credenciais para acessar o sistema. Após o envio das informações pela interface, os dados são validados e encaminhados ao controlador, que realiza a busca do usuário no banco de dados. Caso o usuário seja encontrado e as credenciais estejam corretas, a autenticação é realizada com sucesso, e o sistema redireciona o usuário para a tela principal.

O diagrama também contempla dois cenários de exceção. O primeiro ocorre quando o e-mail informado não está cadastrado no sistema, fazendo com que uma mensagem de erro seja exibida, acompanhada da sugestão de realizar o cadastro. O segundo cenário ocorre quando os dados preenchidos são inválidos ou inconsistentes, sendo então apresentada ao usuário uma mensagem de erro com a solicitação de correção.

Figura 10 – Diagrama de Sequência “Login”



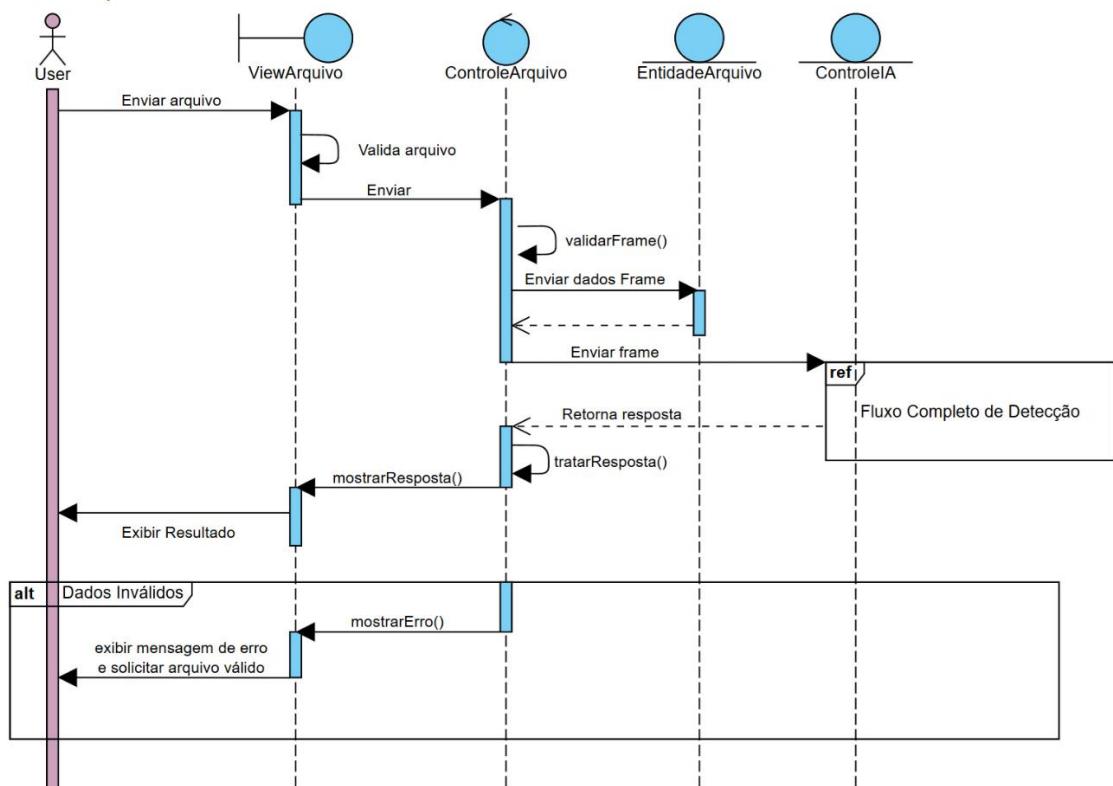
Fonte: Autoria própria, 2025

#### 4.6.4.3 DS Enviar Arquivo

O diagrama de sequência "Enviar Arquivo" (Figura 11) representa o processo em que o usuário realiza o envio de um arquivo para o sistema, contendo os dados visuais que serão analisados. O arquivo é inicialmente validado pela interface e, em seguida, encaminhado ao controlador responsável, que realiza a verificação do conteúdo e extrai os frames relevantes. Após essa etapa, os dados são direcionados ao módulo de detecção representado por um diagrama de sequência separado, identificado no modelo pelo fragmento de referência "ref".

Ao final do processamento, uma resposta é gerada e retornada ao controlador, que trata o resultado e envia o retorno para o usuário. O fluxo também contém um cenário de exceção, no qual o arquivo enviado contém dados inválidos. Nessa situação, o sistema informa o erro e solicita ao usuário o reenvio de um arquivo válido.

Figura 11 – Diagrama de Sequência “Enviar Arquivo”  
sd Enviar Arquivo



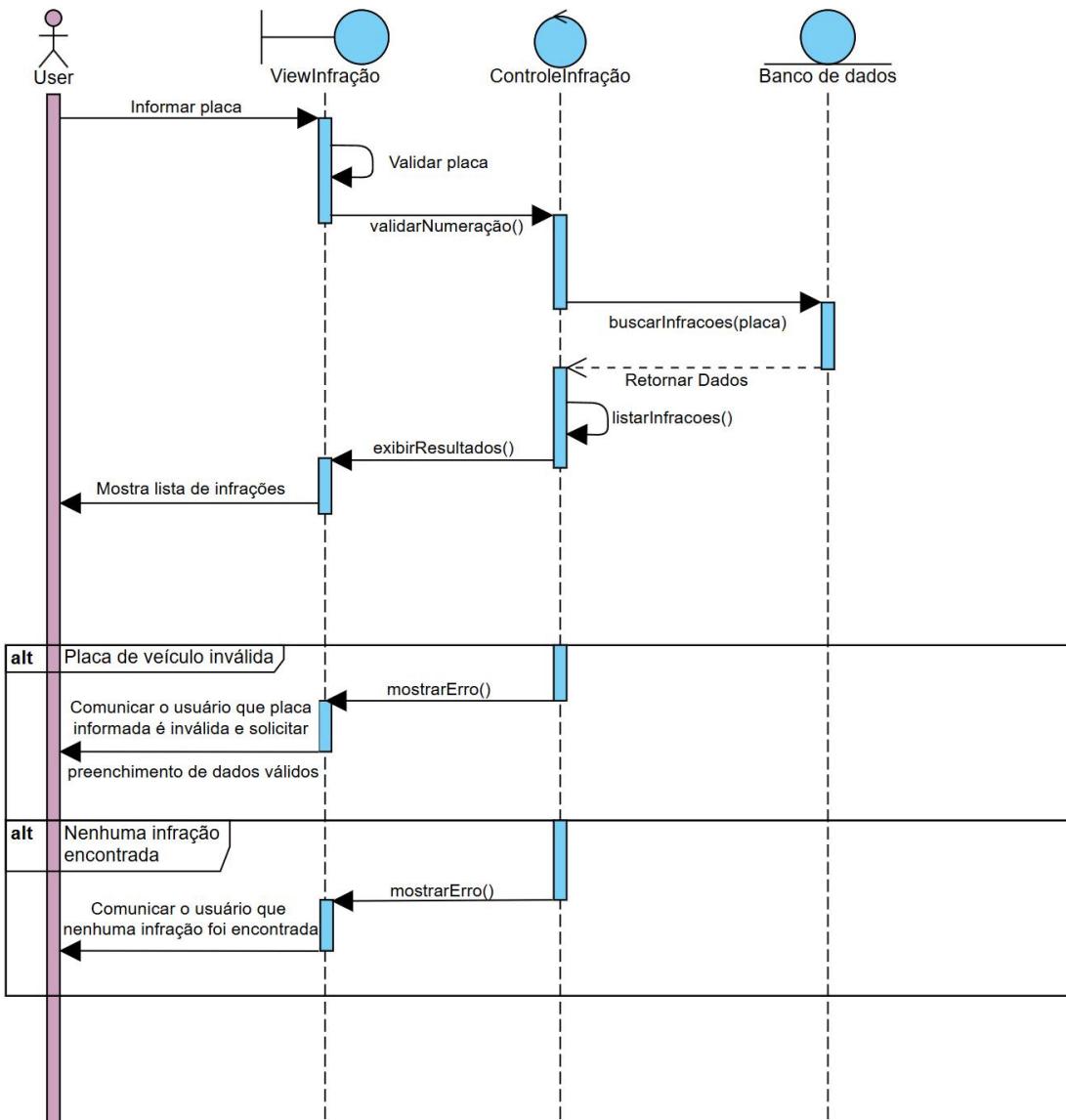
Fonte: Autoria própria, 2025

#### 4.6.4.4 DS Consultar Infrações

O diagrama de sequência "Consultar Infrações" (Figura 12) representa o fluxo de interação em que o usuário informa a placa de um veículo para consultar se há infrações registradas. O dado é inicialmente validado pela interface e, em seguida, encaminhado ao controlador, que por sua vez consulta o banco de dados. Caso a placa seja válida e existam infrações associadas, os resultados são listados e exibidos ao usuário.

O fluxo também contempla dois cenários de exceção. O primeiro ocorre quando a placa informada é inválida, o que aciona uma resposta de erro solicitando ao usuário o preenchimento de dados corretos. O segundo cenário trata da ausência de infrações registradas para uma placa válida, em que o sistema comunica ao usuário que nenhuma infração foi encontrada.

Figura 12 – Diagrama de Sequência “Consultar Infrações”  
sd Consultar Infrações



Fonte: Autoria própria, 2025

#### 4.6.4.5 DS Fluxo completo de detecção

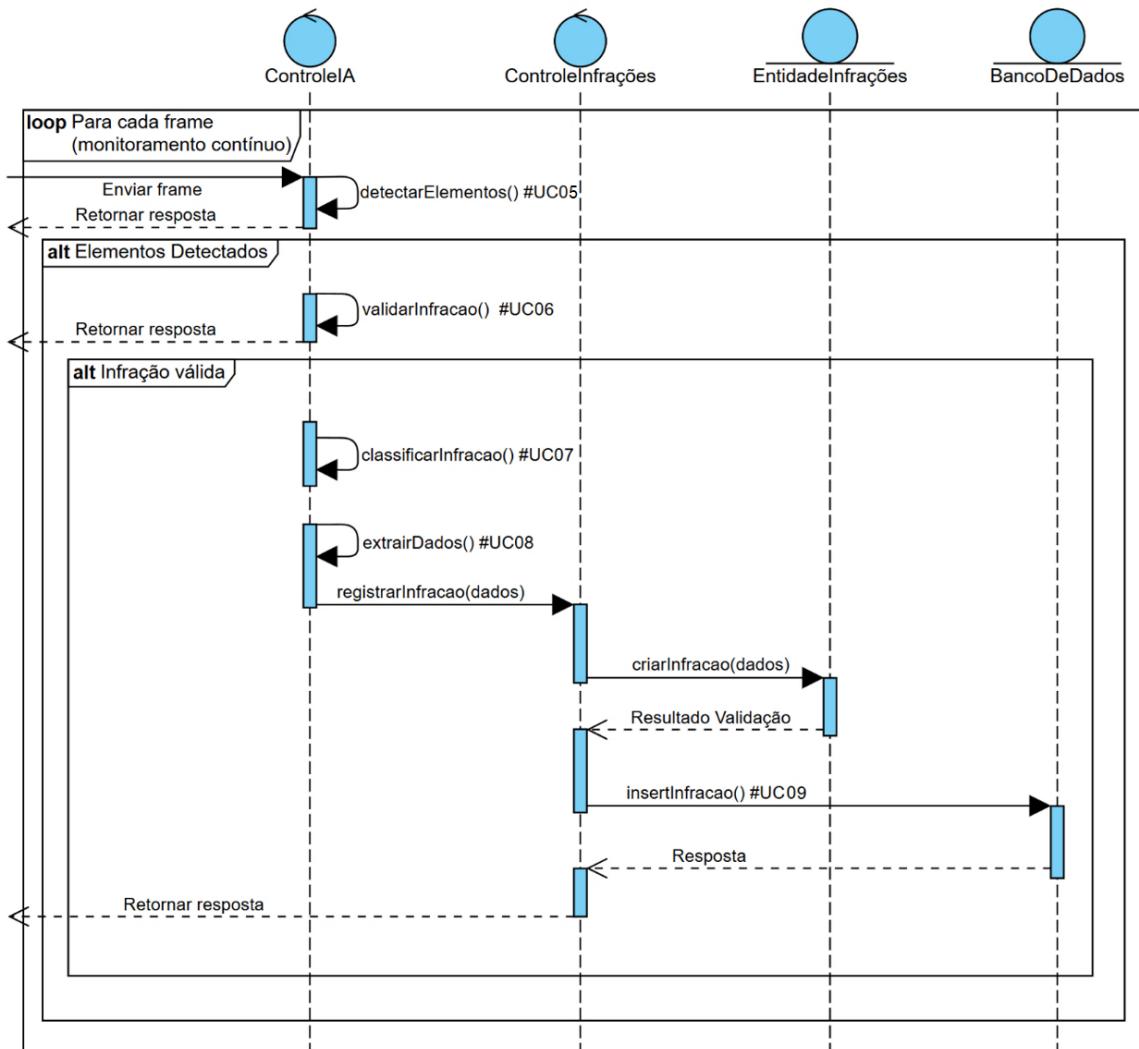
O diagrama de sequência "Fluxo completo de detecção" (Figura 13) representa o núcleo funcional do sistema de monitoramento automatizado de infrações, sendo o principal responsável pela identificação e registro de irregularidades a partir de imagens capturadas. Este fluxo ocorre de forma contínua, iniciando com um laço de repetição (loop) que monitora constantemente os frames recebidos. A cada iteração, o sistema realiza a análise do conteúdo visual com o objetivo de identificar elementos relevantes, como veículos, placas e sinalizações de trânsito.

O processo tem início com a chamada do caso de uso de detecção de elementos (UC05), o qual avalia se há objetos reconhecíveis nos frames analisados. Caso nenhum elemento seja encontrado, o fluxo é interrompido de forma segura, retornando uma resposta informativa ao sistema externo. No entanto, se a detecção for bem-sucedida, o fluxo avança para a próxima etapa, que consiste na validação da ocorrência de uma infração (UC06). Nesta fase, os elementos detectados são avaliados em relação a critérios legais e contextuais, como posição do veículo e sinalizações visuais.

Se a validação não indicar nenhuma infração, o sistema finaliza a análise daquele frame, enviando uma resposta que sinaliza a ausência de irregularidades. Contudo, se for identificada uma infração válida, o sistema passa para uma sequência de etapas obrigatórias. Inicialmente, é realizada a classificação da infração (UC07), determinando seu tipo específico (por exemplo, estacionamento em local proibido sinalizado por placa), como também sua gravidade (leve, média, grave ou gravíssima). Em seguida, ocorre a extração dos dados relevantes (UC08), como placa e cor do veículo, localização da ocorrência, data e horário.

Por fim, os dados extraídos são utilizados para a criação e o registro formal da infração (UC09). Isso envolve a instância da entidade correspondente e a persistência das informações no banco de dados do sistema. Uma vez armazenados, esses dados poderão ser consultados e notificados ao usuário. A cada final de ciclo, o fluxo retorna uma resposta conclusiva, seja informando a ocorrência de uma infração registrada, seja apontando a ausência de elementos ou irregularidades detectadas. Este modelo dinâmico e modular garante flexibilidade e autonomia ao sistema, permitindo que ele opere de forma contínua, sem intervenção humana direta.

Figura 13 – Diagrama de Sequência “Fluxo completo de detecção”  
sd Fluxo completo de detecção



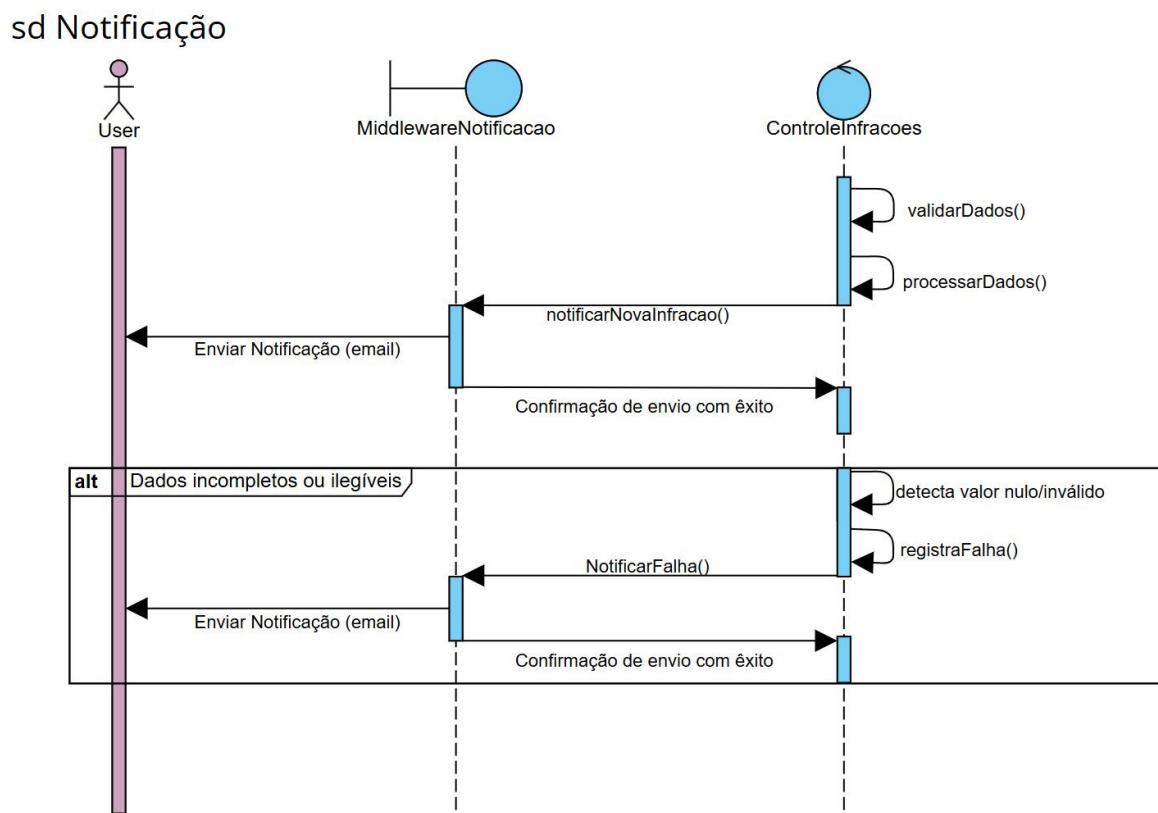
Fonte: Autoria própria, 2025

#### 4.6.4.6 DS Notificação

O diagrama de sequência "Notificação" (Figura 14) descreve o fluxo responsável por enviar ao usuário uma notificação relacionada ao registro de uma nova infração. A partir dos dados processados pelo controlador de infrações, uma solicitação é enviada ao middleware responsável pela entrega, que então dispara o envio da notificação ao usuário por e-mail, confirmando o sucesso da operação.

O diagrama também contempla um cenário de exceção. Caso os dados recebidos estejam incompletos ou ilegíveis, o sistema detecta a falha, regista o erro e executa uma notificação alternativa informando a ocorrência do problema. Ainda assim, há uma confirmação de envio, assegurando que o usuário será informado da tentativa, mesmo que sem sucesso total.

Figura 14 – Diagrama de Sequência “Notificação”



Fonte: Autoria própria, 2025

## 5. PROTÓTIPO DO PROJETO

### 5.1 Tela inicial

A interface de usuário da tela inicial do sistema IPD é clean e intuitiva. No topo, uma barra branca com o logo "IPD" (que inclui um ícone de carro) e links para "Login" e "Criar Conta". O título principal, "Monitoramento inteligente de infrações de trânsito", é exibido em uma grande fonte com destaque para "inteligente" e "infrações". Abaixo do título, uma descrição curta explica o propósito do sistema: "Identifique placas, valide infrações automaticamente com IA e gerencie ocorrências em um único painel intuitivo e seguro." Duas opções de ação estão disponíveis: "Acessar Painel" (botão verde com ícone de usuário) e "Começar Agora" (botão branco com ícone de carro). Abaixo desses botões, três seções destacadas fornecem informações sobre a funcionalidade: "Envie Evidências" (com ícone de câmera), "Validação via IA" (com ícone de faísca) e "Histórico Completo" (com ícone de relógio). Cada seção contém uma breve descrição. No rodapé, uma barra cinza com o aviso de direitos autorais: "© 2025 IPD - Detecção de Infrações. Todos os direitos reservados."

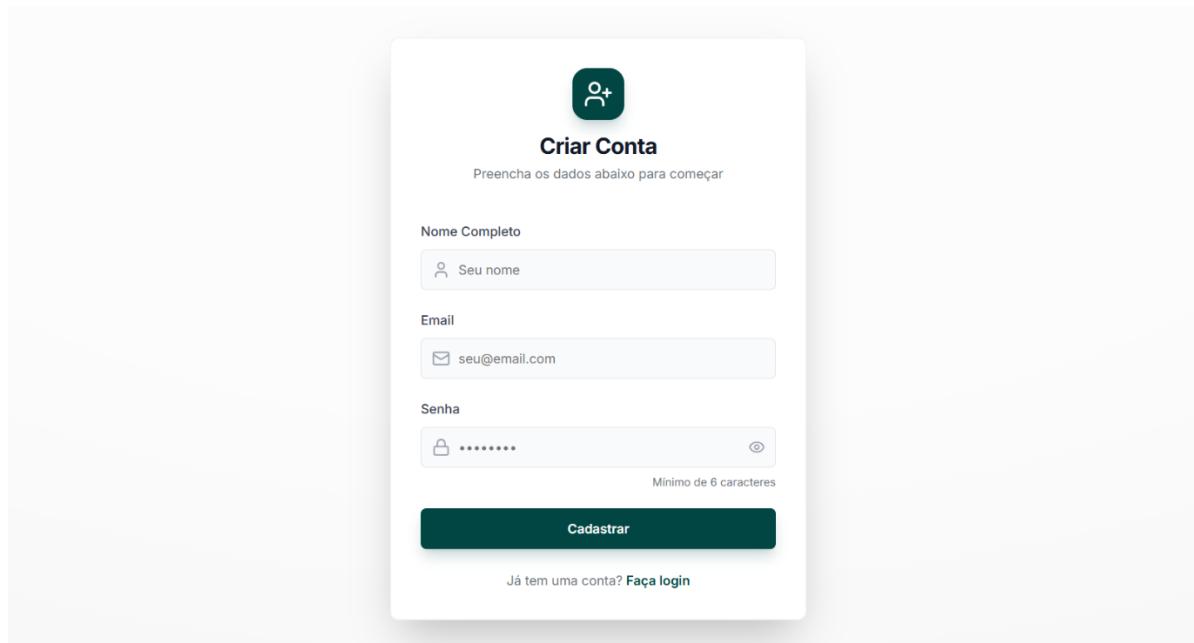
Fonte: Autoria própria, 2025

### 5.2 Tela de login

A interface de usuário da tela de login é simples e direta. Um formulário centralizado pede credenciais para "Entrar na conta". Ele contém campos para "Email" (exemplo@email.com) e "Senha" (campo com placeholder "\*\*\*\*\*" e ícone para alternar visibilidade). Um botão "Entrar na conta" em destaque encerra o formulário. Abaixo do formulário, uma mensagem informa que novos usuários podem "Cadastrar-se agora". A tela é rodeada por uma barra cinza.

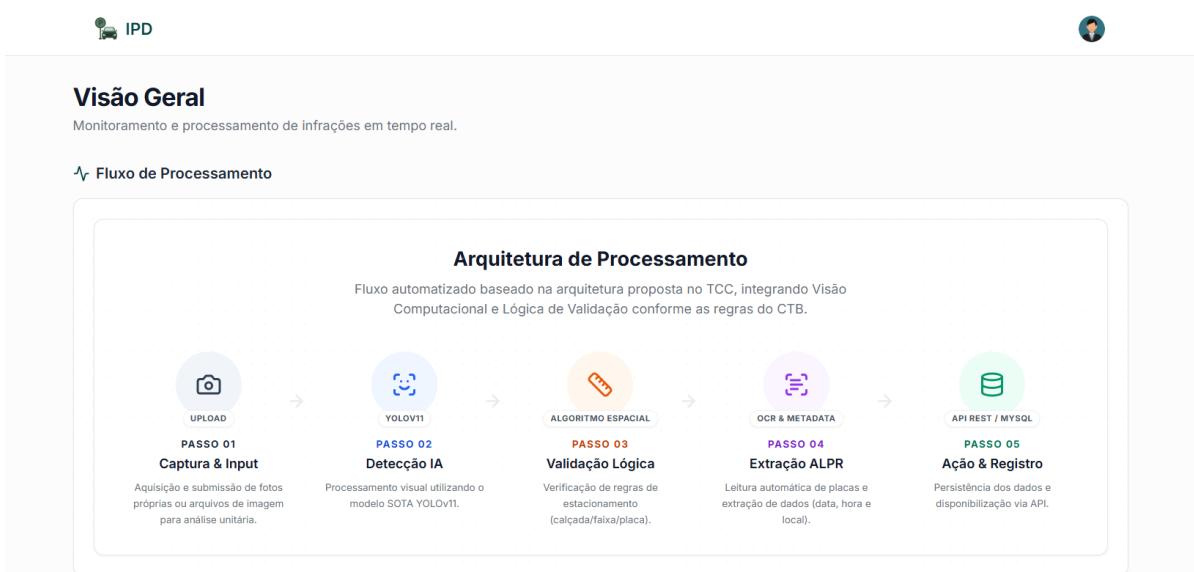
Fonte: Autoria própria, 2025

### 5.3 Tela de cadastro



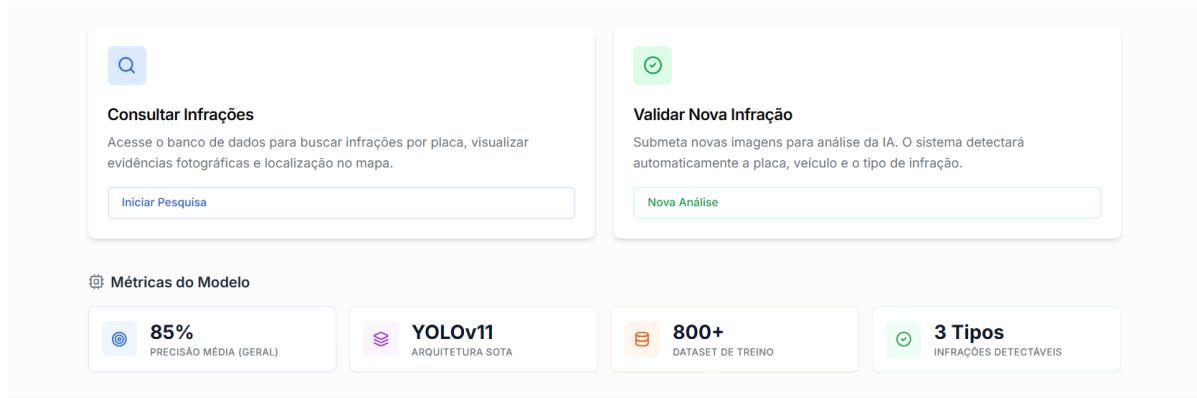
Fonte: Autoria própria, 2025

### 5.4 Tela inicial após autenticação (fluxo de processamento)



Fonte: Autoria própria, 2025

### 5.4.1 Tela inicial após autenticação (funcionalidades e métricas)



Fonte: Autoria própria, 2025

### 5.5 Tela de consultar infrações (vazia)



Fonte: Autoria própria, 2025

### 5.5.1 Tela de consultar infrações onde placa não encontrada

The screenshot shows a web-based traffic violation search system. At the top, there's a header with a car icon and the text 'IPD'. Below it is a main title 'Consulta de Infrações' and a subtitle 'Pesquise o histórico de infrações pela placa do carro.' A search input field contains the license plate 'COM6864', and a search button with a magnifying glass icon is to its right. A pink message bar below the input says 'Nenhuma infração encontrada para esta placa' (No violations found for this plate). The background is white with some light gray shadows.

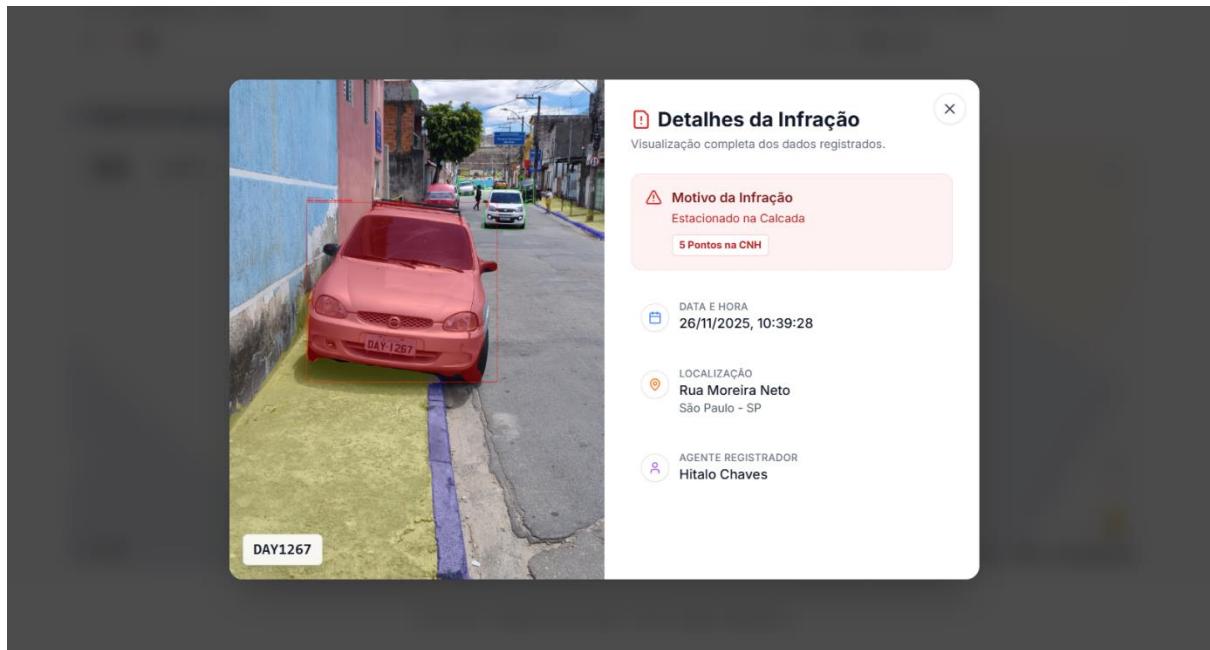
Fonte: Autoria própria, 2025

### 5.5.2 Tela de consultar infrações com ocorrências encontradas

This screenshot shows the same traffic violation search interface but with results. The search input now contains 'DAY1267'. The main title 'Consulta de Infrações' and subtitle 'Pesquise o histórico de infrações pela placa do carro.' are still present. Below the input, a green box displays the license plate 'DAY1267' with the word 'BRASIL' above it. A pink message bar indicates '4 infração(es) encontrada(s)' (4 violations found). The results are displayed in three separate cards, each showing a photo of a red car parked on a sidewalk, with the caption 'Estacionado na Calçada' (Parked on the sidewalk), a 'Grave' (Serious) status, and a '5 pts' (5 points) fine. Each card also includes a timestamp ('26/11/2025, 10:39:28'), location ('Rua Moreira Neto, São Paulo'), and reporter information ('Por: Hitalo Chaves'). The background features a light gray grid pattern.

Fonte: Autoria própria, 2025

## 5.6 Tela com o modal da infração



Fonte: Autoria própria, 2025

## 5.7 Tela de validação de Infrações (vazia)

**Validação de Infrações**

Faça o upload da imagem para análise automática.

**3 Tipos Detectáveis:**

- Carro estacionado sobre a calçada
- Carro estacionado sobre a faixa de pedestre
- Carro estacionado sob a placa de proibido

Selecionar Imagem

Escolher arquivo Nenhum arquivo escolhido

Formatos aceitos: JPG, PNG

Validar Agora

Fonte: Autoria própria, 2025

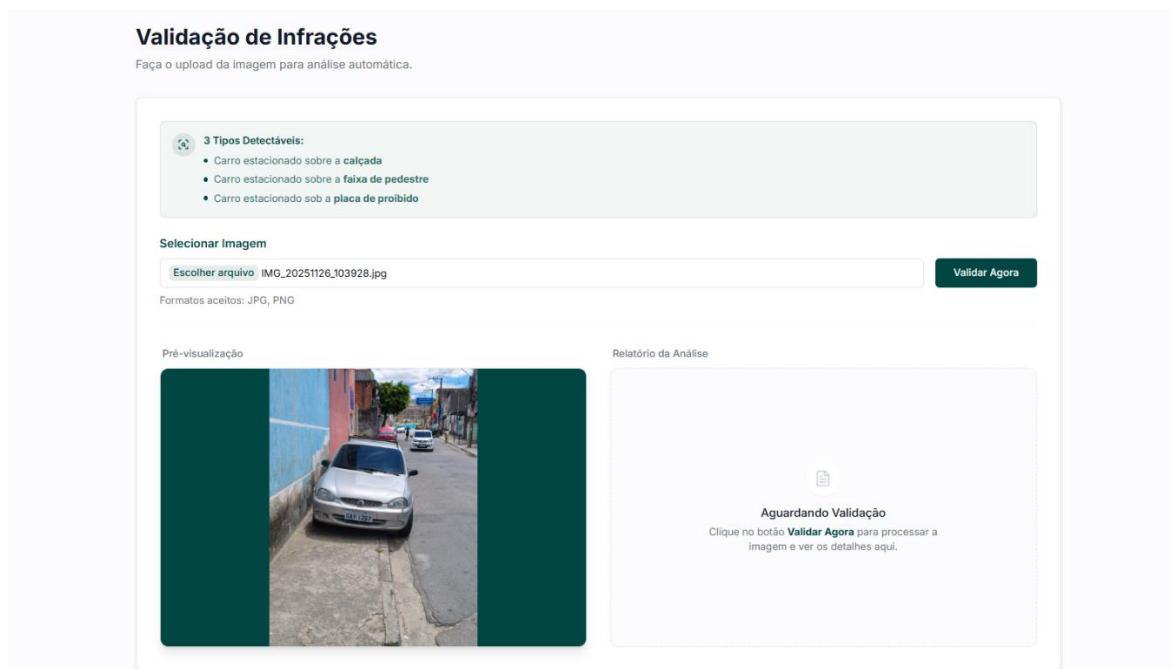
### 5.7.1 Tela de validação de Infrações (tutorial)



© 2025 IPD - Detecção de Infrações. Todos os direitos reservados.

Fonte: Autoria própria, 2025

### 5.7.2 Tela de validação com uma imagem para processar



Fonte: Autoria própria, 2025

### 5.7.3 Tela de validação com uma imagem de infração processada

**Validação de Infrações**

Faça o upload da imagem para análise automática.

**3 Tipos Detectáveis:**

- Carro estacionado sobre a calçada
- Carro estacionado sobre a faixa de pedestre
- Carro estacionado sob a placa de proibido

**Selecionar Imagem**

Escolher arquivo:  **Validar Agora**

Formatos aceitos: JPG, PNG

**Imagen Analisada**

**Relatório da Análise**

**Infração Detectada**  
Atenção requerida para os dados abaixo.

PLACA - COR <b>DAY1267 - Prata</b>	GRAVIDADE <b>Grave</b>
<b>⚠️ Tipo da Infração</b> Estacionado na Calçada	
<b>📍 Localização</b> Rua Moreira Neto, São Paulo - SP	
<b>🕒 Data e Hora</b> 26/11/2025, 10:39:28	

Fonte: Autoria própria, 2025

### 5.7.4 Tela de validação com uma imagem sem infrações detectadas

**Validação de Infrações**

Faça o upload da imagem para análise automática.

**3 Tipos Detectáveis:**

- Carro estacionado sobre a calçada
- Carro estacionado sobre a faixa de pedestre
- Carro estacionado sob a placa de proibido

**Selecionar Imagem**

Escolher arquivo:  **Validar Agora**

Formatos aceitos: JPG, PNG

**Imagen Analisada**

**Relatório da Análise**

**Regular**  
Carro detectado, mas sem infração identificada.

Fonte: Autoria própria, 2025

### 5.7.5 Tela de validação com uma imagem sem veículos detectados

**Validação de Infrações**

Faça o upload da imagem para análise automática.

**3 Tipos Detectáveis:**

- Carro estacionado sobre a calçada
- Carro estacionado sobre a faixa de pedestre
- Carro estacionado sob a placa de proibido

**Selecionar Imagem**

Escolher arquivo:  Validar Agora

Formatos aceitos: JPG, PNG

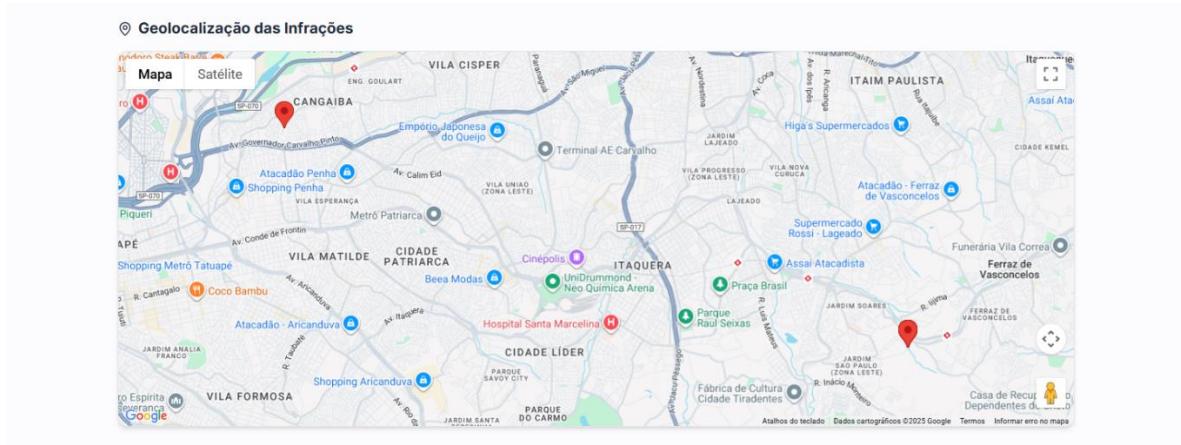
**Imagen Analisada**

**Relatório da Análise**

✓ Regular  
Nenhum carro identificado na imagem.

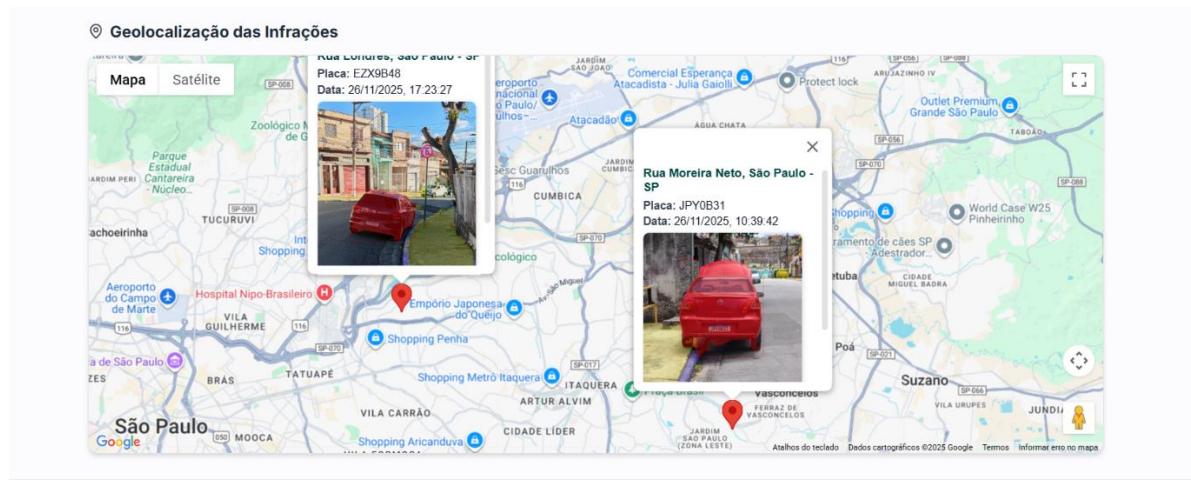
Fonte: Autoria própria, 2025

### 5.8 Geolocalização das Infrações (mapa)



Fonte: Autoria própria, 2025

### 5.8.1 Geolocalização das Infrações (mapa com modais)



Fonte: Autoria própria, 2025

### 5.9 Tela de Perfil

The screenshot shows the "Configurações de Perfil" (Profile Settings) screen:

- User Info:**
  - Profile picture placeholder with letter 'H'.
  - Name: Hitalo Chaves
  - Email: hytalosantos26@gmail.com
  - Text: Recomendado: .jpg ou .png, máx 2MB.
- Informações Pessoais (Personal Information):**
  - Section title: Informações Pessoais
  - Description: Atualize seus dados de identificação.
  - Fields: Nome Completo (Hitalo Chaves), Email (hytalosantos26@gmail.com).
- Segurança (Security):**
  - Section title: Segurança
  - Description: Preencha apenas se desejar alterar sua senha.
  - Fields: Senha Atual (placeholder with masked text), Nova Senha (placeholder with lock icon), Confirmar Senha (placeholder with lock icon).
- Buttons:**
  - Salvar Alterações (Save Changes) button at the bottom right.

Fonte: Autoria própria, 2025

## 5.10 Tela de infrações enviadas

The screenshot shows the 'Monitoramento' (Monitoring) section of the IPD application. At the top, there's a header with the IPD logo and a user profile icon. Below it, a message says 'Gerencie as infrações detectadas e analise o mapa.' (Manage detected violations and analyze the map.) and a button 'Total de registros: 14'. The main area displays six violation reports in a grid:

- DAY1267**: Estacionado na Calçada (Parking on the sidewalk). **Grave**, 5 pts. Date: 26/11/2025, 10:39. Location: Rua Moreira Neto, São Paulo - SP.
- EZX9848**: Estacionado sob Placa Proibida (Parking under a prohibited sign). **Grave**, 5 pts. Date: 26/11/2025, 17:23. Location: Rua Londres, São Paulo - SP.
- DAY1267**: Estacionado na Calçada (Parking on the sidewalk). **Grave**, 5 pts. Date: 26/11/2025, 10:39. Location: Rua Moreira Neto, São Paulo - SP.
- EZX9848**: Estacionado sob Placa Proibida (Parking under a prohibited sign). **Grave**, 5 pts. Date: 26/11/2025, 17:23. Location: Rua Londres, São Paulo - SP.
- DAY1267**: Estacionado na Calçada (Parking on the sidewalk). **Grave**, 5 pts. Date: 26/11/2025, 10:39. Location: Rua Moreira Neto, São Paulo - SP.
- Não Identificada**: Estacionado na Faixa (Parking in the crosswalk). **Grave**, 5 pts. Date: 26/11/2025, 10:39. Location: Endereço não identificado.

Fonte: Autoria própria, 2025

### 5.10.1 Tela de infrações enviadas (nenhuma encontrada)

The screenshot shows the same 'Monitoramento' section of the IPD application. In the center, a large box displays a green exclamation mark icon and the text 'Nenhuma infração enviada' (No violations sent). Below it, a smaller message says 'Nenhuma infração encontrada para este usuário' (No violations found for this user). A green button at the bottom right says 'Tentar novamente' (Try again).

Fonte: Autoria própria, 2025

## 6. CRONOGRAMA PARA TC-II

Nome da Tarefa	07/2025					08/2025					09/2025					10/2025					11/2025					
	01	06	13	20	27	03	10	17	24	31	07	14	21	28	05	12	19	26	02	09	16	23				
Planejar estrutura do código																										
Treinar o modelo de visão computacional																										
Elaboração do banco de dados																										
Interconectar I.A de infrações com modelo de V.C																										
Integração do processamento com a interface																										
Integração da Interface com o banco de dados																										
Testabilidade geral (debug e solucionar falhas)																										
Revisão final do código																										
Revisar documentação final do TCC																										
Preparação para apresentação final do TCC																										

Fonte: Autoria própria, 2025

## 7 TESTES

Este capítulo detalha o processo metodológico adotado para o desenvolvimento e a validação do modelo de detecção de objetos baseado na arquitetura YOLO (You Only Look Once) e do módulo de validação de infrações. Serão descritas as etapas essenciais, que abrangem a aquisição e preparação do dataset, o treinamento, implementação e avaliação do modelo através de suas métricas, além dos testes do módulo de validação de infrações e da interface gráfica do usuário (GUI).

### 7.1 Aquisição e Cura do Dataset

A etapa inicial e fundamental para o sucesso de qualquer modelo de aprendizado supervisionado, como o YOLO, reside na constituição de um dataset robusto e adequadamente anotado. A curadoria do conjunto de dados deve garantir diversidade, representatividade e qualidade das amostras, fatores críticos para a generalização do modelo.

Para o presente trabalho, considerou-se a utilização de duas estratégias principais: a adoção de datasets públicos preexistentes e a criação de um conjunto de dados customizado. A primeira estratégia oferece vantagens em termos de volume e praticidade, enquanto a segunda permite a customização para cenários específicos.

### 7.2 Estratégias de Aquisição de Imagens e Anotação

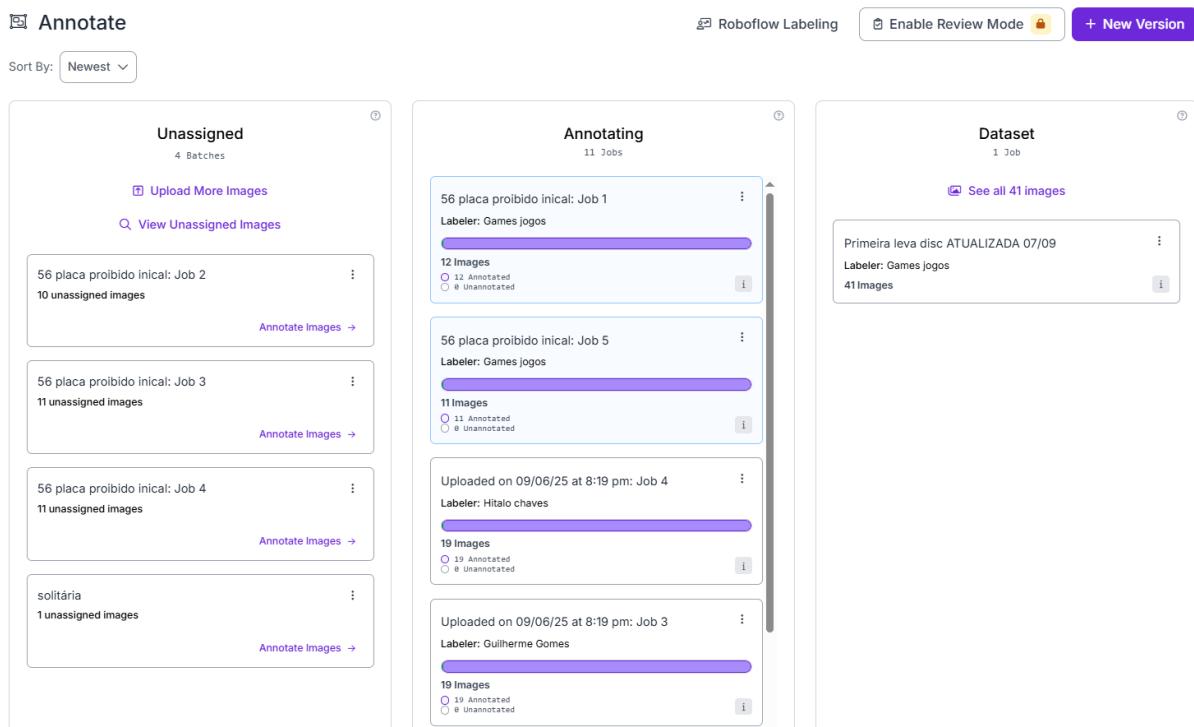
Diante da necessidade de detectar veículos em contextos urbanos específicos, optou-se pela complementação de datasets públicos com imagens customizadas. Este processo de aquisição foi realizado por meio de múltiplas abordagens:

- Coleta de imagens em bancos de dados online e repositórios acadêmicos.
- Captura de *screenshots* em ferramentas de visualização de ambientes urbanos em tempo real ou virtual, como Google Maps Street View e Google Earth, que proporcionam uma ampla variedade de perspectivas, condições de iluminação e climáticas.
- Captura *in situ* utilizando dispositivos móveis e câmeras digitais, garantindo a representatividade do contexto local alvo da aplicação.

Posteriormente à aquisição, as imagens forammeticulosamente anotadas (*labeled*), utilizando ferramentas especializadas como LabelStudio e a própria ferramenta de anotação da plataforma Roboflow. Neste processo, os elementos de interesse (carros; guias normais; guias rebaixadas; placas de sinalização; faixas de pedestre; calçadas e ruas) foram delimitados por segmentação (polígonos) e associados às suas respectivas classes.

Para melhor organização, foram utilizados métodos práticos na plataforma Roboflow, primeiramente organizando o conjunto de imagens em *batches* (fornadas) para serem rotuladas por vez, como também sendo dividido o processo de rotulação em *jobs* (tarefas) para cada integrante do grupo. Assim não sobrecarregando o projeto e facilitando a divisão de conjunto de imagens caso necessite reavaliá-las ou reutilizá-las em outros aspectos do projeto.

Figura 15 – Plataforma de rotulação (*annotate*) do Roboflow



Fonte: Roboflow, 2025

Figura 15 apresenta uma captura de tela (*screenshot*) de uma seção da plataforma Roboflow, na qual é possível observar a divisão de *jobs* e *batches* destinados à rotulação em um projeto. A organização adotada assemelha-se a um quadro Kanban, sendo estruturada em três categorias distintas (representadas como cartões), que correspondem às etapas do processo de anotação de imagens.

No cartão à esquerda, intitulado "*Unassigned*" (Não Atribuídas), encontram-se as tarefas ainda não designadas a nenhum colaborador. Essas tarefas, denominadas *batches*, consistem em conjuntos de imagens que aguardam ser convertidas em *jobs* individuais para, posteriormente, serem rotuladas com as classes de interesse.

No cartão central, denominado "*Annotating*" (Em Anotação), estão listadas as tarefas cujo processo de anotação já foi iniciado. Para cada uma delas, são exibidas informações como o nome do responsável atribuído e o progresso do trabalho, indicado por meio de uma barra colorida que mostra a proporção de imagens já rotuladas em relação ao total.

Por fim, no cartão à direita, na seção "*Dataset*", estão contidas as imagens que passaram por todo o fluxo e encontram-se devidamente anotadas. Esses conjuntos de dados estão prontos para exportação e podem ser utilizados nas etapas subsequentes do projeto, como treinamento e validação do modelo.

Figura 16 – Imagem rotulada com ferramenta do Roboflow



Fonte: Autoria própria, 2025

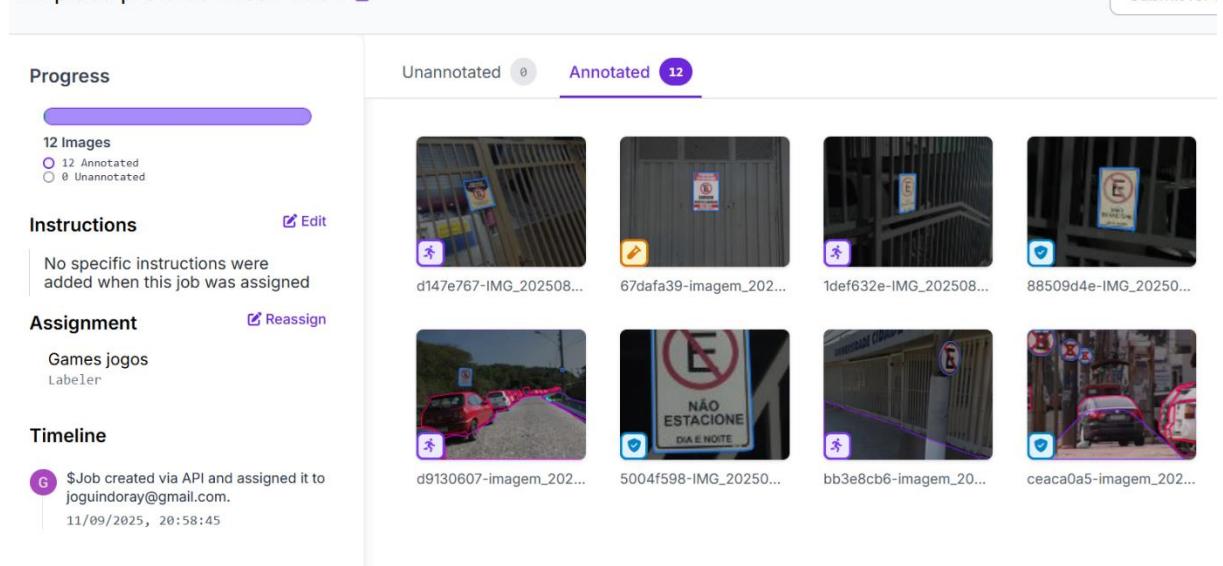
Figura 16 apresenta uma das *screenshots* capturadas através do Google Street View, que foi carregada no projeto para o dataset completo do TCC na plataforma

Roboflow. Após separada corretamente em seu próprio *batch*, e dividida em um *job* para um dos integrantes, foi feita a anotação (rotulação) de cada elemento desejado para contribuir com o objetivo da detecção. Isto é, foram criadas classes para cada elemento e assim atribuídas manualmente através da ferramenta de rotular por segmento de pixels (polígonos).

No momento é necessário rotular a presença de qualquer veículo como classe “carro”, todo asfalto onde os carros estão presentes como “rua”, e toda passagem de pedestres a partir da guia (meio fio) como “calçada”. Também detalhes mais específicos como guias normais (altas), guias rebaixadas e faixas de pedestre foram anotadas para melhor treinamento do modelo.

As imagens podem ser divididas em 3 tipos de funções: Treinamento; Validação e Teste. Treinamento é utilizado como fundamento para o modelo poder ser treinado, com as rotulações devidas. As imagens de validação são utilizadas durante o treinamento após certo período para conferir se o modelo está de fato fazendo a detecção da maneira correta. E as de teste são utilizadas pelo usuário após o treinamento completo do modelo na validação do mesmo.

Figura 17 – Opções para tarefa selecionada do projeto no Roboflow  
56 placa proibido inicial: Job 1



Fonte: Roboflow, 2025

Figura 17 apresenta uma seção de um *job* dos *batches* feitos que foi atribuído a um dos integrantes. A tarefa foi nomeada como “56 placas proibido inicial: Job 1” pois o batch em que esse conjunto de imagens se encontrava, contia 56 imagens ao

total, e em que a maioria eram fotos de placas com a sinalização de “proibido estacionar” variadas.

Foram atribuídas 12 dessas 56 imagens para este integrante e todas necessitam ser rotuladas para contribuir com o dataset final. Pode se notar a presença de símbolos no canto inferior direito de cada miniatura da imagem, pois esses símbolos representam a divisão feita automaticamente pela plataforma para qual função a imagem terá no dataset final. O símbolo roxo representa as imagens que serão para treinamento, o símbolo azul são as imagens para a função de validação durante o treinamento e o símbolo amarelo representa as de teste, que são utilizadas após o treinamento para testar o modelo pronto e obter uma taxa de sucesso.

### 7.2.1 Detalhamento Operacional da Criação e Segmentação do Dataset

A construção do dataset customizado foi centralizada integralmente na plataforma Roboflow, escolhida por oferecer um ambiente colaborativo em nuvem que permite o gerenciamento de versões e a distribuição de tarefas entre os membros do grupo. O processo de criação seguiu um fluxo de trabalho (*pipeline*) estruturado para garantir a consistência dos dados, detalhado nas etapas a seguir:

**Configuração do Projeto e *Workspace*:** Inicialmente, foi criado um *Workspace* privado no Roboflow, onde todos os integrantes foram adicionados como colaboradores. O projeto foi configurado para a tarefa de "*Instance Segmentation*" (Segmentação de Instância), uma vez que a detecção de infrações exige uma compreensão precisa da geometria da cena, não apenas a localização aproximada dos objetos.

**Upload e Pré-processamento:** As imagens capturadas (via Google Street View e *in situ*) foram enviadas para a plataforma ("Upload"). Antes da etapa de anotação, não foram aplicados filtros de pré-processamento destrutivos, mantendo a resolução original das capturas para preservar detalhes vitais, como a legibilidade de placas e texturas de calçadas.

**Metodologia de Anotação (Polígonos):** A etapa de rotulação foi a mais crítica e trabalhosa do processo. Diferentemente do método tradicional de *Bounding Boxes* (Caixas Delimitadoras), que apenas desenha retângulos ao redor dos objetos, optou-se exclusivamente pela Anotação por Polígonos (*Polygon Annotation*).

- Justificativa Técnica: A escolha por polígonos foi imperativa devido à natureza irregular dos elementos urbanos. Uma "calçada" ou uma "faixa de pedestre", por exemplo, raramente se apresenta como um retângulo perfeito na perspectiva da câmera. O uso de polígonos permitiu aos anotadores clicar ponto a ponto ao redor do contorno exato do objeto, excluindo pixels de fundo (como a rua ou grama) que seriam indevidamente incluídos em uma caixa retangular. Isso aumenta drasticamente a precisão do cálculo de interseção (IoU) durante a validação da infração.
- Execução: Para cada imagem, os anotadores contornaram manualmente os veículos (classe carro), as zonas de circulação (rua, calçada) e as sinalizações (faixa\_pedestre, placa\_proibido, guia\_rebaixada), criando máscaras de segmentação precisas.

**Gestão de Lotes e Revisão:** Para organizar o trabalho em equipe, utilizou-se o sistema de *Batches* do Roboflow. Grandes volumes de imagens cruas eram segregados em lotes menores e atribuídos a membros específicos através de *Jobs*. Após a conclusão de um *Job*, implementou-se uma etapa de "Revisão", onde um membro diferente do anotador original verificava a qualidade dos polígonos, corrigindo contornos mal ajustados ou classes atribuídas erroneamente antes que as imagens fossem integradas ao dataset mestre.

**Geração de Versões:** Após a aprovação das anotações, o dataset foi gerado dentro da plataforma. Nesta etapa, o Roboflow processou as anotações poligonais e converteu os dados para o formato compatível com o YOLOv11 (formato TXT contendo as coordenadas normalizadas dos vértices dos polígonos). O conjunto foi então exportado com a divisão automática de arquivos para Treino, Validação e Teste, conforme mencionado anteriormente, consolidando a base de dados para o treinamento da rede neural.

**Pré-processamento e Estratégias de Aumento de Dados (*Data Augmentation*):** Para assegurar a qualidade da entrada de dados na rede neural e mitigar problemas de overfitting (quando o modelo "decora" as imagens de treino, mas não sabe generalizar), foram aplicadas técnicas rigorosas de pré-processamento e enriquecimento de dados diretamente na plataforma Roboflow:

- Pré-processamento (Padronização): Todas as imagens foram submetidas a uma etapa de orientação automática (para corrigir a rotação baseada nos metadados EXIF) e, crucialmente, a um redimensionamento para 1800x1800 pixels. Optou-se pela técnica de "*Fit within*" (Ajustar dentro) com preenchimento de bordas pretas (*black edges/padding*). Essa escolha técnica foi fundamental para manter a proporção de aspecto (*aspect ratio*) original das imagens. Ao contrário do redimensionamento que estica a imagem (*stretch*), o uso de *padding* preserva a geometria real dos veículos e das placas, impedindo distorções que prejudicam a precisão das máscaras poligonais.
- *Data Augmentation* (Aumento de Dados): Para simular a variabilidade encontrada em cenários urbanos reais (como diferentes horários do dia, condições climáticas e imperfeições de câmera), foi aplicado um pipeline de aumento de dados que gerou 3 versões aumentadas para cada imagem original. As transformações incluíram:
  - Variações Geométricas: Aplicação de Flip Horizontal (espelhamento) e Rotação entre -13° e +13°, simulando diferentes direções de tráfego e leves inclinações da câmera ou do terreno.
  - Variações Fotométricas: Ajustes de Brilho (entre -25% e +25%) e Exposição (entre -10% e +10%) para garantir que o modelo reconheça infrações tanto em dias ensolarados quanto em áreas de sombra ou nubladas.
  - Simulação de Ruído: Introdução de Desfoque (*Blur*) de até 0.7px e Ruído (*Noise*) de até 0.73% dos pixels, preparando o modelo para lidar com imagens de câmeras de vigilância de baixa qualidade ou capturas em movimento.

### 7.3 Processo de Treinamento do Modelo

O treinamento de uma rede neural convolucional profunda como a YOLO é computacionalmente intensivo e requer a configuração adequada de uma série de hiperparâmetros. Os principais fatores considerados foram:

- A definição da arquitetura e da profundidade da rede (e.g., YOLOv5s, YOLOv8m).

- A divisão do dataset em subconjuntos de treino, validação e teste (usualmente na proporção 70/20/10).
- O ajuste dos hiperparâmetros de otimização, como taxa de aprendizado (*learning rate*), *batch size* e *momentum*.
- O número de épocas (*epochs*) para evitar *overfitting* ou *underfitting*.

A execução do treinamento foi conduzida em um ambiente virtual que provê aceleração por GPU. Utilizou-se a plataforma Google Colab, que oferece acesso a unidades de processamento gráfico (*Graphics Processing Unit - GPU*), drasticamente reduzindo o tempo necessário para o processo iterativo de ajuste dos pesos da rede neural em comparação com o uso de CPU.

#### **7.4 Implementação e Validação do Modelo**

Ao final do treinamento, o modelo resultante, um arquivo de pesos com extensão .pt (no caso das implementações Ultralytics YOLO), é implementado para inferência. A fase de validação é crítica para conferir o desempenho e a generalização do modelo.

Os testes foram realizados em um conjunto de dados de teste isolado, composto por imagens anotadas que não foram utilizadas em qualquer fase do treinamento. Isso permite uma avaliação imparcial da performance. A métrica principal de avaliação adotada é a precisão média (*mean Average Precision - mAP*), que combina as métricas de precisão (*precision*) e revocação (*recall*) para diferentes valores de *Intersection over Union* (IoU).

Além disso, testes qualitativos com imagens inéditas, não presentes no dataset original, são realizados para verificar o comportamento do modelo em cenários do mundo real, identificando possíveis pontos de falha e oportunidades para refinamento futuro.

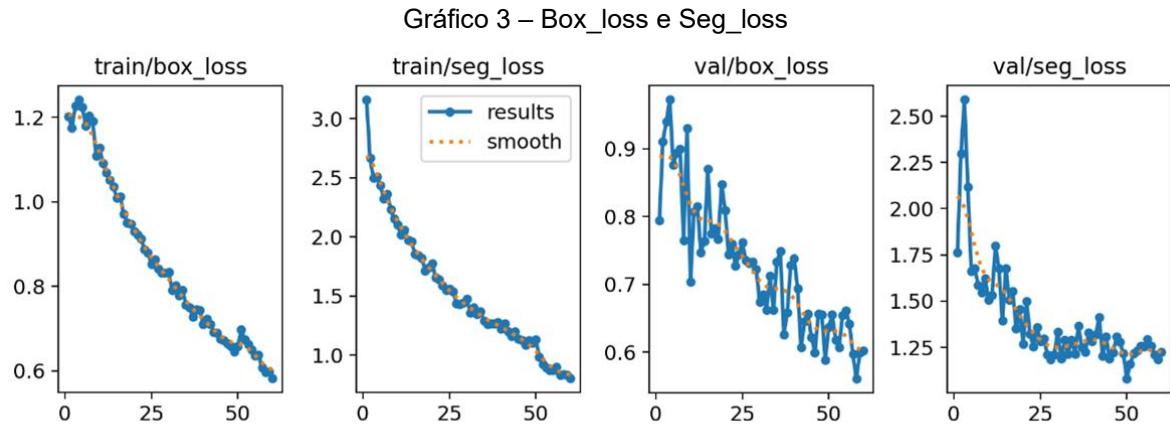
#### **7.5 Gráficos de métricas do treinamento do modelo**

A análise de desempenho será exposta nas seções subsequentes, concentrando-se na avaliação quantitativa do modelo, que será fundamentada em métricas padrões da literatura, como Precisão (*Precision*), Revocação (*Recall*), e a precisão média (*mean Average Precision - mAP*) nos limiares de IoU 50 e 50-95, para

ambas as tarefas de detecção (*Boxes*) e segmentação (*Masks*). Adicionalmente, serão apresentados gráficos de evolução das métricas durante o treinamento, a matriz de confusão e uma análise do desempenho em tempo de inferência.

Este capítulo apresenta e analisa os resultados obtidos no terceiro ciclo de treinamento do modelo, realizado em 08 de outubro. O último treinamento realizado, identificado como "v1+v2+notcars", teve como objetivo principal avaliar o desempenho de um modelo de segmentação de instâncias na tarefa de detecção e segmentação de veículos em ambiente urbano. Para isso, foi empregada a arquitetura YOLOv11s-seg, uma variante compacta e eficiente da família YOLO, caracterizada pela rotulação por segmentos (polígonos). As variações da versão 11 do YOLO foram cogitadas neste teste por ser uma das últimas versões lançadas pela empresa Ultralytics no ano de 2025, ótima para testes de validação de conceito, balanceando velocidade e precisão.

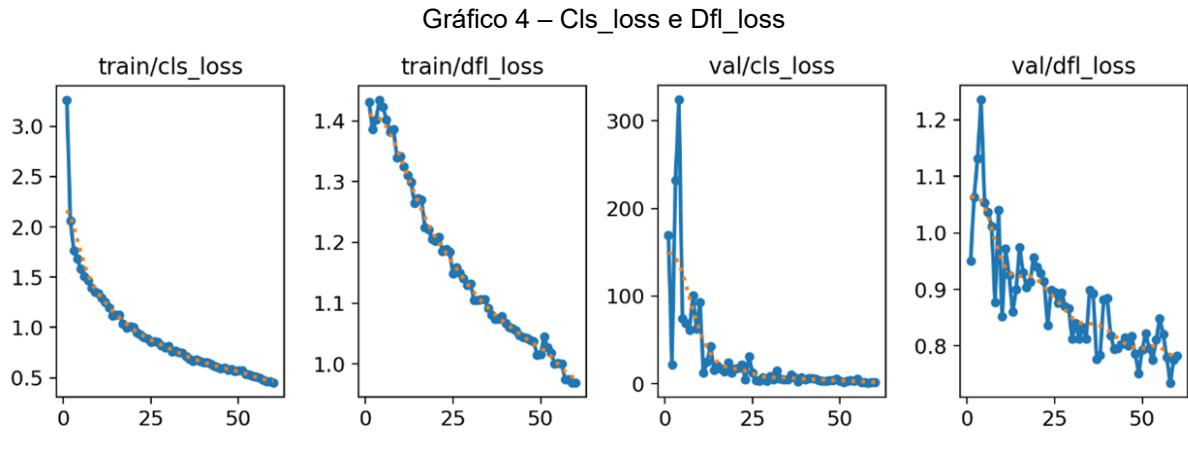
O modelo foi treinado por 60 épocas em um conjunto de dados contendo 873 imagens, totalizando 539 instâncias anotadas. O dataset foi dividido estrategicamente, alinhando-se de maneira aproximada às convenções da literatura que sugerem uma proporção majoritária para o aprendizado (como a regra empírica 70/20/10). Desta forma, os dados foram segmentados em subconjuntos de treino (672 imagens, correspondendo a aproximadamente 77% do total), validação (136 imagens, aproximadamente 16% do total) e teste (65 imagens, aproximadamente 7% do total). Essa distribuição garante uma avaliação robusta e imparcial da capacidade de generalização do modelo, assegurando volume suficiente para a extração de características durante o treino, ao mesmo tempo que preserva amostras inéditas para as etapas de verificação. O treinamento, conduzido em uma GPU Tesla T4 via Google Colab, teve duração total de aproximadamente 52 minutos (3.140 segundos), demonstrando a eficiência computacional da configuração adotada.



Fonte: Autoria própria, 2025

Os gráficos apresentados (Gráfico 3) ilustram o comportamento das perdas (losses) durante o treinamento e validação do modelo, com foco nas métricas box\_loss e seg\_loss. A box\_loss, associada à detecção e ajuste das bounding boxes, apresentou uma redução progressiva e estável ao longo das épocas, atingindo aproximadamente 0,58 no final do treinamento. Essa tendência demonstra que o modelo conseguiu aprender de maneira consistente a delimitar os objetos de interesse, com diminuição gradual dos erros de predição espacial. Já a seg\_loss, relacionada à segmentação de máscaras, iniciou com valores mais elevados, mas apresentou convergência expressiva, alcançando cerca de 0,80. Essa queda reflete um aprendizado eficiente na identificação e separação das regiões correspondentes aos objetos dentro das imagens.

A redução simultânea das perdas de treinamento e validação evidencia que o modelo apresenta boa capacidade de generalização, sem sinais de *overfitting* significativos. A proximidade entre as curvas de treino e validação, aliada à suavidade observada na tendência das métricas, indica que o processo de ajuste dos parâmetros foi bem equilibrado, resultando em um modelo capaz de aprender representações robustas tanto para detecção quanto para segmentação. Esses resultados confirmam a eficiência do processo de treinamento e sugerem que o modelo alcançou um ponto de convergência adequado para prosseguir com análises quantitativas mais detalhadas.

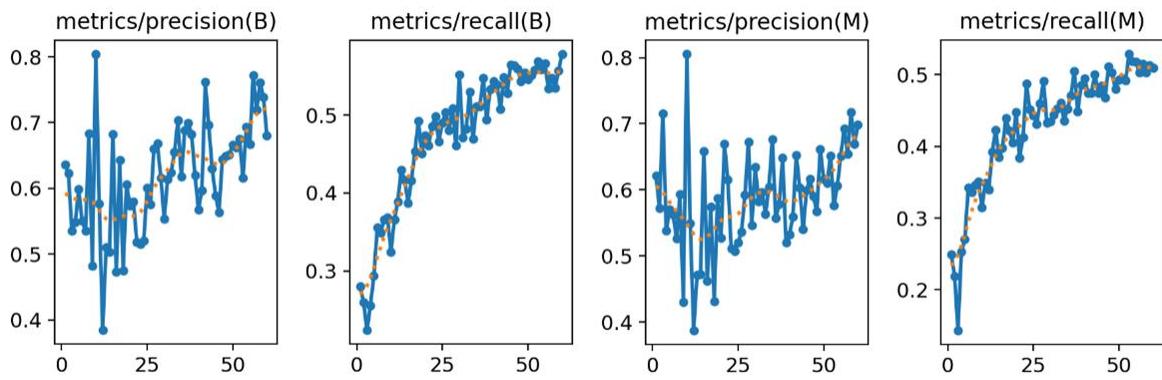


Fonte: Autoria própria, 2025

Os gráficos apresentados (Gráfico 4) representam o comportamento das métricas de perda de classificação (*cls\_loss*) e de regressão de caixas delimitadoras via distribuição focal (*dfl\_loss*) ao longo do treinamento e validação do modelo. A *cls\_loss*, responsável por medir os erros na categorização dos objetos detectados, apresentou uma redução significativa e contínua, atingindo aproximadamente 0,44 na fase final do treinamento. Esse valor indica que o modelo adquiriu uma boa capacidade de distinguir corretamente as classes de interesse. Já a *dfl\_loss*, associada à precisão na regressão das caixas delimitadoras, apresentou uma tendência semelhante, diminuindo gradualmente até cerca de 0,96, demonstrando estabilidade e eficiência no ajuste das coordenadas preditas em relação às reais.

No processo de validação, observa-se que ambas as métricas mantêm uma trajetória de queda coerente com o treinamento, sugerindo consistência entre aprendizado e generalização. O comportamento da *cls\_loss* apresenta uma variação inicial mais acentuada, o que é comum em estágios iniciais de treinamento, mas converge de forma satisfatória após algumas iterações. Já a *dfl\_loss* evidencia uma convergência mais estável, o que reforça a robustez do modelo em tarefas de regressão espacial. Em conjunto, esses resultados apontam para um treinamento equilibrado, com boas perspectivas de desempenho em cenários reais de detecção e segmentação.

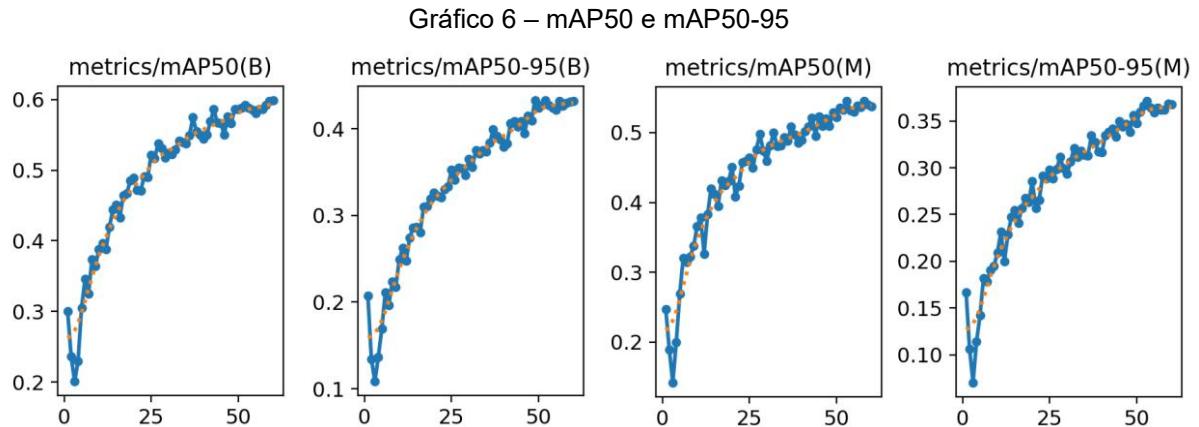
Gráfico 5 – Precision e Recall



Fonte: Autoria própria, 2025

A partir da análise das métricas (Gráfico 5) de precisão (Precision) e revocação (Recall) para as tarefas de detecção e segmentação, observa-se um comportamento consistente de aprendizado ao longo das iterações. A métrica de precisão apresenta oscilações iniciais, estabilizando-se gradualmente em torno de 0,72, o que indica que o modelo foi capaz de reduzir a ocorrência de falsos positivos conforme o treinamento avançou. Já o recall demonstra uma curva ascendente mais contínua, atingindo valores próximos de 0,56, evidenciando o aumento da taxa de acertos na detecção de objetos relevantes. O comportamento conjunto dessas métricas sugere que o modelo alcançou um equilíbrio satisfatório entre precisão e cobertura, aspecto fundamental para sistemas que precisam identificar infrações visuais sem comprometer a confiabilidade das previsões.

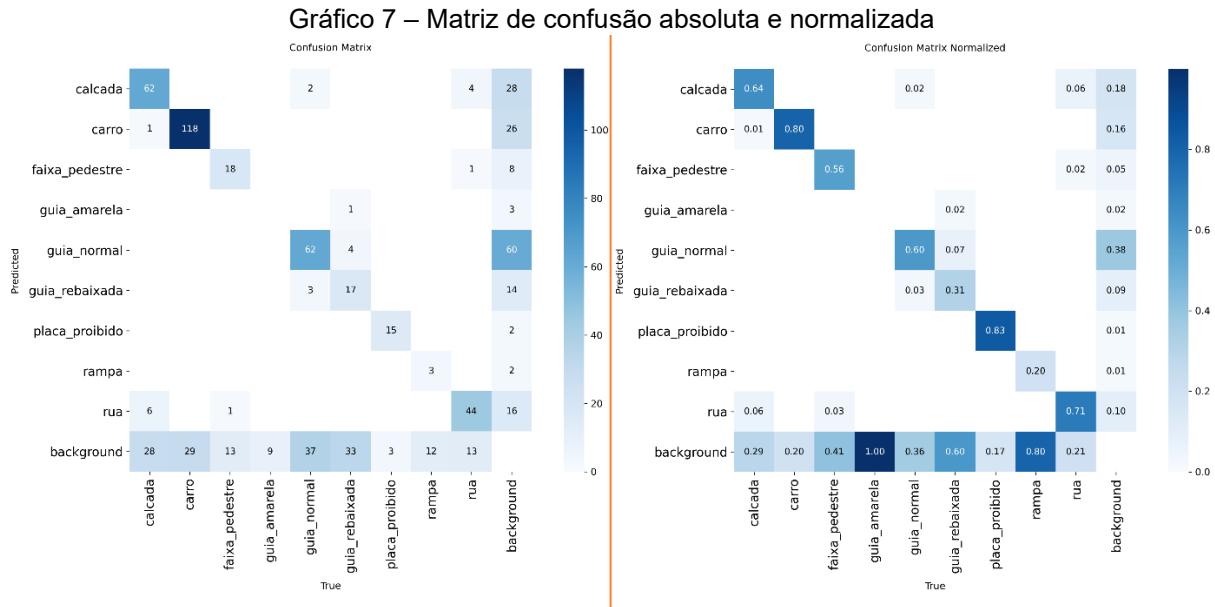
Os resultados indicam que a precisão ( $\approx 0,72$ ) e o recall ( $\approx 0,56$ ) alcançados ao término do treinamento são compatíveis com o equilíbrio desejado entre minimizar falsos positivos e maximizar a detecção de instâncias relevantes. A precisão relativamente alta sugere que o modelo comete poucos erros ao classificar objetos inexistentes, enquanto o recall mais moderado revela que ainda há margem de aprimoramento na recuperação de casos positivos espalhados. Essas métricas refletem diretamente a qualidade do aprendizado: quanto mais elas convergem em patamares elevados e estáveis, mais confiável tende a ser o modelo para a tarefa de identificação visual em condições variadas.



Fonte: Autoria própria, 2025

As métricas de mAP (mean Average Precision), apresentadas (Gráfico 6) para os limiares de IoU 50 e 50–95, refletem a capacidade geral do modelo em detectar e segmentar corretamente as instâncias de interesse. O gráfico de mAP@50 evidencia um crescimento contínuo e estável ao longo das iterações, atingindo aproximadamente 0,60, o que indica um bom desempenho na detecção de objetos com sobreposição significativa entre as predições e as anotações reais. Já o mAP@50–95, com valor médio em torno de 0,43, representa uma avaliação mais rigorosa, considerando múltiplos níveis de IoU. Essa diferença entre as duas métricas é esperada, uma vez que limiares mais elevados penalizam levemente desvios na localização das detecções.

Esses resultados sugerem que o modelo alcançou boa capacidade de aprendizado espacial, sendo capaz de distinguir corretamente os limites das instâncias, ainda que existam variações na iluminação e no enquadramento das imagens. O crescimento progressivo de ambas as métricas demonstram a eficiência do processo de otimização e reforça a consistência do modelo em cenários de detecção e segmentação. Assim, o comportamento dos gráficos indica que o treinamento foi capaz de convergir de maneira adequada, resultando em uma rede com respostas estáveis e coerentes aos dados apresentados.



Fonte: Autoria própria, 2025

A matriz de confusão apresentada (Gráfico 7) está presente em duas versões: absoluta (contagem de instâncias) e normalizada (proporcional), isso oferece uma visão rica sobre o comportamento da rede em relação a falsos positivos e falsos negativos para cada classe. Ao comparar ambas, é possível identificar quais classes têm desempenho robusto e onde há confusões mais recorrentes.

Um dos pontos mais positivos é a classe “carro”, que apresenta uma forte diagonal com alta contagem correta e proporção elevada ( $\approx 0,80$  na versão normalizada). Isso indica que a rede reconhece bem veículos em geral, com baixa taxa de erro nessa classe. Outra classe que se destaca é “guia\_normal”, que também mostra um desempenho razoável, com certas confusões para “guia\_rebaixada” ou “background”, mas ainda mantendo predominância na classificação correta. Na versão normalizada, percebe-se que “placa\_proibido” também atinge uma boa precisão relativa, sugerindo que a rede trata bem situações específicas de infração de placa proibida.

Contudo, a matriz evidencia áreas de fragilidade. Por exemplo, classes como “calçada” e “faixa\_pedestre” apresentam misturas visíveis: “calçada” tem algumas previsões como “background” ou “rua”, e “faixa\_pedestre” mostra confusões com “background” e “calçada”. Isso sugere que, visualmente, essas classes podem ter padrões visuais menos distintos no dataset, causando confusão. Também se observa que “guia\_rebaixada” é confundida parcialmente com “guia\_normal” em algumas

instâncias, o que não é inesperado dado que a segmentação entre rebaixadas e normais pode depender de diferenças sutis.

Importante destacar: as classes “rampa” e “guia\_amarela” devem ser desconsideradas na análise interpretativa, pois tiveram pouquíssimas instâncias no conjunto de teste. Com tão poucos exemplos, seus resultados não são estatisticamente confiáveis nem representam bem o comportamento do modelo nessas categorias.

Em síntese, a matriz de confusão mostra que a rede lida bem com classes mais frequentes e visualmente distintas (como carro e placa proibido), mas esbarra em classes com padrões mais sutis ou menos representadas. Essa análise permite identificar onde concentrar esforços: aumentar a amostragem para classes com poucas instâncias, melhorar anotação/qualidade de imagem para classes difíceis e talvez aplicar técnicas de balanceamento ou aumento nos dados para reduzir a confusão entre categorias próximas.

## 7.6 Testes e Implementação da validação de Infrações

A etapa de implementação e inferência do modelo treinado, baseada na arquitetura YOLO para segmentação de instâncias, culminou no desenvolvimento de um código de processamento que traduz as detecções visuais em um veredito de estacionamento regular ou irregular. Este módulo algorítmico foi especificamente desenhado para analisar a posição e o relacionamento entre a classe de interesse primária, o veículo (carro), e as demais classes que representam zonas de estacionamento proibido no contexto urbano. As classes proibidas foram separadas em dois grupos para a aplicação de lógicas de infração distintas: Infrações por Interseção (calcada, faixa\_pedestre) e Infrações Relacionais (placa\_proibido).

A detecção de infrações por interseção é realizada através de uma análise híbrida de sobreposição de pixels. Para classes de zona proibida que exigem precisão no contato com o solo (como a faixa de pedestre), a checagem é feita utilizando a Máscara de Contato do Veículo (delimitada pelo *CONTACT\_ZONE\_HEIGHT\_RATIO* em 60% da altura inferior do *bounding box*), prevenindo falsos positivos causados por erros de perspectiva. Para a outra classe de infração por intersecção, “calçada”, a máscara de segmentação completa do veículo é utilizada para garantir a detecção de obstrução da área proibida. Uma infração é confirmada somente se o número de pixels de sobreposição, na máscara relevante, exceder um Limiar de Interseção (*INTERSECTION\_THRESHOLDS*) previamente definido, permitindo que o sistema ignore pequenos erros de segmentação. Esses limiares foram ajustados de forma customizada para cada classe de zona proibida, utilizando-se, por exemplo, um threshold menor para a faixa\_pedestre para refletir sua criticidade e maior sensibilidade. O algoritmo prioriza a verificação de infrações em faixa\_pedestre antes de outras interseções, reforçando a importância dessa área crítica para o trânsito.

Por outro lado, a detecção de infração sob placas de proibição utiliza uma lógica relacional entre a bounding box do veículo e a bounding box da placa (placa\_proibido). Esta lógica se baseia em dois critérios fundamentais. O primeiro é a Distância Dinâmica, onde o limite máximo de proximidade para configurar a infração é estabelecido pela largura da bounding box do próprio carro, multiplicada por um Fator de Distância Dinâmica (*DYNAMIC\_DISTANCE\_FACTOR* = 2.0). O segundo critério é a Relação de Área (*Ratio*), um intervalo ideal de proporção

(*MIN\_AREA\_RATIO\_CAR\_TO\_PLACA* e *MAX\_AREA\_RATIO\_CAR\_TO\_PLACA*) entre a área do veículo e a área da placa. Este *ratio* garante que a infração relacional só seja considerada quando a placa e o veículo estão em uma escala visualmente compatível, prevenindo detecções incorretas em cenas complexas.

Por fim, o código inclui uma etapa de priorização para o feedback visual e de dados. Quando múltiplas infrações são detectadas na imagem, o sistema identifica o infrator primário como o veículo que possui a maior área dentre todos os carros que cometeram alguma infração. Essa heurística é utilizada para destacar visualmente o veículo de maior relevância ou que está em maior evidência na cena, facilitando a interpretação do caso de infração pelo agente fiscalizador. O infrator primário é destacado em vermelho, infratores secundários em amarelo e veículos em conformidade em verde.

## **7.7 Testes de Interface Gráfica do Usuário (GUI) e Usabilidade**

Além dos rigorosos testes de criação e validação do modelo de visão computacional, uma avaliação aprofundada da usabilidade e da consistência da Interface Gráfica do Usuário (GUI) foi conduzida. Esta avaliação baseou-se nos protótipos detalhados apresentados no Capítulo 5 e seguiumeticulosamente os requisitos não funcionais estabelecidos para Usabilidade e Confiabilidade. O foco principal desses testes foi garantir uma experiência do usuário otimizada, uma navegação altamente intuitiva e a clareza inquestionável das informações apresentadas em todas as telas da aplicação.

Os casos de teste da interface foram desenvolvidos para cobrir os aspectos mais críticos da interação do usuário com o sistema:

**Fluxo de Login e Cadastro Otimizado:** Este teste avaliou a facilidade e a intuitividade do processo de criação de conta e acesso à plataforma. Foi verificado se as interfaces de login e cadastro (conforme ilustrado nas Figuras 5.2 e 5.3 dos protótipos) guiavam o usuário de forma clara e eficiente, alinhando-se perfeitamente com os passos descritos nos casos de uso UC01 (Cadastro de Usuário) e UC02 (Login de Usuário). O objetivo era garantir uma barreira de entrada mínima para novos usuários.

Navegação Eficiente na Tela Principal: Após o login bem-sucedido, este caso de teste focou na clareza e na organização das opções disponíveis no painel principal. Foram verificadas funcionalidades essenciais como "Consultar Infrações" e "Validar Infração" (representadas na Figura 5.4), assegurando que os botões e links correspondentes direcionavam os usuários para as telas corretas e esperadas (Figuras 5.5 e 5.7), proporcionando uma experiência de navegação fluida e sem ambiguidades.

Visualização Legível de Dados: Este teste foi crucial para validar a eficácia das telas de exibição de dados. A tela de "Consultar Infrações" (Figura 5.5) foi testada para garantir a funcionalidade e precisão de seu campo de busca, permitindo aos usuários encontrar informações específicas rapidamente. Adicionalmente, a tela de "Infrações Enviadas" (Figura 5.10) foi examinada para assegurar que as informações fossem apresentadas de forma organizada, completa e facilmente legível. Isso incluía detalhes essenciais como local, data, hora, motivo e tipo da infração, além da visualização de uma imagem da ocorrência para contextualização.

Perfil do Usuário Funcional: A interface de perfil do usuário (Figura 5.9) foi testada para confirmar que as informações pessoais pudessem ser atualizadas e armazenadas corretamente, mantendo a consistência dos dados do usuário.

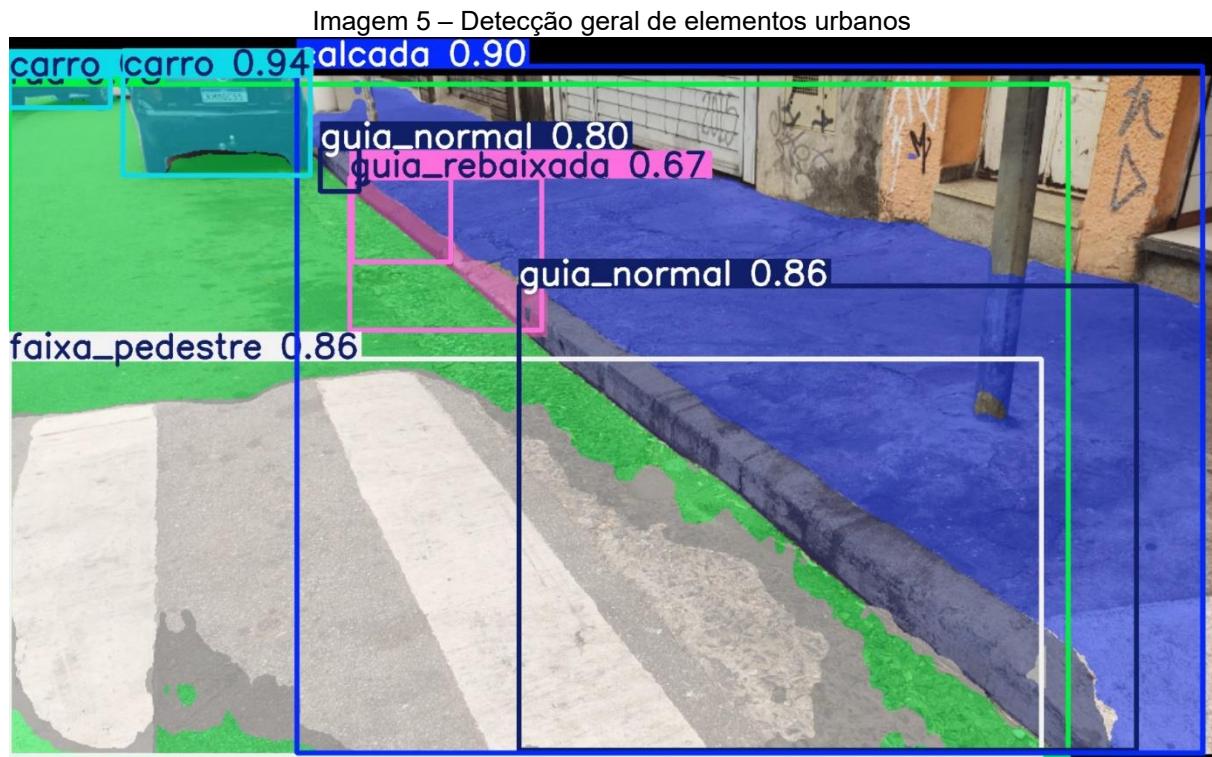
Os resultados desses testes foram extremamente positivos, demonstrando que o design da interface, apresentada nos protótipos, é perfeitamente fluido e direto. Este alinhamento com o objetivo de minimizar cliques e reduzir a curva de aprendizado foi comprovado, indicando que os usuários podem interagir com a aplicação de forma intuitiva e eficiente desde o primeiro contato.

## 8 RESULTADOS

Este capítulo concentra-se na análise qualitativa e funcional do sistema em cenários de uso prático. A Seção 8.1 apresenta a capacidade de generalização do modelo de detecção, exibindo as previsões das classes desejadas (bounding boxes e máscaras) em imagens de testes com variações urbanas e de perspectiva. Em seguida, a Seção 8.2 demonstra a eficácia do Módulo de Validação de Infrações, mostrando como o sistema integra as detecções do modelo com as regras algorítmicas de lógica espacial para classificar, com precisão, a ocorrência de estacionamentos irregulares em zonas proibidas.

### 8.1 Detecção em imagens de casos reais

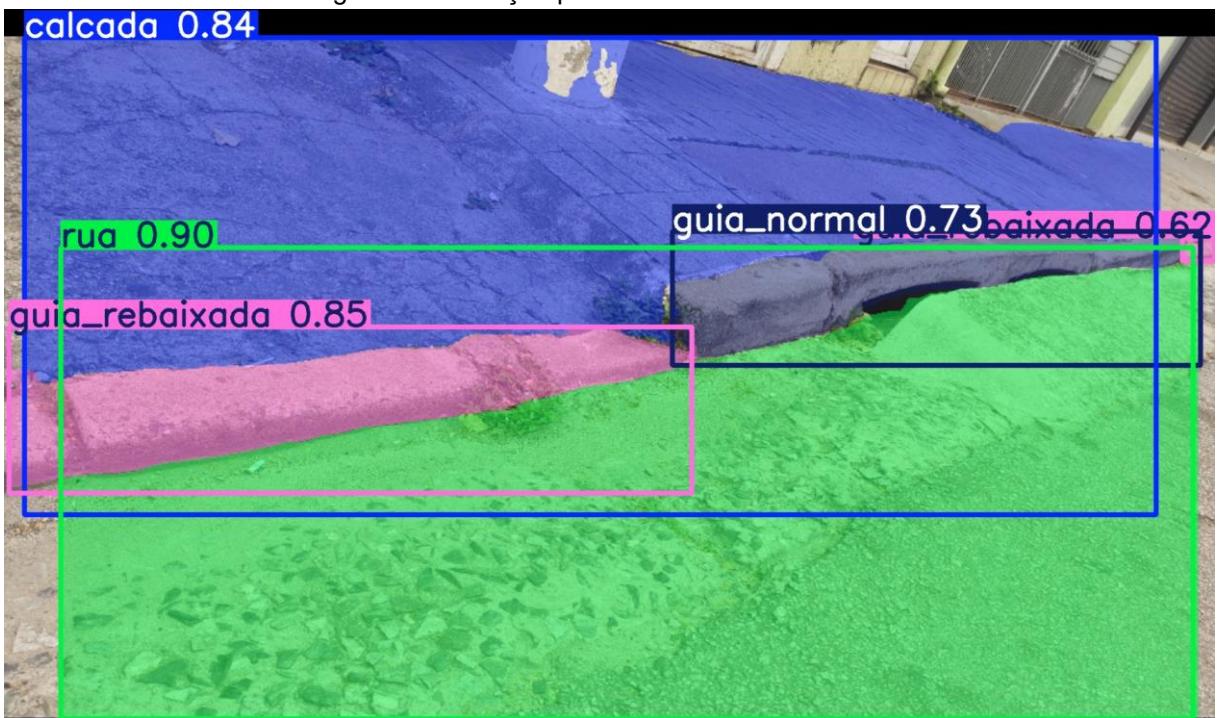
Nesta seção são apresentadas as detecções realizadas pelo modelo em imagens reais pertencentes ao conjunto de teste, utilizadas para avaliar o desempenho qualitativo do sistema em cenários próximos ao uso prático. As figuras exibem as previsões geradas pela rede, incluindo as bounding boxes e máscaras de segmentação associadas às classes identificadas, acompanhadas de seus respectivos valores de confiança. Cada imagem contém elementos distintos do ambiente urbano, como calçadas, veículos, guias, faixas de pedestres e sinalizações, permitindo observar a robustez do modelo diante de variações de perspectiva, iluminação e textura. Essa análise visa complementar as métricas quantitativas apresentadas anteriormente, demonstrando a capacidade do modelo de generalizar padrões visuais e reconhecer objetos de interesse em contextos reais.



Fonte: Autoria própria, 2025

Nesta amostra (Imagen 5), o modelo demonstrou desempenho satisfatório na detecção simultânea de múltiplas classes em um mesmo quadro, com confianças variando entre 0,67 e 0,94. Observa-se especial precisão na identificação de objetos de maior contraste, como veículos e calçadas, enquanto categorias mais sutis, como guia rebaixada, exibem confiança inferior, comportamento esperado em regiões com texturas e tons semelhantes. Esses resultados reforçam a influência da geometria da cena e das condições de iluminação sobre o desempenho do detector, fatores críticos para aplicações em ambientes reais.

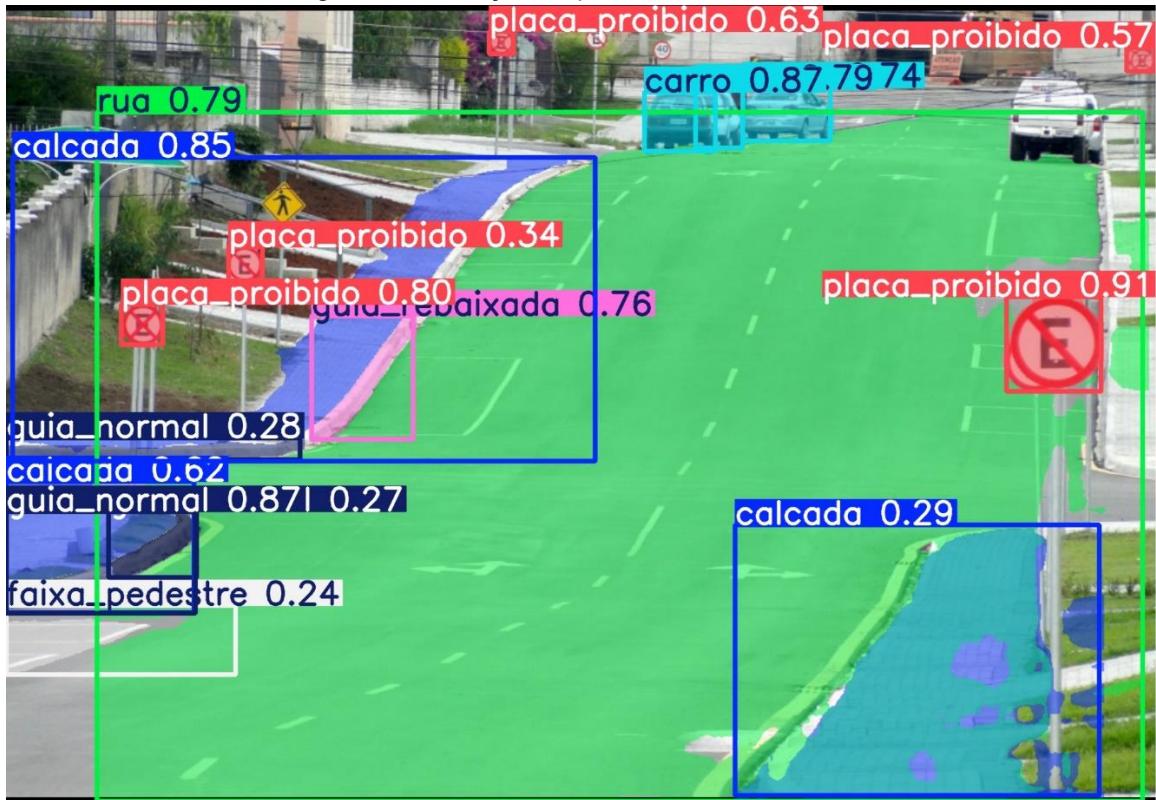
Imagen 6 – Detecção próxima de elementos urbanos



Fonte: Autoria própria, 2025

Nesta amostra (Imagen 6), o modelo apresentou desempenho consistente na segmentação de elementos urbanos estruturais, distinguindo adequadamente as classes “calçada”, “rua”, “guia normal” e “guia rebaixada”, com níveis de confiança entre 0,73 e 0,90. Nota-se uma clara separação entre as regiões da via e da calçada, evidenciando a capacidade do sistema em compreender limites físicos e desníveis do terreno. A detecção da guia rebaixada, com confiança de 0,85, indica que o modelo conseguiu identificar variações sutis na geometria do meio-fio, ainda que essas apresentem tonalidades semelhantes ao pavimento. Pequenas sobreposições entre as classes “guia normal” e “guia rebaixada” sugerem desafios residuais em áreas de transição, onde o relevo e a textura se confundem. De modo geral, o resultado reforça a eficiência do modelo em interpretar corretamente a estrutura viária, fator essencial para a análise automatizada de infrações relacionadas ao estacionamento.

Imagen 7 – Detecção ampla de elementos urbanos



Fonte: Autoria própria, 2025

Nesta amostra (Imagen 7), observa-se um desempenho robusto do modelo na detecção de múltiplos elementos urbanos, com destaque para as classes “rua”, “carro” e “placa de proibido estacionar”, que apresentaram altos níveis de confiança, alcançando até 0,91. Nota-se uma tendência decrescente na confiança à medida que os objetos se afastam da câmera, especialmente nas placas de sinalização mais distantes, o que demonstra a influência da escala e da resolução sobre a precisão das previsões. Apesar de algumas detecções incorretas, como a “faixa de pedestre” identificada com apenas 0,24 de confiança, esses falsos positivos possuem baixo impacto analítico, podendo ser facilmente filtrados por meio de um limiar de confiança (threshold) adequado. A segmentação das calçadas e guias manteve consistência visual, embora pequenas variações de textura e iluminação tenham afetado a diferenciação entre “guia normal” e “guia rebaixada”. De modo geral, o resultado evidencia a capacidade do modelo em reconhecer padrões relevantes em cenas complexas e distantes, demonstrando comportamento coerente com os desafios esperados em ambientes urbanos reais.

## 8.2 Validação de infrações em imagens de casos reais

Nesta seção são apresentados os resultados qualitativos obtidos durante os testes do módulo de validação de infrações, integrando as detecções do modelo às regras algorítmicas responsáveis por classificar veículos em situação regular ou irregular. As imagens reais do conjunto de teste demonstram o comportamento do sistema ao aplicar as lógicas de interseção e relação espacial, verificando tanto a sobreposição entre as máscaras de segmentação do veículo e das zonas proibidas quanto a proximidade e proporção entre as bounding boxes de carros e placas de proibição. Essas análises incluem *thresholds* específicos para cada classe crítica, como calçadas e faixas de pedestres, além da avaliação relacional baseada na distância dinâmica e no *ratio* de áreas entre veículo e placa. As figuras exibem as máscaras, bounding boxes e valores de confiança associados às predições, permitindo observar a robustez da validação em cenários reais com variações de perspectiva, iluminação e complexidade urbana. Esse conjunto de testes evidencia não apenas a capacidade do modelo de identificar corretamente os elementos relevantes, mas também a eficácia do módulo de validação em interpretar essas detecções e determinar, com precisão, a ocorrência de infrações de estacionamento.

### 8.2.1 Veículo estacionado sobre faixa de pedestre

Imagen 8 – Veículo estacionado sobre faixa de pedestre



Fonte: Autoria própria, 2025

Nesta amostra (Imagen 8), o módulo de validação de infrações demonstrou desempenho consistente ao identificar corretamente o veículo, a faixa de pedestres e a calçada, com máscaras e bounding boxes compatíveis com a estrutura real da cena. A lógica de interseção registrou 55 pixels de sobreposição entre o carro e a faixa de pedestres, levando à correta classificação da infração “Estacionado na Faixa”, tratada como prioritária. Esse resultado foi obtido a partir do critério híbrido de detecção: para zonas críticas que exigem contato preciso com o solo como a faixa de pedestres o sistema utiliza a Máscara de Contato do Veículo, correspondente aos 60% inferiores do bounding box, evitando falsos positivos de perspectiva. Mesmo havendo leve interseção com a calçada, a heurística de prioridade foi aplicada corretamente, privilegiando a faixa de pedestres. Para a intersecção com a classe calçada, a verificação é feita com a máscara completa do veículo, assim, a infração só é confirmada quando a sobreposição ultrapassa o limiar específico definido para a mesma. Já o veículo à esquerda foi classificado como regular, sem interseções relevantes com zonas proibidas, evidenciando a capacidade do sistema de diferenciar infrações de situações adequadas em cenários urbanos complexos.

### 8.2.2 Veículo estacionado sobre calçada (passeio)

Imagen 9 – Veículo estacionado sobre calçada (passeio)

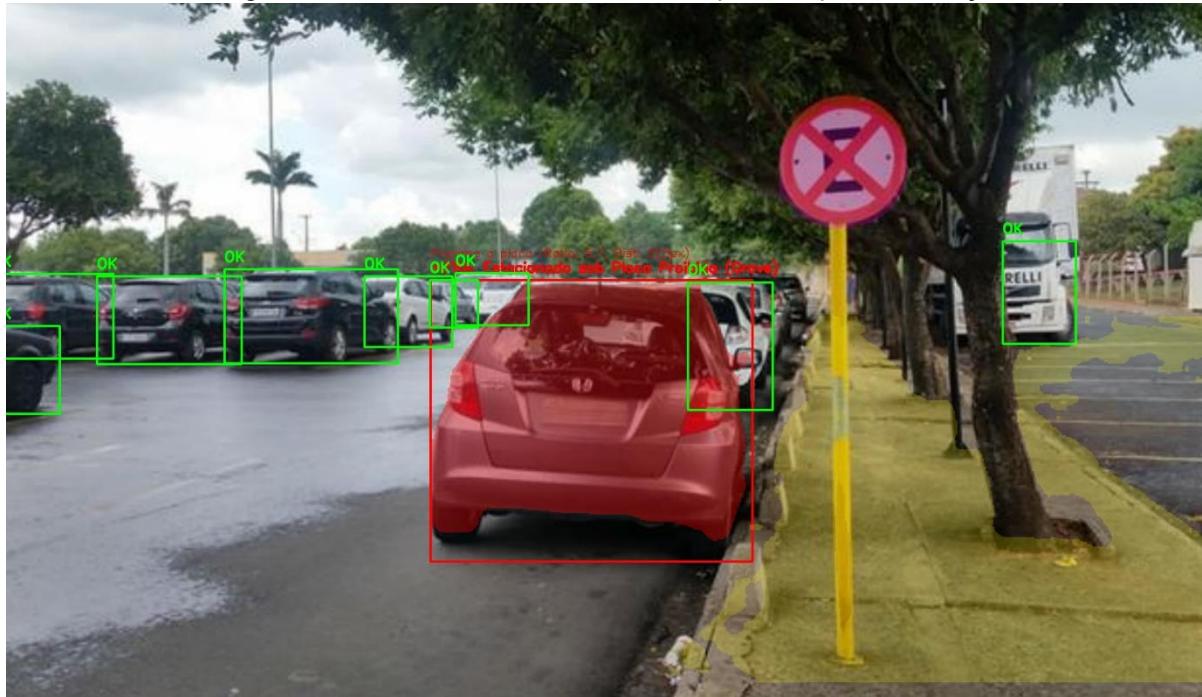


Fonte: Autoria própria, 2025

Nesta amostra (Imagen 9), o módulo de validação de infrações apresentou um desempenho preciso ao identificar de forma clara os elementos urbanos e a irregularidade cometida. O veículo foi corretamente detectado e segmentado, assim como a região da calçada, cuja máscara aparece nitidamente representada na área onde o carro invade o espaço destinado ao pedestre. A sobreposição registrada entre o veículo e a calçada, evidenciada pelo contador de pixels exibindo “102”, confirma de maneira objetiva a infração “Estacionado na Calçada (Grave)”, refletindo exatamente o cenário observado na imagem, onde o carro ocupa indevidamente parte da via destinada à circulação de pessoas. Os elementos estruturais da cena, como o meio-fio, o piso diferenciado da calçada e o relevo do entorno, foram corretamente reconhecidos pelo sistema, reforçando a coerência espacial da detecção. A classificação final destacada em vermelho é compatível com a situação real capturada, demonstrando que o módulo consegue interpretar corretamente não apenas a presença dos objetos, mas também o contexto em que estão inseridos, diferenciando claramente uma condição de infração de um estacionamento regular.

### 8.2.3 Veículo em locais proibidos pela sinalização (placa - Proibido Estacionar)

Imagen 10 – Veículo estacionado em local proibido pela sinalização

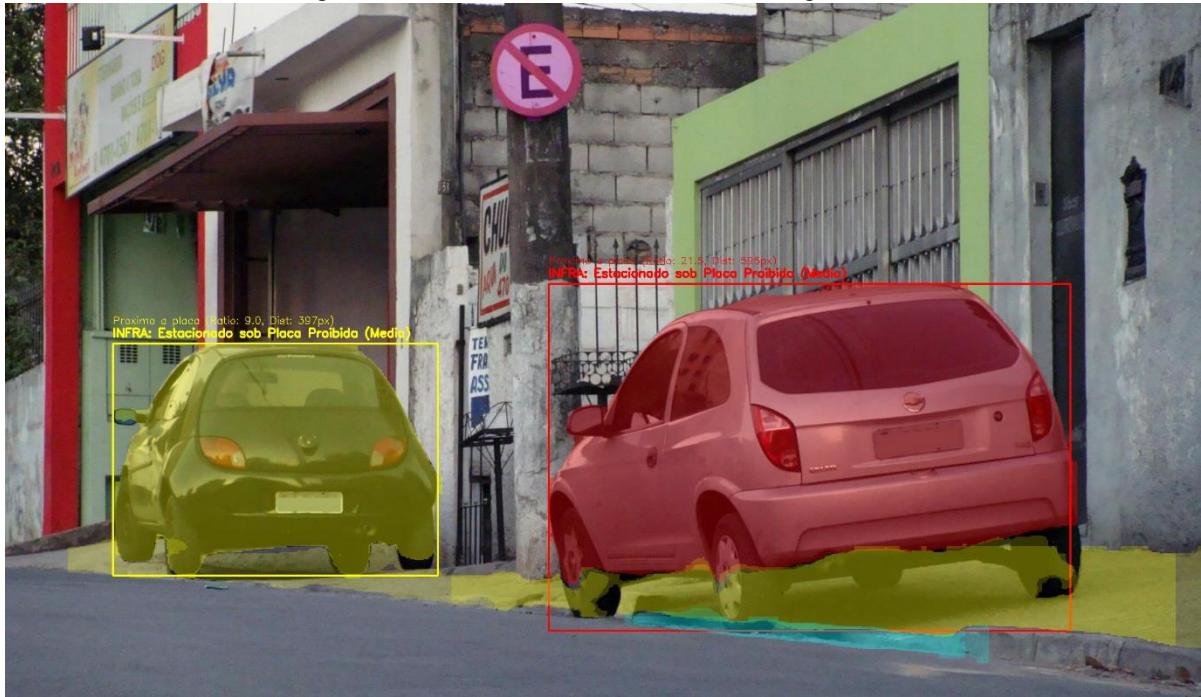


Fonte: Autoria própria, 2025

Nesta amostra (Imagen 10), o módulo de validação de infrações demonstrou precisão e aplicação correta da lógica relacional ao identificar a infração de "Estacionado sob Placa Proibida (Grave)". O sistema detectou e segmentou corretamente os elementos-chave, como o veículo em destaque (pintado em vermelho), a calçada (zona amarela), e a placa de proibido (zona magenta). O carro infrator foi identificado como tal, não por invasão de área (Interseção), mas sim a infração foi confirmada pela proximidade do carro com a sinalização (Critério de Distância Dinâmica e *Ratio* de Área), garantindo que o veículo estivesse na área de influência da placa e não fosse um falso positivo devido a uma placa muito distante ou minúscula. Os outros veículos na imagem foram classificados como "OK" (caixas verdes), indicando a conformidade com as regras de estacionamento para o restante da cena, reforçando a capacidade do modelo de diferenciar corretamente a infração relacional dos estacionamentos regulares.

#### 8.2.4 Casos especiais (múltiplas infrações/múltiplos veículos infratores)

Imagen 11 – Dois veículos estacionados irregularmente

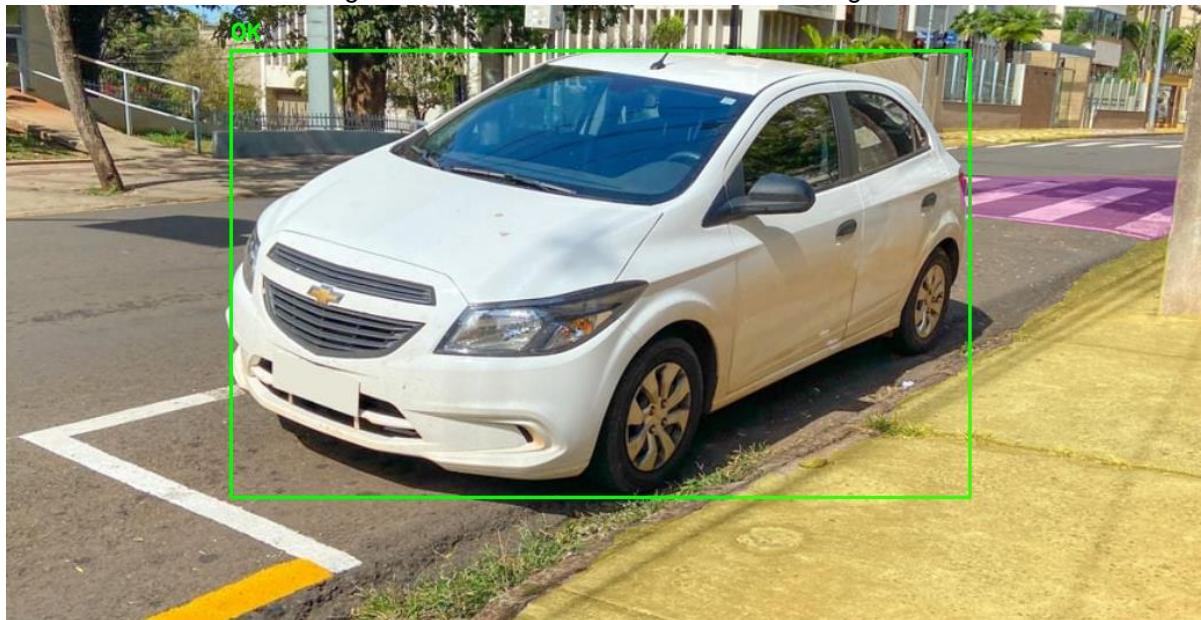


Fonte: Autoria própria, 2025

Nesta nova amostra (Imagen 11), o módulo de validação identificou com sucesso duas infrações relacionais de "Estacionado sob Placa Proibida" (a placa central). O sistema classificou ambos os veículos como infratores por estarem dentro da área de influência da placa. No entanto, o sistema aplicou a regra de priorização visual ao destacar o veículo da direita em vermelho (infrator primário) e o da esquerda em amarelo (infrator secundário). Esta diferenciação baseia-se na heurística que define o infrator primário como o veículo com a maior área de bounding box dentre os infratores, indicando que ele é o elemento de maior destaque na cena. Assim, o modelo consegue não apenas identificar múltiplas infrações simultâneas, mas também fornecer um foco visual imediato ao agente fiscalizador para o caso mais proeminente, facilitando a interpretação do cenário complexo.

### 8.2.5 Veículo estacionado de forma regular

Imagen 12 – Veículo estacionado de forma regular



Fonte: Autoria própria, 2025

Nesta amostra (Imagen 12), o módulo de validação de infrações demonstrou um funcionamento adequado ao reconhecer corretamente que nenhuma irregularidade está presente na cena. Embora a faixa de pedestres tenha sido detectada e exista uma interseção parcial entre ela e o veículo, o sistema aplicou corretamente o critério da Máscara de Contato, que considera apenas os 60% inferiores do bounding box para verificar o contato real com o solo. Como essa interseção não ocorreu na zona de contato do veículo, o módulo evitou um falso positivo e não classificou a cena como infração. Os demais elementos do ambiente área pavimentada, gramado, meio-fio e entorno urbano também foram interpretados corretamente, sem indícios de qualquer zona proibida efetivamente violada.

A ausência de sobreposição entre o veículo e qualquer área restrita, aliada ao posicionamento coerente do carro sobre a via, levou o sistema a classificá-lo corretamente como em conformidade, destacado pelo retângulo verde. A análise espacial confirma que o carro se encontra estacionado dentro dos limites regulares, sem invadir áreas destinadas à circulação de pedestres ou sinalizadas como proibidas. Esse resultado evidencia a capacidade do módulo não apenas de identificar infrações quando presentes, mas também de reconhecer situações totalmente regulares, um aspecto essencial para evitar falsos positivos e garantir confiabilidade no processo de fiscalização automatizada.

## 9 CONCLUSÃO

### 9.1 Retomada do Trabalho e Solução Proposta

Este trabalho enfrentou o desafio da fiscalização ineficiente de estacionamentos irregulares, um problema contínuo nos centros urbanos que compromete a mobilidade, acessibilidade e segurança. Como alternativa à dependência exclusiva de agentes de trânsito, propôs-se o desenvolvimento de um sistema automatizado baseado em visão computacional e inteligência artificial. A metodologia adotada envolveu a criação de um dataset robusto, como também o treino e a implementação de um modelo de visão computacional para detecção de veículos e reconhecimento de placas, integradas a uma plataforma web intuitiva que permite o envio e a análise automatizada de mídias. A abordagem combinou técnicas de métodos de detecção com algoritmos de pré-processamento de imagens, visando criar uma solução escalável e de alta precisão para um problema urbano atual.

### 9.2 Principais Resultados e Contribuições

Os resultados validaram a eficácia da solução proposta. O sistema demonstrou alta confiabilidade na detecção de veículos, com precisão de  $\approx 0.90$  mAP50, e  $\approx 90\%$  de sucesso no reconhecimento de placas quando estas estão visíveis na mídia. A plataforma desenvolvida automatiza integralmente o fluxo de validação, desde o recebimento da imagem até a geração do laudo, exigindo apenas a mídia como input para retornar uma análise completa contendo placa, modelo, cor do veículo, local e horário da suposta infração. Objetivamente, o sistema foi especializado nas cinco infrações de estacionamento mais comuns e impactantes do Artigo 181 do CTB incluindo estacionamento em calçadas, faixas de pedestres e locais com sinalização específica, cobrindo assim a maioria dos cenários problemáticos observados em campo. Vários algoritmos de segurança foram implementados para minimizar falsos negativos, assegurando que potenciais infrações não passem despercebidas.

### **9.3 Limitações e Desafios Enfrentados**

Apesar dos avanços, o trabalho reconhece limitações importantes. O escopo do sistema restringe-se a quatro dos vinte incisos do Artigo 181 do CTB, não abrangendo infrações menos comuns. A performance em vídeos mostrou-se inferior à de imagens estáticas, com funcionalidades como extração de características operando de forma subótima em fluxos contínuos. O tempo de processamento, embora aceitável para análise posterior ao fato, impede o uso para validação em tempo real, podendo chegar a 20 segundos por ocorrência. O modelo também exibe sensibilidade a condições específicas de captura, como ângulos atípicos, obstruções parciais ou iluminação deficiente, resultando numa taxa agregada de falsos positivos e negativos de aproximadamente 30% nestes cenários mais complexos. Estas fragilidades destacam que, embora a automação seja viável e poderosa, a variabilidade do mundo real ainda impõe desafios significativos.

### **9.4 Trabalhos Futuros**

Com vistas a uma melhoria contínua da solução, os trabalhos futuros terão como prioridade imediata, a expansão do conjunto de dados de treinamento para incluir mais variações de ângulos, condições climáticas e iluminação, visando reduzir a taxa de erro. Otimizações de código e a exploração de modelos mais eficientes são necessárias para viabilizar a análise em tempo real e o pleno suporte a vídeos. A longo prazo, a incorporação progressiva dos demais incisos do Artigo 181 do CTB pode se tornar viável, o desenvolvimento de um módulo de análise de contexto de via (ex.: verificação da sinalização vertical) para aumentar a autonomia do sistema, e a integração com bancos de dados oficiais de veículos para automação completa do processo de autuação. Esta evolução consolidaria a plataforma como uma ferramenta indispensável para a gestão urbana inteligente.

## 10 REFERÊNCIAS BIBLIOGRAFICAS

AGÊNCIA CNT, Transporte Atual. **Falta fiscalização de trânsito em três de cada quatro municípios. Confederação Nacional do Transporte**, 4 de novembro de 2015. Disponível em: <https://cnt.org.br/agencia-cnt/falta-fiscalizacao-de-transito-em-tres-de-cada-quatro-municipios-brasileiros-cnt>. Acesso em: 22 de abril de 2025

ALPAYDIN, Ethem. **Introduction to machine learning**. MIT press, 2020.

BERTOLETI, Pedro. **Contagem de objetos em movimento com OpenCV e Python usando Raspberry Pi**. Embarcados, 21 de agosto de 2017. Disponível em: <https://embarcados.com.br/objetos-opencv-e-python-raspberry-pi/>. Acesso em: 24 de abril de 2025.

BIANCHI, Reinaldo Augusto da Costa. **Uso de heurísticas para a aceleração do aprendizado por reforço**. Tese de Doutorado. Universidade de São Paulo, São Paulo, 2004. Disponível em: <https://fei.edu.br/~rbianchi/publications/tese-bianchi.pdf>. Acesso em: 22 de abril de 2025.

BISHOP, Christopher M.; NASRABADI, Nasser M. **Pattern recognition and machine learning**. New York: springer, 2006.

BRASIL. Conselho Nacional de Trânsito. **Manual Brasileiro de Sinalização de Trânsito – Volume I: Sinalização Vertical de Regulamentação**. Departamento Nacional de Trânsito – DENATRAN. Brasília: CONTRAN, 2007. Disponível em: <https://www.gov.br/infraestrutura/pt-br/assuntos/transito>.

BRASIL. *Lei nº 9.503, de 23 de setembro de 1997*. Institui o Código de Trânsito Brasileiro. Diário Oficial da União: seção 1, Brasília, DF, 24 set. 1997. Disponível em: [https://www.planalto.gov.br/ccivil\\_03/leis/l9503compilado.htm](https://www.planalto.gov.br/ccivil_03/leis/l9503compilado.htm). Acesso em: 24 de abril de 2025.

CARLETTI, Vincenzo et al. **Automatic detection of long term parked cars.** 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Karlsruhe, Germany. 2015. Disponível em:  
<https://ieeexplore.ieee.org/document/7301722>. Acesso em: 6 de maio de 2025.

CARVALHO, Carlos Henrique Ribeiro de et al. **A mobilidade urbana no Brasil.** 2011.

Comunicação Social, **Prefeitura vai capacitar guardas da GCM como agentes de fiscalização de trânsito. Prefeitura de Piracicaba.** Disponível em:  
<https://piracicaba.sp.gov.br/noticias/prefeitura-vai-capacitar-guardas-da-gcm-como-agentes-de-fiscalizacao-de-transito/>. Acesso em 24 de abril de 2025

Conselho Nacional de Trânsito Contran. **Imagen 1 – Placa de proibido estacionar (R-6a).** Disponível em: [https://www.gov.br/transportes/pt-br/assuntos/transito/arquivos-senatran/educacao/publicacoes/manual\\_vol\\_i\\_2.pdf](https://www.gov.br/transportes/pt-br/assuntos/transito/arquivos-senatran/educacao/publicacoes/manual_vol_i_2.pdf)

CTB. **Art.181 - Capítulo XV - DAS INFRAÇÕES.** 2021. Disponível em:  
<https://www.ctbdigital.com.br/comentario/comentario181>. Acesso em: 10 de abril de 2025

DENATRAN. "Políticas Nacionais de Trânsito: Estratégias para Redução de Acidentes". 2019.

DOMINGOS, Pedro. **The master algorithm: How the quest for the ultimate learning machine will remake our world.** Basic Books, 2015.

DU, Shan et al. **Automatic license plate recognition (ALPR): A state-of-the-art review.** IEEE Transactions on circuits and systems for video technology, v. 23, n. 2, p. 311-325, 2012. Disponível em: <https://ieeexplore.ieee.org/document/6213519>. Acesso em: 23 de abril de 2025.

GOMES, Dennis dos Santos. **Inteligência Artificial: conceitos e aplicações.** Revista Olhar Científico, v. 1, n. 2, p. 234-246, 2010.

GOODFELLOW, Ian et al. **Deep learning.** Cambridge: MIT press, 2016.

HASTIE, Trevor et al. **The elements of statistical learning.** 2009.

HAYKIN, Simon. **Redes neurais: princípios e prática.** Bookman Editora, 2001.

HEIMBERGER, Markus et al. **Computer vision in automated parking systems: Design, implementation and challenges.** Image and Vision Computing, v. 68, p. 88-101, 2017. Disponível em:  
<https://www.sciencedirect.com/science/article/abs/pii/S0262885617301105>. Acesso em: 03 de maio de 2025

Høye, A. "Speed cameras, section control, and kangaroo jumps—a meta-analysis" (2014). Accident Analysis & Prevention.

IBM. **What is computer vision?** 2019. Disponível em:  
<https://www.ibm.com/think/topics/computer-vision>. Acesso em: 23 de abril de 2025.

IZIDIO, Diogo; FERREIRA, Antonyus; BARROS, Edna. **An Embedded Automatic License Plate Recognition System using Deep Learning.** In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SISTEMAS COMPUTACIONAIS (SBESC), 8., 2018, Salvador. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2018. p. 50-57. ISSN 2237-5430. Disponível em:  
<https://link.springer.com/article/10.1007/s10617-019-09230-5>. Acesso em: 30 de abril de 2025

KUNDU, Rohit. **YOLO: Algorithm for Object Detection Explained [+Examples]. What is YOLO architecture and how does it work? Learn about different YOLO algorithm versions and start training your own YOLO object detection models.**

V7Labs. 17 de janeiro de 2023. Disponível em: <https://www.v7labs.com/blog/yolo-object-detection>. Acesso em: 24 de abril de 2025.

LAROCA, Rayson et al. **A robust real-time automatic license plate recognition based on the YOLO detector**. In: 2018 international joint conference on neural networks (ijcnn). IEEE, 2018. p. 1-10. Disponível em: <https://ieeexplore.ieee.org/document/8489629>. Acesso em 2 de maio de 2025

LEFUNDÉS, Gabriel. **AccessALPR. Deep Learning-based ALPR system for vehicle access control trained on brazilian license plates**. 2019. Disponível em: <https://github.com/glefundes/AccessALPR>. Acesso em: 28 de abril de 2025

LI, Jialin et al. **Detection of Illegal Parking Based on Deep Learning**. In: 2023 IEEE International Conference on Electrical, Automation and Computer Engineering (ICEACE). IEEE, 2023. p. 79-83. Disponível em: <https://ieeexplore.ieee.org/document/10442566>. Acesso em: 4 de maio de 2025

LI, Wei et al. **Vacant parking slot detection in the around view image based on deep learning**. Sensors, v. 20, n. 7, p. 2138, 2020. Disponível em: <https://www.mdpi.com/1424-8220/20/7/2138>. Acesso em: 1 de maio de 2025

LIU, Zhemin; CHEN, Weiling; YEO, Chai Kiat. **Automatic detection of parking violation and capture of license plate**. In: 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE, 2019. p. 0495-0500. Disponível em: <https://ieeexplore.ieee.org/document/8936164>. Acesso em: 2 de abril de 2025

MARTINS, Beatriz Helena. **Trânsito urbano: desafios da fiscalização e da educação no Brasil**. Rio de Janeiro: EDUFRJ, 2020.

MITCHELL, Tom M.; MITCHELL, Tom M. **Machine learning**. New York: McGraw-hill, 1997.

PARENZA, Lutiely Neves; CÂMARA, Sheila Gonçalves. **Relações Pessoa-Cidade: Mobilidade Urbana e Qualidade de Vida em Porto Alegre (RS)**. Universidade Federal de Ciências da Saúde de Porto Alegre, Porto Alegre, RS, Brasil. 2022. Disponível em: <https://doi.org/10.1590/1982-3703003238317>. Acesso em: 24 de abril de 2025

PENG, Xinggan et al. **Real-time illegal parking detection algorithm in urban environments**. IEEE Transactions on Intelligent Transportation Systems, v. 23, n. 11, p. 20572-20587, 2022. Disponível em: <https://ieeexplore.ieee.org/document/9797236>. Acesso em: 2 de abril de 2025

PLATE RECOGNIZER, Accurate, Fast, Developer- Friendly ALPR. 2025. Disponível em: <https://platerecognizer.com>. Acesso em: 20 de abril de 2025

Portal do Trânsito. **Infrações de trânsito mais cometidas no Brasil: excesso de velocidade lidera ranking**. Disponível em: <https://www.portaldotransito.com.br/noticias/fiscalizacao-e-legislacao/infracoes-de-transito-mais-cometidas-no-brasil-excesso-de-velocidade-lidera-ranking/>. Acesso em: 21 de abril de 2025

PREFEITURA DE CURITIBA. **Imagen 3 – Guia rebaixada**. Disponível em: <https://www.curitiba.pr.gov.br/>

PUTRA, M. H. et al. **Convolutional neural network for person and car detection using yolo framework**. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), v. 10, n. 1-7, p. 67-71, 2018. Disponível em: <https://jtec.utm.edu.my/jtec/article/view/3599>. Acesso em: 04 de maio de 2025

REDMON, Joseph et al. **You only look once: Unified, real-time object detection**. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 779-788. Disponível em: [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2016/papers/Redmon\\_You\\_Only\\_Look\\_CVPR\\_2016\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf). Acesso em: 23 de abril de 2025.

ROBOFLOW, **Explore the Roboflow Universe**. 2025. Disponível em:  
<https://universe.roboflow.com>. Acesso em 18 de abril de 2025

RUSSELL, Stuart J.; NORVIG, Peter. **Artificial intelligence: a modern approach**. Pearson, 2016.

RUSSO, Maurício Bastos. **Violência no trânsito "À Moda Brasileira": Insegurança, Letalidade e Impunidade**. 2012. 226f. – Tese (Doutorado) – Universidade Federal do Ceará, Programa de Pós-graduação em Sociologia, Fortaleza (CE), 2012. Disponível em:  
[https://repositorio.ufc.br/bitstream/riufc/6304/1/2012\\_tese\\_mbrusso.pdf](https://repositorio.ufc.br/bitstream/riufc/6304/1/2012_tese_mbrusso.pdf). Acesso em: 05 de abril de 2025.

SOARES, W. **Número de agressões a agentes da CET cresce mais de 20% em São Paulo**. G1 São Paulo, 2017. Disponível em: <https://g1.globo.com/sao-paulo/noticia/numero-de-agressoes-a-agentes-da-cet-cresce-mais-de-20-em-sao-paulo.ghtml>. Acesso em: 20 de abril de 2025.

SRIVASTA, Niharika. **What is SOTA in Artificial Intelligence?**. E2E Networks. 2 de abril de 2025. Disponível em: <https://www.e2enetworks.com/blog/what-is-sota-in-artificial-intelligence>. Acesso em: 22 de abril de 2025.

TAULLI, Tom. **Introdução à inteligência artificial: uma abordagem não técnica**. Novatec Editora, 2020.

TAY, Richard. **The effectiveness of enforcement and publicity campaigns on serious crashes involving young male drivers: Are drink driving and speeding similar?**. Accident Analysis & Prevention, v. 37, n. 5, p. 922-929, 2005.

ULTRALYTICS. **Ultralytics Glossary**. 2025. Disponível em:  
<https://www.ultralytics.com/glossary>. Acesso em: 21 de abril de 2025

ULTRALYTICS. **Home**. 2025. Disponível em: <https://docs.ultralytics.com/>. Acesso em: 21 de abril de 2025

ULTRALYTICS. **Ultralytics YOLO11**. Disponível em:  
<https://docs.ultralytics.com/pt/models/yolo11/>. Acesso em: 21 de abril de 2025

VASCONCELLOS, Eduardo Alcântara. **Transporte urbano, espaço e eqüidade: análise das políticas públicas**. Annablume, 2001.

VINA, Abirami. **Comparing Ultralytics YOLO11 vs previous YOLO models**. Ultralytics. 2 de abril de 2025. Disponível em:  
<https://www.ultralytics.com/blog/comparing-ultralytics-yolo11-vs-previous-yolo-models>. Acesso em: 22 de abril de 2025

ZHANG, Lin et al. **Vision-based parking-slot detection: a benchmark and a learning-based approach**. Symmetry, v. 10, n. 3, p. 64, 2018. Disponível em:  
<https://ieeexplore.ieee.org/document/8019419>. Acesso em: 3 de abril de 2025