



MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
BAIANO – CAMPUS CATU

DEMERSON ARAÚJO RODRIGUES

**REDE NEURAL CONVOLUCIONAL APLICADA AO  
MONITORAMENTO DE TRÂNSITO EM CIDADE DE PEQUENO  
PORTE: UM ESTUDO DE CASO NA CIDADE DE POJUCA-BAHIA**

Catu  
2025

DEMERSON ARAÚJO RODRIGUES

**REDE NEURAL CONVOLUCIONAL APLICADA AO  
MONITORAMENTO DE TRÂNSITO EM CIDADE DE PEQUENO  
PORTE: UM ESTUDO DE CASO NA CIDADE DE POJUCA-BAHIA**

Trabalho de Conclusão de Curso apresentado como  
requisito parcial à obtenção do grau de Tecnólogo  
em Análise e Desenvolvimento de Sistemas do  
Instituto Federal de Educação, Ciência e Tecnologia  
Baiano *Campus Catu*.

Nome do Orientador: Prof. Ma. Eneida Alves Rios

Catu  
2025

DEMERSON ARAÚJO RODRIGUES

**REDE NEURAL CONVOLUCIONAL APLICADA AO  
MONITORAMENTO DE TRÂNSITO EM CIDADE DE PEQUENO  
PORTE: UM ESTUDO DE CASO NA CIDADE DE POJUCA-BAHIA**

Trabalho de Conclusão de Curso apresentado como  
requisito parcial à obtenção do grau de Tecnólogo  
em Análise e Desenvolvimento de Sistemas do  
Instituto Federal de Educação, Ciência e Tecnologia  
Baiano *Campus* Catu.

Nome do Orientador: Prof. Ma. Eneida Alves Rios

APROVADO EM \_\_\_\_/\_\_\_\_/\_\_\_\_ CONCEITO FINAL: \_\_\_\_\_

BANCA EXAMINADORA

---

Prof. Ma. Eneida Alves Rios  
(Orientadora - Instituto Federal Baiano - *Campus* Catu)

---

---

Prof. Dr. Romero Mendes Freire de Moura Júnior  
(Instituto Federal Baiano - *Campus* Catu)

---

Prof. Dr. Gilvan Martins Durães  
(Instituto Federal Baiano - *Campus* Catu)

Catu  
2025

Rodrigues, Demerson Araújo

R696r      Rede neural convolucional aplicada ao monitoramento de trânsito em  
cidade de pequeno porte: um estudo de caso na cidade de Pojuca-Bahia./  
Demerson Araújo Rodrigues.- Catu, BA, 2025.  
63p.; il.: color.

Inclui bibliografia.

Trabalho de Conclusão de Curso (Graduação em Tecnologia em  
Análise e Desenvolvimento de Sistemas ) – Instituto Federal de Educação,  
Ciência e Tecnologia Baiano – Campus Catu.

Orientadora: Profa. Ma. Eneida Alves Rios.

1. Trânsito. 2. Monitoramento de tráfego. 3. Reconhecimento de  
veículos. 4. Visão computacional. 5. Rede neural convolucional. I. Instituto  
Federal de Educação, Ciência e Tecnologia Baiano. II. Rios, Eneida Alves  
(Orient.). III. Título.

CDU: 656.13

## AGRADECIMENTOS

O desenvolvimento deste trabalho de conclusão de curso foi possível graças ao meu empenho, mas que sem o apoio das pessoas certas na minha vida, não seria a mesma coisa.

Primeiramente agradeço a Deus, acima de tudo, que na sua enorme benevolência jamais deixou de me abençoar grandiosamente em todos os aspectos da minha vida, principalmente me mantendo sempre firme e resiliente na busca da realização desse sonho.

À Anna Victoria, minha amada e fiel companheira, que sempre foi minha incondicional apoiadora, estando lado a lado, garantindo todo suporte necessário e principalmente acreditando em mim.

A meu pai, Derivaldo e minha mãe, Geilza, agradeço pelo dom da vida e por terem sido meus bons pais, que nunca me deixaram faltar nada e que seus esforços garantiram a realização desta conquista de concluir o curso e mais um sonho. Sem vocês nada seria possível.

Aos meus tios em geral, e em especial a Joseneide e Manoel, meus segundos pais, que serviram como rede de apoio e amparo nos meus momentos mais necessários.

Aos meus irmãos e primos, pela cumplicidade.

Agradeço também às minhas avós, Daura, em vida, e Evangelina, em memória e coração, que sempre me induziram a perseguir meus sonhos.

Aos professores do curso de Tecnólogo em Análise e Desenvolvimento de Sistemas do IFBaiano *Campus - Catu*, por todos os aprendizados adquiridos. Um agradecimento especial para a Prof. Ma. Eneida Rios, minha orientadora, que além da ajuda e paciência no processo, acreditou na minha capacidade do início ao fim me dando todo o suporte necessário.

Por fim, agradeço a mim mesmo por ter me mantido firme, perseverante e jamais ter perdido a crença de que sou capaz apesar de dificuldades e adversidades, pois é no íntimo de cada ser que ocorrem as batalhas mais duras.

Portanto, graças a toda a minha base de apoio incondicional, a realização deste sonho está sendo possível. A todos, meu muito obrigado!

*O sonho das pessoas nunca têm fim!* - Marshall D. Teach  
Eiichiro Oda, One Piece, 2001.

## RESUMO

O crescimento acelerado das áreas urbanas tem intensificado os desafios relacionados à mobilidade urbana e ao gerenciamento do tráfego, inclusive em cidades de pequeno porte. Nesse contexto, a aplicação de tecnologias inteligentes torna-se essencial para otimizar o fluxo de veículos, reduzir congestionamentos e aumentar a segurança viária. Este trabalho de conclusão de curso propôs o desenvolvimento de um sistema inteligente, intitulado *Traffic Viewer*, uma ferramenta para análise e monitoramento do trânsito. A aplicação investiga o uso de técnicas de visão computacional de rede neural convolucional, responsáveis por detectar e rastrear objetos para o monitoramento do tráfego urbano em áreas centrais e movimentadas de municípios de pequeno porte. A fim de validar o desempenho do sistema e compreender uma realidade prática, um estudo de caso foi realizado na cidade de Pojuca - BA, com foco no reconhecimento, classificação e contagem de fluxo de veículos, de forma a disponibilizar dados e informações para gestão de trânsito local no apoio ao controle e tomada de decisões. Os resultados da pesquisa após a aplicação do estudo de caso indicaram que o fluxo nas vias urbanas de cidade de pequeno porte também é uma questão a ser explorada, principalmente devido predominância de veículos leves, como automóveis e motocicletas, em locais onde se concentram as atividades comerciais, reforçando a necessidade de estratégias específicas para melhorar a mobilidade urbana em municípios pequenos.

**Palavras-chave:** Trânsito; Monitoramento de tráfego; Reconhecimento de veículos; Visão Computacional; Rede Neural Convolutional.

## *ABSTRACT*

The rapid growth of urban areas has intensified challenges related to urban mobility and traffic management, even in small cities. In this context, the application of smart technologies becomes essential to optimize vehicle flow, reduce congestion, and increase road safety. This final course project proposed the development of an intelligent system called Traffic Viewer, a tool for analyzing and monitoring traffic. The application investigates the use of convolutional neural network computer vision techniques, responsible for detecting and tracking objects for monitoring urban traffic in central and busy areas of small municipalities. In order to validate the system's performance and understand a practical reality, a case study was conducted in the city of Pojuca, Bahia, focusing on the recognition, classification, and counting of vehicle flow, in order to provide data and information for local traffic management to support control and decision-making. The results of the research after the application of the case study indicated that traffic flow on urban roads in small towns is also an issue to be explored, mainly due to the predominance of light vehicles, such as cars and motorcycles, in places where commercial activities are concentrated, reinforcing the need for specific strategies to improve urban mobility in small municipalities.

**Keywords:** Traffic; Traffic monitoring; Vehicle recognition; Computer vision; Convolutional neural network.

## Lista de Figuras

Figura 1 - Fluxograma da Metodologia do trabalho

Figura 2 - Fluxograma da Implementação de uma Visão Computacional

Figura 3 - Funcionamento de uma Rede Neural Convolucional

Figura 4 - Comparativo entre as versões *YOLO*

Figura 5 - Estrutura de desenvolvimento do Sistema *Traffic Viewer*

Figura 6 - *Visualização do dataset no Roboflow*

Figura 7 - *Workspace de Versões*

Figura 8 - Ambiente no *Google Colab* para treinamento do modelo *YOLO*

Figura 9 - Documentação *Swagger Traffic Viewer API*

Figura 10 - Modelagem do banco de dados

Figura 11 - Tela de *Dashboard*

Figura 12 - Tela de Registros

Figura 13 - Tela de Registros com aplicação de filtro

Figura 14 - Tela do Processamento com Visão computacional

Figura 15 - Volume por tipo de veículo

Figura 16 - Velocidade média por tipo de veículo

Figura 17 - Direções de curso mais comuns

Figura 18 - Volume de veículos por hora

Figura 19 - Volume por local

Figura 20 - Volume de veículos por dia

Figura 21 -

Valores das métricas do modelo

## **Lista de Tabela**

**Tabela 1 - Tabela com os valores das métricas do modelo**

## **Lista de Abreviaturas**

<b>AMT -</b>	Autoridade Municipal de Trânsito
<b>AP -</b>	Administração Pública
<b>CTB -</b>	Código de Trânsito Brasileiro
<b>CNN -</b>	Convolutional Neural Networks
<b>CUDA -</b>	Compute Unified Device Architecture
<b>DENATRAN -</b>	Departamento Nacional de Trânsito
<b>SENATRAN -</b>	Secretaria Nacional de Trânsito
<b>SNT -</b>	Sistema Nacional de Trânsito
<b>STT -</b>	Superintendência de Trânsito e Transportes
<b>VC -</b>	Visão Computacional
<b>YOLO -</b>	You Only Look Once

## SUMÁRIO

1. **INTRODUÇÃO**
  - 1.1. CONTEXTUALIZAÇÃO
  - 1.2. JUSTIFICATIVA
  - 1.3. OBJETIVO GERAL
  - 1.4. OBJETIVOS ESPECÍFICOS
  - 1.5. HIPÓTESE
2. **METODOLOGIA**
  - 2.1. REVISÃO DA LITERATURA
  - 2.2. ANÁLISE DAS FERRAMENTAS E TECNOLOGIAS
  - 2.3. DESENVOLVIMENTO DO SISTEMA DE MONITORAMENTO, DETECÇÃO, CLASSIFICAÇÃO E CONTAGEM DE VEÍCULOS
    - 2.3.1. PLANEJAMENTO E TREINAMENTO
      - 2.3.1.1. PREPARO DA BASE DE DADOS NO *ROBOFLOW*
      - 2.3.1.2. TREINAMENTO DO MODELO ATRAVÉS DO *GOOGLE COLAB*
    - 2.3.2. ELABORAÇÃO DOS CÓDIGOS
  - 2.4. INTEGRAÇÃO DO SISTEMA
  - 2.5. TESTES DO SISTEMA
3. **REFERENCIAL TEÓRICO**
  - 3.1. VISÃO COMPUTACIONAL
  - 3.2. REDES NEURAIS CONVOLUCIONAIS - *CNNs*
    - 3.2.1. *YOU ONLY LOOK ONCE (YOLO)*
4. **TRABALHOS RELACIONADOS**
5. **PROPOSTA DE TECNOLOGIA**
  - 5.1. DESCRIÇÃO DA SOLUÇÃO PROPOSTA
    - 5.1.1. ETAPA DA IA
    - 5.1.2. ETAPA API
    - 5.1.3. ETAPA BANCO DE DADOS
    - 5.1.4. ETAPA INTERFACE WEB
      - 5.1.4.1. FUNCIONAMENTO DO SISTEMA *TRAFFIC VIEWER*

- 5.1.4.2. CAPTURA E ENVIO DE DADOS
- 5.1.4.3. PROCESSAMENTO COM VISÃO COMPUTACIONAL
- 5.1.4.4. ARMAZENAMENTO E ORGANIZAÇÃO DOS DADOS
- 5.1.4.5. VISUALIZAÇÃO E INTERAÇÃO COM OS DADOS

## **6. RESULTADOS E DISCUSSÕES**

### **6.1. ESTUDO DE CASO**

- 6.1.1. ESTRATÉGIA DE VALIDAÇÃO
- 6.1.2. EXECUÇÃO DO TESTE
- 6.1.3. RESULTADOS OBTIDOS
- 6.1.4. DISCUSSÃO DOS RESULTADOS

- 6.2. AVALIAÇÃO DO SISTEMA
- 6.3. COMPARATIVO DO SISTEMA *TRAFFIC VIEWER* COM TRABALHOS RELACIONADOS

## **7. CONSIDERAÇÕES FINAIS**

## **8. REFERÊNCIAS**

## 1. INTRODUÇÃO

### 1.1 Contextualização

Dentre os principais problemas no trânsito brasileiro, o congestionamento de veículos é um grande desafio, principalmente em grandes cidades. Apesar da existência das leis previstas no Código de Trânsito Brasileiro (CTB), que visam viabilizar um bom funcionamento do tráfego, de ações operacionais e da implantação de tecnologias para controle de trânsito, é difícil gerenciar a aglomeração de veículos, principalmente devido ao crescimento populacional e aumento recorrente do número da frota sem que a infraestrutura das cidades e o transporte público acompanhem esse crescimento.

Diversos estudos (Nobre, 2023) sobre a mobilidade urbana e o monitoramento do tráfego têm surgido com o intuito de resolver as lacunas existentes e apoiar a gestão pública. Contudo, os problemas no trânsito se intensificaram, principalmente em razão do aumento crescente do número de veículos nacionais. Segundo uma análise da Data OLX Autos, de acordo com o censo do IBGE, de 2022, houve um aumento significativo da frota, representando um acréscimo percentual de 43,7%, que em 2010 era de 64,8 milhões de veículos, passando para 115,1 milhões em 2022.

Diante da grande quantidade de veículos circulando, o fluxo no tráfego é cada vez mais intenso, principalmente em horários de pico, resultando em congestionamentos de grandes proporções, e como consequência desse cenário, obtém-se um quadro caótico propício a acidentes. As autoridades de trânsito e órgãos competentes atuam diante do tráfego para evitar/minimizar quaisquer ocorrências, tarefa difícil diante do cenário brasileiro atual.

Acontece que, apesar da discussão ser homogênea sobre as dificuldades relativas ao trânsito, estas se manifestam de formas distintas, a depender do porte das cidades. No caso das grandes e médias cidades, pode-se deduzir maiores transtornos devido ao contingente de pessoas e veículos, porém estas desfrutam de alternativas mais eficazes para solução de problemas devido ao maior aporte financeiro. Entretanto, municípios de pequeno porte também enfrentam diversos desafios no trânsito, como por exemplo, falta de sinalização adequada, estacionamento irregular, ausência de semáforos em cruzamentos movimentados,

excesso de velocidades em vias locais entre outros que certamente podem aumentar o risco de acidentes. Segundo dados do DataSus (CONFEDERAÇÃO NACIONAL DE MUNICÍPIOS, 2023), apesar da concentração de sinistros estarem em cidades grandes, a maior taxa de internação e óbitos está voltada para os municípios pequenos e de médio porte.

Diante de tantos problemas de mobilidade urbana, outro dado relevante a ser considerado, principalmente em cidades de pequeno porte, é que as motos ganharam um destaque de uso, por sua ágil mobilidade para contornar engarrafamentos, realizar viagens rápidas e o viés econômico favorável sendo uma aquisição mais rentável, entretanto, o crescimento desse meio de transporte, por sua vez, tem aumentado os problemas no tráfego das vias urbanas. Segundo a coleta do SENATRAN (2023), as motos ocupam o segundo lugar de veículos em maior quantidade de frota, com 28% equivalentes a 32.474.102 em circulação, representando uma (1) moto para cada seis (6) pessoas.

Diante das problemáticas que circundam o trânsito e a mobilidade urbana, existe o programa de Municipalização que amparado pelo Sistema Nacional de Trânsito (SNT) busca integrar novos municípios ao seu corpo, contando com o apoio da Secretaria Nacional do Trânsito (SENATRAN). Foi com base nesta iniciativa, que municípios de pequeno porte passaram a aderir ao programa de Municipalização, para se adequarem e buscarem melhores condições de gestão dos tráfegos municipais das suas cidades, como é o caso de Pojuca, no interior da Bahia.

A cidade de Pojuca, situada na região metropolitana de Salvador, é definida como uma cidade de pequeno porte, que, segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), são aquelas que possuem até 50 mil habitantes (VIEIRA *et al.*, 2004, p. 23). Esta cidade será modelo de estudo nesta pesquisa por englobar critérios tais como: integração do município ao Sistema Nacional de Trânsito (SNT), sua classificação pela concentração populacional urbana que se caracteriza como pequena cidade e o expoente crescimento do contingente de veículos do município nos últimos anos.

Apesar do pequeno porte, cidades pequenas como Pojuca-BA estão suscetíveis de enfrentar desafios semelhantes aos das grandes metrópoles com relação à gestão do trânsito e à mobilidade urbana. A implementação de tecnologias

avançadas, como as Redes Neurais Convolucionais (CNNs), pode ser uma solução eficaz para essas dificuldades. Estudos (IBM, 2023) indicam que a utilização de CNNs em sistemas de monitoramento de tráfego tem se mostrado eficiente na detecção e classificação de veículos, contribuindo para uma gestão mais precisa e segura do trânsito .

As Redes Neurais Convolucionais (CNNs) têm se destacado como uma das principais arquiteturas no campo da visão computacional, principalmente no reconhecimento de imagens. Inspiradas na organização do córtex visual dos animais, as CNNs são projetadas para processar dados visuais de maneira eficiente, utilizando camadas convolucionais que capturam características espaciais e hierárquicas das imagens (IBM, 2023). Segundo LeCun et al. (1998), as CNNs são capazes de aprender representações de dados em múltiplos níveis de abstração, o que as torna muito eficazes para tarefas de classificação e detecção de objetos.

A aplicação de CNNs no reconhecimento de imagens tem revolucionado diversas áreas. Por exemplo, Krizhevsky et al. (2012) demonstraram que as CNNs podem superar métodos tradicionais em competições de reconhecimento de imagens, como o *ImageNet*. Além disso, a capacidade das CNNs de aprender com grandes volumes de dados rotulados permite que esses modelos melhorem continuamente seu desempenho, tornando-se ferramentas indispensáveis para o aprendizado de máquina e a inteligência artificial.

Assim como as grandes cidades têm investido em inovações tecnológicas para gerir o trânsito e melhorar a mobilidade urbana, pequenas cidades também podem se beneficiar da implantação de tecnologias, já não dispõem em quantidade de alternativas tecnológicas modernas como as grandes cidades, para avaliar internamente o trânsito da cidade, como por exemplo, câmeras de monitoramento dedicados exclusivamente ao tráfego de veículos. A adoção de um sistema de visão computacional baseado em CNNs, pode auxiliar em um monitoramento mais preciso e eficiente do tráfego. Esses sistemas podem ser capazes de identificar e classificar veículos, pedestres e outros elementos através de um treinamento para o reconhecimento de imagens por meio de vídeos.

Sendo assim, diante dos desafios reconhecidos e enfrentados por cidades pequenas, tais como a cidade de Pojuca, que por muitas vezes carecem de

infraestrutura tecnológica robusta para realizar o monitoramento do tráfego, esta pesquisa visa propor o desenvolvimento de um sistema de monitoramento, contagem e classificação de veículos baseado em Redes Neurais Convolucionais (CNNs). Desta forma, o objetivo é investigar e compreender como um sistema de reconhecimento de veículos pode gerar dados para analisar o tráfego da cidade, sendo capaz de munir o órgão responsável pelo gerenciamento do trânsito local com dados que orientem nas tomadas de decisões em busca de uma gestão mais eficiente e acessível das vias urbanas.

## **1.2 Justificativa**

O número de veículos vem crescendo de forma contínua, impulsionado pelo avanço da urbanização. Com isso, mesmo os municípios de pequeno porte, como a cidade de Pojuca-BA, passam a enfrentar desafios relacionados à gestão do tráfego. Nesse cenário, torna-se determinante buscar soluções acessíveis e eficazes para lidar com o aumento da circulação de veículos, especialmente em locais com estrutura limitada, visando a manutenção da mobilidade.

De acordo com o estabelecido no Decreto de nº 8.324, de 27 de outubro de 1910, nota-se a importância em gerir o trânsito, justamente em função do aumento do fluxo. Conforme o avanço dos anos, foi perceptível o aumento do número de veículos e a crescente no fluxo destes nas regiões urbanas, propiciando um ambiente complexo de exposição cotidiana, visto que o uso de meios de deslocamento automotores se tornou essencial para a vida humana com os benefícios ofertados para o deslocamento ágil.

A utilização de tecnologias alternativas, como os sistemas de monitoramento baseados em redes neurais convolucionais, surge como uma possibilidade viável e promissora. Essas redes, com sua capacidade de reconhecimento de objetos, permitem a contagem e classificação automática do fluxo de veículos em tempo real. Trata-se de uma solução de baixo custo, que pode ser implementada mesmo em cidades com recursos mais restritos, como é o caso de Pojuca-BA.

Segundo o Guia de Municipalização (2024, p. 24), um município necessita dispor de estrutura e capacidade para executar as atividades e competências legais, minimamente sendo: engenharia de tráfego, fiscalização e operação de trânsito,

educação de trânsito, coleta, controle e análise estatística de trânsito e julgamento de recursos contra penalidades por eles impostas.

A integração de um município ao que rege o SNT gera o favorecimento na solução de questões para áreas diversas de uma cidade, não se abstendo necessariamente ao trânsito. Ocorre a melhora da mobilidade urbana pela melhor administração da articulação do tráfego, da sinalização, estacionamentos e locomoção dos pedestres. Há também o estreitamento das relações com demais órgãos do SNT e outros setores. Além de que registros apontam que municípios há tempos integrados ao SNT obtiveram a redução de acidentes e mortes em decorrência do trânsito, consequentemente reduzindo despesas hospitalares. (Guia de Municipalização, 2024, p. 16 e 17).

Contudo, ainda existe uma quantidade considerável de municípios brasileiros que não aderiram ao programa de Municipalização do SNT. Desta forma, é importante o desenvolvimento de iniciativas para integrar as cidades ao SNT, de modo que a prefeitura assuma responsabilidades de gestão de trânsito local.

A cidade de Pojuca, como exemplo de município integrado, aderiu ao processo de integração e se tornou efetivamente competente com o SNT a partir de 2022, por meio do estabelecimento da Lei Municipal Complementar nº 012/2022, de 29 de dezembro de 2022. Foi instituído o Órgão da Superintendência de Trânsito e Transportes (STT), que dispõe de um Superintendente, o qual é a Autoridade Municipal de Trânsito (AMT) da cidade.

A motivação para aplicar essa tecnologia em Pojuca está diretamente ligada à necessidade de modernizar os processos de gestão do trânsito local. Atualmente, a STT dispõe de recursos limitados para monitoramento e controle do tráfego, atendo-se à fiscalização realizada por guardas de trânsito, o que dificulta e limita o monitoramento e a tomada de decisões baseadas em dados concretos coletados de maneira geral.

Com a implementação de um sistema inteligente de análise de tráfego, seria possível não apenas acompanhar o fluxo de veículos com maior precisão, mas também planejar intervenções mais eficazes, melhorar a mobilidade urbana e garantir maior segurança viária. Além disso, a coleta contínua de dados

proporcionada por essas tecnologias permite uma análise mais ampla e estratégica do trânsito na cidade, possibilitando uma gestão mais eficiente das vias públicas.

### **1.3 Objetivo Geral**

Implementar uma solução de reconhecimento de veículos baseada em redes neurais convolucionais para análise de vídeos, produzindo informações para proporcionar o melhor monitoramento e orientação da gestão do trânsito da cidade de Pojuca-Ba. Esta ferramenta deverá ser capaz de identificar, segmentar, classificar e contar veículos, coletar dados sobre o fluxo de trânsito e gerar relatórios detalhados que auxiliem as autoridades municipais na tomada de decisões estratégicas para a melhoria da fluidez do tráfego e otimização das vias públicas.

### **1.4 Objetivos Específicos**

- Desenvolver um *dataset* utilizando banco de imagens gratuitas para treinamento de uma IA via Google Colab;
- Desenvolver e treinar uma IA para ser capaz de identificar, detectar, classificar e contar veículos, utilizando a rede neural convolucional YOLO v8;
- Desenvolver a Interface *Web* para consumo, análise e exportação de dados, por meio de relatórios;
- Aplicar a solução desenvolvida no contexto da cidade de Pojuca-Ba.

### **1.5 Hipótese**

A utilização de uma rede neural convolucional para análise de vídeos das vias públicas realizará o reconhecimento de veículos, coleta e geração de informações do fluxo do trânsito na cidade de Pojuca-Ba, que orientará a autoridade de trânsito do município na tomada de decisões sobre a gestão do tráfego.

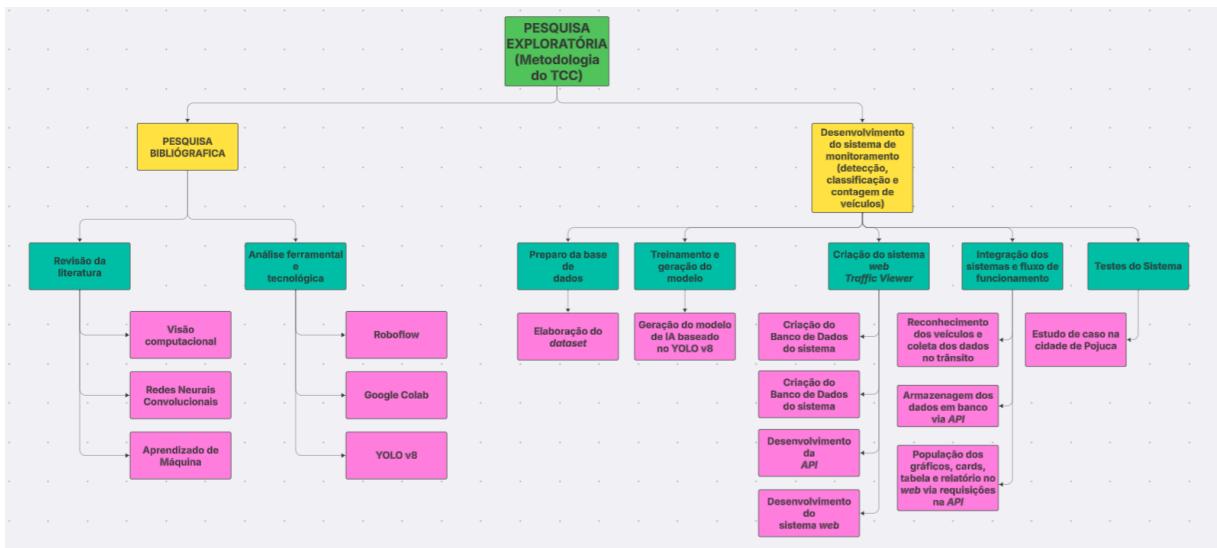
## **2. METODOLOGIA**

A metodologia de pesquisa adotada neste trabalho consiste na pesquisa exploratória e a técnica utilizada é a pesquisa bibliográfica e o estudo de caso.

Para realização das atividades a metodologia foi dividida nas seguintes etapas: revisão da literatura com a coleta de dados e levantamento de trabalhos correlatos, análise de ferramentas e tecnologias para amparar a implementação do

sistema, desenvolvimento do sistema de monitoramento, detecção, classificação e contagem de veículos que englobou as seguintes atividades: planejamento e treinamento dos dados, codificação, integração e testes do sistema com a aplicação do estudo de caso na cidade de Pojuca-Ba, e, por fim, tem-se a análise dos resultados obtidos e discussões, conforme ilustrado na Figura 1.

**Figura 1** - Fluxograma da Metodologia do trabalho



Fonte: próprio autor, 2025.

## 2.1 Revisão da Literatura

Nesta etapa foi realizado o levantamento bibliográfico para proporcionar ao autor uma familiaridade com o tema de visão computacional e as tecnologias utilizadas de Rede Neural Convolucional e Aprendizagem de Máquina na resolução de problemas de localização e classificação de objetos em vídeos e imagens. Os artigos e trabalhos já publicados ajudaram na identificação de lacunas existentes em relação a tecnologias específicas para auxiliar na gestão de trânsito de pequenas cidades.

## 2.2 Análise de Ferramentas e Tecnologias

Nesta fase a intenção foi estudar e analisar como uma CNN como a YOLO V8 com seus modelos pré-treinados faz para a detecção e classificação de objetos por meio da visão computacional e o aprendizado de máquina.

Os dados coletados permitiram visualizar os diferentes cenários e condições para desenvolver e treinar uma IA capaz de identificar, segmentar, classificar e contar veículos, e que neste trabalho, especificamente foram divididos em 4 classes: caminhões, carros, motocicletas e ônibus.

Para criar e gerenciar modelos de visão computacional foi necessário analisar a estrutura para organizar *datasets* de imagens, e que favorecesse o treinamento de um modelo *YOLO v8*. Para isto, pode-se contar com a utilização de plataformas como a *Roboflow*, a qual foi utilizada neste trabalho. Ela trata-se de uma plataforma *online* que consegue consumir imagens para serem convertidas em visão computacional, facilitando a criação do conjunto de dados para treinar um modelo de IA e do conjunto de validação para testar os resultados e medir a qualidade do modelo treinado. Os dados oferecidos por esta plataforma são disponibilizados com permissão para o uso público podendo ser aplicados para pesquisas e desenvolvimento de inovações.

Por meio da análise das tecnologias, informações importantes foram colhidas para não somente fundamentar a pesquisa, mas sim contar com dados de extrema relevância para projetos que visam trabalhar com IA.

Diante da necessidade de criar um banco de imagens e treinar o modelo com essas imagens, é importante destacar que a coleta desses dados se deu por meio da busca em ambientes de livre acesso, ou seja, repositórios públicos para consumo de imagens diversas que foram aplicadas na tentativa de oferecer qualidade e bom volume ao treinamento do modelo.

Com base nisso, o ferramental utilizado busca se apoiar nas condições pré-estabelecidas para desenvolver um sistema inteligente. O projeto usa da linguagem *Python v3.10.7*, linguagem que possui uma sintaxe prática, com vasto suporte à diferentes paradigmas de programação e também conta com bibliotecas para aprendizado de máquina que operam bem com o modelo *YOLO V8*, para o processamento dos dados e o processo de treinamento da IA. Assim, o projeto foi desenvolvido em *Python v3.10.7*, aproveitando sua sintaxe simples e o amplo suporte a bibliotecas voltadas ao aprendizado de máquina, essenciais para o processamento dos dados e o treinamento do modelo *YOLO v8*.

## 2.3 Desenvolvimento do sistema de monitoramento, detecção, classificação e contagem do fluxo de veículos

### 2.3.1 Planejamento e treinamento

#### 2.3.1.1 Preparo da base de dados no *Roboflow*

A etapa de preparação dos dados consistiu na elaboração de um banco de imagens dos objetos a serem reconhecidos pelo sistema, com o propósito de serem incorporados a uma coleção de imagens na plataforma *Roboflow*. As imagens incorporadas à plataforma passaram por um processo de categorização e foram classificadas nas respectivas classes: caminhões, carros, motos e ônibus. A classificação consiste no procedimento de seccionar uma área retangular nas imagens, denominadas *bouding boxes*, delimitando o objeto de análise para aprendizado e reconhecimento da IA.

Uma vez concluídas as etapas citadas, foi então assim estabelecido um *dataset* que foi exportado via *token url* e submetido como base para o treinamento do modelo. Para exportação deste modelo e uso no treinamento como descrito, é gerado o *token* que contém os dados do *dataset* para poder ser copiado e associado no *Google Colab*.

#### 2.3.1.2. Treinamento do modelo através do *Google Colab*

Com o *dataset* produzido, a etapa de treinamento consistiu em utilizar o *Google Colab*, que se trata de uma plataforma gratuita oferecida pela *Google* e possibilita um ambiente de desenvolvimento *Python online*, além de também oferecer recursos como *GPU's* e *TPU's*.

Ao trabalhar com o *Google Colab* para fazer um treinamento e obter o modelo treinado de visão computacional é fundamental preparar o ambiente para estar apto a recepcionar o modelo *dataset*. As instalações necessárias foram, respectivamente, as seguintes bibliotecas: *Nvidia-smi*, *Ultralytics* e *Roboflow*. A *Nvidia-smi* prepara um modelo de *GPU* para ser utilizado no ambiente, a *Ultralytics* é a plataforma responsável por proporcionar o trabalho com um grande volume de dados e gerar partilha dos mesmos e por fim o próprio *Roboflow* que possibilita criar, treinar e implementar modelos de visão computacional. Além disso, é importante destacar que o utilitário *nvidia-smi* é amplamente utilizado para verificar o funcionamento da

*GPU* durante o treinamento. A biblioteca Ultralytics, por sua vez, oferece suporte direto ao modelo *YOLO v8*, permitindo gerenciar o processo de treinamento e validação. Por fim, o Roboflow contribui para a criação e organização dos conjuntos de dados, facilitando a integração com o modelo e garantindo eficiência na etapa de preparação.

### **2.3.2 Elaboração dos códigos**

A codificação para o projeto consistiu em três (3) etapas (*IA*, *API* e *Web*) e um banco de dados para comportar os dados do projeto de modo geral. O desenvolvimento resultou na utilização das linguagens, frameworks, bibliotecas e banco: *Python* (biblioteca *Ultralytics*, *OpenCV*), *TypeScript*, *React Js + Vite*, *NestJS*, *Prisma*, *PostgreSQL*.

A etapa da *IA*, conforme relatado na seção anterior, consistiu em usar o Google Colab para treinar o modelo da IA capaz de detectar, classificar e contar as classes de veículos esperadas: carros, motos, caminhões e ônibus.

Em seguida foi preparado a Etapa da *API*, para estabelecer as conexões entre os dados gerados da *IA* para o banco de dados com o projeto *Web*. A API foi estruturada para atender requisições de criação (POST), consulta e filtragem (GET) nos *endpoints* Veículos, Registros, Estatísticas e Exportação, contando com documentação via *Swagger* para facilitar a visualização e teste das rotas. Para o banco de dados, utilizou-se o ORM *Prisma*, garantindo a modelagem adequada e integração eficiente com os demais componentes do projeto.

Por final, a última etapa consistiu na implementação da Interface *Web*, em que foi elaborado um site simples com duas telas. A tela inicial apresenta um *Dashboard* para acompanhamento dos dados por meio de gráficos e a segunda tela exibe registros para consulta geral e filtragem dos dados. Além disso, a interface permite a exportação via *PDF* destes dados visualizados na tabela, tendo como saída os relatórios.

## **2.4 Integração do sistema**

A atividade de integração do sistema, consistiu em realizar a conexão entre todos elementos elaborados. Após a finalização de cada componente individual modelo de *IA*, *API* e Interface *Web*, foi necessário estabelecer arquivos de

comunicação entre os mesmos, para que todo o operacional ficasse coeso e pudesse então funcionar.

Unificando os dados alimentados em banco pela operação do projeto *IA*, a *API* pode estar realizando as devidas requisições para alimentar gráficos, tabelas e documentos de exportação existentes no sistema de *web* que foi o produto final obtido. Esta implementação refletiu num sistema *web* completamente funcional e de acordo com o planejamento elaborado.

## **2.5 Testes do sistema**

Para validar o sistema um estudo de caso foi realizado com vídeos de algumas vias urbanas da cidade de Pojuca - BA. Os testes consistiram em submeter a visão computacional a uma coleção total de oitenta e quatro (84) vídeos, referentes a três localizações específicas do município. As regiões escolhidas, proporcionaram análises e perspectivas distintas para que o sistema pudesse ser avaliado com expectativas variadas e então validar seu funcionamento.

As distinções previstas nos oitenta e quatro vídeos utilizados foram: trabalhar diferentes ambientes e seus ângulos de visualização, quantitativo de veículos e as variedades dos tipos, com seus dados específicos como velocidade média, e direcionamento no trânsito, além dos períodos de exposição.

## **3. REFERENCIAL TEÓRICO**

### **3.1 Visão Computacional**

Dentre a diversidade existente no ramo da Inteligência Artificial, há um meio capaz de mimetizar a capacidade humana de análise, aprendizado, verificação e validação através de recursos computacionais, este ramo é chamado de Visão Computacional (VC) (Rodrigues, 2021).

A visão computacional busca capacitar os computadores para interpretar e compreender o mundo visual utilizando técnicas, como *Machine Learning* (aprendizado de máquina) e Redes Neurais Convolucionais (*CNNs*). Pela sua ampla área de atuação, a visão computacional tem se destacado podendo auxiliar diversos processos e de maneiras variadas, permitindo que as máquinas extraiam

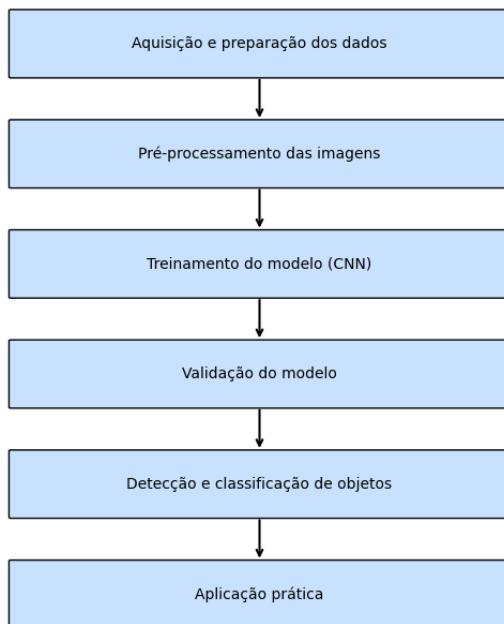
informações significativas de imagens e vídeos e simule a capacidade humana de ver e entender o ambiente.

Segundo Rodrigues (2017), a partir do aperfeiçoamento das ferramentas de detectar e analisar objetos, a visão computacional tornou os sistemas capazes de tomarem decisões apenas por meio de elementos visuais, isso favoreceu, dentre alguns exemplos a criação de veículos autônomos, a produção de mapas de ambientes e detecção de obstáculos (*apud* MACHADO, 2017), a detecção de anomalias em exames por imagens de tomografias computadorizadas (*apud* BARELLI, 2018) e até mesmo classificação de peças mecânicas por geração de imagens sintéticas (*apud* RODRIGUES, 2020).

A aplicação da visão computacional passa por um conjunto de etapas (Figura 2) que envolvem desde a coleta de dados até a análise final, que consiste em interpretar os resultados obtidos pelo modelo, avaliar métricas de desempenho e sintetizar informações relevantes para apoiar decisões ou aplicações práticas. Inicialmente, ocorre a aquisição e preparação dos dados, onde são capturadas imagens ou vídeos que servirão de base para o treinamento. Em seguida, realiza-se o pré-processamento das imagens, que consiste em ajustes como redimensionamento, remoção de ruídos e padronização de cores, com o intuito de melhorar a qualidade dos dados de entrada .

Após essa etapa, os dados são utilizados no treinamento do modelo, onde algoritmos de aprendizado de máquina, como as Redes Neurais Convolucionais (CNNs), aprendem a reconhecer padrões visuais. O modelo então passa por uma fase de validação, que verifica sua capacidade de generalização. Por fim, o sistema é aplicado para detecção e classificação de objetos, sendo capaz de identificar elementos visuais em tempo real, contribuindo para a tomada de decisões em diferentes contextos.

**Figura 2 - Fluxograma da Implementação de uma Visão Computacional**



Fonte: próprio autor, 2025.

Com base nessa tecnologia, o presente trabalho imerge na temática de VC propondo a criação de sistema baseado na tecnologia de CNN para realizar atividades de identificação, detecção, classificação e contagem de veículos, especificamente caminhões, carros, motos e ônibus.

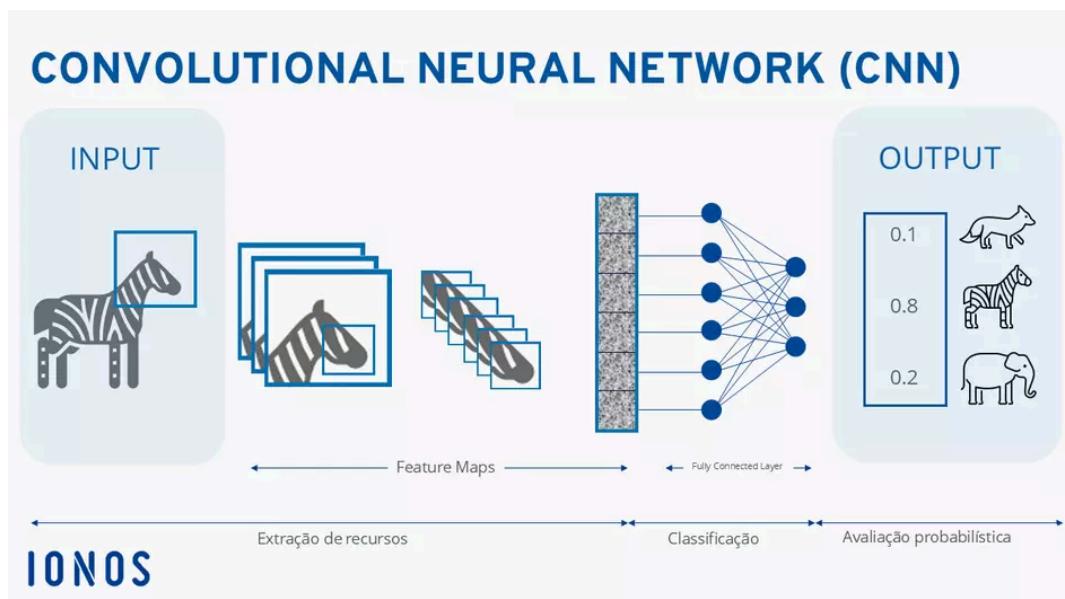
### **3.2 Redes Neurais Convolucionais - CNNs**

As redes neurais convolucionais (Convolutional neural network/CNNs) é uma tecnologia do ramo da Inteligência Artificial, oriunda do *Deep Learning* e *Machine Learning*, a qual foi desenvolvida através do processamento biológico de dados virtuais (VARGAS et. al, 2016) e é umas das estruturas mais comumente utilizadas para problemas envolvendo processamento de imagens.

As redes neurais convolucionais também podem ser classificadas como uma variação das redes de Perceptrons de Múltiplas Camadas, esses “perceptrons” basicamente são um modelo de rede neural análogo à estrutura neural humana (SANTOS, 2022, p. 26).

Estas redes utilizam dados tridimensionais para trabalhos de classificar imagens e reconhecer objetos, como demonstrado na Figura 3. Este processo atualmente se dá pela utilização de métodos escaláveis como a aplicação de álgebra linear com a multiplicação de matrizes na identificação de padrões estabelecidos nas imagens e objetos analisados. Anteriormente estes processos eram realizados de forma manual, como em “*Backpropagation Applied to Handwritten Zip Code Recognition*”, trabalho de Yann LeCun de 1989, no qual ele utilizou das redes neurais convolucionais para desenvolver um método eficiente de reconhecimento de números manuscritos.

**Figura 3** - Funcionamento de uma Rede Neural Convolucional



Fonte: IONOS editorial team, 2024. Disponível em:  
<https://www.ionos.com/pt-br/digitalguide/sites-de-internet/desenvolvimento-web/convolutional-neural-network/>

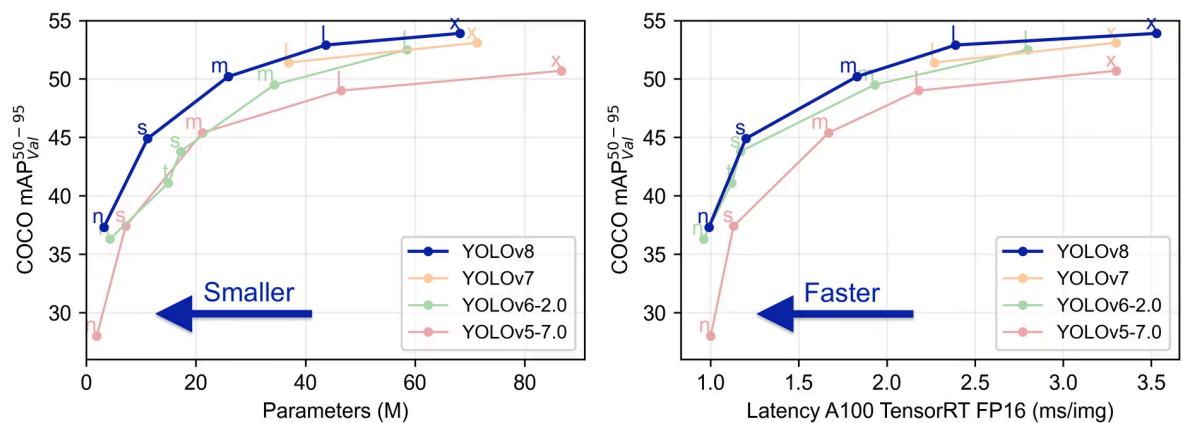
Portanto, é importante destacar que o uso de tecnologias aplicadas ao reconhecimento e detecção de objetos é possível graças à visão computacional que engloba técnicas como a CNN, que envolve identificar, localizar e classificar os objetos em imagens ou vídeos, previamente gravados ou em tempo real. Essa técnica inclusive passou a ser utilizada em diversas áreas, como saúde, esporte e setores da agropecuária brasileira. Ressalta-se também que uma das redes neurais convolucionais mais populares é o YOLO, que detém várias versões - YOLO, YOLO

v2, YOLO v3, etc., até a versão mais atual, a v12 (NELSON, 2025). A escolhida para esse trabalho de pesquisa foi a YOLO v8, visto que uma das suas principais características está amparada na sua abrangência de uso em diversas áreas além de possuir uma arquitetura mais leve e otimizada em relação às versões anteriores.

### 3.2.1 YOU ONLY LOOK ONCE (YOLO)

O modelo YOLO v8 da empresa Ultralytics foi uma versão lançada no dia 10 de janeiro de 2023 apontando como principal diferencial a velocidade e precisão com relação às versões anteriores (Figura 4). Na nova versão foram introduzidas novas funcionalidades e otimizações que resultaram num modelo ideal para atividades de detecção de objetos para diversas aplicações (ULTRALYTICS, 2024).

**Figura 4 - Comparativo entre as versões YOLO**



**Fonte:** Ultralytics Yolo V8, 2024. Disponível em <https://docs.ultralytics.com/pt/models/yolov8/#can-i-benchmark-yolov8-models-for-performance>

O modelo v8 possui quatro (4) características principais que resultaram na sua diferenciação dos modelos antigos. A arquitetura de *backbone* e *neck* foi otimizada para ter uma modelagem mais leve e rápida, podendo manter uma alta capacidade para extração de informações visuais.

O *backbone* e *neck*, tratam respectivamente da parte inicial e a intermediária que compõem o modelo, sendo inicialmente responsável o *backbone* por extrair da imagem de entrada as características visuais, gerando o chamado “mapas de características” (*features maps*), que representa os padrões como formas, objetos e bordas. Enquanto o *neck* capta as informações trazidas do *backbone* e as refina

para obter a combinação das características extraídas em diferentes escalas, identificando objetos em diferentes tamanhos e com isso melhorar a sua detecção de múltiplas escalas com menor custo computacional.

O *head* é a parte final e está inclusa na segunda característica principal do modelo, é o chamado Cabeça dividida sem âncoras (*Anchor-free head*), que consiste na tarefa de detectar e classificar a presença e o tipo de objetos capturados na imagem, com isso uma predição da localização da caixa delimitadora é gerada a partir da identificação do centro do objeto e seu tamanho, evitando sobreposições e erros.

Em modelos antigos, o uso de âncoras (*anchor boxes*) era comum, consistindo em trabalhar com boxes pré-definidos em tamanho e proporções, a fim de prever onde o objeto estaria disposto. Esse procedimento gerava muitos cálculos preditivos e ajustes a serem realizados, devido a esta maior taxa de assertividade, ocasionando em um modelo mais lento e com maior custo computacional também.

A terceira característica do *YOLO v8* é a otimização entre velocidade e precisão, garantindo equilíbrio entre desempenho e custo computacional. Essa melhoria torna o modelo adequado para aplicações em tempo real e diferentes cenários de detecção de objetos.

Por fim, destaca-se a variedade de modelos pré-treinados disponibilizados pela Ultralytics, organizados em três tipos: classificação (*c/s*), detecção (*det*) e segmentação (*seg*). Cada tipo oferece cinco tamanhos — *Nano* (n), *Small* (s), *Medium* (m), *Large* (l) e *Extra Large* (x) — permitindo ajustar o modelo conforme os recursos computacionais disponíveis e a necessidade de precisão. Para este trabalho, foi escolhido o *YOLO v8n (Nano)*, por apresentar menor custo computacional e maior velocidade, características essenciais para aplicações em tempo real com *hardware* limitado (ULTRALYTICS, 2024).

#### **4. TRABALHOS RELACIONADOS**

A questão do trânsito brasileiro tem sido discutida por diversos segmentos nos últimos anos, principalmente devido às significativas mudanças que vêm ocorrendo com o aumento populacional e a aquisição de veículos, que segundo o

DENATRAN em 2021 estimou-se cerca de um carro para cada 4,4 habitantes (MAUS et al, 2021, p. 49).

Reyna (2024), em sua dissertação de mestrado, discorre sobre a necessidade da implantação de ferramentas para detecção e análise de anomalias ocorridas no tráfego brasileiro, embasada na busca para evitar possíveis acidentes e congestionamentos. A resolução de tais problemáticas do trânsito brasileiro para a autora perpassa pela utilização de um gráfico bipartido e o algoritmo *Convex Hull* para a delimitação de áreas móveis em que se teria o rastreamento dos veículos, além do uso de estruturas de dados para identificar as possíveis anomalias. Para a autora, tais ferramentas proporcionariam “[...] aos sistemas de vigilância de tráfego, sistemas de visão computacional capazes de analisar e determinar, em tempo real, o fluxo veicular” (REYNA, 2024, p. 25).

A busca por tecnologias de reconhecimento e análise de objetos, como veículos, tem motivado a criação de inúmeras pesquisas, como é o caso de Maus, et al (2021), que buscou “conhecer a quantidade de veículos, suas rotas e gerar estatísticas (...) que colaboraram para o bom gerenciamento do trânsito, principalmente em grandes cidades” (2021, p. 49).

A intenção do autor era a elaboração de uma ferramenta capaz de tornar mais prática a contagem e classificação de veículos na cidade, pois os métodos usados eram análises manuais. Desta forma, o objetivo foi

(...) desenvolver uma aplicação web, utilizando bibliotecas e soluções de visão computacional, para o reconhecimento de veículos, onde será possível visualizar as informações do trânsito através de relatórios e dashboards, com os quais os gestores municipais poderão analisar a quantidade de veículos que trafegam nas rotas mapeadas e assim gerar soluções para melhor o fluxo de veículos da cidade. (MAUS et al, 2021, p. 49)

O projeto contou com o uso do seguinte conjunto ferramental: O modelo de rede neural *Darknet*, responsável por realizar a identificação de algum objeto, e que opera em sincronia com o modelo *YOLO v3*, que detecta objetos específicos para análise; a plataforma *CUDA* (*Compute Unified Device Architecture*), que processa os *GPUs* (*Graphics Processing Units*) na realização de detecção dos objetos. Maus, et al. (2021)

O sistema proposto por Maus, et al. (2021) também desenvolveu uma interface *web* para poder analisar os dados de veículos e rotas por meio de vídeos, resultando em um sistema que classificava a partir do YOLO v3 a quantidade e categoria dos veículos, além da velocidade média dos mesmos e a trajetória num cruzamento semafórico na cidade de Joinville (SC), cidade com classificada como de grande porte.

Outra pesquisa que buscou analisar a realidade do trânsito brasileiro foi a de Barbado, et al (2022). Dentro do contexto da cidade de Foz de Iguaçu no Paraná, de médio porte, os autores se debruçaram no problema de volume de veículos e tempo de deslocamento que existia no trânsito da cidade. O trabalho propôs desenvolver uma aplicação para “para categorização dos veículos, sua contagem e a determinação dos respectivos tempos de trânsito” (BARBADO et al, 2022, p. 2).

Para tal, foi

(...) considerado como entrada os fluxos de imagens provenientes de câmeras de monitoramento já instaladas nas áreas de interesse, e a utilização de técnicas e algoritmos de processamento de imagens, reconhecimento de padrões capazes de identificar e rastrear veículos em cruzamentos viários, urbanos, aplicando os recursos e bibliotecas disponíveis na linguagem de programação Python. (BARBADO et al, 2022, p. 2)

A aplicação de Barbado, et al (2022) concentrou o reconhecimento de veículos em apenas 3 classes (carros, motos e caminhões) e utilizou a CNN YOLO V4. Além disso foram utilizados também, os seguintes materiais e métodos: a biblioteca *OpenCV*, *NumPy*, *MatPlotLib*, *YOLO: Real-time Object Detection* e *CUDA Toolkit*. De acordo com os autores, “foi possível atribuir identificadores para os veículos, ao entrarem nas regiões de interesse delimitadas nas imagens, e acompanhar a sua movimentação, registrando os momentos de entrada e saída destas regiões de interesse” (BARBADOS et al, 2022, p. 6), o que demonstra a importância de trabalhos voltados para a detecção e análise de veículos no contexto dos problemas relacionados ao trânsito brasileiro.

Todos os trabalhos discutidos nesta seção trouxeram diferentes sistemas e aplicações que buscaram propor medidas de resolução para a mobilidade urbana brasileira em cidades de médio e grande portes. Ao utilizarem diferentes interfaces *web* para análise de veículos e um modelo de IA, as pesquisas demonstram como

tal instrumento pode ser utilizado no tráfego municipal, em prol de detecção e monitoramento do trânsito a nível nacional. A proposta de investir em tecnologias semelhantes para cidades de pequeno porte, até então pouco analisadas, será investigada nesta pesquisa para aprofundar o debate proposto.

## 5. PROPOSTA DE TECNOLOGIA

A crescente busca por soluções eficazes para a mobilidade urbana estabelece uma relação diretamente proporcional com o aumento das problemáticas relacionadas ao tráfego de veículos, ocasionando numa busca por tecnologias capazes de melhorar o monitoramento e gestão dos trânsitos nas cidades, principalmente nos grandes centros urbanos. No entanto, cidades de pequeno porte, como o caso de Pojuca-Ba, também têm enfrentado o aumento da frota de veículos e dentre os desafios que estão relacionados a este problema, tem-se a carência de infraestrutura tecnológica adequada e a ausência de uma fonte de dados que embase as tomadas de decisões pelo órgão responsável pela gestão do trânsito na cidade.

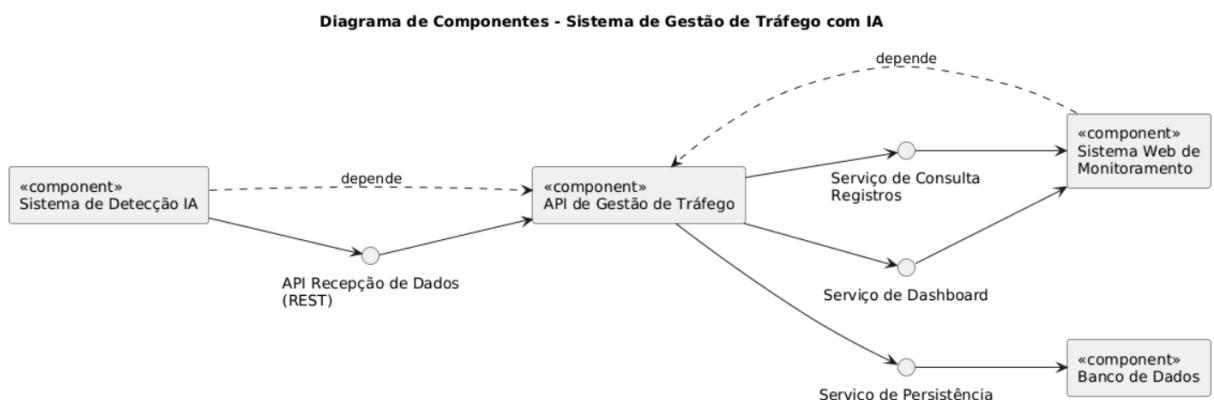
Diante desse fato, este trabalho visou desenvolver um sistema inteligente de análise de tráfego, com o objetivo de propor uma solução de baixo custo capaz de realizar a detecção, classificação e contagem de veículos (incluindo motocicletas), gerando registros em tempo real através de análise de vídeo.

Este sistema busca se destacar pela utilização de tecnologias modernas, como as Redes Neurais Convolucionais (CNNs), *frameworks web* e bancos relacionais, além de contar com uma arquitetura facilmente replicável para demais municípios de características semelhantes por ser modular e escalável. Ao realizar a integração simultânea da detecção de veículos com a visualização de dados em um *dashboard* e tabelas para exportação de relatórios, o sistema busca não somente oferecer uma plataforma de monitoramento do tráfego municipal, mas também ser uma fonte de informações confiável e capaz de basear a tomada de decisões estratégicas para melhor gerir o trânsito da cidade.

## 5.1 Descrição da solução proposta

O sistema proposto foi denominado *Traffic Viewer*, e conforme já relatado envolveu o desenvolvimento do modelo IA, da API, do banco de dados da Interface Web (Figura 5). Cada etapa foi trivial para compor o sistema, sendo o primeiro na linha de desenvolvimento e de maior importância, o modelo IA, seguido da integração com a API e a Interface Web.

**Figura 5** - Estrutura de desenvolvimento do Sistema *Traffic Viewer*



Fonte: próprio autor, 2025.

### 5.1.1 Etapa da IA

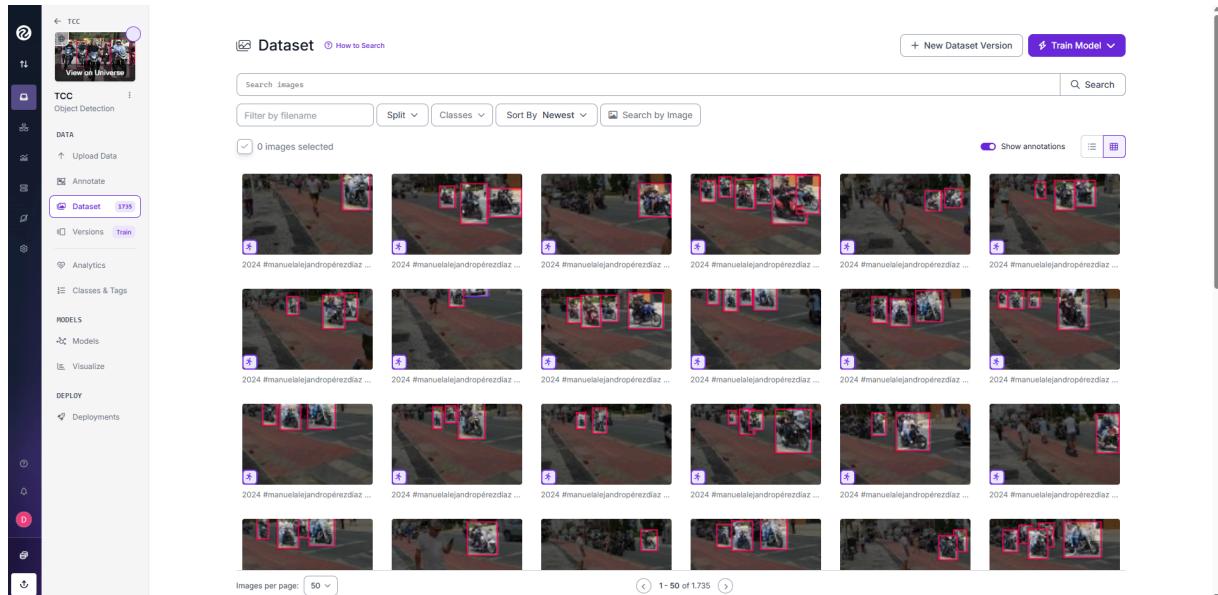
A Etapa da IA constituiu o núcleo do sistema, responsável por processar vídeos e extrair dados para o monitoramento do tráfego. Nela foram definidos o ambiente isolado para execução e o modelo *YOLO v8n (Nano)*, escolhido por sua leveza e eficiência na detecção, classificação e contagem de veículos, atendendo às demandas de cidades de pequeno porte com equilíbrio entre precisão e baixo custo computacional.

Nesta etapa foram desenvolvidos os códigos com base no uso da biblioteca *OpenCV* que fornece “um conjunto abrangente de ferramentas e algoritmos especificamente concebidos para tarefas de visão computacional (CV)” (ULTRALYTICS, 2025) para captar as seguintes informações dos vídeos analisados pela IA: classificar, contar, detectar, identificar direção de curso, o deslocamento do veículo no trajeto e a velocidade do veículo.

Como resposta à necessidade de uma tecnologia de fácil acesso para atender à necessidade de uma solução acessível e de baixo custo, e conforme já relatado, o modelo *YOLO v8n (Nano)* é uma versão mais leve da família *YOLO v8*. Esse modelo oferece alta velocidade e menor demanda computacional, mantendo precisão adequada para aplicações em tempo real, características essenciais para sistemas inteligentes voltados ao monitoramento de tráfego em cidades de pequeno porte.

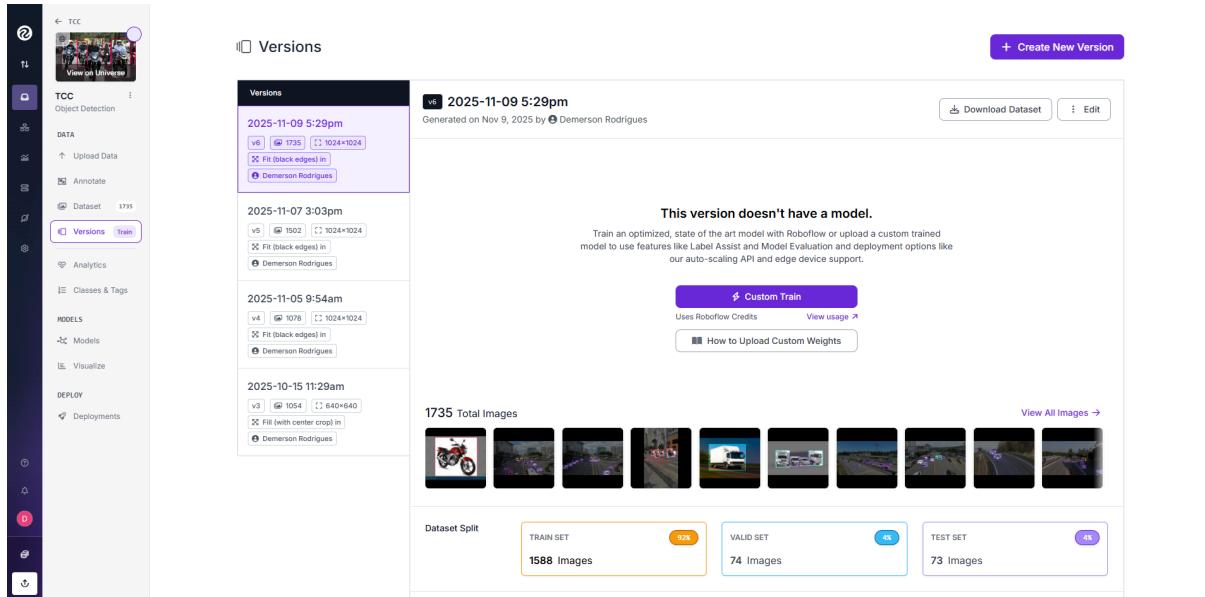
A aquisição e preparação dos dados iniciou com a criação de um *dataset* na plataforma *Roboflow* (Figura 6), site de uso livre para criação e compartilhamento de banco de imagens, em que estas podem ser categorizadas em classes e identificadas por uma *box*, processo chamado de “anotação”. Com a anotação realizada, o *dataset* foi então concluído (Figura 7), ficando disponível para ser baixado para uso em treinamento no Google *Colab*, ou utilizar a *api\_key*.

**Figura 6 - Visualização do dataset no Roboflow**



Fonte: próprio autor, 2025. Disponível em: <https://app.roboflow.com/tcc-eoyqb/tcc-dzopp>

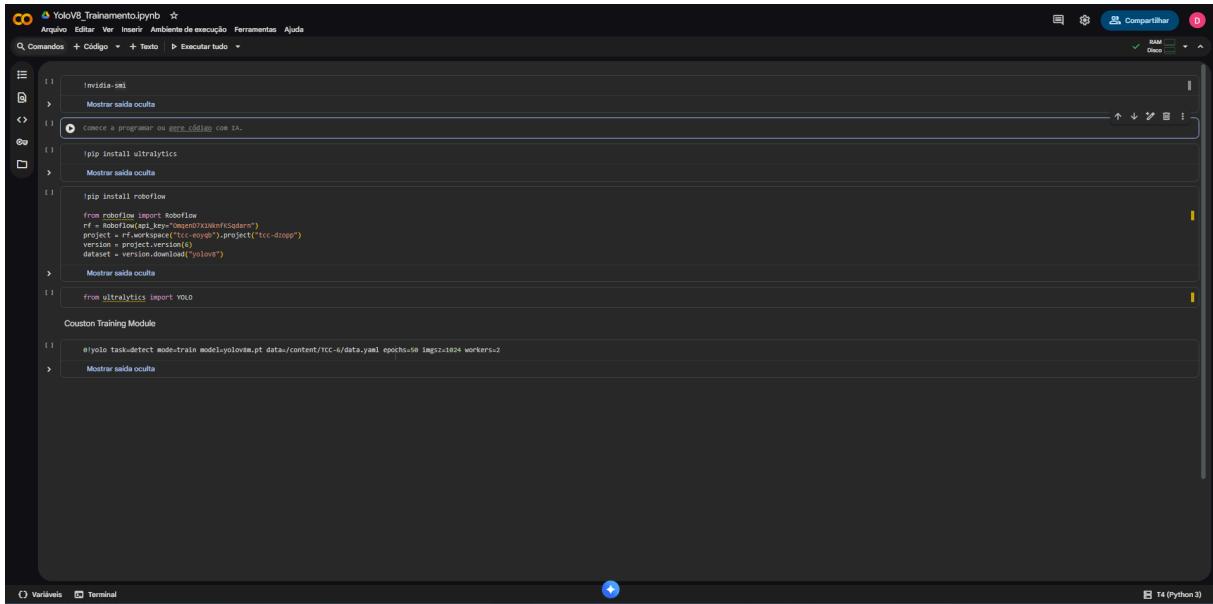
**Figura 7 - Workspace de Versões**



Fonte: próprio autor, 2025. Disponível em: <https://app.roboflow.com/tcc-eoyqb/tcc-dzopp/6>

Com o *dataset* criado e organizado no *Google Colab*, inicialmente foi realizada a conexão com um modelo de *GPU* gratuito para melhorar o desempenho da atividade de treinamento do modelo. Em seguida foram executados os comandos em sequência: “*!nvidia-smi*”, permite o monitoramento da *GPU* utilizada no ambiente; “*!pip install ultralytics*” instala o pacote necessário para trabalhar com modelos *YOLO*; “*from ultralytics import YOLO*” importa a base dos modelos *YOLO* para uso; “*!pip install roboflow*” instala os parâmetros vindo do dataset do roboflow; e “*!yolo task=detect mode=train model=yolov8n.pt data=archive.yaml epochs=50 imgsz=1024 workers=2*” é o treinamento customizado com os parâmetros necessários para gerar o modelo esperado (Figura 8).

**Figura 8 - Ambiente no Google Colab para treinamento do modelo YOLO**



```

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda
Comando + Código + Texto Executar tudo
Início - srt
Mostrar saída oculta
Comece a programar ou pega código com IA.
!pip install ultralytics
Mostrar saída oculta
!pip install roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="XXXXXXXXXXXXXXXXXXXXXX")
project = rf.workspace("yolov8").project("tcc-dspp")
version = project.version(6)
dataset = version.download("yolov8")
Mostrar saída oculta
from ultralytics import YOLO
Custom Training Module
@yo lo task: detect model: train model: yolov8.pt data: /content/YCC-4/data.yaml epochs: 50 images: 1024 workers: 2
Mostrar saída oculta

```

Fonte: Google Colab, 2025. Disponível em: <https://colab.research.google.com/>

### 5.1.2 Etapa API

Para desenvolvimento da API, foi utilizado o *framework NestJS* do *NodeJS*, que se destaca pelos recursos de arquitetura modular, linguagem *typescript* e suporte à microsserviços para agilizar o desenvolvimento e manutenção do sistema.

A API foi construída também contando com a utilização das ferramentas do *Swagger*, que trata justamente de “um conjunto de ferramentas de código aberto desenvolvido com base na especificação OpenAPI que pode ajudar você a projetar, construir, documentar e consumir APIs REST” (SMARTBEAR, 2025), sendo a *OpenAPI* um formato de descrição de *API* para *API REST*.

Os *endpoints* foram definidos de acordo com as funcionalidades principais do sistema, como a listagem e o cadastro de dados, além de permitir a filtragem de registros e a exportação de relatórios no formato *PDF*, conforme na Figura 9. Também foi criado um endpoint específico para receber os dados processados pela inteligência artificial, que realiza a detecção dos veículos.

Essa organização contribuiu para que a API funcionasse de forma eficiente, atendendo às necessidades do sistema e permitindo que o *frontend* e os demais serviços se comuniquem da maneira mais clara e eficiente possível.

**Figura 9 - Documentação Swagger Traffic Viewer API**

Fonte: próprio autor, 2025.

Para complementar a arquitetura da API, foi integrado o ORM Prisma ao banco PostgreSQL, simplificando a modelagem e garantindo comunicação eficiente com os serviços, onde a combinação das tecnologias NestJS, Swagger e Prisma proporcionou um *backend* robusto, escalável e de fácil manutenção, atendendo aos requisitos do projeto e assegurando integração fluida entre a inteligência artificial, o banco de dados e a interface *web*.

### 5.1.3 Etapa Banco de dados

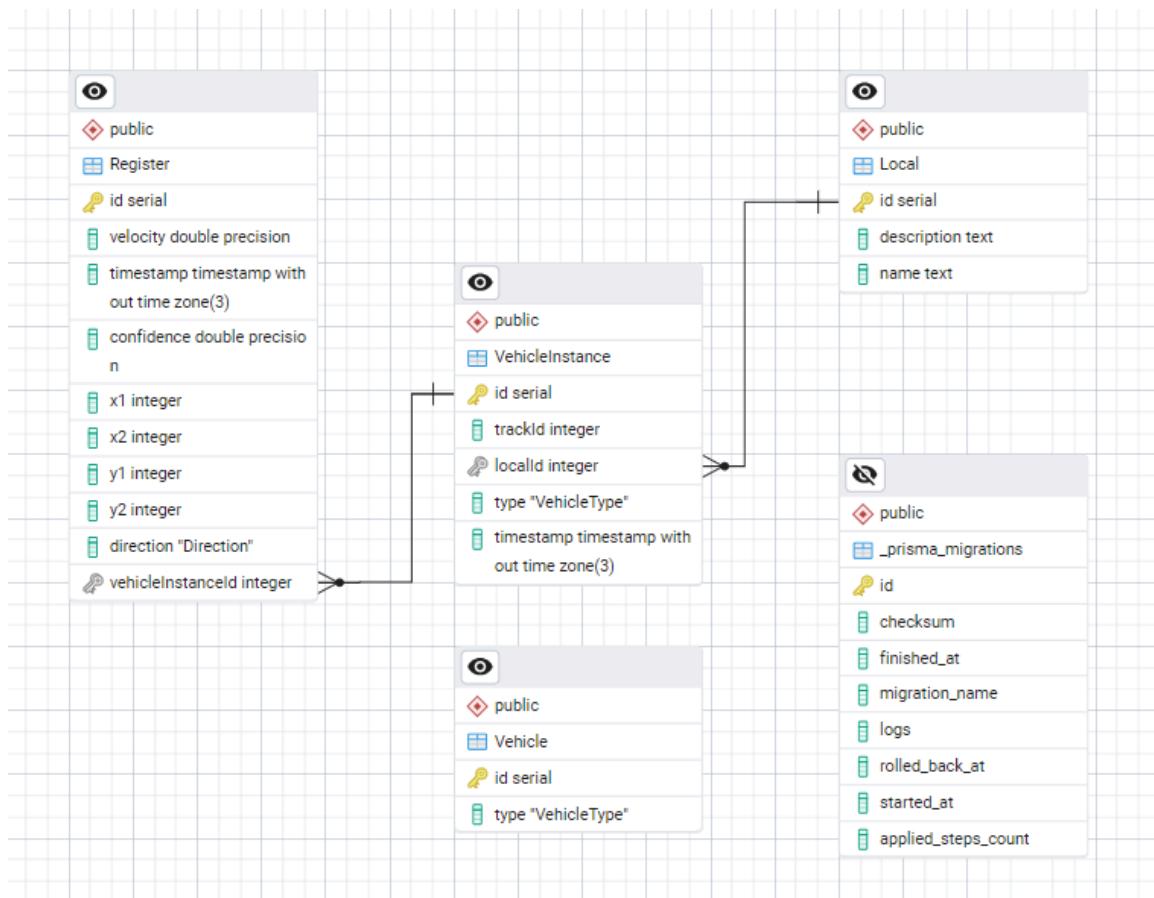
O banco de dados do sistema foi implementado com a utilização do banco relacional PostgreSQL, que foi escolhido baseado na sua capacidade robusta, confiabilidade e conformidade com os princípios ACID (*Atomity, Consistency, Isolation, Durability*), que são fundamentais para garantir a integridade dos dados e sua consistência em aplicações críticas (POSTGRESQL, 2025).

A modelagem do banco (Figura 9) se baseou nas três entidades principais: Veículo, Registro e Localização, para permitir a organização de forma eficiente dos dados quando extraídos pela *IA*, facilitando assim a aplicação das filtragens (data, tipo de veículo, local e horário).

Como tratado no tópico da etapa da API, os dados podem ser manipulados por meio do ORM Prisma, que trabalha como uma camada que abstrai o banco e o *backend*. A integração entre as partes, especificamente entre o Prisma com *TypeScript* de forma nativa, oferece uma API declarativa para definir esquemas e execução de *queries*, além de gerar clientes de acesso com tipagem estática para reduzir erros e proporcionar a melhora da produtividade (PRISMA, 2025).

O banco de dados tem como responsabilidade sustentar a camada de persistência do sistema, armazenando os dados que são processados pela *IA* e deixando-os à pronta entrega para visualização no *frontend* por meio das devidas requisições.

**Figura 10** - Modelagem do banco de dados



Fonte: próprio autor, 2025.

### 5.1.4 Etapa Interface Web

A parte visual do sistema, ou seja, o *frontend*, foi desenvolvido usando o *framework React*, *TypeScript* e a biblioteca de componentes *Material UI* (MUI). Esta união de tecnologias foi escolhida baseada na praticidade de desenvolvimento, devido a sua modernidade e por ser escalável e segura para o desenvolvimento de interfaces interativas, sendo para o caso deste sistema proposto, o foco está na visualização de dados de tráfego urbano.

O *framework React* trata de uma biblioteca baseada em componentes, que é amplamente utilizada no mercado para criar interfaces de usuário dinâmicas. Sua arquitetura favorece o reuso para os códigos além de separar as responsabilidades, contribuindo para a manutenção e progressão do sistema (REACT, 2025). Ao trabalhar de forma integrada com a linguagem *TypeScript*, o projeto do sistema se torna mais suscetível à redução de erros e melhora a sua legibilidade e previsibilidade do código já que conta com a definição explícita de tipagens (TYPESCRIPT, 2025).

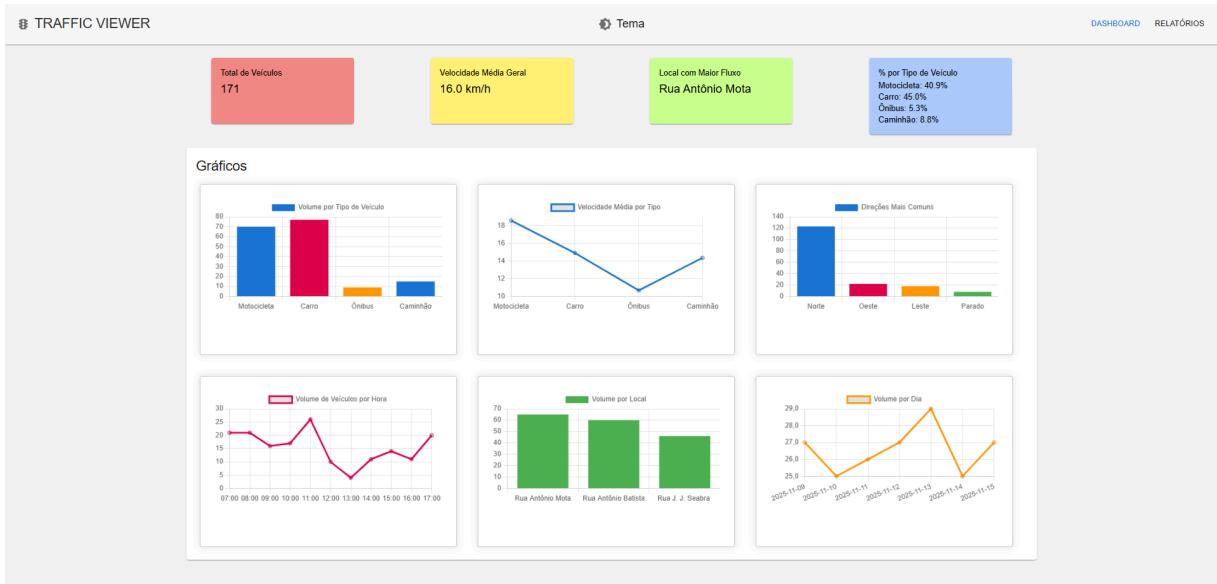
A *lib Material UI* foi escolhida como uma solução ágil para a construção da interface visual, por contar com os princípios *Material Design* de forma nativa ao oferecer uma vasta coleção de componentes já elaborados e responsivos de forma muito acessível. Além disso, o MUI possui suporte nativo para a linguagem *TypeScript*, garantindo maior consistência e segurança no desenvolvimento (MUI, 2025).

A interface do *Traffic Viewer* foi criada em módulos e organizada em rotas para separar os componentes, páginas e serviços. A interface visual está organizada contando com duas telas que estão integradas ao *backend* trazendo os dados processados em tempo real, são a tela de *Dashboard* e a tela de *Registros* e nelas estão previstas:

- Tela de *Dashboard*: Nesta tela (Figura 11) estão apresentados os gráficos que são gerados por meio da biblioteca *Chart.js*, que está integrada aos componentes do MUI, trazendo em exibição informações como o volume de veículos por hora, os tipos de veículos e registros de suas localizações. Ao trabalhar com uma tela de *dashboard* com

gráficos objetivou-se trazer uma perspectiva mais simplificada e fácil para interpretação rápida dos dados, do tráfego a ser analisado.

**Figura 11 - Tela de Dashboard**



Fonte: próprio autor, 2025.

- **Tela de Registros:** Esta tela (Figura 12 e 13) conta com uma tabela apresentando dados brutos que foram coletados do modelo IA operando na visualização do trânsito. A tabela utilizou o componente *DataGrid* do MUI, por oferecer recursos de paginação, ordenação além de filtros dinâmicos nativos da tabela. A escolha por organizar os dados em uma tabela foi pela necessidade de detalhar precisamente os registros de modo individual, além da capacidade de exportação destes dados no formato PDF para relatórios de modo geral ou filtrados. A interface visual do *Traffic Viewer* é a camada essencial do sistema por ser aquilo que capacita o usuário a tomar conhecimento de todos os dados gerados que fomentam na capacidade de tomar decisões para o tráfego urbano.

**Figura 12 - Tela de Registros**

Fonte: próprio autor, 2025.

**Figura 13 - Tela de Registros com aplicação de filtro**

Fonte: próprio autor, 2025.

#### 5.1.4.1 Funcionamento do Sistema *Traffic Viewer*

O funcionamento do sistema envolve as etapas de captura de dados, processamento com IA, armazenamento em banco de dados e visualização em interface web. Cada etapa dessas operam se comunicando entre si, garantindo um fluxo contínuo de dados e geração das informações para análise do tráfego urbano.

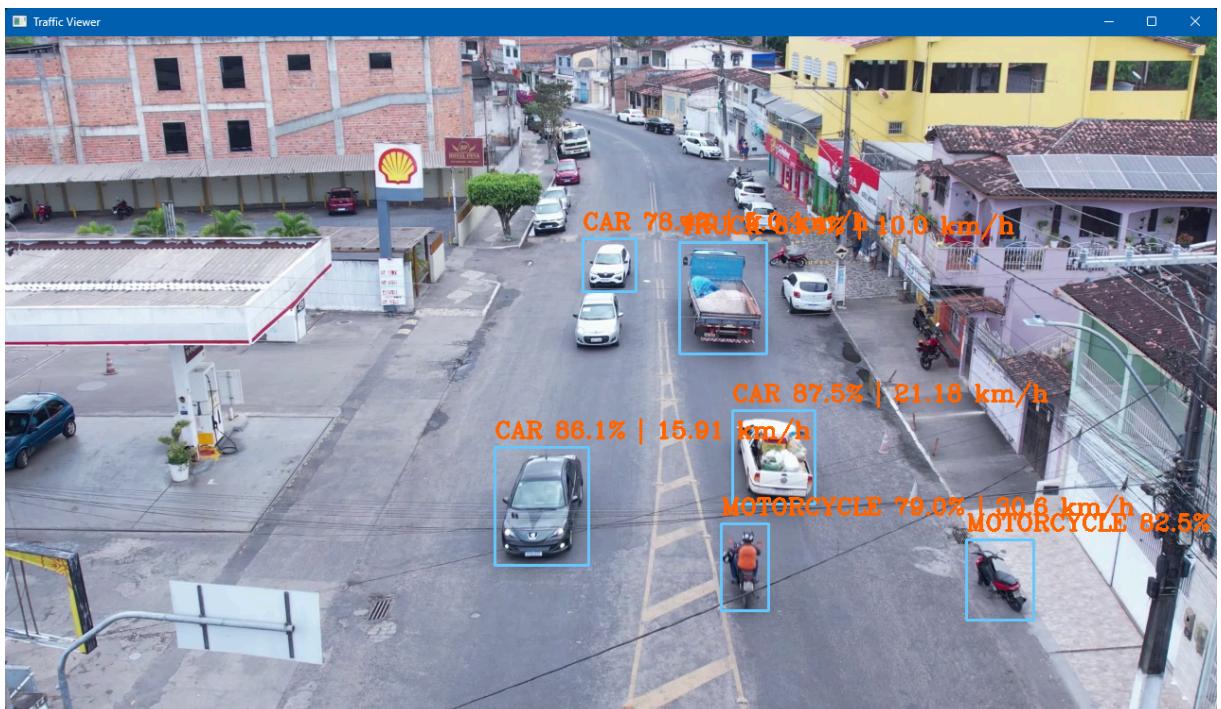
#### 5.1.4.2 Captura e Envio de Dados

Esta etapa inicia a partir da obtenção de vídeos por meios convencionais ou até mesmo por produção própria via câmeras, que podem ser armazenados e utilizados posteriormente no módulo de visão computacional. Desse modo, a submissão manual destes vídeos é realizada para dar início à produção dos dados.

#### 5.1.4.3 Processamento com Visão Computacional

Os arquivos de vídeo uma vez adquiridos são processados no modelo de detecção criado baseado no YOLO v8 oriundo do treinamento via *Gooble Colab*. O modelo é executado no ambiente virtual isolado (*venv*), com uso da *lib OpenCV* para ler e extrair as informações, por exemplo: o tipo do veículo detectado, quantidade de veículos por classe, velocidade estimada, conforme ilustrado na Figura 14.

**Figura 14** - Tela do Processamento com Visão computacional



Fonte: próprio autor, 2025.

Todos os dados captados nesta etapa, são estruturados para formato *JSON* e acabam sendo enviados por meio de requisição *HTTP* para a API.

#### **5.1.4.4 Armazenamento e organização dos dados**

Nesta etapa, o *backend* recebe todos os dados processados e armazena-os no banco de dados do sistema e pode ser gerenciado via *PgAdmin 4*. As principais entidades do sistema são: Veículos, Registros e Localização, como previstas na seção 5.1.3 Banco de Dados.

A modelagem foi elaborada contando com a capacidade de permitir a realização de consultas otimizadas, tendo suporte para filtros dinâmicos por data, tipo de veículo, local e horário.

#### **5.1.4.5 Visualização e interação com os Dados**

A camada de visualização consiste na apresentação dos dados de modo interativo após o processamento do modelo da IA.

Nesta etapa o usuário consegue visualizar de modo prático e rápido os dados por meio da tela inicial de *Dashboard* do sistema permitindo identificar padrões e tomar decisões com base nas evidências visuais. Enquanto, na tela de Registros o usuário encontra informações mais detalhadas que podem ser exploradas para trazer uma análise mais específica dos dados.

É importante destacar que a estrutura da API foi organizada para separar as responsabilidades entre os módulos. Cada parte do sistema, como os dados de veículos, registros e localizações, foi dividida em seus próprios módulos a fim de facilitar tanto o desenvolvimento quanto a manutenção.

Por fim, as informações registradas podem ser exportadas por meio de relatório PDF que apresenta inclusive algumas recomendações baseadas na legislação de trânsito brasileira para ajudar os gestores do trânsito na tomada de decisões.

## 6.0 RESULTADOS E DISCUSSÕES

### 6.1 Estudo de caso

Com o objetivo de validar a eficácia do sistema proposto para monitoramento do tráfego em cidades de pequeno porte, foi conduzido um estudo de caso na cidade de Pojuca-BA. A validação buscou comprovar se a ferramenta baseada em redes neurais convolucionais seria capaz de detectar, classificar e contabilizar veículos em vídeos curtos, fornecendo dados confiáveis sobre fluxo, velocidade média e distribuição por tipo de veículo. Além disso, pretendeu avaliar se os resultados obtidos poderiam subsidiar decisões estratégicas da autoridade municipal de trânsito, atendendo à necessidade de soluções acessíveis e de baixo custo para gestão do tráfego urbano local.

#### 6.1.1 Estratégia de validação

A estratégia de validação foi estabelecida com o intuito de alcançar uma representatividade significativa do tráfego urbano real na cidade de Pojuca-BA, contemplando as localizações com boa movimentação e variedade de veículos durante diferentes períodos do dia.

Os locais foram definidos com base na premissa de ruas que apresentassem boa amostragem do trânsito em Pojuca, o que resultou na decisão por ruas estabelecidas em regiões de comércio da cidade. Dessa forma, foram escolhidas as seguintes localidades: Rua Antônio Batista (próximo à feira municipal da cidade), Rua Antônio Mota (sinaleira com cruzamento no bairro Cruzeiro) e Rua J. J. Seabra (comércio da cidade).

O processo de preparo dos dados consistiu em traçar um planejamento capaz de atender uma diversidade de cenários e amostragem do comportamento do trânsito em diferentes locais da cidade. Para isso, foram coletados um total de oitenta e quatro (84) vídeos, produzidos durante uma (1) semana em diferentes turnos do dia. Assim ficando organizado da seguinte forma: um total de quatro (4) vídeos ao dia, dois (2) por cada turno do dia, para cada um dos três (3) locais de análise, totalizando doze (12) vídeos para cada dia da semana.

A duração dos vídeos ficou decidida que deveria ser de um (1) minuto, para proporcionar uma boa representação para análise, assim como para a devida aquisição de registros a serem exibidos no *Traffic Viewer*.

É importante destacar que os vídeos foram gravados em resolução 720p por drones posicionados em altura equivalente a postes de monitoramento urbano, simulando ângulos reais de câmeras municipais. A orientação da câmera seguiu uma perspectiva diagonal orientada ao solo, garantindo ampla cobertura da via. Durante o processamento, adotou-se a estratégia de analisar um frame a cada três, equilibrando desempenho e custo computacional.

### 6.1.2 Execução do Teste

Após a preparação do dataset, os testes foram conduzidos de forma progressiva para avaliar a evolução do modelo *YOLO v8n*. Inicialmente, o modelo denominado *detect-vehicle-v1* foi treinado com 480 imagens, apresentando dificuldades na detecção das classes caminhão e ônibus, devido à variação de perspectiva e dimensões desses veículos.

Para melhorar a assertividade, foram realizadas duas intervenções: ampliação do dataset em 200 imagens adicionais e ajuste da dimensão padrão de entrada dessas imagens de  $640 \times 640$  para  $1024 \times 1024$  pixels. Assim foram surgindo novas versões do *detect-vehicle* e no modelo *v2*, observou-se melhora nas detecções, mas os resultados mais consistentes ocorreram no modelo *v3*, após a alteração da dimensão.

A partir dessa análise, novas adições de imagens foram incorporadas ao dataset, resultando em versões *v4* e *v5* do modelo *detect-vehicle*, que reforçaram a tendência de melhoria. Finalmente, a versão *detect-vehicles-v6*, com totalização de 1735 imagens, consolidou o melhor desempenho, validando a capacidade do sistema para detecção e classificação das quatro classes definidas (carros, motos, caminhões e ônibus).

### 6.1.3 Resultados Obtidos

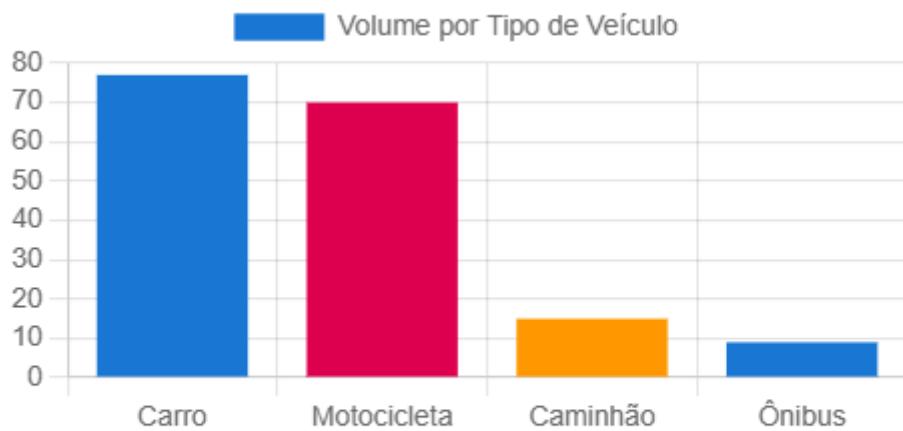
Os resultados obtidos no *Traffic Viewer* após o processamento dos vídeos refletem o comportamento do tráfego nos locais analisados na cidade de Pojuca, conforme Apêndice 1.

Quanto à classificação por tipo, os percentuais foram: caminhões (8,8%), carros (45%), motocicletas (40,9%) e ônibus (5,3%). Esses dados indicam predominância de veículos leves, especialmente automóveis e motocicletas,

característica comum em cidades de pequeno porte, que inclusive, segundo estudo de frota do DETRAN, BA desde 2022 até 2024 apresentou um crescimento relativo no número destes veículos para o município de Pojuca (Anexo 1).

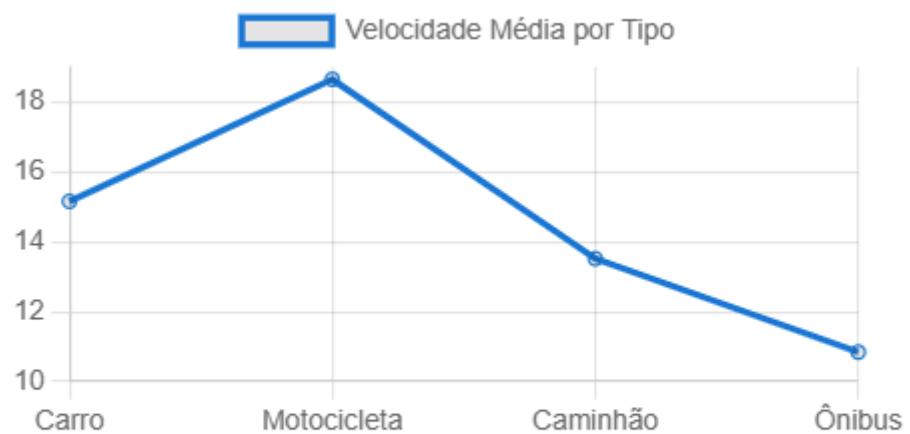
Para melhor visualização, os gráficos gerados pelo sistema apresentam diferentes perspectivas do tráfego analisado: volume por tipo de veículo (*Figura 15*), velocidade média por tipo (*Figura 16*), direções de curso mais comuns (*Figura 17*), volume por hora (*Figura 18*), volume por local (*Figura 19*) e volume por dia (*Figura 20*). Cada gráfico contribui para uma análise detalhada do fluxo, permitindo identificar padrões e horários de maior movimentação.

**Figura 15** - Volume por tipo de veículo



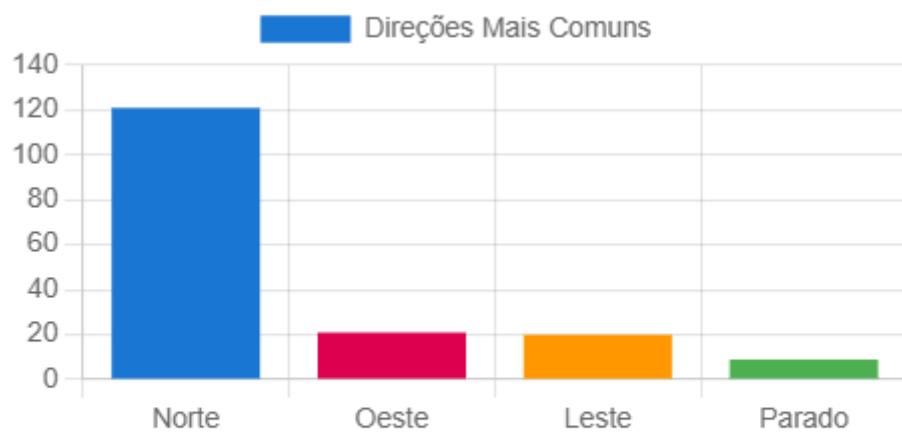
Fonte: próprio autor, 2025.

**Figura 16** - Velocidade média por tipo de veículo



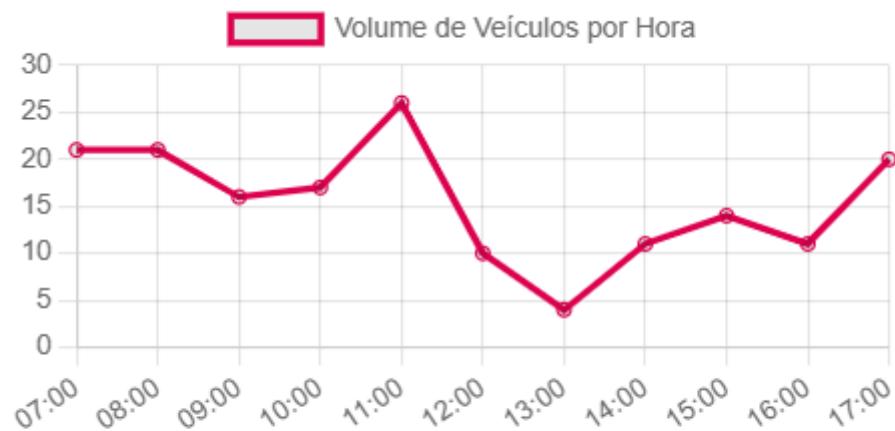
Fonte: próprio autor, 2025.

**Figura 17** - Direções de curso mais comuns



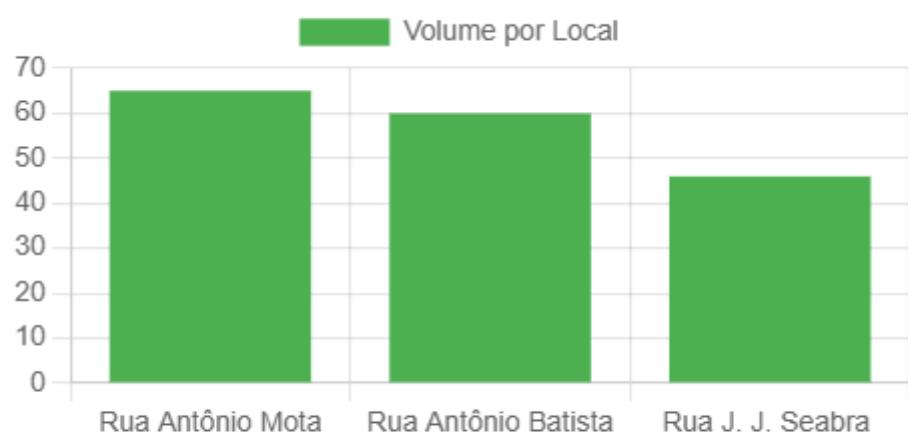
Fonte: próprio autor, 2025.

**Figura 18** - Volume de veículos por hora



Fonte: próprio autor, 2025.

**Figura 19** - Volume por local



Fonte: próprio autor, 2025.

**Figura 20** - Volume de veículos por dia



Fonte: próprio autor, 2025.

De modo geral, pode-se observar a partir dos dados na plataforma que o contingente total de veículos únicos foi de 171, registrando uma velocidade média geral de 16 km/h, tendo como a rua com maior fluxo predominante a Rua Antônio Mota, dentro do período datado de 09/11/2025 à 15/11/2025.

Especificamente, a Rua Antônio Mota, como consta no Apêndice 2, apresentou um total de 65 veículos durante o período de 09/11 a 15/11/2025, com velocidade média geral de 17,87 km/h, indicando um fluxo relativamente estável para uma via com semáforos e de encruzilhada. Os horários de maior movimento ocorreram às 09h (16 veículos) e às 11h (26 veículos), refletindo deslocamentos típicos do início da manhã e próximo ao meio-dia. A predominância da direção Norte (48 registros) sugere que a via apresenta uma maior movimentação orientada para o interior da cidade, especificamente região do centro da cidade. Entre os tipos de veículos, destacaram-se carros (28) e motos (27), com velocidades médias de 16,2 km/h e 20,4 km/h, respectivamente, enquanto caminhões e ônibus tiveram menor participação, ambos com apenas 5 registros cada. Esse padrão indica que, apesar da presença de controle semafórico, a via mantém boa fluidez, mas requer atenção nos horários de pico para evitar retenções.

Por sua vez, a Rua Antônio Batista, Apêndice 3, registrou 60 veículos no período analisado, com velocidade média geral de 15,85 km/h, inferior à observada na Rua Antônio Mota, sugerindo um singelo aumento na lentidão do trânsito, possivelmente devido à presença da rotatória localizada na região. Os horários de maior movimento ocorreram às 08h (21 veículos) e às 10h (17 veículos), coincidindo com os períodos de maior atividade comercial na região da feira municipal e mercados próximos. A predominância da direção Norte (43 registros) confirma a função da via como corredor de acesso às áreas centrais do município, enquanto os acessos Leste (8) e Oeste (6) complementam a distribuição do tráfego para ruas adjacentes. Entre os tipos de veículos, destacaram-se carros (28) e motos (23), seguidos por caminhões (7) e ônibus (2), evidenciando a presença de transporte de carga e coletivo na região. A velocidade média dos veículos foi ligeiramente inferior à dos demais em relação à Rua Antônio Mota, sugerindo retenções pontuais, como já induzido, potencialmente associadas à rotatória próxima e à movimentação de pedestres. Esse padrão reforça a necessidade de atenção para a fluidez no tráfego, sinalização na via e movimentação dos pedestres, especialmente nos horários de pico.

Já a Rua J.J. Seabra, no Apêndice 4, apresentou um cenário distinto, com 46 veículos registrados e velocidade média geral de 12,2 km/h, significativamente inferior às demais vias, indicando realmente maior lentidão no fluxo. Os horários de pico ocorreram às 07h (21 veículos) e às 17h (11 veículos), períodos que coincidem com deslocamentos para trabalho e retorno, além da intensa movimentação comercial típica do centro da cidade. A predominância da direção Norte (32 registros) confirma a função estratégica da via como eixo de mobilidade urbana, complementada pelos acessos Leste (9) e Oeste (6), principalmente pela orientação do fluxo (mão única) nesta região do centro da cidade. Entre os tipos de veículos, os carros (23) e motos (19) foram maioria, mas a presença de ônibus (2) e caminhões (2) indica circulação mista, associada ao transporte coletivo e abastecimento comercial também necessário na região, devido à presença de lojas e mercados próximos. A baixa velocidade média, especialmente para ônibus (7,9 km/h) e caminhões (9,1 km/h), reforça a necessidade de medidas para reduzir retenções e melhorar a fluidez, visto que são veículos longos e robustos com maior capacidade

de travar vias em função da movimentação lenta ou pela atividade de carga e descarga nos pontos comerciais da região.

A comparação estabelecida entre as três vias evidencia diferenças significativas na dinâmica do tráfego. Enquanto nas ruas Antônio Mota e Antônio Batista apresentaram volumes semelhantes (65 e 60 veículos) e velocidades médias próximas (17,87 km/h e 15,85 km/h), a Rua J.J. Seabra registrou menor quantidade de veículos (46) e velocidade média inferior (12,2 km/h), refletindo que nas ruas Antônio Mota e Batista houve maior concentração de fluxo, proveniente de maior concentração comercial, além de rotas mais difundidas. Os horários de pico também variaram: nas duas primeiras ruas, destacaram-se os períodos matutinos e intermediários, enquanto na J.J. Seabra os picos ocorreram no início da manhã e final da tarde. Esses resultados indicam que fatores como função da via, presença de comércio e infraestrutura viária influenciam diretamente o comportamento do tráfego, reforçando a necessidade de estratégias específicas para cada contexto urbano.

#### **6.1.4 Discussão dos Resultados**

Os resultados obtidos demonstraram que o sistema atendeu aos objetivos propostos para o trabalho, realizando de forma satisfatória a detecção, classificação e contagem dos veículos, bem como a geração de dados relevantes sobre o fluxo dos veículos e demais informações, como a velocidade média dos automóveis. A predominância de carros e motocicletas, em contraste com a baixa presença de caminhões e ônibus, confirma um padrão esperado para cidades de pequeno porte.

Apesar do desempenho positivo, algumas limitações foram identificadas. Entre elas, destacam-se: dependência de boas condições de iluminação para garantir a precisão nas detecções; a restrição do ângulo das câmeras, estabilizando-se fixamente, pois pode resultar em comprometimento na identificação de veículos em situações de sobreposição; e a necessidade de ampliar o *dataset* com imagens reais da cidade para aumentar a robustez do modelo.

A execução do sistema ocorreu em um notebook Samsung, equipado com processador Intel i5 de 11<sup>a</sup> geração, 256 GB SSD NVMe e GPU integrada Intel Iris Xe, demonstrando que a solução é viável em equipamentos de baixo custo. Apesar das limitações de *hardware*, o sistema manteve estabilidade e apresentou tempo de

processamento adequado para análises offline, reforçando sua aplicabilidade em cenários com infraestrutura restrita.

Sendo assim, como o processamento foi realizado em um ambiente controlado, não havendo integração com sistemas de monitoramento em tempo real, isso pode trazer uma limitação a aplicação prática de modo imediato no cenário atual desenvolvido.

Os impactos potenciais dos resultados para a gestão do tráfego local são significativamente proveitosos. A análise automatizada do fluxo do trânsito permite identificar horários e locais de maior movimentação, bem como detalhamento de informações sobre os veículos, propiciando decisões estratégicas como ajustes na sinalização, definição de rotas alternativas e planejamento de intervenções para reduzir congestionamentos. A integração futura do sistema com câmeras de monitoramento de tráfego em pequenas cidades poderia viabilizar alertas em tempo real e relatórios contínuos, ampliando a capacidade de gestão da autoridade municipal do trânsito e tornando o processo mais eficiente e orientado por dados.

Como sugestão de trabalhos futuros, propõe-se evoluir para um modelo integrado com infraestrutura existente, utilizando por exemplo de câmeras públicas de monitoramento já disponíveis no município, aliado a mecanismos de armazenamento em nuvem e *dashboards* interativos para acompanhamento dinâmico do tráfego. Essa abordagem consolida o sistema como uma ferramenta prática e escalável para cidades de pequeno porte, com viabilizações de investimento de baixo custo para implantação.

## 6.2 Avaliação do Sistema

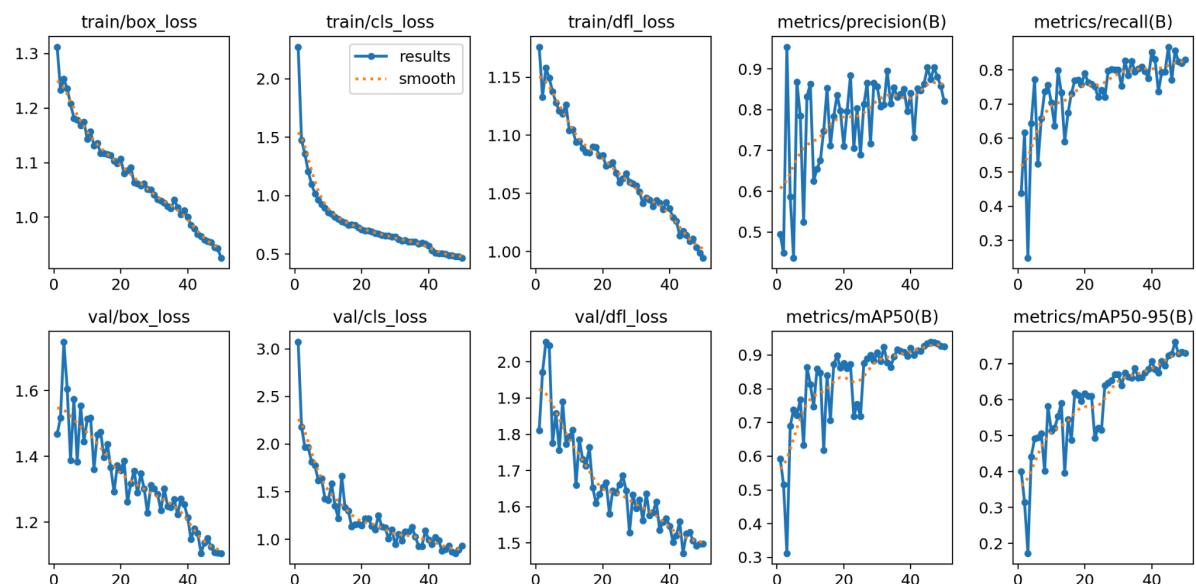
O sistema proposto, *Traffic Viewer*, foi avaliado considerando aspectos técnicos relacionados ao treinamento do modelo, execução do *pipeline* e desempenho geral.

O processo de treinamento ocorreu no ambiente Google Colab, utilizando a *GPU* disponibilizada pela plataforma, o que permitiu reduzir o tempo de processamento e garantir estabilidade durante as iterações. Ao final do treinamento, as métricas indicaram resultados satisfatórios, com precisão (*Precision*) aproximada

de 85%, revocação (*Recall*) em torno de 80%, mAP50 próximo de 85% e mAP50-95 em torno de 65%, conforme ilustrado na Figura 21 e detalhado na Tabela 1.

Além das métricas apresentadas, é fundamental considerar os impactos dos falsos positivos e falsos negativos na análise do sistema. Falsos positivos ocorrem quando o modelo identifica um veículo inexistente no contexto, enquanto falsos negativos representam casos em que um veículo presente não é detectado. Esses erros afetam diretamente a precisão e a revocação do modelo, podendo comprometer a confiabilidade das estatísticas de tráfego. Conforme Andrade (2020), tais inconsistências são comuns em aplicações de visão computacional para contagem e rastreamento, especialmente em cenários com alta densidade ou condições adversas, exigindo ajustes contínuos nos parâmetros do modelo para mitigar esses efeitos.

**Figura 21** - Valores das métricas do modelo



Fonte: próprio autor, 2025.

**Tabela 1** - Tabela com os valores das métricas do modelo

Métrica	Valor aproximado
Precision	0,85
Recall	0,80
mAP 50	0,85
mAp50-95	0,65

Fonte: próprio autor, 2025.

### 6.3 Comparativo do Sistema *Traffic Viewer* com Trabalhos Relacionados

Os trabalhos analisados nesta pesquisa apresentam abordagens importantes para a gestão do tráfego urbano, porém direcionadas, na sua maioria, a cidades de médio e grande porte, com infraestrutura tecnológica mais robusta. Em razão disto, o sistema desenvolvido neste estudo estabelece diferenciais importantes, sobretudo pela sua aplicabilidade em cidades de pequeno porte, realidade ainda pouco explorada na literatura.

Dentre os principais pontos de diferença da solução proposta, destaca-se a viabilidade em equipamentos de baixo custo, uma vez que todo o processamento foi realizado em um notebook com *GPU* integrada, sem necessidade de recursos avançados como *CUDA*, uma plataforma de computação paralela e modelo de programação criada pela *NVIDIA* ou placas gráficas dedicadas. Essa característica reforça a possibilidade de adoção por cidades com restrições orçamentárias.

Outro diferencial consiste na integração completa com interface *web*, permitindo não apenas a detecção e classificação de veículos, mas também a visualização dinâmica dos dados em *dashboards* e a exportação de relatórios técnicos. Essa funcionalidade amplia o potencial de uso do sistema como ferramenta de apoio à tomada de decisões estratégicas pelos órgãos gestores do trânsito.

Comparativamente, enquanto Reyna (2024) concentrou sua proposta na análise de anomalias utilizando algoritmos específicos, sem integração com uma interface que proporcionasse visualização de dados por meio de relatórios, o

presente trabalho oferece uma solução prática com essa funcionalidade e orientada à gestão municipal. Maus et al. (2021), por sua vez, desenvolveu uma aplicação voltada para cidades de grande porte, com dependência de *CUDA* e *YOLO v3*, enquanto a solução aqui apresentada utiliza *YOLO v8n*, garantindo maior precisão e eficiência em *hardware* acessível. Já Barbado et al. (2022) limitou sua análise a três classes de veículos e câmeras fixas, ao passo que este estudo amplia a categorização de classes com a inserção de motocicletas e adota uma arquitetura modular, favorecendo a replicabilidade futura.

Esses elementos evidenciam que a proposta não apenas atende às necessidades específicas de cidades menores, mas também se posiciona como alternativa economicamente viável e tecnicamente consistente frente às soluções existentes.

## 7.0 CONSIDERAÇÕES FINAIS

A mobilidade urbana é vista como um dos maiores desafios das cidades brasileiras nas últimas décadas e varia muito entre municípios pequenos, médios e grandes. Entre alguns desafios, estão o crescimento da frota de veículos que não acompanha a organização e infraestrutura urbana, causando congestionamentos e elevada taxa de acidentes. Entretanto, ao contrário do que comumente se pensa, cidades de pequeno porte podem também ter problemas no trânsito.

De acordo com uma pesquisa realizada pelo Instituto de Pesquisa Econômica Aplicada (IPEA, 2023), o Brasil registrou um aumento de 13,5% de mortes em acidentes de trânsito nos entre 2010 e 2019, número que inclui cidades pequenas, como o caso de Pojuca, estudo de caso desta pesquisa. A busca por melhorias passa por monitorar o trânsito e criar redes de infraestrutura que tornem uma cidade preparada tanto para a expansão da frota de veículos, como carros e motocicletas, quanto para permitir um ambiente urbano sem sinistros e situações que coloquem em risco a vida da população.

O objetivo deste trabalho foi propor um sistema de monitoramento inteligente do tráfego urbano para a cidade de Pojuca, localizada na região metropolitana de Salvador. Para isso, foi desenvolvida a aplicação nomeada *Traffic Viewer*, plataforma web que condensa os dados obtidos a partir da detecção de imagens em

vídeos previamente gravados no decorrer da semana nas vias centrais e mais movimentadas do município.

O desenvolvimento do sistema *Traffic Viewer* ocorreu em etapas bem definidas. Inicialmente, foram coletados vídeos que serviram como base para a criação do *dataset*. Em seguida, os dados foram preparados e utilizados no treinamento do modelo de Visão Computacional no ambiente Google Colab, utilizando a arquitetura *YOLO v8* na sua versão *Nano*. Após a obtenção do modelo, foi implementada a API responsável por receber os dados processados e armazená-los em banco relacional. Por fim, desenvolveu-se a plataforma *web*, composta por telas de *dashboard* e registros, permitindo a visualização interativa das informações e a exportação de relatórios técnicos em formato PDF.

Com os resultados obtidos, foi possível analisar a dinâmica do tráfego urbano em uma cidade de pequeno porte, identificando horários de pico, quantidade de veículos, velocidade média, locais de maior fluxo e percentual por tipo de veículo. Essas informações podem servir como base para os órgãos gestores de trânsito, como o STT, auxiliando na tomada de decisões estratégicas para melhorar a mobilidade urbana.

Apesar dos avanços tecnológicos, algumas especificações foram identificadas durante o estudo. Entre eles, destaca-se a dependência de condições específicas de iluminação, que influenciam diretamente a especificidade do modelo, e o uso de ângulo fixo de câmera, que restringe a abrangência espacial da análise. Soma-se a isso a limitação do conjunto de dados, composto por vídeos curtos e específicos, e a ausência de integração em tempo real, que manteve o sistema secundário para análises offline.

Como perspectivas futuras, propõe-se a integração com câmeras urbanas para permitir o monitoramento contínuo, a ampliação do conjunto de dados (*dataset*) com diferentes cenários e condições climáticas, além do desenvolvimento de módulos para análise preditiva, capazes de antecipar padrões de congestionamento. Também se considera a possibilidade de incorporar recursos de IoT de forma gradual, aproveitando a infraestrutura existente que conta com câmeras públicas municipais, e mantendo a proposta inicial de baixo custo, garantindo escalabilidade sem comprometer a estratégia econômica.

## REFERÊNCIAS

ANDRADE, Matheus Molin de. **Aplicação de visão computacional para rastreamento e contagem de veículos em rodovias. 2020.** Monografia (Graduação) - Escola Politécnica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2020. Disponível em: <<http://www.repositorio.poli.ufrj.br/monografias/monopolis10032456.pdf>>. Acesso em: 12 dez. 2025.

BARBADO, Larissa; SANTOS, Lucas Medeiros R. dos; MATRAKAS, Miguel Diógenes; MOREIRA, Jasmine. **Aplicação da rede convolucional Yolo para análise de fluxo de veículos.** In: 19º Congresso Latino-americano de Software livre e Tecnologias abertas - Parque Tecnológico Itaipu, - Foz do Iguaçu, Paraná, 2022.

BRASIL. **Decreto Lei nº 2.994, de 28 de janeiro de 1941.** Código Nacional de Trânsito. Disponível em <https://www2.camara.leg.br/legin/fed/declei/1940-1949/decretolei-2994-28-janeiro-1941-412976-publicacaooriginal-1-pe.html>. Acesso em: 31 ago. 2024.

BRASIL. **Decreto nº 8.324, de 27 de outubro de 1910.** Aprova o regulamento para o serviço subvencionado de transportes por automóveis. Disponível em [http://www.planalto.gov.br/ccivil\\_03/\\_Ato2011-2014/2014/Decreto/D8324.htm](http://www.planalto.gov.br/ccivil_03/_Ato2011-2014/2014/Decreto/D8324.htm). Acesso em: 31 ago. 2024.

BRASIL. **Lei nº 5.108, de 21 de setembro de 1996.** Institui o Código Nacional de Trânsito.

BRASIL. **Lei nº 9.503, de 23 de setembro de 1997.** Institui o Código de Trânsito Brasileiro. Diário Oficial da União, Brasília, DF, 24 set. 1997. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/leis/l9503.htm](http://www.planalto.gov.br/ccivil_03/leis/l9503.htm). Acesso em: 31 ago. 2024.

CONFEDERAÇÃO NACIONAL DE MUNICÍPIOS (CNM). **Análise do impacto da frota de veículos nos municípios brasileiros, Brasília - DF:** CNM, 2023. Disponível em [https://cnm.org.br/storage/biblioteca/2023/Estudos\\_tecnicos/202309\\_ET\\_MOB\\_Impacto\\_frota\\_veiculos.pdf](https://cnm.org.br/storage/biblioteca/2023/Estudos_tecnicos/202309_ET_MOB_Impacto_frota_veiculos.pdf). Acesso em: 02 jul. 2025.

IBM. **O que são redes neurais convolucionais?** Disponível em: [https://www.ibm.com/br-pt/topics/convolutional-neural-networks#:~:text=As%20redes%20neurais%20convolucionais%20\(Convolucional,de%20classifica%C3%A7%C3%A7%C3%A3o%20e%20vis%C3%A3o%20computacional](https://www.ibm.com/br-pt/topics/convolutional-neural-networks#:~:text=As%20redes%20neurais%20convolucionais%20(Convolucional,de%20classifica%C3%A7%C3%A7%C3%A3o%20e%20vis%C3%A3o%20computacional). Acesso em: 22 ago. 2024.

NELSON, Joseph. **O que é YOLO? O Guia Definitivo [2025]**. Blog da Roboflow, 09 de Janeiro de 2025. Disponível em <https://blog.roboflow.com/guide-to-yolo-models/>

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. **ImageNet classification with deep convolutional neural networks**. Advances in Neural Information Processing Systems, 25, 1097-1105, 2012.

MAUS, André Vítor; CAMILLO Jr, Ademir; LINHARES Jr, Valdir; RAZERA, Tereza Martins; Kuntz, Roney. **Contagem e Classificação de Veículos por Visão Computacional**. In: XII Computer on the Beach, 07 a 09 de Abril de 2021, Online, SC, Brasil.

MUI. **Material UI - TypeScript Guide**. Disponível em: <https://mui.com/material-ui/guides/typescript/>. Acesso em: 25 jul. 2025.

NOBRE, Karine. **Frota de veículos cresce 43,7% em 12 anos no Brasil**. A Gazeta, 23 nov. 2023. Disponível em: <https://www.agazeta.com.br/colunas/karine-nobre/frota-de-veiculos-cresce-437-em-12-anos-no-brasil-1123>. Acesso em: 18 ago. 2024.

NESTJS. **Using Prisma with NestJS**. Disponível em: <https://docs.nestjs.com/recipes/prisma>. Acesso em: 25 jul. 2025.

POJUCA. **Lei Municipal Complementar nº 012, de 29 de dezembro de 2022**. Cria a Superintendência de Trânsito e Transportes (STT) e define suas atribuições. Disponível em: <https://www.pojuca.ba.gov.br/stt/institucional>. Acesso em: 18 ago. 2024.

POSTGRESQL. **PostgreSQL Documentation**. Disponível em: <https://www.postgresql.org/docs/>. Acesso em: 25 jul. 2025.

PRISMA. **Prisma ORM Documentation**. Disponível em: <https://www.prisma.io/docs>. Acesso em: 25 jul. 2025.

REACT. **Using TypeScript - React**. Disponível em: <https://react.dev/learn/typescript>. Acesso em: 25 jul. 2025.

REYNA, Ana Rosalia Huaman. **Desenvolvimento de um sistema de detecção e rastreamento de veículos para análise de anomalias de tráfego em rodovias utilizando estruturas espaciais e temporais por meio de visão computacional**.

Dissertação (Mestrado em Ciência da Computação e Matemática) - Instituto de Ciências Matemáticas e da Computação, Universidade de São Paulo (USP), 2024.

SMARTBEAR. **O que é o OpenAPI?** Disponível em: [https://swagger.io/docs/specification/v3\\_0/about/](https://swagger.io/docs/specification/v3_0/about/). Acesso em: 25 jul. 2025

TYPESCRIPT. **TypeScript: Documentation - React.** Disponível em: <https://www.typescriptlang.org/pt/docs/handbook/react.html>. Acesso em: 25 jul. 2025.

ULTRALYTICS. **Explorar o Ultralytics YOLOv8.** Disponível em: <https://docs.ultralytics.com/pt/models/yolov8/>. Acesso em: 02 jul. 2025.

VIEIRA, Alexandre Bergamin; ROMA, Cláudia Marques; MIYAZAKI, Vitor Koiti. **Cidades Médias e Pequenas: Uma Leitura Geográfica.** Programa de Pós-graduação em Geografia, Universidade Estadual Paulista – UNESP, Faculdade de Ciências e Tecnologia – FCT Campus de Pres. Prudente, 2004.

## ANEXOS

### Anexo 1

## Anexo 1 - Frota de Veículos Cadastrados na Bahia (por Município X Tipo) Agrupado por CIRETRAN

### Taxa de crescimento de motocicletas em Pojuca, Ba



Referência: Dezembro / 2022

Ciretran Número	Local	Tipo de Veículo							Frota	%
		Automóveis	Caminhonetas	Caminhões	Ônibus	Micro-ônibus	Motocicletas	Outros		
28	POJUCA	5.509	1.191	809	116	130	3.077	2.173	13.005	0,2585



Referência: Julho / 2023

CIRETRAN NUMERO	Local	Tipo de Veículo							Frota	%
		Automóveis	Caminhoneta	Caminhão	Ônibus	Micro Ônibus	Motocicletas	Outros		
28	POJUCA	5.665	1.222	828	121	129	3.161	2.253	13.379	0,26



Referência: Julho/2024

Ciretran Número	Local	Tipo de Veículo							Frota	%
		Automóveis	Caminhonetas	Caminhões	Ônibus	Micro-ônibus	Motocicletas	Outros		
28	POJUCA	5.911	1.311	861	118	129	3.334	2.482	14.146	0,26

# APÊNDICES

## Apêndice 1

### Resumo Executivo

Este relatório apresenta uma análise do tráfego urbano na cidade de Pojuca-BA. Com base nos dados coletados entre 10/11/2025 e 17/11/2025Estando no intervalo de 09:45 às 12:00, no local Rua Antônio Mota.

Foram detectados 21 veículos, com velocidade média de 17.92 km/h.

O tipo de veículo mais comum foi Carro.

### Metodologia

Os dados foram coletados por câmeras inteligentes com detecção de veículos via IA. A coleta foi segmentada por hora, tipo de veículo e direção de fluxo.

A plataforma Traffic Viewer utiliza algoritmos de visão computacional para identificar e classificar os veículos com alta precisão.

### Análise Quantitativa



### Indicadores Estratégicos

Horário de maior fluxo  
11:00

Tipo de veículo predominante  
Carro

Local mais congestionado  
Rua Antônio Mota

### Conclusões e Recomendações

Com base nos dados coletados, recomenda-se:

**Máxima** - Investigar causas de lentidão nas vias com média inferior a 15 km/h.

### Apêndices

Abaixo estão os dados brutos utilizados para gerar este relatório.

Hora	Quantidade
11:00	21

## Apêndice 2

### Resumo Executivo

Este relatório apresenta uma análise do tráfego urbano na cidade de Pojuca-BA. Com base nos dados coletados entre 09/11/2025 e 15/11/2025, no local Rua Antônio Mota. Foram detectados 65 veículos, com velocidade média de 17.87 km/h. O tipo de veículo mais comum foi Carro.

### Metodologia

Os dados foram coletados por câmeras inteligentes com detecção de veículos via IA. A coleta foi segmentada por hora, tipo de veículo e direção de fluxo. A plataforma Traffic Viewer utiliza algoritmos de visão computacional para identificar e classificar os veículos com alta precisão.

### Análise Quantitativa



### Indicadores Estratégicos

Horário de maior fluxo  
09:00

Tipo de veículo predominante  
Carro

Local mais congestionado  
Rua Antônio Mota

### Conclusões e Recomendações

Com base nos dados coletados, recomenda-se:

- Média - Investigar causas de lentidão nas vias com média inferior a 15 km/h.
- Média - Reforçar a sinalização nos horários de pico (09:00).

### Apêndices

Abaixo estão os dados brutos utilizados para gerar este relatório.

Hora	Quantidade
09:00	15
11:00	25
15:00	12
17:00	8

## Apêndice 3

### Resumo Executivo

Este relatório apresenta uma análise do tráfego urbano na cidade de Pojuca-BA. Com base nos dados coletados entre 09/11/2025 e 15/11/2025, no local Rua Antônio Batista.

Foram detectados 60 veículos, com velocidade média de 15.85 km/h.

O tipo de veículo mais comum foi Carro.

### Metodologia

Os dados foram coletados por câmeras inteligentes com detecção de veículos via IA. A coleta foi segmentada por hora, tipo de veículo e direção de fluxo.

A plataforma Traffic Viewer utiliza algoritmos de visão computacional para identificar e classificar os veículos com alta precisão.

### Análise Quantitativa



### Indicadores Estratégicos

Horário de maior fluxo  
08:00

Tipo de veículo predominante  
Carro

Local mais congestionado  
Rua Antônio Batista

### Conclusões e Recomendações

Com base nos dados coletados, recomenda-se:

**Máx** - Investigar causas de lentidão nas vias com média inferior a 15 km/h.

### Apêndices

Abaixo estão os dados brutos utilizados para gerar este relatório.

Hora	Quantidade
08:00	21
10:00	17
14:00	11
16:00	11

## Apêndice 4

### Resumo Executivo

Este relatório apresenta uma análise do tráfego urbano na cidade de Pojuca-BA. Com base nos dados coletados entre 09/11/2025 e 15/11/2025, no local Rua J. J. Seabra.

Foram detectados 46 veículos, com velocidade média de 12,20 km/h.

O tipo de veículo mais comum foi Carro.

### Metodologia

Os dados foram coletados por câmeras inteligentes com detecção de veículos via IA. A coleta foi segmentada por hora, tipo de veículo e direção de fluxo.

A plataforma Traffic Viewer utiliza algoritmos de visão computacional para identificar e classificar os veículos com alta precisão.

### Análise Quantitativa



### Indicadores Estratégicos

Horário de maior fluxo  
17:00

Tipo de veículo predominante  
Carro

Local mais congestionado  
Rua J. J. Seabra

### Conclusões e Recomendações

Com base nos dados coletados, recomenda-se:

**Média** - Investigar causas de lentidão nas vias com média inferior a 15 km/h.

### Apêndices

Abaixo estão os dados brutos utilizados para gerar este relatório.

Hora	Quantidade
07:00	21
12:00	10
13:00	4
17:00	11