

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

**FRAME: SEMÁFORO DINÂMICO PARA A MELHORIA DO  
TRÁFEGO TERRESTRE DE VEÍCULOS EM CRUZAMENTOS DE  
ÁREAS URBANAS**

**Murilo Cambuzzi Paiva**

Ji-Paraná - RO

2025

**Murilo Cambuzzi Paiva**

**FRAME: SEMÁFORO DINÂMICO PARA A MELHORIA DO  
TRÁFEGO TERRESTRE DE VEÍCULOS EM CRUZAMENTOS DE  
ÁREAS URBANAS**

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

Orientador: Prof. Dr. Wanderson Roger Azevedo Dias

Ji-Paraná - RO

2025

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO.

Paiva, Murilo Cambuzzi.

Frame: semáforo dinâmico para a melhoria do tráfego terrestre de veículos em cruzamentos de áreas urbanas / Murilo Cambuzzi Paiva. - Ji-Paraná, 2025.

59 f.

Orientador(a): Dr. Wanderson Roger Azevedo Dias.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO, Ji-Paraná, 2025.

1. Semáforo dinâmico. 2. Gerenciamento de tráfego. 3. Mobilidade urbana. 4. Processamento de imagens. 5. Inteligência artificial. I. Dias, Wanderson Roger Azevedo (orient.). II. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO. III. Título.

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

## **BANCA AVALIADORA**

---

Prof. Dr. Wanderson Roger Azevedo Dias (CCSTADS/IFRO)  
Presidente e Orientador

---

Prof. Me. Walter Ferreira Siqueira  
Membro

---

Prof. Me. Gleison Guardia  
Membro

**Data da Defesa:** 19 de agosto de 2025

**Nota Final:** 97

Ji-Paraná - RO  
2025

## RESUMO

Atualmente, os semáforos desempenham um papel fundamental no gerenciamento do tráfego urbano, especialmente em cruzamentos, contribuindo de forma significativa para a redução de acidentes envolvendo veículos e pedestres. Em diversas cidades ao redor do mundo, já se encontram implantados sistemas automatizados de controle de tráfego, que representam um avanço em relação ao modelo manual. No entanto, a maioria desses sistemas ainda opera de maneira estática, com tempos fixos para a abertura e o fechamento dos sinais luminosos. Essa característica resulta, frequentemente, em esperas desnecessárias por parte dos condutores, sobretudo em situações em que o fluxo de veículos em determinada via é reduzido. Como alternativa a essa limitação, surge a proposta de um sistema de semáforo dinâmico, capaz de operar de forma independente e inteligente, ajustando os tempos de sinalização de acordo com a demanda real de tráfego. Para isso, o sistema realiza a detecção e a contagem de veículos por meio de câmeras, permitindo que os sinais sejam gerenciados de maneira mais eficiente. Essa abordagem promove maior fluidez no trânsito, reduz congestionamentos e melhora a mobilidade urbana. Assim, neste trabalho, foi desenvolvido um protótipo de semáforo dinâmico voltado para o aperfeiçoamento do tráfego em cruzamentos urbanos. A solução consistiu na utilização de câmeras para identificar e contabilizar a quantidade de veículos em circulação, considerando diferentes categorias de transporte, de modo que o tempo de abertura dos sinais luminosos fosse ajustado proporcionalmente à demanda detectada em cada via. Dessa forma, buscou-se aliar tecnologia e inovação para contribuir com a otimização da mobilidade urbana e a segurança viária.

**Palavras-chave:** Semáforo dinâmico; Gerenciamento de tráfego; Mobilidade urbana; Processamento de imagens; Inteligência artificial.

## ABSTRACT

Currently, traffic lights play a fundamental role in managing urban traffic, especially at intersections, significantly contributing to the reduction of accidents involving vehicles and pedestrians. In several cities around the world, automated traffic control systems have already been implemented, representing an advancement over manual models. However, most of these systems still operate in a static manner, with fixed timing for the opening and closing of light signals. This often results in unnecessary waiting for drivers, especially in situations where traffic flow on a given road is low. As an alternative to this limitation, the proposal of a dynamic traffic light system emerges, capable of operating independently and intelligently by adjusting signal timings according to real-time traffic demand. To achieve this, the system performs vehicle detection and counting through cameras, allowing the signals to be managed more efficiently. This approach promotes better traffic flow, reduces congestion, and improves urban mobility. Thus, this work developed a dynamic traffic light prototype aimed at improving traffic flow at urban intersections. The solution involved using cameras to identify and count the number of vehicles in circulation, considering different types of transportation, so that the green light duration could be adjusted proportionally to the demand detected on each road. Therefore, this project aimed to combine technology and innovation to contribute to the optimization of urban mobility and traffic safety.

**Keywords:** Dynamic traffic light; Traffic management; Urban mobility; Image processing; Artificial intelligence.

## LISTA DE FIGURAS

Figura 1 – Semáforos	15
Figura 2 – Utilização de uma câmera no tráfego de veículos	17
Figura 3 – Arquitetura do sistema do semáforo dinâmico	19
Figura 4 – Projeto das vias da maquete	24
Figura 5 – Versão final da maquete	25
Figura 6 – Rede neural profunda	30
Figura 7 – Comparação de desempenho entre diferentes metodologias de detecção	31
Figura 8 – Plataforma Arduino - modelo Mega 2560	35
Figura 9 – IDE Arduino	35
Figura 10 – Protoboard	36
Figura 11 – Display de Sete Segmentos	37
Figura 12 – Módulo Semáforo	37
Figura 13 – Jumper	38
Figura 14 – Cabo USB-A	38
Figura 15 – Cabo USB-B	39
Figura 16 – Metodologia Scrum	41
Figura 17 – Papéis do Scrum	42
Figura 18 – Os 17 objetivos da ODS	44
Figura 19 – Diagrama de caso de uso	51

## LISTA DE QUADROS

Quadro 1 – Divisão da equipe	46
Quadro 2 – Product backlog	47
Quadro 3 – Sprint backlog	48
Quadro 4 – Requisito Funcional: Reconhecer Veículo	49
Quadro 5 – Requisito Funcional: Realizar a Contagem dos Veículos	49
Quadro 6 – Requisito Funcional: Realizar o controle do semáforo	50
Quadro 7 – Requisito Funcional: Emitir contagem regressiva	50

## LISTA DE ABREVIATURAS E SIGLAS

<i>CPU</i>	<i>Central Processing Unit</i>
<i>DNN</i>	<i>Deep Neural Network</i>
<i>IA</i>	<i>Inteligência Artificial</i>
<i>IDE</i>	<i>Integrated Development Environment</i>
<i>LED</i>	<i>Light Emitting Diode</i>
<i>mAP</i>	<i>Mean Average Precision</i>
<i>ML</i>	<i>Machine Learning</i>
<i>MS COCO</i>	<i>Microsoft Common Objects in Context</i>
<i>NMS</i>	<i>Non Maximum Suppression</i>
<i>OpenCV</i>	<i>Open Source Computer Vision Library</i>
<i>USB</i>	<i>Universal Serial Bus</i>
<i>VS Code</i>	<i>Visual Studio Code</i>
<i>YOLO</i>	<i>You Only Look Once</i>

## SUMÁRIO

1 TEMA.....	12
1.1 Delimitação do Tema.....	12
1.2 Problemática .....	12
2 JUSTIFICATIVA .....	12
3 HIPÓTESE .....	13
4 OBJETIVOS.....	13
4.1 Objetivo Geral .....	13
4.2 Objetivos Específicos .....	13
5 SEMÁFORO .....	14
5.1 Semáforo Tradicional .....	14
5.2 Semáforo Dinâmico .....	16
5.2.1 Funcionamento .....	16
5.2.2 Arquitetura.....	18
5.2.3 Condições para Implementação.....	19
5.2.4 Riscos a Integridade do Sistema.....	20
5.2.4.1 Fatores Ambientais.....	20
5.2.4.2 Vandalismo.....	20
5.2.4.3 Acidentes de Trânsito .....	21
5.2.4.4 Funcionamento do Semáforo.....	21
5.2.4.5 Componentes Eletrônicos .....	22
5.2.5 Intervenção Humana para Manutenção .....	22
5.2.6 Funcionalidades Adicionais.....	23
6 MAQUETE.....	24
7 FUNDAMENTAÇÃO TEÓRICA .....	25
7.1 Back-end.....	25

7.2 Programação Paralela .....	27
7.3 Visão Computacional .....	27
7.4 Machine Learning.....	28
7.5 Rede Neural .....	29
7.6 YOLO .....	30
7.7 Dataset COCO .....	33
7.8 OpenCV .....	33
7.9 Arduíno Mega.....	34
7.10 Componentes Eletrônicos do Arduino.....	36
7.10.1 Protoboard .....	36
7.10.2 Display de Sete Segmentos.....	36
7.10.3 Módulo Semáforo .....	37
7.10.4 Jumper .....	38
7.10.5 USB padrão A/B.....	38
7.11 Python.....	39
7.12 C++ .....	39
7.13 Visual Studio Code.....	40
7.14 Metodologia Ágil Scrum .....	40
7.15 Cidades Inteligentes.....	43
8. OBJETIVOS DE DESENVOLVIMENTO SUSTENTÁVEL.....	43
9 METODOLOGIA DE DESENVOLVIMENTO .....	45
10 MODELAGEM DA SOLUÇÃO PROPOSTA .....	46
10.1 Product Backlog .....	46
10.2 Sprints Backlog .....	47
10.3 Levantamento de Requisitos.....	48
10.3.1 Reconhecer os Veículos .....	49
10.3.2 Realizar a Contagem dos Veículos.....	49

10.3.3 Realizar o Controle do Semáforo .....	50
10.3.4 Emitir Contagem Regressiva.....	50
10.4 Modelo de Caso de Uso.....	51
10.4.1 Diagrama de Caso de Uso .....	51
10.4.2 Diagrama de Caso de Uso Expandido .....	51
10.4.2.1 Caso de Uso - iniciar o sistema .....	51
10.4.2.2 Caso de Uso - monitora tráfego de veículos.....	51
10.4.2.3 Caso de Uso - observa se o sinal está verde .....	52
10.4.2.4 Caso de Uso - desliga o sistema .....	52
11 CONCLUSÕES E TRABALHOS FUTUROS .....	53
REFERÊNCIAS .....	55

## **1 TEMA**

Semáforo dinâmico para a melhoria do tráfego terrestre.

### **1.1 Delimitação do Tema**

Desenvolver um protótipo de semáforo dinâmico para a melhoria do tráfego terrestre de veículos em cruzamentos de áreas urbanas. A solução consiste na captura da quantidade de veículos de diferentes categorias que trafegam pela via para que seja proporcional ao tempo de abertura dos sinais luminosos do semáforo.

### **1.2 Problemática**

A utilização de semáforos com o tempo estático em seus sinais luminosos, em cruzamentos urbanos, exerce grande influência no desempenho e no fluxo de veículos e pedestres. Entretanto, esse tráfego pode variar em diferentes momentos do dia, podendo aumentar ou diminuir de acordo com a necessidade dos condutores de se deslocarem. Sendo assim, ter o tempo do sinal luminoso estático pode se tornar ineficiente, pois gera congestionamentos em horário de maior movimentação e tempo de espera desnecessário em horários de menor movimento, resultando no aumento do período de traslado dos condutores.

## **2 JUSTIFICATIVA**

Diante do problema apresentado é viável a ideia de desenvolver um semáforo dinâmico, onde o sistema possui ciência da quantidade de veículos que trafegam em determinada via e gerencia o tempo de cada sinal luminoso proporcional a esta quantidade de veículos contados. Dessa forma, o funcionamento do semáforo será moldado de acordo com as diferentes intensidades dos fluxos de um cruzamento.

Essa medida tornará o trânsito mais fluido, evitando que os condutores tenham que aguardar desnecessariamente pela passagem no cruzamento quando o fluxo de veículos estiver baixo. Com essa ideia aplicada, a média do tempo dos traslados será reduzida, o que resulta em uma economia de combustível e redução do impacto ambiental devido a emissão dos poluentes, contribuindo para um ambiente mais sustentável em longo prazo.

Diante dessa proposta, fica evidente que a implementação de semáforos dinâmicos inteligentes pode proporcionar benefícios significativos para a mobilidade urbana, com um

trânsito mais eficiente, seguro e sustentável. Além disso, diversas melhorias podem ser implementadas, contribuindo ainda mais para melhorar a mobilidade urbana.

### 3 HIPÓTESE

- O fluxo de veículos no cruzamento de maneira geral será mais fluido;
- O sistema irá contribuir para um ambiente mais sustentável;
- O sistema não irá melhorar o tráfego caso todas as vias de um cruzamento estejam congestionadas ao mesmo tempo.

### 4 OBJETIVOS

#### 4.1 Objetivo Geral

O sistema FRAME consiste em um gerenciador luminoso dinâmico de tráfego de veículos em cruzamentos de vias urbanas. Seu principal objetivo é melhorar o controle do fluxo de veículos para reduzir congestionamentos, como também o tempo de espera desnecessário nos semáforos, proporcionando deslocamentos mais rápidos.

#### 4.2 Objetivos Específicos

Para alcançar o objetivo geral, será necessário seguir com uma série de objetivos específicos. São eles:

- Definir e validar a arquitetura de hardware para reconhecimento de objetos e controle semafórico;
- Integrar as duas arquiteturas de hardware distintas através de um USB Serial, garantindo uma comunicação confiável entre elas;
- Analisar, compreender e selecionar a melhor tecnologia, bibliotecas, técnica de visão computacional e rede neural para aplicar o reconhecimento de objetos;
- Desenvolver um *Script* em Python responsável pelo reconhecimento e contagem de veículos automotores garantindo eficiência;
- Implementar um *Script* para o funcionamento do semáforo e a contagem regressiva em ordem decrescente na plataforma de prototipagem Arduino, integrando-a com a contagem de veículos;

- Montar a maquete de apresentação, demonstrando a integração funcional entre o reconhecimento e contagem de objetos com a representação semafórica.

## 5 SEMÁFORO

### 5.1 Semáforo Tradicional

O semáforo luminoso é uma das principais formas de gestão de tráfego terrestre utilizado em diversas cidades pelo mundo e em pontos estratégicos que necessitam deste gestor para controlar a passagem de veículos e pedestres garantindo um trânsito mais fluído, garantindo a queda na taxa de acidentes, principalmente se contar com um temporizador de tempo para fechar o sinal luminoso verde. “Uma pesquisa da USP, em São Carlos, mostra que a utilização de semáforos com temporizador - sistema que indica o tempo restante para troca das luzes vermelha/verde - pode reduzir o número de acidentes.” (ARAUJO, 2011, p.1).

Segundo o CONTRAN (2022), ele possui diferentes combinações de cores, formas e símbolos, cada uma passando uma mensagem ao condutor e ao pedestre. Elas possuem o objetivo de facilitar a compreensão das pessoas que trafegam pela via, evitando um acúmulo de diversas combinações diferentes, podendo ocasionar acidentes. Para que surjam novas combinações de sinais em um semáforo, o órgão responsável pela fiscalização deverá ser acionado para autorizar a nova medida. “O uso de sinais não previstos, em caráter definitivo, somente poderá ocorrer após a devida regulamentação pelo CONTRAN.” (CONTRAN, 2022, p.10).

O semáforo convencional para veículos é composto por 3 (três) focos de luz no formato circular que indicam, através de suas cores, o que o condutor do veículo deverá fazer. A cor vermelha indica a proibição do direito de passagem, já a cor amarela indica que o condutor deve parar o veículo salvo se não for possível imobilizá-lo em condições de segurança, enquanto a verde indica a permissão do direito de passagem. Quando o indicador luminoso possuir uma seta na forma circular, ela indica a direção da via que a mensagem está representando. Sempre 1 (um) sinal luminoso ficará aceso de cada vez, enquanto os outros estarão apagados, dessa forma a mensagem passada ao condutor fica claro o que cada cor representa. Há a possibilidade de somente o indiciador amarelo ficar intermitente por um tempo de 1 (um) segundo, indicando a existência de situação perigosa ou obstáculo.

As cores estão distribuídas na forma de linha reta em um painel que emoldura o grupo focal chamado de anteparo e possui a finalidade de melhorar a visibilidade do condutor

destacando-a em meio à paisagem urbana e na incidência de raios solares. Toda esta estrutura deverá estar apoiada em um elemento de sustentação, que são elementos que possui a finalidade de sustentar todo o anteparo contendo os sinais luminosos, tais como colunas, braços projetados, cordoalhas e pórticos. A Figura 1 apresenta exemplos de Semáforos.

Figura 1 – Semáforos



Fonte: DETRAN-DF (s.d)

Como afirma o CONTRAN (2022), a sinalização semáfora faz parte de um grupo que compõem a sinalização viária. Este grupo possui por finalidade a passagem de diferentes informações aos usuários da via pública, sejam eles condutores ou pedestres. A sinalização semáfora é dividida em duas classes chamadas de sinalização semafórica de regulamentação e sinalização semafórica de advertência.

- Sinalização semafórica de regulamentação: consiste em gerenciar o controle do trânsito em uma intersecção de via através de combinações de indicações luminosas nas cores verde, amarelo e vermelho e símbolos, conforme explicado neste tópico;
- Sinalização semafórica de advertência: possui a finalidade de advertir o usuário sobre a existência de um obstáculo ou uma situação perigosa, exigindo um nível maior de atenção por parte dele. Pode haver um único sinal luminoso na cor amarelo intermitente dentro de um anteparo, como também pode ser utilizada essa funcionalidade em um anteparo tendo todas as cores, mas os focos verde e vermelho são apagados e o foco amarelo opera de forma intermitente.

Como aponta na Revista Superinteressante (2017), desenvolvido por John Peake Knight, o primeiro semáforo foi instalado em uma praça em frente ao Parlamento Britânico em 1868 e até os dias atuais são amplamente utilizados pela sociedade como forma de gerenciador de tráfego. Esta invenção possui formas de melhorar a ideia original utilizando a tecnologia para desenvolver novas ideias, como o semáforo dinâmico, que pode colaborar significativamente com a melhoria na área social, econômico e ambiental.

## 5.2 Semáforo Dinâmico

Ao contrário dos semáforos tradicionais, um semáforo dinâmico possui a flexibilidade de adaptar o tempo de abertura do sinal verde de cada via de acordo com a quantidade de veículos que transitam por ela, graças à combinação de tecnologias que analisam e quantificam o fluxo dos veículos e tomam decisões a partir destes dados coletados em tempo real. Essa funcionalidade proporciona um fluxo de veículos mais fluído, reduzindo o tempo de espera gasto pelos condutores dos veículos e agrega para o desenvolvimento de uma cidade inteligente.

### 5.2.1 Funcionamento

O funcionamento deste sistema necessita de um detector de tráfego como fonte de dados sobre o fluxo dos veículos. Segundo a CONTRAN (2022), estes detectores possuem a função de detectar a demanda de tráfego através diferentes tecnologias para esta funcionalidade, tais como laços detectores indutivos, botoeiras, detecção magnética entre outros. A tecnologia utilizada neste sistema é chamada de laços virtuais por tratamento de imagem, tendo o princípio a detecção de vídeo como entrada de informações para a central controladora.

A coleta do fluxo de veículos que definem o tempo de operação dos grupos focais é realizada por uma câmera treinada e posicionada em uma distância calculada do cruzamento que pode variar do cenário onde o semáforo será instalado, onde são capturados os frames que são processados com o intuito de identificar classes específicas de veículos, onde cada classe representa um grupo de um determinado objeto. Além de realizar a contagem de cada veículo que transita pela via e envia este dado para a central de processamento, que por sua vez, é responsável de receber o dado e tomar a decisão. A Figura 2 apresenta a utilização de uma câmera no monitoramento do tráfego de veículos em vias.

Cada via possui uma câmera que trabalha de forma independente, detectando diferentes veículos e realizando diferentes contagens de uma via em relação às outras. Durante o intervalo de tempo entre o momento em que o sinal verde de seu fluxo fechou até o momento que ele deverá ser aberto novamente, ela estará identificando e realizando a contagem dos veículos para repassar este dado para a central controladora gerenciar os componentes eletrônicos. Durante a abertura do sinal verde de seu fluxo, ela deve ser preparada para o início de uma nova etapa de detecção, zerando a quantidade de objetos contados para não haver inconsistência de dados na próxima entrega de dados.

Figura 2 – Utilização de uma câmera no tráfego de veículos



Fonte: Vargas (2025)

Cada grupo focal é acompanhado por um *display* numérico para mostrar a contagem regressiva do tempo em que o sinal luminoso verde ficará aberto, pois esta é uma informação que pode ser prevista pelo sistema, sendo uma informação importante para deixar o condutor ciente do tempo restante que ele possui para realizar a travessia do cruzamento. O cenário é diferente para o tempo restante de um sinal vermelho de uma determinada via ficará aberto, pois é um valor indeterminado e dependerá do fluxo de veículos das outras vias em cada ciclo daquele momento da captura.

Todo o gerenciamento dos dados de veículos recebidos de cada câmera, como também o controle dos componentes eletrônicos que compõe o semáforo, tomada de decisão para os

diferentes cenários, sincronização entre os processos independentes de cada fluxo que devem atuar de forma sincronizada é feito por uma central de processamento. Ela é responsável por gerenciar cada ciclo do semáforo, que é definido como cada volta completa entre todos os fluxos que compõe o cruzamento no qual o semáforo está incluso. Este gerenciamento é responsável pela interação de diferentes cenários de cada fluxo de forma sincronizada e dinâmica, além da comunicação entre as diferentes tecnologias utilizadas na arquitetura do sistema.

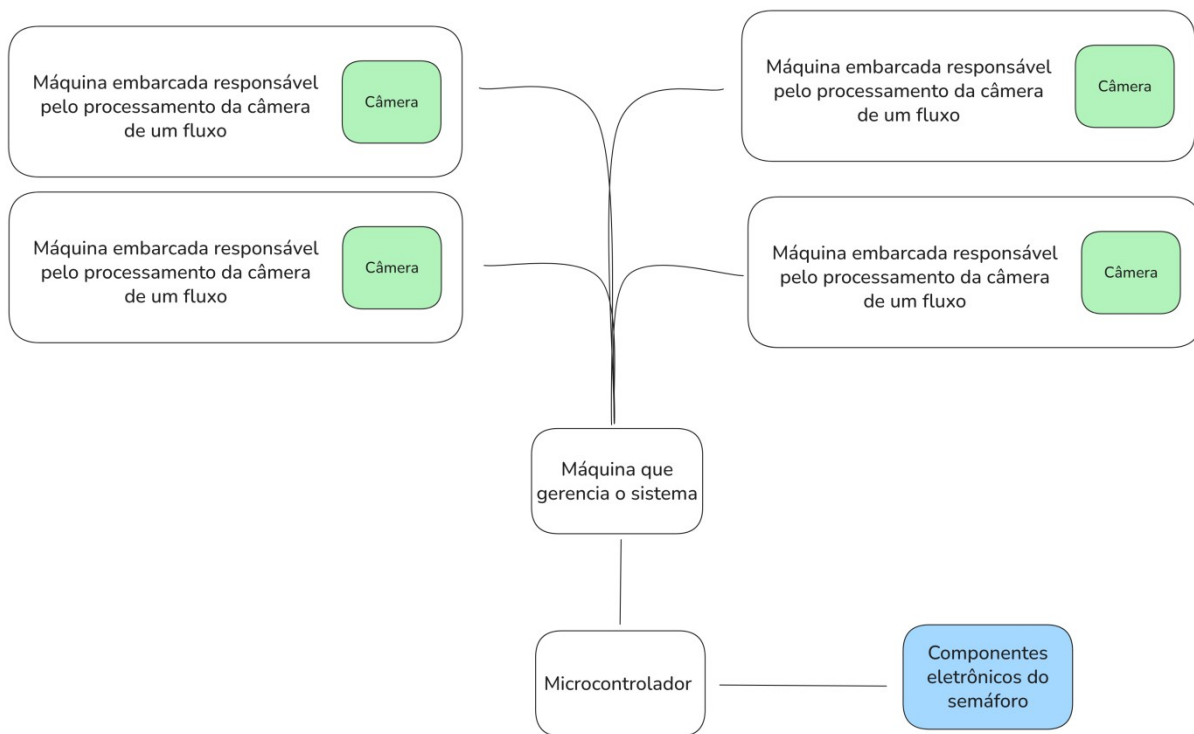
Apesar das inovações, o semáforo pode ter o funcionamento semelhante ao do tradicional em ocasiões específicas. Esse comportamento pode ser usado em situações em que haja alguma falha na integridade do sistema, como a perda da comunicação entre as unidades de processamento ou problema na identificação dos veículos, necessitando de intervenção humana para solucionar o problema. Essa funcionalidade permite que mesmo que o sistema apresente uma falha, o trânsito não seja impactado de forma negativa, mantendo a segurança viária até que o sistema volte para o seu pleno funcionamento. Outro comportamento que se mantém no semáforo dinâmico é o uso do sinal luminoso amarelo intermitente em horários de baixo fluxo de veículos, como nas madrugadas. Além de ser uma funcionalidade já familiar aos condutores, evita o processamento desnecessário, aumentando a vida útil dos recursos computacionais.

### 5.2.2 Arquitetura

A arquitetura do semáforo pode ser implementada de duas formas principais: com todo o processamento da câmera e o gerenciador do sistema centralizados em uma única máquina, ou ter uma unidade de processamento embarcada para cada câmera e ter uma máquina isolada para atuar como gerenciador, dividindo a carga total de processamento. O primeiro cenário exige um alto poder computacional, o que pode acarretar em um alto custo orçamentário, além de estar suscetível a uma pane geral caso a unidade de processamento central apresentar falhas. Já a segunda opção permite realizar um balanceamento em diferentes unidades, não necessitando de um hardware muito potente, além do impacto para o sistema ser menor caso haja a falha de um equipamento, podendo ser previamente programado para lidar com este cenário. Construir o sistema tendo o segundo cenário implantado é a melhor opção, pois além de possuir um menor custo financeiro, por haver arquiteturas com menor poder de processamento, proporciona uma tolerância a falha planejada para evitar que o funcionamento do sistema como um todo seja impactado.

Também facilita a manutenção preventiva e corretiva, não havendo a necessidade de parada total do serviço do semáforo. Estes fatores aumentam a confiabilidade do sistema como um todo. A Figura 3 apresenta o melhor cenário para a arquitetura do sistema.

Figura 3 – Arquitetura do sistema do semáforo dinâmico



Fonte: Autoria própria (2025)

### 5.2.3 Condições para Implementação

Para implementar este tipo de semáforo, é necessária uma avaliação da necessidade deste tipo de sistema gerenciador de tráfego, pois algumas exigências específicas devem ser atendidas para garantir o seu pleno funcionamento. Como é utilizado o método de detecção de vídeo, é fundamental que as vias possuam uma iluminação adequada, especialmente em períodos noturno, para evitar falhas na contagem dos veículos, ocasionando uma perda significativa na integridade dos dados gerados pelo sistema. Outro aspecto importante que necessita de avaliação é sobre o posicionamento da câmera, que necessita de uma distância do cruzamento para que ela contabilize os veículos antes de chegar ao semáforo. Além disso, nesse trecho não podem existir interseções, pois estas poderiam resultar em contagens incorretas, comprometendo a eficácia do sistema.

#### 5.2.4 Riscos a Integridade do Sistema

É essencial considerar os riscos que podem afetar a integridade e funcionamento do semáforo dinâmico, apesar das tecnologias utilizadas, planejamento preventivo a falhas e etapas de segurança implementadas. A seguir, são apresentados os principais fatores que exigem atenção e que colocam a plenitude do sistema em risco.

##### 5.2.4.1 Fatores Ambientais

Apesar da construção da infraestrutura ser robusta, alguns fatores externos podem trazer riscos. Eventos como chuvas fortes e enchentes podem danificar a parte elétrica por causa da umidade gerada, sendo necessária a elaboração de um mecanismo de proteção para os componentes eletrônicos tornando-os resistentes a água. Terremotos com magnitudes elevadas podem comprometer os elementos de sustentação das câmeras e grupos focais. O excesso de poeira ou geadas ocasionadas pela diminuição da temperatura nas lentes da câmera, podem diminuir a visibilidade dela e impactando na detecção e contagem dos veículos. O aumento de temperaturas ocasiona o aquecimento dos equipamentos de hardware, podendo diminuir a vida útil dos componentes. Altas descargas elétricas podem gerar picos de tensão queimando as fontes de alimentação das placas. Por fim, corrosão pela salinidade de regiões praieiras. Estes fatores irão depender na região em que o semáforo está instalado e trazem riscos a integridade do sistema.

##### 5.2.4.2 Vandalismo

Estando em uma região urbanizada, os componentes eletrônicos estão suscetíveis a atos de vandalismo da sociedade, que desejam destruir o patrimônio público afetando o seu funcionamento. Destruição e obstrução das câmeras de detecção, furto de cabos elétricos, destruição dos elementos de sustentação são exemplos de atitudes que impactam negativamente o funcionamento do sistema.

Para evitar estes atos depredatórios, recomenda-se que o material utilizado para a construção de infraestrutura seja resistente, como também, posicionar os componentes eletrônicos em uma altura adequada, dificultando o acesso não autorizado. A instalação de câmeras de segurança para realizar a captura das imagens pode aumentar a resistência dos indivíduos. Por fim, também é recomendado não deixar os cabos de comunicação a mostra, para evitar que pessoas não autorizadas tenham acesso.

#### 5.2.4.3 Acidentes de Trânsito

Por se tratar de um contexto de fluxo de veículos, a ocorrência de um acidente automobilístico gera riscos à estrutura do semáforo. Eles podem deslocar, entortar e até mesmo destruir os elementos de sustentação do grupo focal, *display*, e da câmera. Além do dano físico na infraestrutura, esse tipo de ocorrência pode gerar um impacto negativo a todo o gerenciamento do fluxo em cruzamentos, aumentando os riscos de ocasionar novos acidentes pela falta do funcionamento íntegro da sinalização do semáforo. Para evitar essa situação, barreiras de proteção resistentes podem ser adicionadas em volta da estrutura, absorvendo o impacto inicial, evitando o primeiro contato com as bases de sustentação do semáforo.

#### 5.2.4.4 Funcionamento do Semáforo

O início da contabilização dos veículos ocorre quando o sistema for iniciado, por isso é recomendado que seja realizado um ciclo completo com o tempo máximo do sinal verde, para que as contabilizações sejam realizadas para o próximo ciclo e não interfira no cruzamento do trânsito.

Apesar do comportamento do condutor do veículo ser imprevisível, algumas situações podem ser previstas e tratadas no desenvolvimento do sistema. Como descrito no tópico anterior, há a chance de ocorrer um acidente em uma via, impedindo o fluxo normal dos veículos, gerando congestionamento que pode ser identificado pela câmera e tomado uma decisão de não abrir o semáforo naquela via até que o problema seja resolvido. Da mesma forma, mesmo que o fluxo da via siga uma direção definida, o condutor pode inverter o sentido do seu veículo, passando novamente pelo ponto de contagem da câmera e sendo contabilizado novamente, atitude que pode ser evitada caso a câmera armazene o veículo identificado temporariamente (por exemplo, via contorno único) e evite que haja contagens redundantes.

Caso não tenha nenhum veículo transitando no momento da detecção, o semáforo pode evitar emitir o sinal luminoso verde na via, evitando o aumento do tempo de espera dos condutores de outros fluxos. Além disso, o veículo pode cruzar o ponto de contagem da câmera depois que o sinal verde abriu, e caso ele não consiga passar pelo cruzamento, ele deverá ser contabilizado na próxima abertura do sinal verde, reduzindo erros na contagem.

Há a possibilidade de ocorrer um período que não tenha nenhum veículo transitando no cruzamento do semáforo, fazendo com que não seja contabilizado nenhum objeto para todas as vias. Nesse caso, o semáforo permanece fechado em todas as vias até que seja

identificado pelo menos um veículo em algumas delas, registrando a quantidade identificada para a próxima abertura do sinal verde.

Em situações de grande volume de veículos em todas as vias, em que a quantidade contabilizada atinge o tempo máximo do sinal verde, será emitido o maior tempo programado, ocasionando um intervalo maior de espera dos condutores de outras vias, tendo um comportamento semelhante ao de um semáforo tradicional. Além disso, a fila de carros pode exceder a distância entre o semáforo e a câmera devido ao grande número de veículos, porém eles serão contabilizados da mesma forma pela câmera quando o trânsito da fila começar a fluir.

#### 5.2.4.5 Componentes Eletrônicos

Mesmo sem nenhuma interferência externa ao sistema, os componentes eletrônicos podem falhar, ocasionando um dano significativo ao funcionamento do sistema. Por exemplo, o *display* do sinal verde pode não ser mais acionado devido a queima deste componente eletrônico, acarretando em grande confusão para os condutores da via e aumentando o risco de acidentes. Da mesma forma, a conexão entre a máquina gerenciadora e a câmera pode ser interrompida devido a uma falha no cabo de conexão, prejudicando a tomada de decisão do semáforo. Para evitar esses problemas, é fundamental adotar monitoramento contínuo de todos os componentes do semáforo, através de manutenções preventivas e até mesmo, implantar um sistema de monitoramento de *log* entre todas as conexões entre cada componente do semáforo, assim a equipe estará ciente e será avisada instantaneamente de forma automatizada.

Caso a unidade gerenciadora não receber as informações da câmera por motivos de falhas nos componentes, o semáforo entrará em modo de segurança e irá adotar o tempo máximo do sinal verde para aquela via até que o problema seja resolvido. Dessa forma, o funcionamento do semáforo não será interrompido e os condutores não serão afetados.

#### 5.2.5 Intervenção Humana para Manutenção

No tópico anterior foram apresentados alguns eventos que podem comprometer o funcionamento do sistema inteligente por diversos fatores. Embora alguns deles estejam fora do controle da equipe responsável pelo desenvolvimento do sistema, é de sua responsabilidade prestar as devidas manutenções de forma rápida e eficiente, de modo que o sistema volte a operar em pleno funcionamento. Para isso, é crucial dispor de mão de obra

adequada e capacitada, tanto na parte de infraestrutura, quanto no desenvolvimento do software, manter um estoque de peças do semáforo, desenvolver uma política de monitoramento dos *logs* do sistema para que a equipe seja acionada assim que houver algum erro inesperado contribuirá para a agilidade em que a integridade do sistema será recuperada. Além disso, é necessário ter uma política de revisão e manutenção preventiva dos componentes eletrônicos contribui para a redução das chances de ocorrer falhas de componentes, prolongando a vida útil dos equipamentos e assegurando a confiabilidade do Semáforo.

#### 5.2.6 Funcionalidades Adicionais

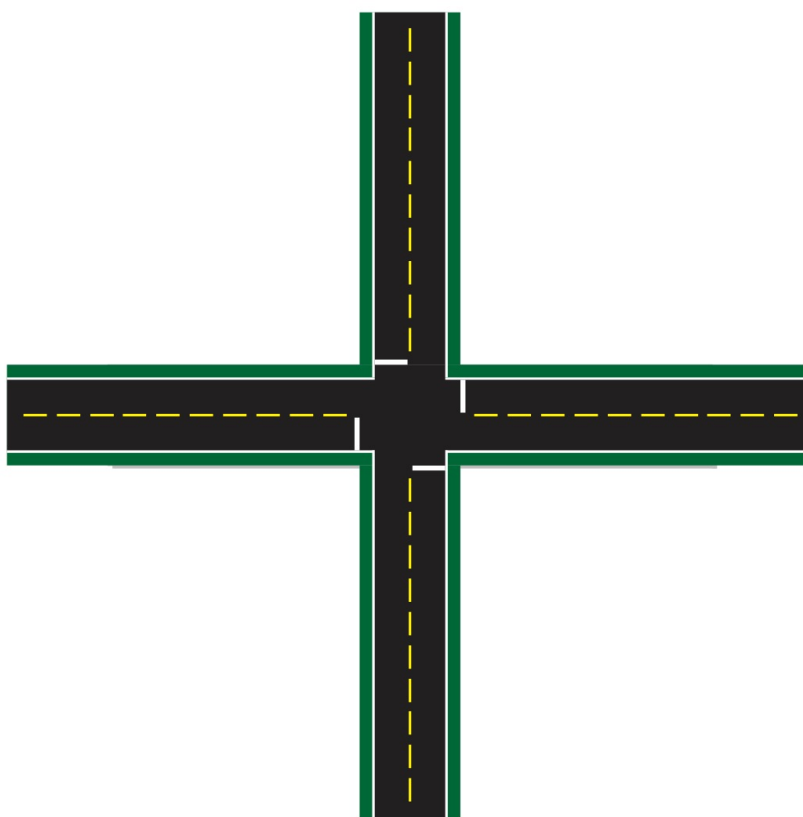
Além de gerenciar a travessia dos veículos automotores de um cruzamento de trânsito, o sistema do Semáforo Dinâmico pode ser alimentado com outras funcionalidades que podem auxiliar o funcionamento de alguns processos que utilizam da via, contribuindo mais para o desenvolvimento de uma cidade inteligente. A seguir, são apresentadas algumas ideias que podem compor o sistema.

- Gerar relatórios detalhados sobre o fluxo de veículos para órgãos competentes monitorem o tráfego, tanto em contextos específicos, quanto gerais;
- Coleta de evidências para atuações, onde câmeras adicionais instaladas no grupo focal identificam veículos e atravessam o sinal vermelho do semáforo, colocando a vida de outros condutores em risco;
- Possui fornecimento de energia em baterias que podem entrar em operação caso a alimentação da rede elétrica da cidade seja suspensa, mantendo o funcionamento do semáforo nos períodos diurnos, visto que no período noturno a identificação dos veículos pode ser impactada negativamente devido à baixa iluminação do local;
- Realizar a identificação de placas dos veículos, auxiliando órgãos de segurança na identificação e localização de veículos específicos;
- Priorizar a abertura do fluxo para veículos especiais quando identificados pela câmera, tais como: carro de bombeiros, ambulâncias e veículos policiais, agilizando seu deslocamento;
- Identifica acidentes, acionando imediatamente os órgãos responsáveis pela intervenção, dentre outros.

## 6 MAQUETE

Para demonstrar o funcionamento básico do semáforo dinâmico, foi desenvolvida uma maquete em escala reduzida, que simula um cruzamento simples de um trânsito. Uma máquina central será responsável pelo processamento dos *frames* captados pelas câmeras, além de atuar como gestor do sistema. Ela está responsável por manter a conexão entre as diferentes tecnologias utilizadas e informar ao Arduino os dados necessários para ele representar o semáforo, juntamente com os seus componentes eletrônicos.

Figura 4 – Projeto das vias da maquete



Fonte: Autoria própria (2025)

A maquete foi construída em um formato de “+” (ver figura 4), sendo possível simular um cruzamento com vias simples, comportando um semáforo para cada uma delas. Também foi utilizado *smartphones* como câmera de detecção de alta resolução, devido à qualidade da imagem que ele fornece, deixando as detecções mais estáveis. Carros em miniaturas são usados na maquete para representar os veículos de um cenário real, eles necessitam de interação humana para trafegarem pela via.

Os *frames* captados são mostrados em tempo real, onde podemos acompanhar cada etapa da detecção e contagem dos veículos. Dessa forma, temos com clareza o funcionamento do sistema, tanto no funcionamento do semáforo dinâmico quanto no funcionamento da câmera realizando as detecções.

Figura 5 – Versão final da maquete



Fonte: Autoria própria (2025)

## **7 FUNDAMENTAÇÃO TEÓRICA**

### **7.1 Back-end**

Foi construído um protótipo físico em forma de maquete, onde são apresentados os equipamentos tecnológicos necessários para a simulação do funcionamento do sistema. Para a localização dos objetos, pelo reconhecimento de suas classes e a contagem de veículos, foi

utilizado um computador *desktop*. A visão computacional em tempo real demanda um alto nível de processamento para que o reconhecimento seja realizado em uma alta taxa de atualização dos *frames*, evitando erros na contagem dos veículos.

As especificações do computador são:

- Intel(R) Core (TM) i5-14600KF de 3.5 GHz;
- Memória RAM de 32GB;
- Placa de Vídeo NVIDIA GeForce RTX 4060 Ti;
- Sistema Operacional Windows 11 Home de 64 *bits*.

O sistema de reconhecimento de objetos foi desenvolvido na linguagem de programação *Python* (versão 3.13.2), que é executada em segundo plano (*back-end*). A linguagem supracitada foi escolhida devido ela possuir uma sintaxe legível, o que facilita a codificação e a manutenção de códigos complexos, além de tornar o desenvolvimento do *Script* mais ágil e de possuir integração com as bibliotecas necessárias para realização do processo da visão computacional.

Para a representação física do semáforo no protótipo, foi utilizada a plataforma Arduino Mega, como sendo o microcontrolador do sistema. A escolha se deve, principalmente, pela possibilidade de conexão com os componentes eletrônicos necessários para a simulação através de seu gerenciamento de sinais de entrada e saída. Os componentes eletrônicos utilizados serão:

- Plataforma Arduino - modelo Mega 2560;
- Módulo de Semáforo;
- Jumper;
- Display de 7 Segmentos;
- Protoboard;
- Cabo USB padrão A/B.

O controle dos periféricos foi modelado pela ferramenta gratuita de código aberto Arduino IDE, que oferece uma interface simplificada na qual é possível desenvolver e compilar códigos nas linguagens C/C++, além de contar com uma biblioteca padrão para acessar facilmente as funcionalidades da placa, como portas de entradas e saídas, *timers*, comunicação serial, controle de LEDs e controle de *display* de 7 segmentos.

Para a representação dos veículos automotores na simulação, foram utilizados automóveis em miniatura, devido a necessidade de proporcionalidade com o tamanho do produto. Eles são manipulados pela via de forma manual, para que eles sejam reconhecidos e contados pelo *Script* implementado.

## 7.2 Programação Paralela

Segundo Paiva (s.d), a Programação Paralela é a realização da divisão de uma aplicação em partes, de maneira que essas partes possam ser executadas simultaneamente. Essa metodologia de desenvolvimento permite que a execução de diferentes tarefas independentes seja realizada simultaneamente por um mesmo computador, sem que seja necessária outra arquitetura.

Um dos principais objetivos da programação paralela é melhorar o tempo de execução e eficiência energética de uma máquina ao executar um script paralelizado devido à divisão de carga de trabalho entre os nós de uma CPU (*Central Processing Unit*). Essa capacidade de alto desempenho provém da necessidade atual de um alto poder de processamento das aplicações atuais.

É importante destacar a necessidade de um hardware compatível com a programação paralela, ou seja, que sua unidade central de processamento possua mais de um núcleo disponível para a utilização. Dessa forma, o programador possui a responsabilidade de administrar o uso de cada núcleo da arquitetura, mantendo a integridade do código, buscando o maior desempenho e evitando as condições de corrida.

Portanto, como cada webcam irá realizar a captura dos veículos que trafegam cada via do cruzamento de forma independente, onde o processamento será unificado em somente uma CPU, será utilizada a programação paralela para gerenciar os scripts de detecção de veículos.

## 7.3 Visão Computacional

Visão computacional é a subárea da IA (Inteligência Artificial) e é utilizada por máquinas para o entendimento e reconhecimento dos diversos objetos do mundo real, simulando o funcionamento do sistema visual humano em nível de hardware e software.

Essa tecnologia tem se tornado cada vez mais presente em diversas áreas, trazendo avanços significativos. Segundo a IBM (s.d), aplicações do mundo real demonstram sua importância para empreendimentos nos negócios, entretenimento, transporte, saúde e vida cotidiana.

Para que a visão computacional seja implementada de forma que auxilie e coopere em nosso cotidiano, é necessária a passagem por um treinamento para que seja possível identificar um objeto, podendo utilizar métodos mais sofisticados, como o *Machine Learning* ou métodos mais tradicionais, baseadas em técnicas clássicas de processamento de imagens, tais como: a segmentação de bordas, segmentação de cores e detecção de formas geométricas.

Essa tecnologia desempenha um importante papel neste projeto, sendo responsável pela visualização, identificação e contagem dos veículos que trafegam determinada pela via analisada através de uma câmera, que será a fonte da captura em tempo real.

#### 7.4 Machine Learning

Segundo a *Google Cloud* (s.d), o *Machine Learning* (ML) é uma técnica de aprendizado de um sistema de forma autônoma ao ser alimentado com uma grande quantidade de dados. Esse aprendizado é realizado utilizando redes neurais ou *Deep Learning* (aprendizado profundo) podendo a máquina seja programada para realizar esta atividade dependendo da estrutura da rede neural utilizada. Essa capacidade de aprender ocorre de maneira contínua e progressiva, à medida que novos dados são processados e integrados ao sistema.

Segundo a Alura (s.d), essa capacidade de aprendizado se deve graças à grande quantidade de dados que possuímos hoje em dia. Eles podem estar em diferentes formatos e fontes, como números, textos, imagens, vídeos e áudios. Assim, quanto maior e mais variado o conjunto de informações, mais preciso será o aprendizado.

De acordo com Alura (s.d), o principal objetivo do ML é buscar detectar vários padrões que estão presentes em um determinado conjunto de dados. Conforme citado pela Oracle (s.d), esse processo pode ser abordado por meio de duas abordagens: o *Machine Learning* Supervisionado e Não Supervisionado. No aprendizado supervisionado, o sistema é treinado por um conjunto de dados que já está rotulado e tem uma saída predefinida, ou seja, a resposta final já é conhecida. Já no aprendizado não supervisionado o computador aprende a identificar processos e padrões complexos sem que um ser humano forneça uma orientação próxima e constante.

Atualmente, a sociedade está em contato direto com o *Machine Learning*, tanto de forma direta quanto indireta. Segundo a *Google Cloud* (s.d), o ML é amplamente utilizado em empresas com o objetivo de aumentar a produtividade em setores como logística, varejo e fabricantes. Diversas atividades podem se beneficiar por meio desta tecnologia, tais como:

detecção de fraudes, identificação de ameaças à segurança, personalização e recomendações e atendimento ao cliente automatizado por meio de *chatbots* são exemplos de funcionalidades que podem ser facilitadas.

## 7.5 Rede Neural

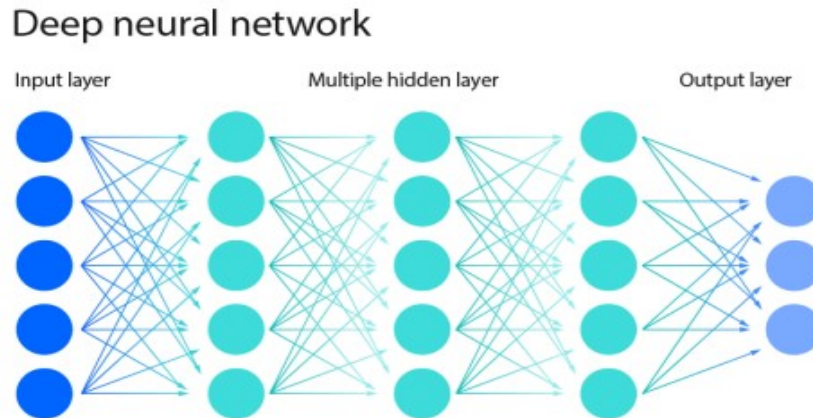
Com o objetivo de simular o funcionamento dos neurônios do cérebro humano em máquinas computacionais, em 1943 foi publicada a primeira pesquisa intitulada “*A Logical Calculus of the Ideas Immanent in Nervous Activity*” que procurou compreender como o cérebro humano poderia produzir padrões complexos através dos neurônios. Essa área do aprendizado de máquina ganhou o nome de Redes Neurais Artificiais e tornou-se cada vez mais estudada pela comunidade científica, o que resultou em diversos aprimoramentos que contribuíram para o seu avanço.

Como aponta a AWS (s.d), uma rede neural ensina computadores a processar dados através de nós ou neurônios interconectados em uma estrutura de camadas, lembrando a arquitetura de um cérebro humano. Com um sistema adaptado para se aprimorar continuamente através do aprendizado com os erros cometidos, as redes neurais artificiais tentam solucionar problemas complicados, como resumir documentos ou reconhecer rostos com grande precisão.

Uma das formas de categorizar uma rede neural é pelo seu número de camadas existentes. De maneira geral, toda rede neural possui uma camada de entrada, que é responsável pela recepção dos dados, e outra camada de saída, possuindo a função de entregar o resultado processado. Entre elas, pode haver uma ou mais camadas ocultas, onde efetivamente ocorre o processamento das informações. Uma rede neural simples é aquela que possui somente uma camada oculta, enquanto uma rede neural profunda possui duas ou mais camadas ocultas, conforme exemplificado na Figura 6.

Há diferentes tipos de rede neural que possuem como parâmetro de sua categorização a forma de lidar com os dados durante o seu treinamento ou funcionamento. O algoritmo de detecção de objetos que foi realizado para a identificação e a contagem dos veículos que trafegam na via do cruzamento do semáforo dinâmico utiliza o tipo chamado Rede Neural Convolutiva (CNN - *Convolutional Neural Network*). A CNN é um algoritmo de Aprendizado Profundo que pode captar uma imagem de entrada, atribuir importância (pesos e vieses que podem ser aprendidos) a vários aspectos e/ou objetos da imagem e ser capaz de diferenciar um do outro.

Figura 6 – Rede neural profunda



Fonte: IBM (s.d)

Conforme a AWS (s.d), o treinamento de uma rede neural baseia-se no processamento inicial de vários conjuntos com grandes volumes de dados rotulados ou não rotulados. Dados rotulados são aqueles em que cada exemplo possui uma etiqueta ou rótulo associado direcionando o treinamento para o aprendizado supervisionado. Em contrapartida, dados não rotulados contêm apenas as características ou informações brutas e são frequentemente usados em técnicas de aprendizado não supervisionado, onde o objetivo é identificar padrões, agrupamentos ou estruturas intrínsecas dos dados sem a orientação de rótulos pré-definidos. Essa abordagem também é conhecida como *Deep Learning*.

Concluindo, as redes neurais lidam com tarefas complexas que envolvem grandes quantidades de dados, permitindo que soluções avançadas sejam desenvolvidas para problemas de classificação, detecção de objetos, previsão e geração de conteúdo. Devido à flexibilidade de aprendizado que ela proporciona diversas áreas se beneficiam para utilizá-la para impulsionar diversas pesquisas, como visão computacional, sistemas de recomendação, saúde, entre outros. Essa interação mudou a forma que interagimos com a tecnologia, simplificando e otimizando diversas atividades cotidianas, além de abrir novos horizontes para futuras inovações.

## 7.6 YOLO

De acordo com o Alves (2020), o YOLO (*You Only Look Once*) é um método de detecção de objetos que utiliza uma rede neural convolucional para realizar a detecção dos objetos em imagens e vídeos. Essa arquitetura ganhou destaque devido aos seus marcos

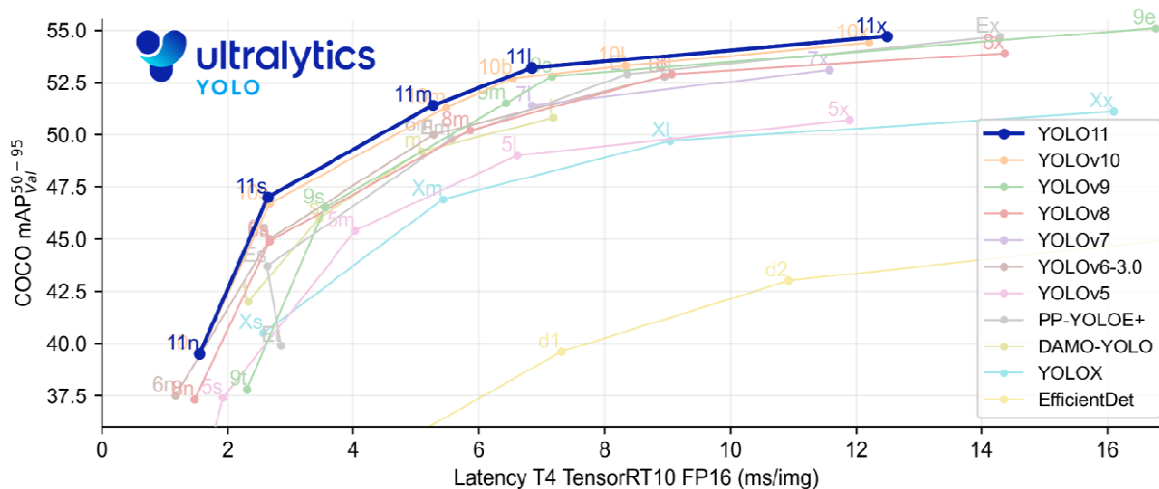
alcançados em relação a outras metodologias, tendo uma velocidade de processamento superior sem possuir uma defasagem na acurácia da detecção.

Há uma diferença entre a detecção e classificação de objetos. De acordo com Alves (2020), localizar objetos significa identificar a presença de itens em uma imagem, sem determinar a que classe eles pertencem. Em contrapartida, a detecção de objetos envolve tanto a localização, quanto a identificação da classe a que aquele objeto pertence.

De acordo com o Kundu (2023), os algoritmos de detecção de objetos podem ser classificados em duas modalidades: detectores de disparo único e detectores de dois estágios, que se diferem pela quantidade de vezes que o algoritmo processa pelas imagens para realizar a detecção. O YOLO é um exemplo da metodologia de detecção de objetos de disparo único, realizando uma única passagem na imagem de entrada para fazer a detecção de objetos na imagem. Entretanto, metodologias como R-CNN (*Region Based Convolutional Neural Networks*) utilizam a de dois estágios, onde a primeira etapa é utilizada para gerar propostas de localidades e a segunda etapa serve para refinar essas propostas e fazer previsões finais.

O YOLO teve sua primeira versão lançada em 2015 e desde então, diferentes autores lançaram versões que apresentaram melhorias nesta metodologia de detecção, aumentando tanto a precisão quanto a velocidade dos modelos para aplicações em tempo real. A Figura 7 apresenta a comparação de desempenho entre diferentes metodologias de detecção de imagens.

Figura 7 – Comparação de desempenho entre diferentes metodologias de detecção



Fonte: Ultralytics (s.d)

Na Figura 7 é apresentada a relação entre a precisão média (mAP) no conjunto COCO representada no eixo vertical, e a latência de inferência (em milissegundos) demonstrada no eixo horizontal. Cada linha do gráfico representa uma família diferente de detecção de objetos, incluindo as versões mais recentes do YOLO e também o *EfficientDet*. Os pontos em cada linha representam variantes específicas dessas famílias, que são diferentes configurações da Rede Neural que impactam na precisão e qualidade da detecção que influencia diretamente na relação entre velocidade e precisão. Como podemos observar, quanto maior a precisão da detecção do modelo será necessário mais tempo para o processamento.

Atualmente temos diferentes versões do YOLO com características peculiares que impactam diretamente nas métricas de desempenho entre elas. Contudo, essa arquitetura segue uma estrutura de funcionamento padrão, mantendo sua essência. Ao receber uma imagem de entrada, o algoritmo redimensiona-se para seguir a padronização da rede neural para que posteriormente ela seja enviada para a rede neural e seja dividida em células com quantidades que podem variar em cada versão do YOLO. Segundo a Visão Computacional (s.d), cada célula ficará responsável por criar caixas delimitadoras, mesmo que não seja localizado nenhum objeto, onde é atribuído um valor de confiança para a existência de um objeto que varia entre 0 e 1 para cada uma delas. Além da tentativa de localização dos objetos, é realizada uma predição para identificar a qual classe ele pertence. O valor de confiança para a caixa delimitadora e a predição da classe é combinado em uma pontuação final, que informa a probabilidade dessa caixa conter um objeto específico. Após este procedimento realizado pela rede neural, será feita uma filtragem dessas caixas de acordo com um valor de referência chamado “*score threshold*”, que é uma referência definida pelo programador, que irá determinar o descarte da caixa delimitadora de acordo com a pontuação final dela. Por fim, é aplicada uma etapa chamada NMS (*Non Maximum Suppression*), que é responsável por identificar e remover caixas delimitadoras redundantes ou incorretas e gerar uma única caixa delimitadora para cada objeto na imagem.

Segundo a Visão Computacional (s.d), o YOLO utiliza uma rede neural profunda (DNN - *Deep Neural Network*), uma rede neural convolucional como base para a extração de características e detecção de objetos. Até a quarta versão, o *framework* utilizado foi o *Darknet*, desenvolvido em C e CUDA que implementa as Redes Neurais convolucionais. A partir do YOLO v5, o principal *framework* utilizado passou a ser o *PyTorch*, tendo também o *TensorFlow* utilizado com menor frequência.

Em suma, o YOLO é um dos algoritmos de detecção de objetos mais utilizados na área de visão computacional, devido a sua eficiência de detecção e velocidade de processamento, tornando a sua escolha ideal para aplicações que utilizam monitoramento em tempo real, como sistemas de segurança, veículos autônomos e Semáforos Dinâmicos.

## 7.7 Dataset COCO

O *dataset* MS COCO (*Microsoft Common Objects in Context*) é um conjunto de dados gratuito para visão computacional, utilizados em larga escala em projetos de detecção de imagem. Segundo a COCO (s.d), esse *dataset* possui anotações para 80 classes de objetos e cerca de 330.000 imagens disponíveis. De acordo com a OpenCV (2021), o COCO é amplamente reconhecido por Redes Neurais de última geração, podendo atuar em aplicações com detecção de objetos, legendagem de imagens, segmentação de instâncias e detecção de pontos chave.

O MS COCO é utilizado como base de dados para a realização de treinamentos da arquitetura de detecção de imagem YOLO. A equipe que gerencia esta metodologia disponibiliza no repositório oficial do projeto os pesos em um arquivo com a extensão **.weights** que contém os parâmetros de redes neurais que foram treinadas utilizando este *dataset*. É possível realizar o treinamento desses pesos do zero, porém, este arquivo é disponibilizado por conveniência na rede de forma gratuita, facilitando o trabalho do desenvolvedor e contribuindo para a comunidade da visão computacional.

## 7.8 OpenCV

O OpenCV (*Open Source Computer Vision Library*) é uma biblioteca de visão computacional e aprendizado de máquina de código aberto utilizada em sistemas por diversas empresas. Segundo a OpenCV (s.d), empresas como Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda e Toyota empregam a biblioteca, devido à abrangência de funcionalidades existentes.

De acordo com a OpenCV, “Os usos implantados do OpenCV abrangem desde a costura de imagens do *streetview*, detecção de intrusões em vídeo de vigilância em Israel, monitoramento de equipamentos de mina na China, auxílio a robôs para navegar e pegar objetos na Willow Garage, detecção de acidentes de afogamento em piscinas na Europa, execução de arte interativa na Espanha e Nova York, verificação de pistas de pouso em busca de detritos na Turquia, inspeção de rótulos em produtos em fábricas ao redor do mundo até detecção rápida de rostos no Japão.”

Segundo Boesch (2024), o objetivo do OpenCV é fornecer uma infraestrutura de visão computacional simplificada, ajudando as pessoas a construir rapidamente aplicativos de visão sofisticados. Assim, esta biblioteca fornece diversos scripts programados para facilitar no desenvolvimento de sistemas que trabalham com a visão computacional.

Com esta biblioteca será possível identificar e marcar os veículos reconhecidos com caixas delimitadoras, além de realizar sua contagem através do cruzamento das caixas por uma linha previamente definida no vídeo em uma localização planejada e permitir utilizar as câmeras do sistema.

A biblioteca possui um módulo DNN (*Deep Neural Network*), que permite carregar e executar modelos de *Deep Learning* sem precisar de outra biblioteca para realizar este processo. Contudo, é necessária a passagem do peso treinado, ou seja, um arquivo contendo as informações dos parâmetros aprendidos pela Rede Neural na hora do treinamento dela para a realização da identificação dos objetos. Outro parâmetro necessário é o modelo de detecção utilizado, como o Yolo v8, por exemplo.

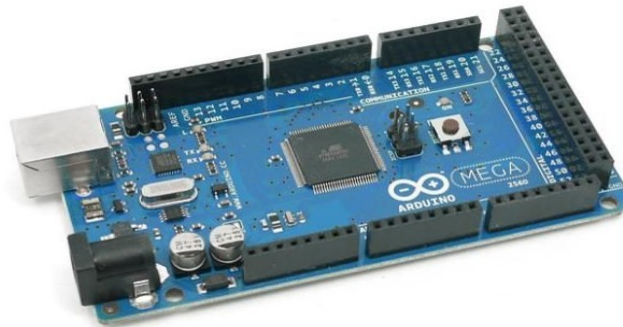
Segundo Boesch (2024), O OpenCV oferece suporte à aceleração NVIDIA CUDA e GPU desde 2011 para o aprimoramento da performance na hora de realizar a detecção, este desempenho é necessário principalmente quando se trata de detecção de vídeo em tempo real.

## 7.9 Arduíno Mega

De acordo com a empresa Arduino (2025), o Arduino é uma plataforma eletrônica baseada em hardware e software e esteve presente em diversos projetos, de objetos cotidianos a instrumentos científicos complexos. Segundo Babos (2025), esta ferramenta de prototipagem vem conquistando cada vez mais adeptos devido a sua simplicidade, podendo atuar em fábricas, no âmbito doméstico e em aulas de Introdução à Eletrônica, Introdução à Robótica, Automação, Sistemas Embarcados, Internet das Coisas dentre outras. Além disso, a plataforma se destaca por ter um custo acessível, o que contribui ainda mais para a sua popularização.

Atualmente, existem no mercado diversos modelos de placas Arduino, cada uma contendo diferenças arquiteturais que se adaptam melhor para cada demanda de um projeto. Neste trabalho, optou-se pelo modelo Arduino Mega 2560 em razão da sua maior quantidade de pinos digitais, responsáveis pelo controle dos componentes eletrônicos. Segundo Babos (2025), esse modelo conta com o processador 16MHz ATmega2560, 8KB SRAM e 256 KB *flash* e 54 pinos digitais (ver Figura 8).

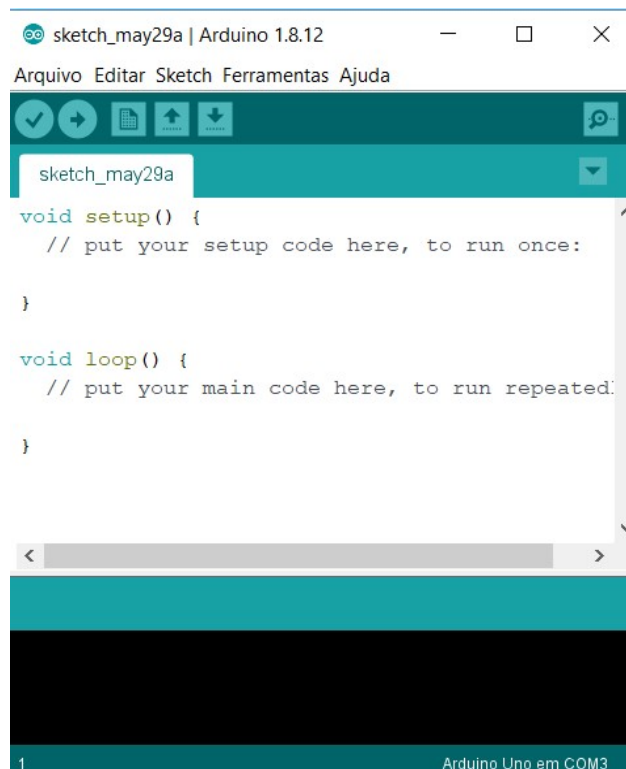
Figura 8 – Plataforma Arduino - modelo Mega 2560



Fonte: Souza (2014)

O ambiente de programação para o Arduino é chamado de Arduino IDE, que é baseado nas linguagens C/C++. Segundo a empresa Arduino (2024), “O *Arduino Integrated Development Environment* - ou Arduino Software (IDE) - conecta-se às placas Arduino para carregar/embarcar programas e se comunicar com elas.”. Os programas criados nesta aplicação são chamados de *Sketches* e são salvos em arquivos com a extensão **.ino** (ver Figura 9).

Figura 9 – IDE Arduino



Fonte: Babos (2025)

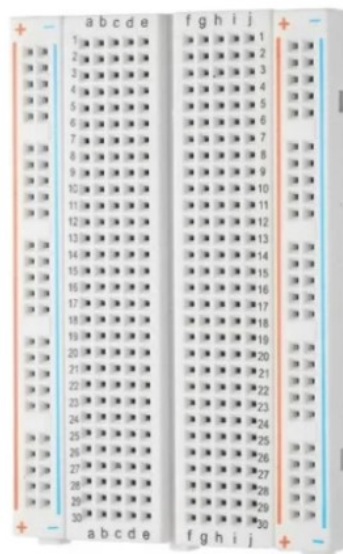
## 7.10 Componentes Eletrônicos do Arduino

O Arduino Mega, usado neste projeto, faz o controle dos componentes eletrônicos para receber a quantidade de veículos contados e realizar a simulação do funcionamento dos sinais luminosos e a contagem em ordem decrescente do semáforo.

### 7.10.1 Protoboard

A *Protoboard* é um equipamento que permite ao usuário simular circuitos eletrônicos com os componentes do Arduino sem a necessidade de solda (ver Figura 10). Segundo Costa (s.d), ela é formada por vários pontos elétricos interligados, permitindo que os componentes eletrônicos façam conexões elétricas entre si. Esse componente é muito utilizado por profissionais e entusiastas da eletrônica, devido à facilidade que ele oferece para a montagem dos protótipos.

Figura 10 – Protoboard



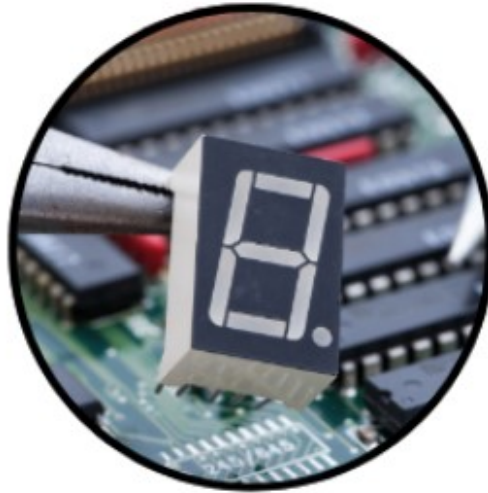
Fonte: Nunes (2023)

### 7.10.2 Display de Sete Segmentos

O *Display* de sete segmentos de catodo comum é formado por sete LEDs independentes, onde cada um pode ser aceso ou apagado de maneira distinta. Sendo assim, a demonstração de informação para o usuário é limitada somente a números inteiros. Esse componente será responsável por representar a contagem dos segundos faltantes para o sinal

luminoso verde do semáforo apagar. A Figura 11 apresenta o modelo de um *display* de sete segmentos.

Figura 11 – Display de sete segmentos



Fonte: Rodrigues (2024)

### 7.10.3 Módulo Semáforo

Com o objetivo de representar os LEDs do semáforo nas cores vermelho, amarelo e verde dispostos no sentido vertical, o módulo semáforo possibilita diferentes aplicações em projetos que envolvam a locomoção urbana (ver Figura 12). Neste projeto, ele será utilizado para a disposição dos sinais luminosos, simulando o funcionamento de um controlador de tráfego em um cruzamento.

Figura 12 – Módulo semáforo



Fonte: Silva (2020)

#### 7.10.4 Jumper

Os *Jumpers* são fios condutores utilizados na prototipagem em eletrônica e estabelece conexão entre Placas Microcontroladoras (Arduinos), *Protoboards*, sensores, atuadores, dentre outros componentes. Os *jumpers* são comumente utilizados na fase de prototipagem para a realização de testes de circuitos, eles costumam ser produzidos de cobre revestido de plástico e possuem pinos de conexão machos, fêmea ou ambos. Neste projeto, este componente terá a responsabilidade de realizar a conexão da plataforma Arduino Mega com a *Protoboard* para o controle do Módulo Semáforo e do *Display* de Sete Segmentos. A Figura 13 apresenta exemplos de *jumpers*.

Figura 13 – Jumper



Fonte: Isaac (s.d)

#### 7.10.5 USB padrão A/B

O cabo USB (ver Figuras 14 e 15), composto por uma porta do tipo A e outra do tipo B, será responsável por conectar a plataforma Arduino Mega ao computador para a integração com o software IDE Arduino, e ao mesmo tempo, a transmissão de dados referente a quantidade de veículos contabilizados para a placa micro controladora.

Figura 14 – Cabo USB-A



Fonte: Samsung (s.d)

Figura 15 – Cabo USB-B



Fonte: Samsung (s.d)

### 7.11 Python

Segundo a empresa Python (2025), o Python é uma linguagem de programação interpretada, interativa e orientada a objetos e pode ser executado em várias distribuições do sistema operacional Unix, incluindo o Linux e Mac, além do sistema operacional Windows. Segundo Alura (2024), o Python se tornou uma das linguagens de programação mais populares do mundo graças a sua versatilidade, praticidade e agilidade, tendo como as principais áreas de atuação o aprendizado de máquinas, desenvolvimento *web*, automação de tarefas, análise de dados e *DevOps*.

O Python é considerado uma linguagem de programação de alto nível, possuindo uma sintaxe simples e clara, facilitando o aprendizado de desenvolvedores, além de ser multiparadigma, ou seja, proporciona uma programação em vários paradigmas, tais como o procedural, funcional e orientação a objetos, conforme Alura (2024).

Esta linguagem será utilizada na etapa de visão computacional deste projeto, onde serão aplicadas técnicas de reconhecimento de objetos e contagem de veículos. Em seguida, será realizado o envio desta informação para a plataforma Arduino Mega realizar a simulação do funcionamento do semáforo dinâmico.

### 7.12 C++

Para a execução das *Sketches*, é utilizada a linguagem de programação C++ com algumas adaptações para atender as funcionalidades que a plataforma Arduino Mega oferece. Segundo a Clemente (2024), esta linguagem foi desenvolvida por *Bjarne Stroustrup* e trouxe uma adaptação da orientação a objetos para a linguagem de programação C e é amplamente utilizada no desenvolvimento de softwares onde o desempenho é essencial, como aplicativos gráficos, jogos e sistemas em tempo real.

Apesar de ser uma evolução de sua antecessora, a linguagem C++ carrega consigo as características da linguagem C. Segundo Clemente (2024), esta linguagem de programação procedural foi desenvolvida nos anos 70 por Dennis Ritchie e se destacou por sua eficiência e controle sobre recursos de hardware e memória.

### 7.13 Visual Studio Code

Segundo o Hanashiro (2021), a ferramenta Visual Studio Code (VS Code) é um editor de código aberto desenvolvido pela Microsoft que está disponível nos sistemas operacionais Windows, Mac e Linux. Amplamente utilizado pela comunidade de desenvolvedores, ele possibilita a utilização de diversas extensões que auxiliam na produtividade do desenvolvimento e manutenção de códigos. O VS Code vem com suporte interno para JavaScript, TypeScript e Node.js e com um ecossistema avançado de extensões para outras linguagens e *runtimes* (como C++, C#, Java, Python, PHP, Go, .NET).

### 7.14 Metodologia Ágil Scrum

Segundo Drumond (s.d), o Scrum é uma estrutura de gerenciamento ágil de projetos que ajuda as equipes a elaborarem e gerenciarem o trabalho delas por meio de um conjunto de valores, princípios e práticas. Embora haja diversas metodologias ágeis, o Scrum se tornou a mais utilizada nos últimos tempos em razão dela ser aplicada não somente em projetos de desenvolvimento de softwares, como também em todos os tipos de trabalho em equipe.

O Scrum foi desenvolvido antes da publicação do Manifesto Ágil, que ocorreu em 2001, contudo os dois estão profundamente conectados. Segundo Drumond (s.d), o *framework* é uma estrutura para realizar o trabalho, enquanto a Metodologia Ágil é uma filosofia. Sendo assim, O manifesto Ágil é uma declaração de valores e princípios essenciais para o desenvolvimento de software, apesar de ser utilizado em outros setores de desenvolvimento de projetos. Esses valores e princípios são encontrados em diferentes *frameworks* de gerenciamento e desenvolvimento de projetos, tais como o *Extreme Programming*, *Crystal Clear* e Scrum.

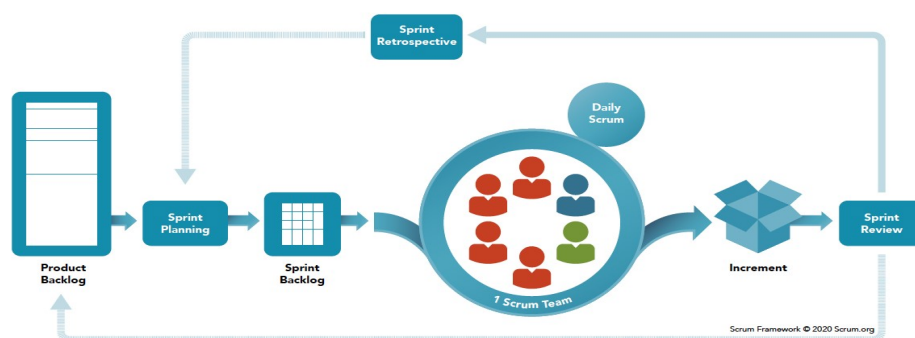
Existe a documentação chamada de Artefatos do Scrum. Segundo Drumond (s.d), são informações importantes utilizadas pela equipe que ajudam a definir o produto e o trabalho a ser feito para criar o produto. Ela é separada em três constantes denominadas Backlog do Produto, Backlog do Sprint e Incremento.

O Product Backlog (Backlog do Produto) é a lista de atividades que devem ser realizadas e que são gerenciadas pelo proprietário do produto. Além disso, o Backlog da Sprint é a lista de realizações que deverá ser concluída na Sprint, que são curtos períodos de realização de atividades e possuem uma vida útil de poucos dias. Essas atividades são escolhidas pela equipe de desenvolvimento antes do início de cada Sprint por meio de uma reunião. Por fim, o Incremento são as atividades concluídas da Sprint pela equipe de desenvolvimento do projeto.

Segundo Drumond (s.d), a estrutura do Scrum inclui práticas, cerimônias e reuniões de Scrum que as equipes fazem com regularidade. Esses eventos servem para alinhamento das atividades entre os integrantes da equipe. Conforme mencionado anteriormente, as atividades são organizadas em curtos ciclos denominados Sprints. Cada Sprint começa com o Planejamento da Sprint e é finalizada com cerimônias de revisão e inspeção, chamadas de Revisão e Retrospectiva da Sprint. Esses eventos possuem o objetivo de planejar, entregar e analisar o trabalho realizado em um pequeno período de tempo (ver Figura 16).

Figura 16 – Metodologia Scrum

## SCRUM FRAMEWORK



Scrum.org

Fonte: Cardoso (2022)

A equipe do projeto que utiliza a metodologia Scrum possui poucas divisões de atuação e normalmente é composta por um pequeno número de colaboradores. Essa divisão consta de três papéis fundamentais, sendo elas o Product Owner, o Scrum master e o time de desenvolvimento.

Segundo o Ministério da Educação (2022), Scrum Master é o responsável que domina os processos e as regras contidas na Metodologia Ágil e tem como missão principal garantir seu cumprimento por todos. É dever desta função possuir o conhecimento teórico e prático para orientar a equipe dentro da linha de raciocínio do framework de gerenciamento Scrum.

O time de desenvolvimento é o responsável pelo desenvolvimento do produto. É composto por desenvolvedores que não possuem uma função específica e tem diferentes conjuntos de habilidades que contribuem para a realização do planejamento de cada etapa do desenvolvimento do projeto.

Segundo o Ministério da Educação (2022), O Product Owner atua como a voz do cliente para orientar as equipes de desenvolvimento na criação de recursos de software. Ele é o responsável pelo conhecimento da regra de negócio do produto, gerenciar o Product Backlog, orientar a melhoria contínua, estar em contato diretamente com o cliente e representar as necessidades e prioridades do projeto (ver Figura 17).

Figura 17 – Papéis do Scrum



Fonte: Souza (s.d)

Essa estrutura visa o cumprimento dos valores incluídos no Guia do Scrum. Incluído no ano de 2016, eles visam orientar o trabalho, as ações e o comportamento de cada integrante dentro da equipe do Scrum. Segundo o Ministério da Educação (2022), esses valores são compostos por comprometimento, coragem, foco, abertura e respeito, e são fundamentais para fornecer clareza na construção da identidade da equipe, para garantir a melhoria contínua e evitar falhas no produto.

### 7.15 Cidades Inteligentes

Segundo o Ministério das Cidades (s.d), cidade inteligente é uma cidade que se compromete com o desenvolvimento urbano e a transformação digital de forma sustentável em aspectos econômicos, ambientais e socioculturais. A utilização da tecnologia para a solução de problemas ou melhorias de processos já estabelecidos é um fator importante dentro da área de cidades inteligentes, contribuindo para o desenvolvimento dela de forma sustentável e inovador. A implementação de um semáforo dinâmico encaixa neste conceito, pois contribui para o desenvolvimento urbano de uma forma tecnológica e inovadora, melhorando o fluxo do tráfego dos veículos, reduzindo o tempo de espera dos condutores e diminuindo a emissão de gases poluentes dos motores dos veículos.

## 8. OBJETIVOS DE DESENVOLVIMENTO SUSTENTÁVEL

De acordo com o Tribunal de Justiça do Estado do Paraná (s.d), o conceito de desenvolvimento sustentável foi introduzido pela Organização das Nações Unidas (ONU) no ano de 1987 através da Comissão Mundial sobre Meio Ambiente e Desenvolvimento, presidida pela ex Primeira Ministra da Noruega, Gro Harlem Brundtland.

Foi então que em 2015 foram desenvolvidos os Objetivos de Desenvolvimento Sustentável (ODS) pela ONU, que é um documento oficial que compõem 17 objetivos e 169 metas que devem ser implementadas por todos os países do mundo até o ano de 2030 e por este motivo, estes objetivos ficou conhecido como Agenda 2030. O compromisso internacional foi assumido pela República Federativa do Brasil com a Agenda 2030, firmado pela Resolução A/Resolução 70/1, de 25 de setembro de 2015, da Assembleia Geral das Nações Unidas.

Publicada no site da ONU, cada objetivo (ver Figura 18) abrange uma área da sociedade que contribui com a harmonia entre o desenvolvimento econômico e a proteção do meio ambiente:

1. Erradicação da pobreza;
2. Fome zero e agricultura sustentável;
3. Saúde e bem-estar;
4. Educação de qualidade;
5. Igualdade de gênero;
6. Água limpa e saneamento;

7. Energia limpa e acessível;
8. Trabalho decente e crescimento econômico;
9. Indústria, inovação e infraestrutura;
10. Redução das desigualdades;
11. Cidades e comunidades sustentáveis;
12. Consumo e produção responsáveis;
13. Ação contra a mudança global do clima;
14. Vida na água;
15. Vida terrestre;
16. Paz, justiça e instituições eficazes;
17. Parcerias e meios de implementação.

Figura 18 – Os 17 objetivos da ODS



Fonte: Nações Unidas Brasil (2025)

Este projeto possui colaboração parcial nas metas deste acordo, cumprindo com os objetivos:

- Número 3 (três), ajudando na agilidade dos veículos especiais de saúde, reduzindo o tempo de espera em cruzamentos semaforicos em situações de emergência;
- Número 7 (sete), onde o abastecimento energético das baterias que alimentam o sistema é realizado de forma limpa e sustentável;

- Número 8 (oito), tendo o auxílio no crescimento econômico através da melhora da agilidade dos meios de transporte que impactam diretamente o trabalho descente e crescimento econômico e utilizam veículos para atenderem as demandas de transporte de cargas e mercadorias;
- Número 9 (nove), trazendo inovação e melhoria para um processo já existente;
- Número 11 (onze) e 13 (treze), contribuindo em longo prazo com a redução de gás poluente que é emitido pela combustão dos veículos, pois o condutor irá diminuir o tempo de espera com o veículo ligado.

Portanto, este projeto contribui com os Objetivos de Desenvolvimento Sustentável da ONU através da inovação tecnológica, que promove impactos sociais, ambientes e econômicos positivos. O projeto de Cidades Inteligentes possui uma grande importância na colaboração deste acordo, trazendo grandes oportunidades de criação de novos processos, como também, a melhoria de procedimentos já existentes, melhorado significativamente o estilo de vida da sociedade e contribuindo com o desenvolvimento econômico atrelado ao desenvolvimento sustentável. Dessa forma, fica clara a importância da união da tecnologia, sustentabilidade e bem-estar coletivo, direcionando o Estado ao compromisso com a Agenda 2030.

## **9 METODOLOGIA DE DESENVOLVIMENTO**

Este projeto será desenvolvido por somente uma pessoa e seguirá a metodologia ágil Scrum para desenvolvê-lo, devido ela proporcionar um ciclo de entregas curto e frequente, tornando a visualização do progresso de forma clara e contínua. No Quadro 1 estão distribuídas as funções estabelecidas para este projeto (*Scrum Master*, *Product Owner* e *Desenvolvedor*), adaptando-as para que todas sejam desempenhadas pelo mesmo indivíduo. Planejar e priorizar as atividades, entender a regra de negócio, aplicar a metodologia adaptada e executar o desenvolvimento das etapas ficará a cargo do responsável do projeto. Essa abordagem, embora fuja do modelo tradicional de equipes, mantém os princípios do *Scrum*, garantindo adaptação e melhoria contínua ao longo de todo o ciclo de desenvolvimento.

Quadro 1 – Divisão da equipe

<b>ID</b>	<b>Nome</b>	<b>Função</b>	<b>Atribuições</b>
01	Murilo Cambuzzi Paiva	<i>Scrum Master</i>	Liderar a equipe durante o andamento do projeto.
02	Murilo Cambuzzi Paiva	<i>Desenvolvedor</i>	Desenvolver o projeto de software.
03	Murilo Cambuzzi Paiva	<i>Product Owner</i>	Entender e esclarecer a regra de negócio.

Esta forma de gerenciamento para este projeto mantém os principais valores e rituais do *Scrum*, mas de forma com que haja uma adaptação para garantir a flexibilidade necessária para atender as demandas de somente uma pessoa. O projeto será dividido em pequenos incrementos (*Sprints*), o que ajuda a manter o foco em metas específicas, além de proporcionar uma documentação detalhada do progresso, alteração e melhoria de cada etapa.

Para organizar o desenvolvimento de cada requisito do sistema, será utilizada a ferramenta *Jira*, desenvolvida pela empresa Atlassian, que é uma das plataformas mais populares no cenário de desenvolvimento ágil. O *Jira* permite a criação do artefato *Product Backlog*, a identificação dos processos em andamento e a conclusão das atividades por meio de um quadro *Kanban*, resultando assim na prática conhecida como *Scrumban* (combinação das abordagens *Scrum* e *Kanban*). Desta forma, cada funcionalidade do sistema será gerenciada de forma a manter a clareza e transparência em todas as etapas do desenvolvimento do produto.

Outro aspecto relevante é a possibilidade de demonstração frequente de resultados parciais, contribuindo para a visualização das partes de maneira detalhada, processo que viabiliza ajustes necessários e identificação de eventuais problemas, onde há a necessidade de uma reflexão no final de cada *Sprint*, por meio de revisão e retrospectivas.

## **10 MODELAGEM DA SOLUÇÃO PROPOSTA**

### **10.1 Product Backlog**

O Quadro 2 apresenta a lista dos itens do *Product Backlog* deste projeto. Ela representa cada item contendo uma descrição resumida, a quantidade de tempo estimado que esta atividade necessita para ser concluída e qual o nível de prioridade desta atividade dentro do projeto.

Quadro 2 – Product backlog

Item	Descrição	Estimativa de Esforço	Prioridade
1	Desenvolver <i>Script</i> para reconhecimento de objetos gerais.	3 dias	1º
2	Desenvolver protótipo, na ferramenta <i>Tinkercad</i> , do funcionamento do semáforo no Arduino.	2 dias	2º
3	Realizar ajuste no <i>Script</i> de reconhecimento de objetos para filtrar as classes desejadas e realizar a contagem dos veículos.	5 dias	3º
4	Montar o protótipo físico do Arduino Mega e implementar o <i>Script</i> em C++.	4 dias	4º
5	Aplicar a funcionalidade de envio da quantidade de veículos contados em Python para a plataforma Arduino Mega.	2 dias	5º
6	Desenvolver a maquete do projeto.	15 dias	6º

## 10.2 Sprints Backlog

Para organizar o *Sprint Backlog* deste projeto, foi criada uma tabela contendo as informações sobre cada *Sprint* (ver Quadro 3). Além disso, o *Jira* será outra ferramenta utilizada para a documentação e acompanhamento de cada atividade. Por fim, a utilização de um quadro *Kanban* facilitará a visualização e os controles das tarefas de cada *Sprint*, como também, de todo o projeto.

Quadro 3 – Sprint backlog

Item do Product Backlog	Nº Sprint e Período	Descrição da Tarefa	Responsável	Estimativa de Esforço
1	<i>Sprint 1:</i> 01/04/2025 a 03/04/2025	<i>Desenvolver Script que reconheça objetos que foram treinados no Dataset Coco.</i>	<i>Murilo Cambruzzi Paiva</i>	<i>72 horas</i>
2	<i>Sprint 2:</i> 04/04/2025 a 05/04/2025	<i>Montar o funcionamento do semáforo na plataforma de simulação de Arduino chamada Tinkercad.</i>	<i>Murilo Cambruzzi Paiva</i>	<i>24 horas</i>
3	<i>Sprint 3:</i> 06/04/2025 a 11/04/2025	<i>Aplicar funcionalidade de filtragem de classes específicas (carros, motos, caminhonetes, ônibus, caminhão e micro-ônibus) e realizar a contagem dos veículos.</i>	<i>Murilo Cambruzzi Paiva</i>	<i>120 horas</i>
4	<i>Sprint 5:</i> 12/04/2025 a 16/04/2025	<i>Montar a estrutura do protótipo físico com a plataforma Arduino Mega utilizando todos os componentes eletrônicos necessários.</i>	<i>Murilo Cambruzzi Paiva</i>	<i>96 horas</i>
5	<i>Sprint 6:</i> 17/04/2025 a 19/04/2025	<i>Realizar adaptação tanto do código em Python, quanto do código em C++ para que as duas arquiteturas (computador e Arduino) conversem entre si para passagem de dados.</i>	<i>Murilo Cambruzzi Paiva</i>	<i>48 horas</i>
6	<i>Sprint 7:</i> 20/04/2025 a 05/05/2025	<i>Criar uma maquete que irá simular o funcionamento do produto em sua versão final.</i>	<i>Murilo Cambruzzi Paiva</i>	<i>360 horas</i>

### 10.3 Levantamento de Requisitos

Conforme consta no livro Engenharia de Software Moderna, escrito pelo autor Marco Tulio Valente (<https://engsoftmoderna.info/cap3.html>), “Requisitos definem o que um sistema deve fazer e sob quais restrições”. Assim, essas funções essenciais são denominadas de Requisito Funcional e representam as funções do sistema. Por outro lado, as restrições de cada requisito são chamadas de Requisito Não Funcional e abrangem atributos detalhados, como desempenho, segurança, usabilidade, confiabilidade, escalabilidade, portabilidade e manutenibilidade.

### 10.3.1 Reconhecer os Veículos

Quadro 4 – Requisito Funcional: Reconhecer Veículo

Requisito Funcional		
<b>Nome:</b>	Reconhecer Veículo	<b>Código:</b> RF1
<b>Descrição:</b>	Será utilizado um <i>webcam</i> para o reconhecimento de diferentes categorias de veículos em tempo real.	
<b>Estimativa de Esforço:</b>	2 dias	<b>Prioridade:</b> Alta
Requisitos Não Funcionais		
ID NF	Descrição	Categoria
1	Será utilizada a linguagem de programação Python.	Portabilidade
2	Será utilizada a técnica de visão computacional YOLO com a rede neural <i>Pytorch</i> .	Portabilidade
3	Somente alguns objetos específicos serão reconhecidos, são eles: (carros, motos, caminhonetes, caminhões, carretas e ônibus).	Desempenho
4	A taxa de confiança para o reconhecimento será de 0.30.	Confiabilidade
5	Esta funcionalidade será acionada quando o sinal luminoso vermelho for acionado.	Escalabilidade
6	Esta funcionalidade será interrompida quando o sinal luminoso vermelho for apagado.	Escalabilidade

### 10.3.2 Realizar a Contagem dos Veículos

Quadro 5 – Requisito Funcional: Realizar a Contagem dos Veículos

Requisito Funcional		
<b>Nome:</b>	Realizar a Contagem dos Veículos	<b>Código:</b> RF2
<b>Descrição:</b>	Em união com o reconhecimento dos veículos, será realizada uma contagem deles, onde será enviada para a plataforma Arduino Mega.	
<b>Estimativa de Esforço:</b>	4 dias	<b>Prioridade:</b> Alta
Requisitos Não Funcionais		
ID NF	Descrição	Categoria
	Será utilizada a linguagem de programação Python.	Portabilidade
	A contagem será realizada pela passagem da caixa delimitadora por uma área específica, dessa forma a repetição na contagem será evitada.	Segurança
	O envio da quantidade de veículos será realizado via cabo USB padrão A/B do computador para a plataforma Arduino Mega.	Portabilidade

### 10.3.3 Realizar o Controle do Semáforo

Quadro 6 – Requisito Funcional: Realizar o Controle do Semáforo

Requisito Funcional		
<b>Nome:</b>	Realizar o Controle do Semáforo	<b>Código:</b> RF3
<b>Descrição:</b>	A plataforma de prototipagem Arduino Mega irá representar o funcionamento do semáforo por meio de seus componentes luminosos eletrônicos.	
<b>Estimativa de Esforço:</b> 5 dias	<b>Prioridade:</b> Alta	
Requisitos Não Funcionais		
ID NF	Descrição	Categoria
	O Arduino Mega será programado na linguagem C/C++.	Portabilidade
	Serão utilizados <i>jumpers</i> , <i>protoboard</i> , módulo de semáforo e resistores para a representação luminosa das cores vermelho, amarelo e verde.	Portabilidade
	O Arduino Mega receberá a quantidade de veículos detectados e contados pelo USB padrão A/B.	Portabilidade
	Será feita uma verificação do dado recebido do computador para acionar a luz verde no módulo de semáforo por um determinado tempo na via que foi realizada a contagem.	Desempenho

### 10.3.4 Emitir Contagem Regressiva

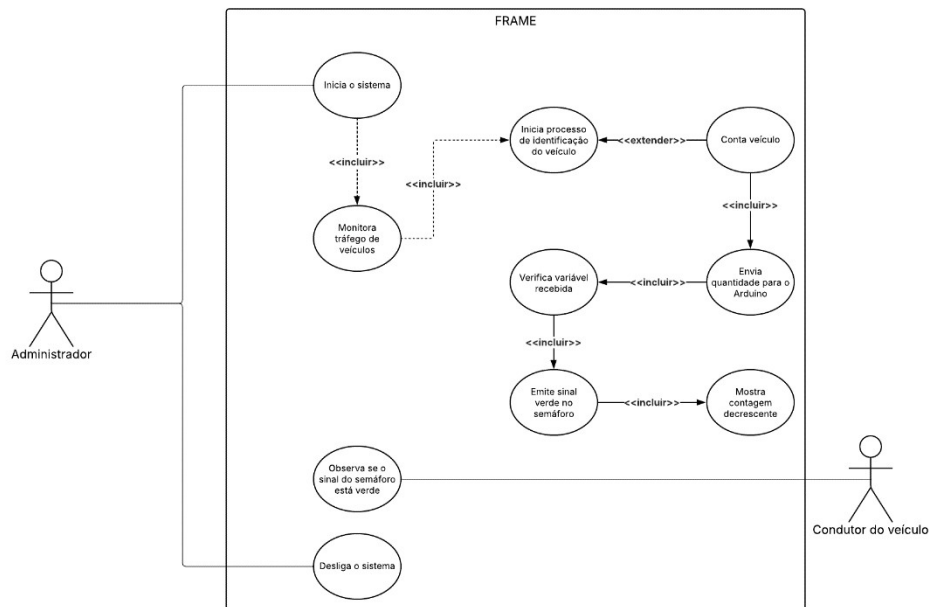
Quadro 7 – Requisito Funcional: Emitir Contagem Regressiva

Requisito Funcional		
<b>Nome:</b>	Emitir Contagem Regressiva	<b>Código:</b> RF4
<b>Descrição:</b>	A plataforma Arduino Mega irá emitir uma contagem regressiva do tempo restante para o sinal luminoso verde apagar.	
<b>Estimativa de Esforço:</b> 2 dias	<b>Prioridade:</b> Baixa	
Requisitos Não Funcionais		
ID NF	Descrição	Categoria
	O Arduino Mega será programado na linguagem C/C++.	Portabilidade
	Serão utilizados <i>jumpers</i> , <i>protoboard</i> e <i>display</i> de sete segmentos para a contagem regressiva.	Portabilidade
	O <i>display</i> será acionado após a verificação da quantidade de veículos contados recebida pelo Arduino Mega.	Segurança

## 10.4 Modelo de Caso de Uso

### 10.4.1 Diagrama de Caso de Uso

Figura 19 – Diagrama de caso de uso



Fonte: Autoria Própria (2025)

### 10.4.2 Diagrama de Caso de Uso Expandido

#### 10.4.2.1 Caso de Uso - iniciar o sistema

**Descrição:** O sistema é iniciado para começar o controle do tráfego.

**Ator Primário:** Administrador.

**Precondições:** O administrador deverá possuir conhecimento do funcionamento do sistema.

#### **Fluxo Principal**

1. O Administrador inicia o sistema.
2. O administrador certifica que o sistema está funcionando corretamente.

#### 10.4.2.2 Caso de Uso - monitora tráfego de veículos

**Descrição:** O sistema inicia o processo de controle do tráfego terrestre.

**Precondições:** O sistema deverá ser iniciado.

#### **Fluxo Principal**

1. O Arduino acende o LED vermelho para cada via do cruzamento, deixando somente uma via com o LED verde acendido.
2. O Arduino mostra os segundos em ordem decrescente pelo *display* de sete segmentos o tempo restante para o LED verde desta via apagar.
3. A *Webcam* realiza a identificação do veículo nas vias que estão com o sinal luminoso vermelho acendido.
4. É realizada a contagem de cada veículo identificado pela *Webcam*.
5. O Arduino apaga o LED verde.
6. O Arduino acende o LED amarelo por alguns segundos.
7. O Arduino apaga o LED amarelo.
8. O Arduino acende o LED vermelho.
9. A quantidade contada da próxima via que irá permitir o fluxo de veículos é enviada para o Arduino através do cabo serial.
10. O Arduino recebe o valor de veículos.
11. O Arduino realiza uma verificação da quantidade de veículos recebida pelo computador.
12. O Arduino acende o LED verde da próxima via.
13. O fluxo volta para o item 2 na próxima via.

#### 10.4.2.3 Caso de Uso - observa se o sinal está verde

**Descrição:** O condutor irá se atentar quando o sinal luminoso verde estiver emitido para ele seguir viagem pela via.

**Ator Primário:** Condutor do veículo.

**Precondições:** O sistema deverá ter realizado o controle do tráfego e todas suas etapas.

#### **Fluxo Principal**

1. O condutor estará aguardando enquanto o sinal luminoso da via que ele estiver não estiver no verde.
2. Quando o semáforo da via que ele estiver ficar verde, ele seguirá caminho.

#### 10.4.2.4 Caso de Uso - desliga o sistema

**Descrição:** O Administrador desliga o sistema interrompendo o funcionamento do sistema.

**Ator Primário:** Administrador.

**Precondições:** O sistema deverá estar ligado.

## **Fluxo Principal**

1. O Administrador para com a execução do *Script*.
2. Verifica que o sistema foi interrompido como esperado.

## **11 CONCLUSÕES E TRABALHOS FUTUROS**

O desenvolvimento do semáforo dinâmico, proveniente da integração entre visão computacional e sistemas embarcados, demonstrou viabilidade ao ser implementado em um processo já existente que não possuía a característica de adaptação dinâmica ao volume de veículos que trafegam pela via. Através da demonstração na maquete apresentada, foi possível apresentar o funcionamento do sistema, que simulou o processo real de detecção e contagem de veículos, evidenciando como esses parâmetros influenciam positivamente o gerenciador luminoso. Dessa forma, contribui-se para uma maior eficiência na gestão do trânsito do cruzamento, reduzindo o tempo de espera desnecessário dos condutores. Assim, este trabalho cumpriu o objetivo de propor uma melhoria em relação aos semáforos com tempos estáticos de sinalização luminosa, revelando grande potencial para a aplicabilidade dessa tecnologia em cenários reais.

Considerando que o ambiente urbano envolve não apenas veículos, mas também pedestres, torna-se fundamental que o sistema contemple a segurança dos transeuntes nas faixas de travessia. Nesse sentido, como trabalhos futuros é proposto a inserção de câmeras adicionais capazes de detectar a presença de pessoas durante a emissão do sinal verde para os automóveis, interrompendo temporariamente a abertura do fluxo veicular. Essa funcionalidade garante que a vida do pedestre não seja colocada em risco, tornando o sistema mais inclusivo e seguro para todos os usuários da via.

Outra possibilidade de expansão do projeto consiste no desenvolvimento de técnicas alternativas de captação da presença de veículos, especialmente para cenários urbanos menos complexos, onde a implementação de múltiplas câmeras pode não ser viável. Nesse contexto, uma solução prática será posicionar sensores ou câmeras próximas ao grupo focal, voltados para a via, de modo a reconhecer a existência de veículos enquanto o sinal estiver aberto. Assim, o sistema poderá identificar a interrupção do fluxo e, automaticamente, liberar a passagem para as outras vias do cruzamento.

Além dos aspectos técnicos, é relevante destacar o impacto social e ambiental da aplicação de semáforos inteligentes. A redução no tempo de espera e no acúmulo de veículos

parados em cruzamentos contribui para diminuir o consumo de combustível e a emissão de poluentes, favorecendo a sustentabilidade urbana. Paralelamente, a fluidez no trânsito reduz situações de estresse entre condutores e pedestres, promovendo uma convivência mais harmoniosa no espaço público.

Por fim, a adoção de sistemas dinâmicos de controle semaforico abre caminho para uma integração mais ampla com o conceito de cidades inteligentes (*smart cities*). A utilização de visão computacional associada a sistemas embarcados pode ser complementada com redes de sensores IoT e algoritmos de inteligência artificial, possibilitando uma análise em tempo real do tráfego em escala urbana. Essa abordagem permitiria não apenas a otimização local de cruzamentos, mas também a coordenação global de corredores viários, tornando o transporte mais eficiente, seguro e alinhado às demandas do crescimento urbano.

## REFERÊNCIAS

ALURA. Machine Learning. Alura, s.d. Disponível em: <https://www.alura.com.br/artigos/machine-learning?srsltid=AfmBOoo2YyH0DUOVDKCTzeba3hZAS0ppLjzgnrUnhhGdPSMvKn8pJxmk>. Acesso em: 13 mar. 2025.

Alves, Gabriel. Detecção de Objetos com YOLO – Uma abordagem moderna, 2020. Disponível em: <https://iaexpert.academy/2020/10/13/deteccao-de-objetos-com-yolo-uma-abordagem-moderna/>. Acesso em 21 mar. 2025.

AMAZON WEB SERVICES. O que é uma rede neural? AWS, s.d. Disponível em: <https://aws.amazon.com/pt/what-is/neural-network/>. Acesso em: 23 mar. 2025.

ARDUINO. The Arduino Software (IDE). Arduino Documentation, s.d. Disponível em: <https://docs.arduino.cc/learn/starting-guide/the-arduino-software-ide/>. Acesso em: 05 mar. 2025.

BABOS, Flávio. O que é Arduino?: tudo que você precisa saber [guia completo], 2025. Disponível em: <https://flaviobabos.com.br/o-que-e-arduino/>. Acesso em: 05 mar. 2025.

BRASIL. Ministério da Educação. Secretaria Executiva STIC. MDSAGIL MEC: versão 1-1, dez. 2022. Disponível em: [https://www.gov.br/mec/pt-br/acao-a-informacao/institucional/estrutura-organizacional/orgaos-especificos-singulares/secretaria-executiva/stic/documentos/mdsagil\\_mec\\_v1-1\\_dez22.pdf](https://www.gov.br/mec/pt-br/acao-a-informacao/institucional/estrutura-organizacional/orgaos-especificos-singulares/secretaria-executiva/stic/documentos/mdsagil_mec_v1-1_dez22.pdf). Acesso em: 07 mar. 2025.

CAMARGO, Robson. Manifesto Ágil: entenda como surgiu e conheça os 12 princípios, 2019. Disponível em: <https://robsoncamargo.com.br/blog/Manifesto-Agil-entenda-como-surgiu-e-conheca-os-12-principios>. Acesso em: 07 mar. 2025.

CARVALHO, Caroline. O que é Python? — um guia completo para iniciar nessa linguagem de programação, 2024. Disponível em:

<https://www.alura.com.br/artigos/python?srsltid=AfmBOoomrciCAzJ13cXcCMB2546lqnxHIG51qq7waAUP5ITJF4k9wwnk>. Acesso em 05 mar. 2025.

COCO DATASET. Common Objects in Context. COCO, s.d. Disponível em: <https://cocodataset.org/#home>. Acesso em: 19 mar. 2025.

Costa, Fabio. Como começar com Arduino: os componentes mais importantes que você precisa conhecer, s.d. Disponível em: <https://fabiocosta.net/arduino/como-comecar-com-arduino/>. Acesso em: 05 mar. 2025.

DATA.SCIENCE.ACADEMY, O Que é Visão Computacional?, 2022. Disponível em: [https://blog.dsacademy.com.br/o-que-e-visao\\_computacional/](https://blog.dsacademy.com.br/o-que-e-visao_computacional/). Acesso em: 11 mar. 2025.

Disponível em: <https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar>. Acesso em: 05 mar. 2025.

DRUMOND, Claire. Manifesto Ágil, s.d. Disponível em: <https://www.atlassian.com/br/agile/manifesto>. Acesso em: Acesso em: 07 mar. 2025.

GOOGLE CLOUD. What is Machine Learning? Google Cloud, s.d. Disponível em: <https://cloud.google.com/learn/what-is-machine-learning?hl=pt-BR>. Acesso em: 13 mar. 2025.

HANASHIRO, Akira. VS Code: o que é e por que você deve usar?. Treinaweb Blog, s.d.

IBM. Neural Networks. IBM, s.d. Disponível em: <https://www.ibm.com/br-pt/topics/neural-networks>. Acesso em: 23 mar. 2025.

KUNDU, Rohit. YOLO: Algoritmo para detecção de objetos explicado [+exemplos]. Disponível em: <https://www.v7labs.com/blog/yolo-object-detection>. Acesso em 21 mar. 2025.

OPENCV. About. OpenCV, s.d. Disponível em: <https://opencv.org/about/>. Acesso em: 10 mar. 2025.

OPENCV. Introduction to the COCO Dataset. OpenCV Blog, 2021. Disponível em: <https://opencv.org/blog/introduction-to-the-coco-dataset/>. Acesso em: 19 mar. 2025.

ORACLE. What is Machine Learning? Oracle, s.d. Disponível em: <https://www.oracle.com/br/artificial-intelligence/machine-learning/what-is-machine-learning>. Acesso em: 13 mar. 2025.

PAIVA, Murilo Cambuzzi et al. Análise Computacional de um Cluster Executando o Algoritmo de Cálculo de Números Primos Implementado com MPI. ERAD-ERAMIA, n. 2, [s.l.], 2025. Disponível em: <https://sol.sbc.org.br/index.php/erad-eramia-no2/article/view/22745/22568>. Acesso em: 18 mar. 2025.

PYTHON. O que é Python? Documentação Python, s.d. Disponível em: <https://docs.python.org/pt-br/dev/faq/general.html#what-is-python>. Acesso em: 05 mar. 2025.

ULTRALYTICS. YOLO11 NOVO. Ultralytics YOLO Docs, 2025. Disponível em: <https://docs.ultralytics.com/pt/models/yolo11/>. Acesso em 21 mar. 2025.

VISÃO COMPUTACIONAL, YOLO para Detecção de Objetos – Visão Geral, s.d. Disponível em: <https://visaocomputacional.com.br/yolo-para-deteccao-de-objetos-visao-geral/>. Acesso em 22 mar. 2025.

VISÃO COMPUTACIONAL, YOLO Versões 1 e 2 (Arquitetura), s.d. Disponível em: <https://visaocomputacional.com.br/yolo-versoes-1-e-2-arquitetura/>. Acesso em 22 mar. 2025.

VISO.AI. OpenCV, s.d. Disponível em: <https://viso.ai/computer-vision/opencv/>. Acesso em: 10 mar. 2025.

CONSELHO NACIONAL DE TRÂNSITO (CONTRAN). Manual Brasileiro de Sinalização de Trânsito: Volume V – Sinalização Semafórica. 2022. Disponível em: [https://www.gov.br/transportes/pt-br/assuntos/transito/arquivos-senatran/docs/copy\\_of\\_\\_05\\_\\_MBST\\_Vol.\\_V\\_\\_Sinalizacao\\_Semaforica.pdf](https://www.gov.br/transportes/pt-br/assuntos/transito/arquivos-senatran/docs/copy_of__05__MBST_Vol._V__Sinalizacao_Semaforica.pdf). Acesso em: 11 jul. 2025.

ARAÚJO, HELIA. Semáforo que conta o tempo reduz os acidentes, diz estudo. 2011. Disponível em: <https://www1.folha.uol.com.br/fsp/cotidian/ff1301201138.htm>. Acesso em 16 jul. 2025

ORACULO, Quem inventou e onde foi instalado o primeiro semáforo?. 2017. Superinteressante. Disponível em: <https://super.abril.com.br/coluna/oraculo/quem-inventou-o-semaforo-de-transito/>. Acesso em 17 jul. 2025.

TRIBUNAL DE JUSTIÇA DO PARANÁ. Objetivos de Desenvolvimento Sustentável. s.d. Disponível em: [https://www.tjpr.jus.br/en/objetivos-de-desenvolvimento-sustentavel?utm\\_source=chatgpt.com](https://www.tjpr.jus.br/en/objetivos-de-desenvolvimento-sustentavel?utm_source=chatgpt.com). Acesso em: 20 jul. 2025.

NAÇÕES UNIDAS BRASIL. Sobre o nosso trabalho para alcançar os Objetivos de Desenvolvimento Sustentável no Brasil. 2025. Disponível em: <https://brasil.un.org/pt-br/sdgs>. Acesso em: 21 jul. 2025.

VARGAS, Julia. Câmeras de trânsito ao vivo; o que elas podem ou não fiscalizar. 2025. Disponível em: <https://autopapo.com.br/noticia/cameras-de-transito-ao-vivo-podem-fiscalizar/>. Acesso em: 22 jul. 2025.

MINISTÉRIO DAS CIDADES. Carta Brasileira para Cidades Inteligentes (CBCI). s.d. Disponível em: <https://www.gov.br/cidades/pt-br/acao-a-informacao/acoes-e-programas/desenvolvimento-urbano-e-metropolitano/projeto-andus/carta-brasileira-para-cidades-inteligentes>. Acesso em: 23 jul. 2025.