

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

VICTOR EMANOEL DA COSTA FERREIRA

**IMPLEMENTAÇÃO E ANÁLISE DE PERFIS ÉTICOS EM VEÍCULOS
AUTÔNOMOS**

PONTA GROSSA

2026

VICTOR EMANOEL DA COSTA FERREIRA

**IMPLEMENTAÇÃO E ANÁLISE DE PERFIS ÉTICOS EM VEÍCULOS
AUTÔNOMOS**

Implementation and Analysis of Ethical Profiles in Autonomous Vehicles

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do título
de Bacharel em Ciência da Computação, da
Universidade Tecnológica Federal do Paraná
(UTFPR).

Orientador: Prof. Dr. Gleifer Vaz Alves.

PONTA GROSSA

2026



Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

VICTOR EMANOEL DA COSTA FERREIRA

**IMPLEMENTAÇÃO E ANÁLISE DE PERFIS ÉTICOS EM VEÍCULOS
AUTÔNOMOS**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do título
de Bacharel em Ciência da Computação, da
Universidade Tecnológica Federal do Paraná
(UTFPR).

Data de aprovação: 21/maio/2026

Gleifer Vaz Alves
Doutor
Universidade Tecnológica Federal do Paraná

Vinícius Camargo Andrade
Doutor
Universidade Tecnológica Federal do Paraná

André Pinz Borges
Doutor
Universidade Tecnológica Federal do Paraná

**PONTA GROSSA
2026**

RESUMO

O avanço da inteligência artificial tem inserido veículos autônomos em cenários urbanos dinâmicos, levantando questões cruciais sobre a tomada de decisão em situações de risco. A simples aplicação de normativas rígidas de trânsito não é suficiente para lidar com dilemas morais complexos, como a iminência de acidentes envolvendo pedestres ou engavetamentos, exigindo sistemas capazes de ponderar regras e consequências. Neste contexto, este trabalho tem como objetivo desenvolver um ambiente de simulação para implementar, analisar e comparar diferentes perfis éticos em veículos autônomos, utilizando a arquitetura BDI e Aprendizado por Reforço. A metodologia consistiu na criação de um simulador bidimensional em Python, utilizando a biblioteca Pygame para reproduzir cenários de cruzamentos sinalizados e rodovias. Os veículos e pedestres foram modelados como agentes autônomos fundamentados na arquitetura BDI (Crenças, Desejos e Intenções) por meio do framework MASPY. Foram estabelecidos cinco perfis de comportamento moral: Deontológico, Legalista, Utilitarista, Egoísta e um Agente Aprendiz RL. Enquanto os agentes puramente simbólicos atuaram com base em regras lógicas de utilidade predefinidas, o agente aprendiz empregou técnicas de Aprendizagem por Reforço (Q-Learning) para adaptar suas decisões lógicas com base em punições e recompensas ligadas à preservação da vida, respeito às leis e eficiência de fluxo. O sistema garantiu a extração automática de métricas críticas no formato JSON, incluindo taxa de atropelamentos, colisões veiculares, infrações e tempo médio de espera. As simulações demonstraram a viabilidade técnica de incorporar o pluralismo moral na deliberação de sistemas multiagentes, evidenciando os impactos operacionais e éticos que cada perfil produz na segurança viária. Conclui-se que a união de métodos simbólicos e empíricos contribui significativamente para o desenvolvimento de agentes computacionais mais flexíveis, transparentes e alinhados à proteção humana na mobilidade autônoma.

Palavras-chave: inteligência artificial; sistemas multiagentes; arquitetura BDI; ética autônoma; aprendizagem por reforço.

ABSTRACT

The advancement of artificial intelligence has introduced autonomous vehicles into dynamic urban scenarios, raising crucial questions about decision-making in high-risk situations. The mere application of strict traffic regulations is insufficient to handle complex moral dilemmas, such as imminent accidents involving pedestrians or multi-vehicle collisions, demanding systems capable of weighing rules and consequences. In this context, this work aims to develop a simulation environment to implement, analyze, and compare different ethical profiles in autonomous vehicles using the BDI architecture and Reinforcement Learning. The methodology consisted of creating a two-dimensional simulator in Python, using the Pygame library to reproduce scenarios of signalized intersections and highways. Vehicles and pedestrians were modeled as autonomous agents based on the BDI (Beliefs, Desires, and Intentions) architecture through the MASPY framework. Five moral behavior profiles were established: Deontological, Legalistic, Utilitarian, Egoistic, and an RL Learning Agent. While the purely symbolic agents acted based on predefined logical utility rules, the learning agent employed Reinforcement Learning techniques (Q-Learning) to adapt its logical decisions based on punishments and rewards linked to the preservation of life, compliance with laws, and traffic flow efficiency. The system ensured the automatic extraction of critical metrics in JSON format, including pedestrian collision rates, vehicle collisions, traffic infractions, and average waiting times. The simulations demonstrated the technical feasibility of incorporating moral pluralism into the deliberation of multi-agent systems, highlighting the operational and ethical impacts that each profile produces on road safety. It is concluded that the union of symbolic and empirical methods contributes significantly to the development of more flexible and transparent computational agents aligned with human protection in autonomous mobility.

Keywords: artificial intelligence; multi-Agent systems; BDI architecture; autonomous ethics; reinforcement learning.

LISTA DE ALGORITMOS

Algoritmo 1 – Sensor virtual de detecção de pedestres com parametrização diferenciada por perfil ético	29
Algoritmo 2 – Ciclo BDI e decisão por regras fixas	34
Algoritmo 3 – Heurística de resolução de <i>deadlocks</i>	36
Algoritmo 4 – Implementação da Função de Utilidade Ponderada no método calcular_utilidade da classe ModuloMoral.....	38
Algoritmo 5 – Atualização <i>Q-Table</i> e política ϵ -greedy	46

LISTA DE FIGURAS

Figura 1 – Mapa de Cruzamento	20
Figura 2 – Mapa da Rodovia	21
Figura 3 – Rotas dos Pedestres	22
Figura 4 – Deslocamento dos Agentes	23
Figura 5 – Rotação da Imagem	24
Figura 6 – Cone de Visão dos Veículos.....	25
Figura 7 – Cone de Visão dos Veículos em Relação aos Pedestres	26
Figura 8 – Detecção Lateral dos Veículos.....	27
Figura 9 – Fluxograma do ciclo de raciocínio dos agentes baseados em regras	32
Figura 10 – Representação da Arquitetura do Sistema Multiagentes no padrão MASPY	35
Figura 11 – Fluxograma do ciclo de raciocínio do agente com aprendizado por reforço	48
Figura 12 – Diagrama de transição de estados e recompensas do agente aprendiz.....	49
Figura 13 – Tela de Menu (MenuScreen)	54
Figura 14 – Tela de Opções com Agentes Éticos	55
Figura 15 – Tela de Sobre	56
Figura 16 – Tela de Simulação (SimulationScreen).....	57
Figura 17 – Tela de Métricas (MetricsScreen).....	58

LISTA DE GRÁFICOS

Gráfico 1 – Gráfico de Radar	41
Gráfico 2 – Média de Tempo de Viagem	65
Gráfico 3 – Média de infrações semaforicas por perfil ético	67
Gráfico 4 – Média de atropelamentos por perfil ético	69
Gráfico 5 – Evolução dos atropelamentos por perfil ético	69
Gráfico 6 – Tempo de espera acumulado por perfil ético.....	71
Gráfico 7 – Média de Tempo de Viagem na Rodovia.....	73
Gráfico 8 – Tempo de espera por perfil ético (Rodovia)	75
Gráfico 9 – Mapa de Calor Global dos Perfis Éticos	78

LISTA DE FÓRMULAS

Fórmula 1 – Distância Euclidiana	23
Fórmula 2 – Ângulo de Rotação Visual	23
Fórmula 3 – Detecção de Veículos	25
Fórmula 4 – Distância Lateral Crítica	26
Fórmula 5 – Função de Utilidade Ponderada	37
Fórmula 6 – Definição do Estado no Cruzamento	44
Fórmula 7 – Definição do Estado na Rodovia.....	44
Fórmula 8 – Equação de Bellman (<i>Q-Learning</i>).....	45

LISTA DE TABELAS

Tabela 1 – Valores utilizados na parametrização ética dos perfis implementados.	40
Tabela 2 – Parâmetros globais do ambiente de simulação	60
Tabela 3 – Estatísticas do tempo de viagem por perfil ético (Cruzamento)	64
Tabela 4 – Infrações semaforicas por perfil ético	66
Tabela 5 – Atropelamentos de pedestres por perfil ético	68
Tabela 6 – Tempo de espera por perfil ético (Cruzamento)	70
Tabela 7 – Tempo de viagem por perfil ético (Rodovia)	72
Tabela 8 – Tempo de espera por perfil ético (Rodovia)	74
Tabela 9 – Síntese comparativa global das métricas de desempenho de ambos os cenários	76

SUMÁRIO

1	INTRODUÇÃO	12
2	REFERENCIAL TEÓRICO.....	14
2.1	Sistemas Multiagentes.....	14
2.2	Arquitetura BDI.....	15
2.3	Aprendizagem por reforço.....	15
2.4	Modelos éticos e pluralismo moral.....	16
2.5	Trabalhos Relacionados	17
2.5.1	Comparação com Trabalhos Relacionados.....	18
3	METODOLOGIA E DESENVOLVIMENTO	19
3.1	Visão Geral da Arquitetura do Sistema	19
3.2	Modelagem do ambiente urbano e navegação	19
3.2.1	Mapa de Cruzamento	20
3.2.2	Mapa de Rodovia	21
3.2.3	Rotas dos Pedestres	21
3.2.4	Navegação dos agentes.....	22
3.3	Sensores Virtuais e Percepção Espacial.....	24
3.3.1	Detecção de Obstáculos e Cone de Visão	24
<u>3.3.1.1</u>	<u>Detecção de Veículos.....</u>	<u>25</u>
<u>3.3.1.2</u>	<u>Detecção de Pedestres</u>	<u>26</u>
3.3.2	Cálculo da Distância Lateral Crítica.....	26
3.3.3	Parametrização Sensorial por Perfil Ético	27
3.4	Arquitetura BDI dos Agentes.....	30
3.4.1	Crenças, Desejos e Intenções no Simulador.....	30
3.4.2	Ciclo de Raciocínio e Tomada de Decisão	31
3.4.3	Resolução de <i>Deadlocks</i>	35
3.5	O Módulo Moral e a Função de Utilidade Ponderada	36
3.5.1	Os Quatro Pilares Éticos e a Equação de Utilidade	36
3.5.2	Parametrização dos Perfis Éticos.....	39
3.6	Perfis Éticos dos Agentes Veiculares	41
3.6.1	Agente Deontológico	41
3.6.2	Agente Utilitarista	42
3.6.3	Agente Legalista.....	42
3.6.4	Agente Egoísta	43

3.6.5	Agente Aprendiz RL	43
<u>3.6.5.1</u>	<u>Mapeamento de Estados e Política de Exploração</u>	<u>44</u>
<u>3.6.5.2</u>	<u>Atualização da Q-Table</u>	<u>45</u>
3.6.6	Quadro Comparativo dos Perfis	49
3.7	Agente Pedestre	50
3.8	Coleta e Processamento de Dados	52
3.9	Interfaces do Sistema	53
3.9.1	Tela de Menu	53
3.9.2	Tela de Opções	54
3.9.3	Tela de Sobre	55
3.9.4	Tela de Simulação	56
3.9.5	Tela de Métricas	57
4	RESULTADOS E DISCUSSÃO	59
4.1	Configurações e Ambiente de Teste	59
4.2	Método Estatístico Adotado	60
4.2.1	Teste de Kruskal-Wallis	60
4.2.2	Teste de Mann-Whitney com Correção de Bonferroni	61
4.2.3	Estrutura das Métricas Analisadas	62
4.3	Cenário 1: Cruzamento Urbano Semaforizado	62
4.3.1	Análise de Eficiência e Fluidez	62
4.3.2	Segurança e Conformidade Normativa	65
4.4	Cenário 2: Rodovia e Interseções de Fluxo Contínuo	71
4.4.1	Eficiência em Vias Expressas	71
4.4.2	Tempos de Retenção e Resolução de <i>Deadlocks</i>	73
4.5	Discussão Geral dos Resultados	75
4.6	Limitações do Trabalho e Testes de Estresse	78
5	CONCLUSÃO	80
	REFERÊNCIAS	82

1 INTRODUÇÃO

O contínuo avanço da inteligência artificial tem viabilizado a operação de veículos autônomos (VAs) em cenários urbanos altamente dinâmicos e imprevisíveis, o que representa um desafio tecnológico que exige verificação robusta devido à complexidade do comportamento no tráfego (Dennis; Fisher, 2023; Markovicz; Alves, 2023). A inserção desses sistemas no trânsito real demanda o processamento simultâneo de múltiplas variáveis para assegurar a fluidez e a segurança (Alves *et al.*, 2021; Li *et al.*, 2022). No entanto, à medida que a automação assume o controle em contextos críticos, emergem dilemas éticos fundamentais sobre como essas decisões são tomadas (Awad *et al.*, 2018; Bonnefon *et al.*, 2016), especialmente em situações onde o cumprimento estrito das regras de trânsito e a preservação da vida entram em conflito (Thornton *et al.*, 2016).

A simples implementação de normativas rígidas não é suficiente para abranger a imprevisibilidade do ambiente urbano, como a travessia inesperada de pedestres em cruzamentos ou o risco de engavetamentos por disputas de preferência em rodovias (Arnold; Scheutz, 2023; Rüb; Dunin-Kępicz, 2020). Nesse contexto, a ausência de critérios éticos bem definidos e adaptáveis pode comprometer diretamente a confiança pública, aceitabilidade e confiabilidade da tecnologia (Kuipers, 2018).

Um sistema puramente legalista, por exemplo, pode falhar em evitar um acidente fatal se a única saída viável exigir o cometimento de uma infração de trânsito, situação na qual a obediência cega à lei pode contrariar o próprio "espírito da norma" e a doutrina legal da necessidade de preservar a vida (Arnold; Scheutz, 2023; De Sio, 2017). Por outro lado, abordagens estritamente focadas na eficiência do fluxo e na maximização do tráfego podem ignorar princípios morais de cooperação e justiça, falhando na proteção aos usuários mais vulneráveis da via ao não incorporar um comportamento pró-social adequado (Mashayekhi *et al.*, 2022).

Nesse contexto, torna-se essencial o desenvolvimento de arquiteturas computacionais capazes de avaliar e orientar o comportamento autônomo com base no pluralismo moral (Rhim *et al.*, 2021). Para preencher essa lacuna, este trabalho propõe a implementação e análise de diferentes perfis éticos (Deontológico, Legalista, Utilitarista, Egoísta e de Aprendizagem Moral) na tomada de decisão de veículos autônomos. Através de um ambiente simulado, a pesquisa utiliza uma arquitetura BDI

(Crenças, Desejos e Intenções) e técnica de Aprendizagem por Reforço (*Q-Learning*) para avaliar o impacto operacional e moral de cada perfil frente a obstáculos e sinalizações.

Durante a preparação deste trabalho, o autor utilizou o modelo de inteligência artificial Gemini (Google), versão de maio de 2026, para auxiliar na estruturação lógica dos capítulos, na compreensão e resolução de falhas no código-fonte da simulação (*debugging*), na revisão gramatical de fragmentos textuais e na sugestão de redação acadêmica baseada nos dados estatísticos extraídos. Após o uso deste serviço, o autor revisou e editou integralmente o conteúdo em conformidade com o método científico e assume total responsabilidade pelo conteúdo da publicação e pelo código desenvolvido.

Os códigos desenvolvidos neste trabalho podem ser encontrados no GitHub, por meio do seguinte repositório: <https://github.com/VictorEmanuel7/simulacao-de-agentes>.

2 REFERENCIAL TEÓRICO

Este capítulo estabelece a base teórica da pesquisa ao integrar a estrutura de Sistemas Multiagentes e arquitetura BDI para a modelagem de agentes autônomos à técnica de Aprendizagem por Reforço, focada na adaptação comportamental. Por fim, utiliza o pluralismo moral e correntes éticas clássicas para fundamentar os parâmetros de tomada de decisão em veículos autônomos.

2.1 Sistemas Multiagentes

Um Sistema Multiagentes (SMA) é definido como um conjunto de agentes autônomos capazes de perceber o ambiente, tomar decisões e interagir entre si para atingir objetivos individuais ou coletivos, constituindo a base para o desenvolvimento de sistemas distribuídos compostos por entidades inteligentes (Wooldridge, 2001). No contexto da mobilidade inteligente, esses agentes podem representar veículos autônomos, infraestruturas urbanas e usuários da via, atuando de forma independente em cenários dinâmicos e distribuídos (Rüb; Dunin-Kępicz, 2020).

Diferente dos sistemas de controle de tráfego centralizados, a abordagem descentralizada dos SMA permite que cada entidade possua a sua própria lógica de decisão e perfil moral autônomo (Mashayekhi *et al.*, 2022), reagindo dinamicamente às mudanças no ambiente, como a alteração de sinais semafóricos ou a presença de obstáculos imprevisíveis (Rüb; Dunin-Kępicz, 2020). De acordo com Boyles (2018), um agente em um SMA deve apresentar quatro características fundamentais para ser considerado inteligente:

- **Autonomia:** Capacidade de operar e tomar decisões sem a intervenção direta de humanos ou de sistemas controladores globais, avaliando internamente quando avançar, ceder a vez ou parar.
- **Habilidade Social:** Capacidade de interagir com outros agentes e com a infraestrutura por meio de protocolos de comunicação, cooperação ou simples observação do comportamento alheio no ambiente.
- **Reatividade:** Capacidade de perceber o ambiente por meio de sensores e responder antecipadamente a estímulos e a mudanças críticas.
- **Proatividade:** Comportamento orientado por objetivos, em que o agente toma a iniciativa para cumprir a sua missão principal, como o planejamento de rotas para alcançar um destino em vez de apenas reagir ao acaso.

Desse modo, a utilização da teoria de SMA é crucial para a simulação de tráfego, pois fornece o embasamento matemático e lógico para observar como decisões descentralizadas impactam a segurança e a fluidez global da rede viária.

2.2 Arquitetura BDI

A arquitetura BDI é um modelo de raciocínio para agentes inteligentes baseado na filosofia da ação humana, que permite representar o estado cognitivo interno do agente por meio de três componentes principais: crenças, desejos e intenções (Lorini, 2015).

As crenças (*Beliefs*) correspondem ao conjunto de informações que o agente possui sobre o ambiente no qual está inserido, bem como o seu próprio estado interno. No domínio de condução autônoma, essas crenças englobam a percepção de obstáculos, normas de trânsito, sinalizações ativas e a posição espacial de outros veículos. Os desejos (*Desires*) representam os objetivos que o agente busca alcançar a longo prazo, tipicamente traduzidos na conclusão segura e eficiente de um trajeto. Por fim, as intenções (*Intentions*) referem-se aos planos de ação específicos e imediatos aos quais o agente se compromete com o intuito de atingir os seus desejos (Alves *et al.*, 2021).

Dessa forma, a arquitetura BDI destaca-se na inteligência artificial ao permitir que os agentes tomem decisões de maneira deliberativa e interpretável (Rüb; Dunin-Kępicz, 2020). Essa transparência lógica é fundamental para a modelagem de comportamentos autônomos em situações de risco, na qual é necessário auditar o raciocínio que levou o agente a executar uma determinada manobra.

2.3 Aprendizagem por reforço

Complementando os modelos puramente deliberativos e baseados em regras, a Aprendizagem por Reforço (*Reinforcement Learning – RL*) permite que um agente inteligente aprenda padrões complexos de comportamento por meio da interação contínua e direta com o ambiente (Tennant *et al.*, 2024). Nesse paradigma de *Machine Learning*, o agente não recebe instruções explícitas sobre como agir, mas sim recompensas ou penalizações em função das ações executadas em cada estado, ajustando progressivamente a sua política de decisão para maximizar o ganho a longo prazo (Vishwanath *et al.*, 2024).

Uma das técnicas clássicas empregadas na resolução desses Processos de Decisão de Markov (MDP) é o *Q-Learning*, um método baseado em valor no qual o agente mantém uma estrutura matemática (como uma *Q-Table*) utilizada para estimar a utilidade esperada de cada par de estado-ação possível (Chaput *et al.*, 2023; Vishwanath *et al.*, 2024).

A atualização geométrica dos valores de utilidade é regida pela Equação de Bellman (seção 3.6.5.2 Atualização da *Q-Table*), que utiliza parâmetros fundamentais como a taxa de aprendizagem e o fator de desconto, que pondera a relevância de recompensas futuras contra ganhos imediatos (Mousavi *et al.*, 2025). Esse mecanismo matemático impõe ao agente o desafio de equilibrar a exploração de novas ações desconhecidas e a exploração do conhecimento já consolidado (Mashayekhi *et al.*, 2022), resultando no refinamento contínuo de decisões durante a interação autônoma com o ambiente.

2.4 Modelos éticos e pluralismo moral

A delegação da tomada de decisão crítica a veículos autônomos levanta a necessidade de parametrizar o raciocínio computacional a partir de perspectivas normativas humanas. A modelagem do pluralismo moral permite analisar como distintas teorias filosóficas clássicas influenciam o comportamento algorítmico do agente e, conseqüentemente, os seus impactos na segurança viária coletiva (Li *et al.*, 2022; Markovicz; Alves, 2023). Na literatura de ética da máquina, destacam-se três abordagens comportamentais fundamentais (Tennant *et al.*, 2024):

- **Deontologia:** Abordagem normativa focada no cumprimento rigoroso de deveres e regras universais. Um agente estruturado sob esta ótica prioriza o seguimento estrito das leis de trânsito e dos códigos de conduta. Em modelos mais avançados, o raciocínio deontológico pode ser hierarquizado para que a violação de uma norma de trânsito só seja matematicamente justificável se for a única ação possível para preservar o valor supremo da vida humana.
- **Utilitarismo:** Doutrina ética baseada na maximização do bem-estar coletivo e na mitigação de danos totais. Um agente utilitarista foca-se no resultado de suas ações. No contexto viário, busca a máxima fluidez do tráfego e a minimização do risco absoluto para todos os envolvidos na cena, aceitando sacrifícios próprios ou violações pontuais de regras se o cálculo do impacto global for positivo.

- Egoísmo Moral: Representa um comportamento maximizador focado exclusivamente no benefício individual da própria entidade. Este paradigma teórico atua de forma amoral quanto ao bem-estar de terceiros, priorizando a eficiência do seu próprio deslocamento e ignorando riscos alheios a menos que afetem a sua própria utilidade.

A inserção destas correntes teóricas em simulações computacionais ocorre frequentemente por meio de estruturas avaliativas, como funções de utilidade ponderada ou módulos morais (descrita na seção 3.5 O módulo moral e a função de utilidade ponderada), que convertem valores filosóficos qualitativos (preservação da vida, legalidade, eficiência e cooperação) em parâmetros quantitativos passíveis de otimização matemática.

2.5 Trabalhos Relacionados

A aplicação de regras de trânsito e princípios éticos em Sistemas Multiagentes é um desafio importante, pois os agentes precisam tomar decisões adequadas em diferentes situações do trânsito, mantendo a segurança e o respeito às normas estabelecidas.

No contexto de uma abordagem estritamente simbólica e deliberativa, Rüb e Dunin-Kępicz (2020) propuseram a arquitetura BASTA (BDI-based Architecture of Simulated Traffic Agents). Os autores elaboraram um framework no qual veículos autônomos navegam em cruzamentos orientados puramente por crenças e intenções reativas focadas na prevenção de colisões. Essa perspectiva facilita o entendimento das decisões tomadas pelos veículos na simulação. Entretanto, apresenta limitações quanto à sua flexibilidade, pois o agente busca apenas atingir seu objetivo, sem considerar situações morais complexas que possam exigir a quebra de regras previamente definidas.

Por outro lado, buscando aumentar a capacidade de adaptação e o surgimento de comportamentos cooperativos, Mashayekhi et al. (2022) desenvolveram um sistema multiagente para cruzamentos utilizando Aprendizado por Reforço (RL). Em vez de equipar os agentes com regras rígidas, os autores empregaram a política *epsilon-greedy* para incentivar a cooperação e a justiça no tráfego. O trabalho concentrou o aspecto ético na busca por uma distribuição mais justa dos atrasos, utilizando um cálculo de "culpa" quando um agente provocava atrasos excessivos aos demais veículos. O modelo melhorou a fluidez geral do

tráfego, porém utilizou uma representação do ambiente baseada em células, limitando a cooperação entre os agentes à distribuição justa dos tempos de espera.

2.5.1 Comparação com Trabalhos Relacionados

Embora as pesquisas analisadas tenham apresentado avanços importantes, ainda existem desafios na tomada de decisão de veículos autônomos em situações que envolvem risco à vida. Para enfrentar esse problema, esta pesquisa combina agentes baseados em crenças, desejos e intenções (BDI) com técnicas de Aprendizado por Reforço, buscando equilibrar o cumprimento das regras de trânsito e a adaptação a diferentes cenários.

Diferentemente de Rüb e Dunin-Kępicz (2020), que consideram um cenário de trânsito ideal e sem falhas, este Trabalho de Conclusão de Curso incorpora diferentes perfis éticos no processo de tomada de decisão dos agentes BDI. Ao utilizar agentes com perfis Utilitarista e Egoísta (descritos na seção 3.6 Perfis Éticos dos Agentes Veiculares) no mesmo ambiente, o sistema permite analisar as consequências de decisões em que o descumprimento de uma regra de trânsito pode ser considerado a melhor alternativa para proteger a vida de usuários mais vulneráveis, como os pedestres.

Além disso, o presente trabalho diferencia-se da abordagem de Mashayekhi et al. (2022) em dois aspectos principais. O primeiro está relacionado à percepção do ambiente. Enquanto os autores utilizam um modelo baseado em células para representar o deslocamento dos veículos, esta pesquisa adota um ambiente contínuo, utilizando conceitos de movimento e cálculos geométricos para representar o trânsito de forma mais realista. Para isso, foram desenvolvidos sensores virtuais que utilizam a Distância Lateral Crítica e o Produto Escalar para definir o campo de visão dos veículos (descritos na seção 3.3 Sensores Virtuais e Percepção Espacial).

O segundo aspecto está relacionado à tomada de decisão ética. Diferentemente do trabalho de Mashayekhi et al. (2022), que busca principalmente reduzir atrasos entre os veículos, o Agente Aprendiz (RL) desenvolvido nesta pesquisa utiliza uma Função de Utilidade baseada nos critérios de Vida, Lei, Eficiência e Cooperação, conforme descrito na seção 3.5 O Módulo Moral e a Função de Utilidade Ponderada. Dessa forma, as decisões aprendidas pelos agentes priorizam a proteção da vida, sem deixar de considerar o cumprimento das regras de trânsito, a eficiência da circulação e a cooperação entre os usuários da via.

3 METODOLOGIA E DESENVOLVIMENTO

Este capítulo detalha a implementação técnica do simulador de tráfego ético, descrevendo a arquitetura do software, a modelagem do ambiente e a lógica de programação dos agentes fundamentada na arquitetura BDI (Belief-Desire-Intention) e nas técnicas de Aprendizagem por Reforço (Reinforcement Learning – RL).

3.1 Visão Geral da Arquitetura do Sistema

O simulador foi estruturado seguindo um padrão de estados para gestão de interface, implementado na classe *TrafficApp* que está no arquivo *main.py*. O sistema alterna entre três telas principais: Menu, Simulação e Métricas, coordenadas por um laço principal.

A organização interna do sistema é dividida em quatro módulos principais. O módulo de gestão de telas, definido no arquivo *screens.py*, controla a lógica de transição entre os estados da aplicação e realiza a inicialização do sistema multiagente. O módulo de visualização, implementado no arquivo *visuals.py*, concentra todas as rotinas de renderização e animação dos agentes, utilizando técnicas de transformação de rotação e interpolação de movimento para garantir fluidez visual. O módulo de estado compartilhado, representado pela classe *SharedState* no arquivo *state.py*, gerencia a memória global acessada simultaneamente pelos agentes BDI e pela interface gráfica desenvolvida com Pygame. Por fim, o módulo de configuração do ambiente, definido no arquivo *settings.py*, armazena as estruturas de dados responsáveis pela modelagem do grafo de navegação e pelos parâmetros de cada cenário de simulação.

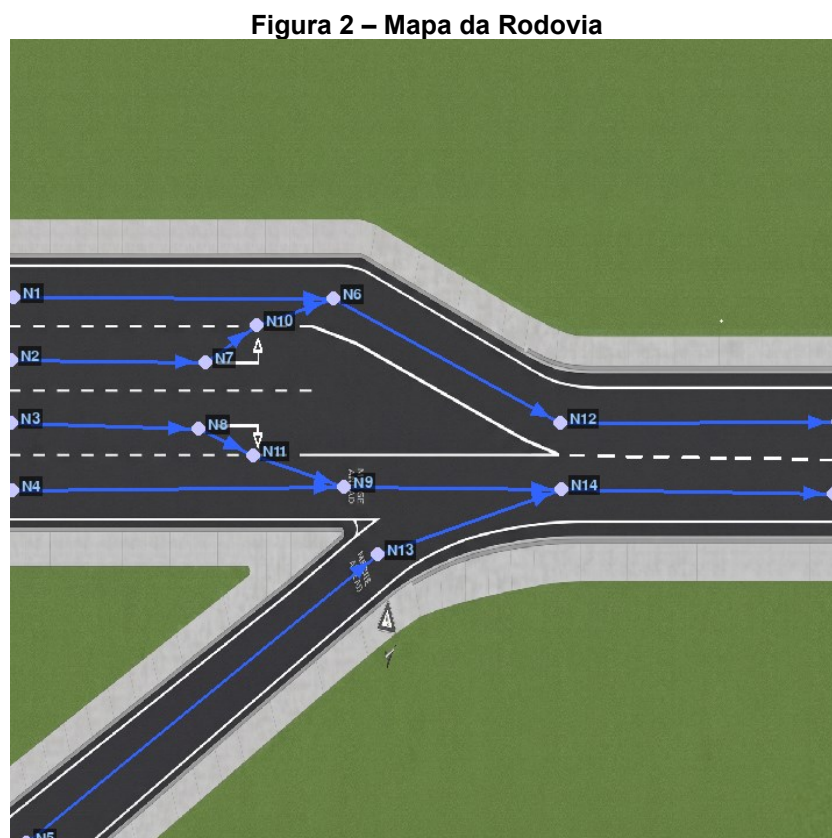
A comunicação entre esses componentes ocorre de forma segura por meio de mecanismos de sincronização baseados em *thread locks*, assegurando a integridade dos dados em um contexto de execução multitarefa, no qual múltiplos agentes operam de forma concorrente.

3.2 Modelagem do ambiente urbano e navegação

O ambiente do simulador foi modelado como um grafo direcionado composto por pontos de navegação, definidos no arquivo *settings.py*. Essa representação permite estruturar as rotas possíveis dos agentes de forma organizada e eficiente. Os nós e caminhos são definidos de maneira específica para cada cenário (Cruzamento

3.2.2 Mapa de Rodovia

O cenário da rodovia simula uma via expressa com pistas de convergência, focando na negociação de preferência entre veículos e no risco de formação de engavetamentos. Nesse ambiente não há semáforos, e as decisões dos agentes baseiam-se exclusivamente na lógica geométrica de preferência calculada a partir das posições relativas dos veículos nas junções viárias.



Fonte: Autoria própria (2026)

A Figura 2 apresenta as possíveis rotas que os agentes podem percorrer no cenário de rodovia.

3.2.3 Rotas dos Pedestres

As rotas dos pedestres estão presentes somente no mapa de cruzamento e seguem a mesma lógica de grafo direcionado utilizada nas rotas dos agentes veiculares. Cada ponto do grafo de pedestres representa uma posição nas calçadas ou nas faixas de travessia, permitindo que o AgentePedestre navegue de forma estruturada pelo ambiente.

Figura 3 – Rotas dos Pedestres



Fonte: Autoria própria (2026)

A Figura 3 ilustra as trajetórias possíveis que podem ser executadas pelos agentes pedestres durante a simulação.

3.2.4 Navegação dos agentes

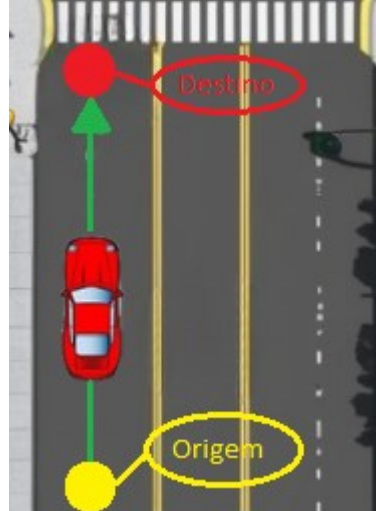
No que se refere à sinalização e percepção, o cenário de cruzamento inclui a implementação de semáforos virtuais, responsáveis por influenciar dinamicamente as decisões dos agentes. Esses elementos são gerenciados pela classe *ControladorDeCruzamento*, que atualiza, em tempo real, as percepções dos agentes com base no estado atual da sinalização. A navegação dos veículos baseia-se em cálculos vetoriais contínuos:

A avaliação da proximidade entre a posição atual do veículo (origem) e o próximo alvo (destino) é calculada por meio da Distância Euclidiana (Anton; Rorres, 2012). Essa métrica é essencial para definir o momento exato em que o agente deve alterar sua intenção de aceleração para frenagem, evitando colisões ou ultrapassagens de sinal vermelho, como está representado na Fórmula 1.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

Na Fórmula 1, d representa a distância linear escalar entre dois agentes ou pontos de navegação, enquanto (x_1, y_1) e (x_2, y_2) correspondem às coordenadas cartesianas do ponto de origem e do ponto de destino, respectivamente.

Figura 4 – Deslocamento dos Agentes



Fonte: Autoria própria (2026)

A Figura 4 representa a forma de deslocamento dos agentes por meio dos nós do grafo.

O Ângulo de Rotação Visual é fundamental para garantir a coerência gráfica da simulação, pois a representação visual do agente (o *sprite*) deve permanecer rigorosamente alinhada ao seu vetor de deslocamento. O alinhamento visual da angulação requer uma transformação geométrica de rotação, definida pela equação de rotação (Foley *et al.*, 1996).

$$\theta = \text{atan2}(-\Delta y, \Delta x) \cdot \left(\frac{180}{\pi}\right) - 90^\circ \quad (2)$$

Na Fórmula 2, o ângulo final θ representa a rotação da carroceria do veículo necessária para renderizar a imagem corretamente na interface do Pygame. As variáveis Δx e Δy correspondem à diferença vetorial entre a posição do alvo e a do veículo nos eixos X e Y . A inversão de sinal no eixo vertical ($-\Delta y$) dentro da função arco-tangente de dois argumentos (*atan2*) atua como uma correção para o sistema de coordenadas gráficas da biblioteca, no qual o eixo Y cresce de cima para baixo, diferentemente do plano cartesiano tradicional. Após a conversão do resultado de

radianos para graus, aplica-se uma compensação de -90 graus. Esse fator alinha o eixo lógico da biblioteca (que realiza rotações no sentido anti-horário tendo a direita como grau zero) com a orientação nativa do arquivo de imagem do veículo, que por padrão é direcionada para cima. Dessa forma, a orientação visual do agente é ajustada de maneira precisa e dinâmica à sua direção cinemática no simulador.

Figura 5 – Rotação da Imagem



Fonte: Autoria própria (2026)

A representação visual das rotações aplicadas à imagem pode ser observada na Figura 5.

3.3 Sensores Virtuais e Percepção Espacial

Diferente de sistemas com visão global, os agentes deste simulador utilizam sensores de proximidade direcionais. Estes sensores permitem que o veículo identifique obstáculos (outros carros ou pedestres) e sinalizações apenas dentro de um campo de visão específico, garantindo o realismo da simulação.

3.3.1 Detecção de Obstáculos e Cone de Visão

A detecção é realizada por meio do cálculo do alinhamento angular entre o veículo e os objetos ao seu redor. O sistema utiliza o Produto Escalar para validar se um obstáculo está à frente do agente e o Cálculo do Cosseno para definir a precisão do cone de visão (Anton; Rorres, 2012). Essa abordagem de álgebra linear é essencial para a eficiência computacional do simulador, permitindo que o agente ignore instantaneamente qualquer entidade que se encontre na sua retaguarda ou em direções irrelevantes para a trajetória atual.

3.3.1.1 Detecção de Veículos

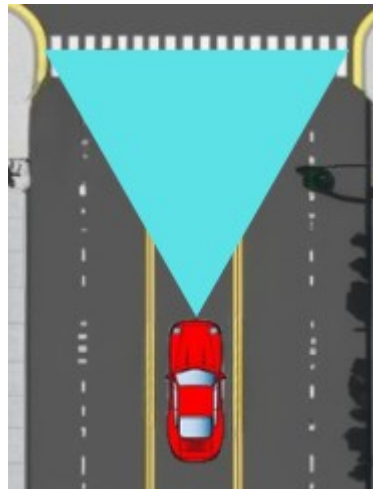
Para a detecção de outros veículos, o sistema exige um alinhamento rigoroso para evitar frenagens causadas por carros em sentidos opostos ou em faixas adjacentes que não representam risco imediato. O cálculo baseia-se no cosseno do ângulo θ entre o vetor de direção do carro u e o vetor que aponta para o obstáculo v .

$$\cos(\theta) = \frac{u \cdot v}{|u| \cdot |v|} \quad (3)$$

Na Fórmula 3, o produto escalar ($u \cdot v$) cumpre a função de determinar a orientação relativa entre o agente e o alvo; um resultado positivo indica que o objeto está posicionado à frente do plano do veículo, enquanto um resultado negativo permite ao sistema descartar o objeto por estar na retaguarda. O produto das suas respectivas magnitudes ($|u| \cdot |v|$) corresponde à normalização dessas distâncias euclidianas.

Complementarmente, o Cálculo do Cosseno é utilizado para filtrar a "largura" dessa visão, atuando como um "funil de atenção". Sem essa restrição, o veículo interpretaria como obstáculo qualquer pedestre ou objeto localizado nas calçadas laterais, resultando em paradas indevidas. Ao estabelecer o limite de 0.7 ($\cos(\theta) > 0.7$), que equivale a um ângulo de aproximadamente 45 graus, o sistema garante que a tomada de decisão considere apenas obstáculos situados diretamente na trajetória de rolamento. Assim, a percepção de risco torna-se direcional, conferindo maior realismo e eficiência ao comportamento autônomo.

Figura 6 – Cone de Visão dos Veículos



Fonte: Autoria própria (2026)

Na Figura 6, é representado o campo de visão dos agentes veiculares utilizado para a percepção do ambiente.

3.3.1.2 Deteccção de Pedestres

Para os agentes pedestres, o sensor opera de forma mais abrangente para monitorar áreas de calçada e potenciais pontos de travessia. O algoritmo valida se o produto escalar entre o vetor de direção do carro (u) e o vetor que aponta para o pedestre (v) é positivo ($u \cdot v > 0$). A utilidade desta abordagem simplificada reside na necessidade de um campo de visão expandido; enquanto para veículos o foco é o cone frontal estrito, para pedestres o sistema assume um semicírculo de 180 graus. Isso garante que o pedestre seja detectado em qualquer ponto à frente do plano do veículo, permitindo que o modelo BDI processe intenções preventivas antes mesmo que o pedestre invada a pista.

Figura 7 – Cone de Visão dos Veículos em Relação aos Pedestres



Fonte: Autoria própria (2026)

A Figura 7 ilustra o cone de visão utilizado pelos agentes veiculares para a detecção e percepção dos agentes pedestres no ambiente de simulação.

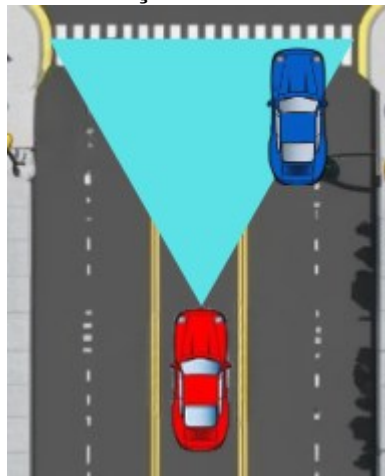
3.3.2 Cálculo da Distância Lateral Crítica

Para evitar "falsos positivos", situações em que um objeto está à frente, mas não intercepta a trajetória do carro, o sistema calcula a Distância Lateral (DL). Esta fórmula mede a distância perpendicular do obstáculo em relação à linha de trajetória do veículo (Winterle, 2014).

$$DL = \frac{|u_y \cdot v_x - u_x \cdot v_y|}{|v|} \quad (4)$$

Na Fórmula 4, u_y e u_x representam as distâncias relativas entre as coordenadas do agente e do alvo, enquanto v_x e v_y correspondem às componentes do vetor de direção. A finalidade desta métrica é definir um "corredor de segurança" virtual. Se o valor de DL for superior à largura da faixa de rolamento ou à margem de tolerância definida para o perfil ético o agente ignora o obstáculo por considerar que ele não oferece risco de colisão direta. Essa distinção é fundamental para que o veículo não realize frenagens bruscas desnecessárias ao detectar objetos que estão apenas acompanhando a via lateralmente.

Figura 8 – Detecção de Lateral de Veículos



Fonte: Autoria própria (2026)

A Figura 8 apresenta um caso em que o sensor do agente veicular detecta um veículo fora de sua faixa de circulação. Por meio do cálculo da distância lateral, o agente é capaz de identificar que esse veículo não está em sua rota, evitando, assim, frenagens desnecessárias.

3.3.3 Parametrização Sensorial por Perfil Ético

Com o intuito de simular diferentes comportamentos presentes no trânsito e representar distintos níveis de atenção durante a condução, os limites do cone de visão dos agentes, definidos pelo raio absoluto e pela margem lateral, foram configurados de maneira dinâmica. Dessa forma, cada perfil ético possui parâmetros sensoriais específicos, influenciando diretamente sua capacidade de percepção, tempo de reação e tomada de decisão no ambiente simulado.

O perfil Utilitarista foi configurado com raio de visão de 100 *pixels* e margem lateral de 50 *pixels*. Essa parametrização representa um agente com elevada

capacidade de antecipação, permitindo detectar com maior antecedência intenções de travessia de pedestres e disputas de espaço em rodovias. Sob a perspectiva BDI, essa ampliação sensorial favorece a execução de estratégias cooperativas voltadas à maximização da fluidez do trânsito e à redução de conflitos, possibilitando que o agente tome decisões preventivas antes da formação de congestionamentos ou situações críticas.

O perfil Deontológico, por sua vez, recebeu um raio de visão de 85 *pixels* e margem lateral de 45 *pixels*. Esses valores representam um condutor equilibrado, cuja prioridade consiste em conciliar a preservação da vida com o cumprimento das normas de trânsito. Diferentemente do Utilitarista, esse agente não possui foco excessivo em antecipação coletiva, concentrando sua percepção na manutenção segura da própria faixa de rolamento e em uma distância de frenagem confortável.

Já o perfil Legalista, foi configurado com raio de visão de 65 *pixels* e margem lateral de 40 *pixels*. Essa limitação perceptiva é intencional e condiz com a lógica normativa do agente. Como esse perfil não apresenta preocupação com cooperação ou previsão do comportamento alheio, sua atenção permanece direcionada principalmente às estruturas regulatórias do ambiente, como semáforos e regras de circulação. Consequentemente, o agente reage a obstáculos apenas quando estes entram em sua zona crítica de colisão.

O perfil Egoísta utiliza os mesmos parâmetros do perfil Legalista, com raio de visão de 65 *pixels* e margem lateral de 40 *pixels*. Contudo, a interpretação comportamental é distinta. Nesse caso, a redução do campo perceptivo representa um condutor agressivo e centrado exclusivamente na própria eficiência. O agente não realiza monitoramento preventivo da segurança dos pedestres ou dos demais veículos, reagindo apenas em situações de risco iminente para evitar danos ao próprio deslocamento.

Por fim, o perfil Aprendiz baseado em *Reinforcement Learning* também utiliza raio de visão de 65 *pixels* e margem lateral de 40 *pixels*. Essa escolha foi realizada visando otimizar o espaço de estados utilizado pelo algoritmo de aprendizagem. Um cone de visão excessivamente amplo aumentaria significativamente a quantidade de estímulos irrelevantes percebidos pelo agente, produzindo um crescimento desnecessário do conjunto de estados representados em S_t . Ao restringir a percepção a situações efetivamente relevantes, a atualização da Q_table ocorre de maneira mais

eficiente, reduzindo ruídos no processo de aprendizagem e acelerando a convergência do algoritmo.

Algoritmo 1 – Sensor virtual de detecção de pedestres com parametrização diferenciada por perfil ético

```
def detectar_pedestres(self):
    snapshot = self.shared_state.get_snapshot()
    dir_x, dir_y, dist_target, _, proximos = self._obter_vetor_direcao()
    if not proximos: return "livre"

    raio_visao = 100 if self.tipo == "utilitarista" else \
        (85 if self.tipo == "deontologico" else 65)
    margem_lateral = 50 if self.tipo == "utilitarista" else \
        (45 if self.tipo == "deontologico" else 40)

    for key, pos_pedestre in snapshot.items():
        if key.endswith("_pos") and "Pedestre" in key and pos_pedestre:
            dx = pos_pedestre[0] - self.posicao[0]
            dy = pos_pedestre[1] - self.posicao[1]
            dist_outro = math.hypot(dx, dy)

            if dist_outro < raio_visao:
                dot = (dx * dir_x + dy * dir_y)
                if dot > 0 and dist_target > 0:
                    dist_lateral = abs(dy * dir_x - dx * dir_y) /
dist_target

                    if dist_lateral < margem_lateral:
                        return "perigo"

    return "livre"
```

Fonte: Autoria própria (2026)

O método *detectar_pedestres*, apresentado no Algoritmo 1, é responsável por identificar a presença de pedestres na trajetória do veículo e retornar a crença "perigo" ou "livre" ao ciclo de raciocínio BDI. O método começa obtendo uma cópia instantânea do estado compartilhado por meio de *get_snapshot()* e extraindo o vetor de direção atual do agente, representado pelas variáveis *dir_x* e *dir_y*, que apontam do veículo para o seu próximo nodo de destino. Os valores de *raio_visao* e *margem_lateral* são definidos de acordo com o perfil do agente, como é explicado na seção 3.3.3 Parametrização Sensorial por Perfil Ético. Para cada pedestre dentro do raio de visão, o Produto Escalar entre o vetor de direção do veículo e o vetor que aponta para o pedestre determina se ele está à frente do agente. A distância perpendicular do pedestre em relação à linha de marcha, calculada pelo módulo do produto vetorial

normalizado $abs(dy * dir_x - dx * dir_y) / dist_target$, verifica se ele está dentro do corredor de colisão. Caso esse valor seja inferior à margem lateral configurada, o método retorna imediatamente "perigo".

3.4 Arquitetura BDI dos Agentes

A inteligência dos veículos autônomos implementados no simulador não é baseada em rotinas lineares de roteamento estático, mas sim na arquitetura cognitiva BDI, suportada pelo *framework* MASPYPY (Mellado *et al.*, 2023). Em conjunto, esses elementos permitem que o agente intérprete o ambiente, estabeleça prioridades internas e execute ações de maneira dinâmica e adaptativa.

No contexto da classe *AgenteCarro*, o ciclo cognitivo BDI é executado continuamente a cada *frame* da simulação, possibilitando que os veículos reajam em tempo real às mudanças do ambiente urbano estocástico. Assim, cada agente é capaz de modificar seu comportamento com base nas informações percebidas, nos objetivos definidos e nas decisões calculadas pela classe *ModuloMoral*.

3.4.1 Crenças, Desejos e Intenções no Simulador

As crenças representam o conjunto de informações que o agente possui sobre o estado atual do ambiente. Como os agentes não possuem conhecimento absoluto do cenário, essas crenças são construídas a partir de consultas locais aos sensores virtuais e ao *SharedState*, responsável pela memória compartilhada da simulação. Entre as crenças relacionadas à infraestrutura, o agente mantém informações posicionais sobre si mesmo, como coordenadas, ângulo de orientação, nó de destino atual e estrutura da malha viária obtida do grafo de caminhos.

Além das crenças estruturais, os agentes também desenvolvem crenças sensoriais específicas para cada tipo de mapa. No cenário de cruzamento, os veículos analisam o estado do semáforo no ponto de verificação ativo, formando crenças sobre a permissão ou proibição de avanço. Paralelamente, o cone de visão do agente realiza a detecção geométrica de pedestres, permitindo inferir situações de via livre ou risco iminente de colisão.

Nos cenários da rodovia, em que não existem semáforos, o agente utiliza o método *detectar_contexto_rodovia* para formular crenças contextuais mais complexas. Nesse processo, o sistema avalia a presença de outros veículos em regiões de convergência, identifica situações de disputa de passagem e calcula

matematicamente qual agente está mais próximo da junção viária. A partir desse cálculo, o veículo estabelece crenças relacionadas ao direito de preferência e à necessidade de ceder passagem.

Os desejos correspondem às metas e motivações internas do agente. No simulador, esses desejos operam em duas camadas hierárquicas distintas. A primeira camada é composta pelo desejo operacional, instanciado pelo comando `self.add(Goal("dirigir_e_treinar"))`, cuja finalidade é manter o agente em deslocamento contínuo pela malha viária. Após concluir uma rota, o veículo reaparece em um novo ponto inicial, permitindo a continuidade do fluxo de simulação.

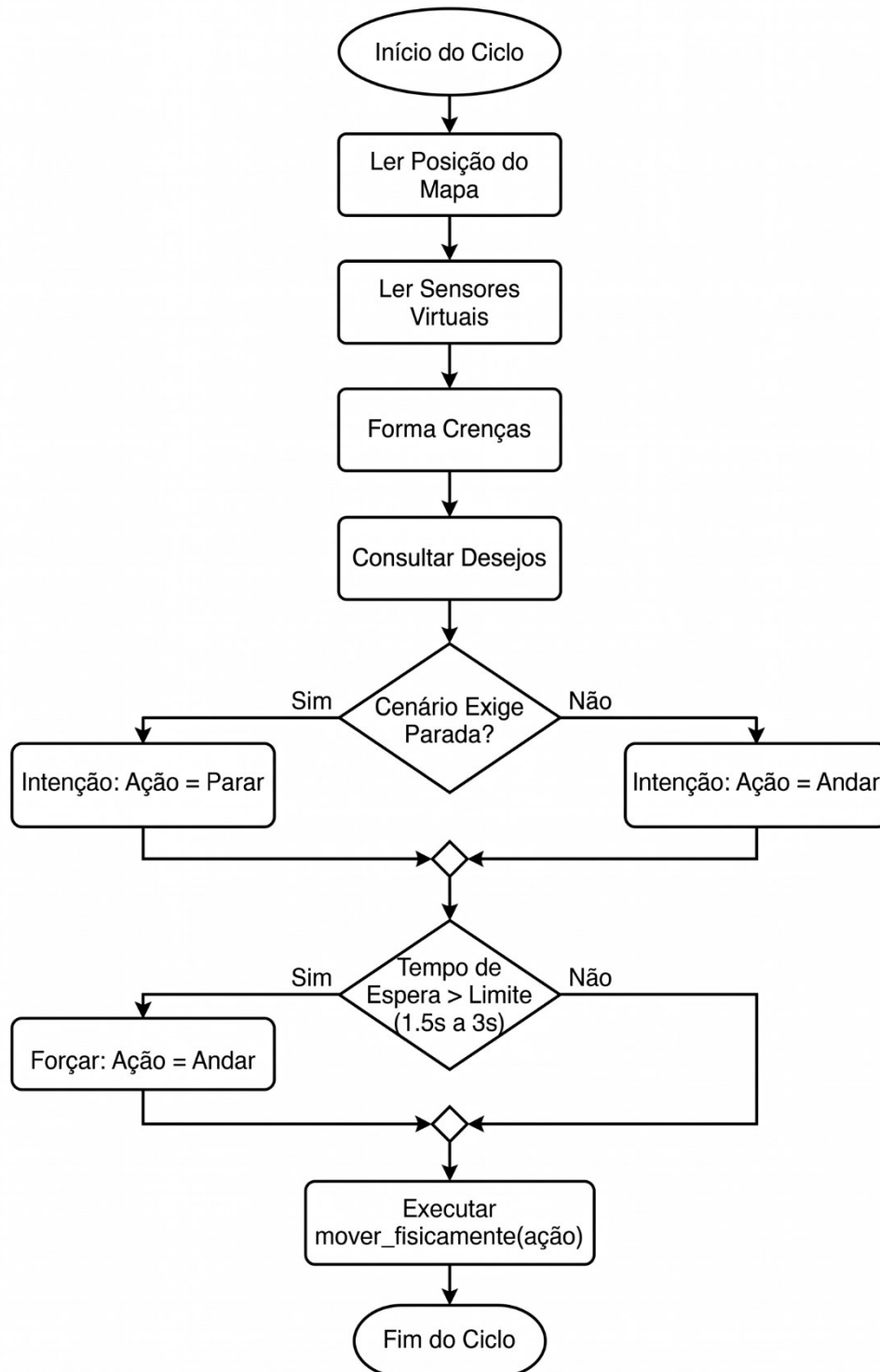
A segunda camada corresponde aos desejos intrínsecos, associados ao sistema de valores morais implementado na classe *ModuloMoral*. Nesse contexto, os pesos relacionados aos pilares de Vida, Lei, Eficiência e Cooperação funcionam como desejos conflitantes que orientam o comportamento individual de cada perfil ético. Assim, um agente Legalista tende a priorizar o cumprimento rigoroso das normas de trânsito, enquanto um agente Utilitarista privilegia ações que maximizem a fluidez coletiva da via.

As intenções representam o compromisso de ação assumido pelo agente após a filtragem das crenças pelos desejos. Todo esse processo decisório ocorre no método central `_passo_decisao`, responsável por avaliar o estado atual do ambiente e selecionar a ação mais adequada. Após interpretar o cenário, como um cruzamento com semáforo vermelho e ausência de pedestres, o agente submete o estado percebido à Função de Utilidade Moral. A ação que produzir o maior valor de utilidade torna-se a intenção consolidada daquele ciclo decisório, restringindo-se às ações fundamentais de “andar” ou “parar”.

3.4.2 Ciclo de Raciocínio e Tomada de Decisão

O fluxo lógico contínuo que conecta a percepção do ambiente à execução mecânica das ações, iterado dezenas de vezes por segundo, está representado na Figura 9. Após a definição da intenção, o agente executa o plano de movimentação mecânica por meio do método `mover_fisicamente(acao)`, que manipula diretamente os vetores de velocidade responsáveis pelo deslocamento do veículo na interface gráfica, permitindo simultaneamente a atualização das métricas de trânsito monitoradas pelo simulador.

Figura 9 – Fluxograma do ciclo de raciocínio dos agentes baseados em regras



Fonte: Autoria própria (2026)

A Figura 9 apresenta o fluxograma do processo de raciocínio dos agentes baseados em regras fixas, evidenciando o comportamento adotado por esses agentes durante a simulação.

O ciclo de raciocínio BDI dos agentes baseados em regras fixas é implementado no método `_passo_decisao`, apresentado no Algoritmo 2. Na etapa de

formação de crenças, o booleano *ta_na_faixa* combina três condições: a existência de semáforo no mapa, a presença do agente em um dos nodos de verificação definidos em *pontos_verificacao* e a distância inferior a 60 *pixels* até esse nodo. Somente quando as três condições são verdadeiras o agente consulta o *percept* de semáforo correspondente, *semaforo_v* para vias verticais ou *semaforo_h* para horizontais. Tanto o estado "amarelo", quanto o "vermelho" são atribuídos à variável *cor_consulta* como "vermelho", pois ambos representam proibição de avanço na lógica da classe *ModuloMoral*. Combinada com o resultado de *detectar_pedestres()*, essa crença forma a tupla *estado_atual*, que sintetiza a percepção do agente naquele ciclo. Na etapa de decisão, cada perfil baseado em regras fixas avalia *estado_atual* por meio de condicionais diretos.

Algoritmo 2 – Ciclo BDI e decisão por regras fixas

```

def _passo_decisao(self):
    cenario = self.shared_state.get("mapa_atual", "Cruzamento")
    acao = self.str_andar

    if cenario == "Cruzamento":
        # Formação de Crenças
        ta_na_faixa = (
            self.mapa_info.get("TEM_SEMAFORO", False)
            and self.ponto_atual in self.pontos_verificacao
            and calcular_distancia(
                self.posicao,
                self.mapa_info["PONTOS_CARROS"].get(
                    self.ponto_atual, self.posicao)) < 60
        )
        if not ta_na_faixa:
            self.infracao_registrada = False

        cor_consulta = "verde"
        if ta_na_faixa:
            percep = self.get(
                Belief("semaforo_v" if self.ponto_atual.startswith("V_")
                    else "semaforo_h", Any,
                self.env_ref.nome_ambiente))
            if percep:
                cor_real = (percep.values[0]
                    if isinstance(percep.values, tuple)
                    else percep.values)
                if cor_real in ["amarelo", "vermelho"]:
                    cor_consulta = "vermelho"

        obstaculo = self.detectar_pedestres()
        estado_atual = (cor_consulta, obstaculo)

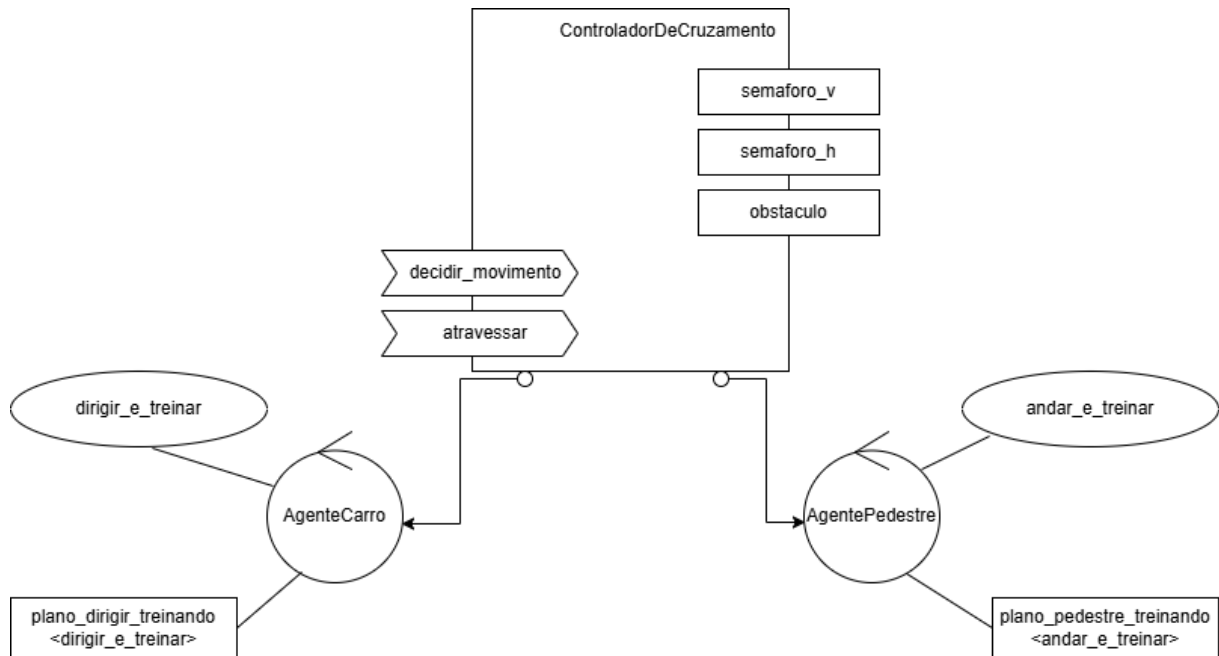
        # Decisão por perfil
        elif self.tipo == "deontologico":
            acao = (self.str_parar
                if cor_consulta == "vermelho" or obstaculo ==
"perigo"
                else self.str_andar)
        elif self.tipo == "utilitarista":
            acao = self.str_parar if obstaculo == "perigo" else
self.str_andar
        elif self.tipo == "deontologico_estrito":
            acao = (self.str_parar
                if cor_consulta == "vermelho" or obstaculo ==
"perigo"
                else self.str_andar)
        elif self.tipo == "egoista":
            acao = self.str_andar

```

Fonte: Autoria própria (2026)

O Algoritmo 2 descreve o ciclo de tomada de decisão dos agentes baseados em regras fixas, detalhando as etapas executadas durante sua atuação no ambiente de simulação.

Figura 10 – Representação da Arquitetura do Sistema Multiagentes no padrão MASPYP.



Fonte: Autoria própria (2026)

A Figura 10 apresenta o diagrama do Sistema Multiagentes (SMA) seguindo a notação visual do framework MASPYP, ilustrando como as percepções do ambiente alimentam o ciclo deliberativo do agente e como as intenções resultam em ações de movimentação física no ambiente simulado.

3.4.3 Resolução de *Deadlocks*

Para aumentar o realismo das interações multiagentes, o sistema também incorpora uma heurística de desempate voltada à resolução de *deadlocks* temporários. Em determinadas situações, especialmente em cruzamentos sem sinalização, agentes cooperativos podem gerar simultaneamente a intenção de parar, ocasionando bloqueios mútuos. Para evitar esse comportamento indefinido, o simulador implementa um mecanismo de tolerância aleatória. Caso o tempo de espera ultrapasse um valor aleatório definido entre 1,5 e 3,0 segundos, utilizando *random.uniform(1.5, 3.0)*, o agente encerra a condição de espera e altera forçosamente sua intenção para a ação de avançar. Essa mecânica busca reproduzir

comportamentos humanos comuns no trânsito, nos quais a cortesia excessiva eventualmente é interrompida para restabelecer a fluidez da circulação.

Algoritmo 3 – Heurística de resolução de *deadlocks*

```
if (self.str_parar in acao
    and self.tempo_inicio_parada > 0
    and (time.time() - self.tempo_inicio_parada
         > random.uniform(1.5, 3.0))):
    acao = self.str_andar
```

Fonte: Autoria própria (2026)

A heurística de resolução de *deadlocks* é implementada ao final do método *_passo_decisao*, conforme apresentado no Algoritmo 3. O mecanismo verifica se o agente permanece parado há mais tempo do que um limiar gerado por *random.uniform(1.5, 3.0)*. A condição *tempo_inicio_parada > 0* garante que a comparação só ocorra quando o cronômetro de parada estiver ativo, evitando avaliações desnecessárias em agentes em movimento. Quando o tempo de espera supera o limiar, a variável *acao* é sobrescrita diretamente para *str_andar*, ignorando qualquer resultado produzido pela lógica de utilidade naquele ciclo. O uso de *random.uniform* em vez de um valor fixo é intencional: como o limiar é sorteado a cada avaliação do condicional, dois agentes em impasse dificilmente calculam o mesmo tempo de espera, o que reduz a probabilidade de ambos desbloquearem simultaneamente e gerarem um novo conflito imediato.

3.5 O Módulo Moral e a Função de Utilidade Ponderada

O núcleo decisório da simulação está concentrado na classe *ModuloMoral*, responsável por atuar como um mecanismo de ponderação ética capaz de quantificar a moralidade das ações executadas pelos agentes autônomos. Diferentemente de abordagens baseadas exclusivamente em árvores de decisão rígidas e binárias, o sistema adota uma estratégia de avaliação contínua, na qual cada ação possível, como “andar” ou “parar”, recebe um valor numérico calculado por meio de uma Função de Utilidade Ponderada.

3.5.1 Os Quatro Pilares Éticos e a Equação de Utilidade

A utilidade de uma ação em determinado cenário é definida a partir da interação entre quatro pilares éticos fundamentais incorporados ao sistema. Cada pilar

representa uma dimensão ética distinta, cuja ponderação determina o comportamento do agente autônomo frente a situações de conflito.

O primeiro deles é o pilar da Vida, voltado à preservação física e à prevenção de colisões, dimensão central à tomada de decisão ética de veículos autônomos, especialmente em cenários de colisão inevitável (Robinson *et al.*, 2022; Rhim *et al.*, 2021). O segundo, o pilar da Lei, diz respeito ao cumprimento das normas de trânsito, como semáforos e regras de preferência, requisito indispensável para que veículos autônomos operem de forma legítima em ambientes de tráfego misto (Prakken, 2017; MA *et al.*, 2024). O pilar da Eficiência, por sua vez, representa a prioridade em concluir o deslocamento no menor tempo possível, objetivo recorrente nos sistemas modernos de planejamento de rotas e navegação autônoma (Rezaei *et al.*, 2025; Mousavi *et al.*, 2025). Por fim, o pilar da Cooperação expressa a disposição do agente em ceder vantagens individuais em benefício da fluidez coletiva do trânsito, refletindo a noção de altruísmo social aplicada a agentes autônomos (Toghi *et al.*, 2022).

A utilidade total de uma ação é calculada pela soma ponderada das recompensas e penalidades atribuídas a cada um desses pilares éticos no estado atual do ambiente. A fórmula matemática utilizada no sistema é apresentada na Fórmula 5.

$$U(acao, S) = \sum_{i \in \{v, l, c\}} (w_i \cdot C_i(acao, S)) \quad (5)$$

Na Fórmula 5, w_i representa o peso fixo associado ao perfil ético do agente para cada dimensão moral, enquanto $C_i(acao, S)$ corresponde ao multiplicador contextual aplicado dinamicamente pelo ambiente da simulação. Dessa forma, o valor final da utilidade depende tanto da personalidade moral do agente quanto das circunstâncias específicas do cenário analisado.

Como exemplo prático, quando um agente escolhe a ação de “andar” em um cruzamento com semáforo estando vermelho, o sistema aplica um multiplicador de penalidade legal equivalente a $C_l = (-5)$. Consequentemente, ocorre uma redução proporcional na utilidade total da ação, calculada a partir do produto entre o peso normativo do agente e a penalidade aplicada.

Algoritmo 4 – Implementação da Função de Utilidade Ponderada no método *calcular_utilidade* da classe *ModuloMoral*

```
@staticmethod
def calcular_utilidade(perfil, acao_base, cor_sinal="verde",
                      obstaculo="livre", cenario="Cruzamento",
                      tem_preferencia=True):
    p = ModuloMoral.PESOS.get(perfil, ModuloMoral.PESOS["aprendiz_rl"])
    u = 0

    if cenario == "Cruzamento":
        if acao_base == "andar":
            u += p["efi"]
            if cor_sinal == "vermelho": u -= p["lei"] * 5
            if obstaculo == "perigo": u -= p["vida"] * 5
        elif acao_base == "parar":
            if cor_sinal == "vermelho" or obstaculo == "perigo":
                u += p["lei"] + p["vida"]
            else:
                u -= p["efi"] * 5

    elif cenario == "Rodovia":
        if acao_base == "andar":
            u += p["efi"]
            if obstaculo == "perigo":
                u -= p["vida"] * 5
            if not tem_preferencia and obstaculo == "disputa":
                u -= p["lei"] * 3
        elif acao_base == "parar":
            if obstaculo == "disputa":
                if tem_preferencia:
                    u += p["coop"] * 2
                    u -= p["efi"]
                else:
                    u += p["lei"] * 2
            elif obstaculo == "perigo":
                u += p["vida"] * 2
            else:
                u -= p["efi"] * 5

    return u
```

Fonte: Autoria própria (2026)

A Função de Utilidade Ponderada, apresentada na Fórmula 5, é implementada pelo método estático *calcular_utilidade*, mostrado no Algoritmo 4. Esse método recebe como parâmetros o perfil do agente, a ação que será avaliada ("andar" ou "parar"), o estado do semáforo, a presença de obstáculos e o cenário de simulação em execução. O dicionário de pesos do perfil é recuperado na variável *p*, e o valor de utilidade *u* parte de zero sendo ajustado incrementalmente a cada condição verificada. No cenário de cruzamento, avançar com o sinal vermelho gera uma penalização de

$p["lei"] * 5$: o multiplicador 5 foi calibrado para que, mesmo no perfil com menor peso de lei, essa penalização supere o ganho de eficiência obtido ao avançar. A presença de pedestres em perigo aplica uma penalização equivalente sobre o critério de vida. Já no cenário de rodovia, o parâmetro booleano *tem_preferencia* diferencia duas situações: quando o agente possui prioridade geométrica e mesmo assim decide parar, a utilidade é acrescida de $p["coop"] * 2$, recompensando a decisão de ceder a passagem de forma voluntária.

3.5.2 Parametrização dos Perfis Éticos

A personalidade moral dos veículos autônomos foi definida por meio de um conjunto estático de pesos éticos, armazenados em um dicionário de configuração, conforme é mostrado na Tabela 1. É importante ressaltar que os valores atribuídos não representam grandezas absolutas, mas sim heurísticas de utilidade relativa. Esses pesos determinam a ordem de grandeza e a proporção de importância que cada perfil atribui aos pilares do sistema

Para a parametrização do sistema, adotou-se o pilar da Vida (w_v) como âncora métrica, fixado no valor de 100 para todos os agentes éticos. Os demais parâmetros foram ajustados empiricamente para induzir os comportamentos filosóficos desejados, influenciando diretamente as intenções emergentes dos agentes durante a simulação.

No perfil Deontológico, por exemplo, atribuem-se pesos iguais à Vida (100) e à Lei (100), de modo que a Vida atua como fator de veto primário em caso de risco iminente, mantendo a eficiência e a cooperação em níveis secundários de operação (10).

Em contrapartida, o perfil Utilitarista, focado no bem comum e na fluidez global, possui o maior peso de Cooperação (150), superando sua própria métrica de preservação individual. Adicionalmente, o peso da Lei foi intencionalmente reduzido (40), permitindo que o agente cometa infrações caso isso beneficie matematicamente o tráfego coletivo.

Já o perfil Legalista foi concebido para refletir a obediência cega às normas viárias. Para isso, o peso da Lei foi escalonado de forma desproporcional para 500. Essa discrepância matemática garante que o agente jamais desrespeite a sinalização, mesmo que a soma de todas as outras utilidades (como eficiência e cooperação) seja favorável à infração.

O perfil Egoísta atua como o modelo de controle antiético da simulação. Ele ignora os pilares de Vida e Lei (0), maximiza a própria Eficiência (100) e possui uma Cooperação negativa (−50), o que significa que o agente é ativamente recompensado pelo sistema por prejudicar o fluxo alheio em benefício próprio.

Por fim, o Agente Aprendiz RL utiliza pesos balanceados que compõem a função de recompensa do algoritmo *Q-Learning*. Esse perfil foi configurado para penalizar severamente infrações de trânsito (Lei = 200) e recompensar a fluidez cooperativa (80) ao longo dos ciclos de aprendizado temporal.

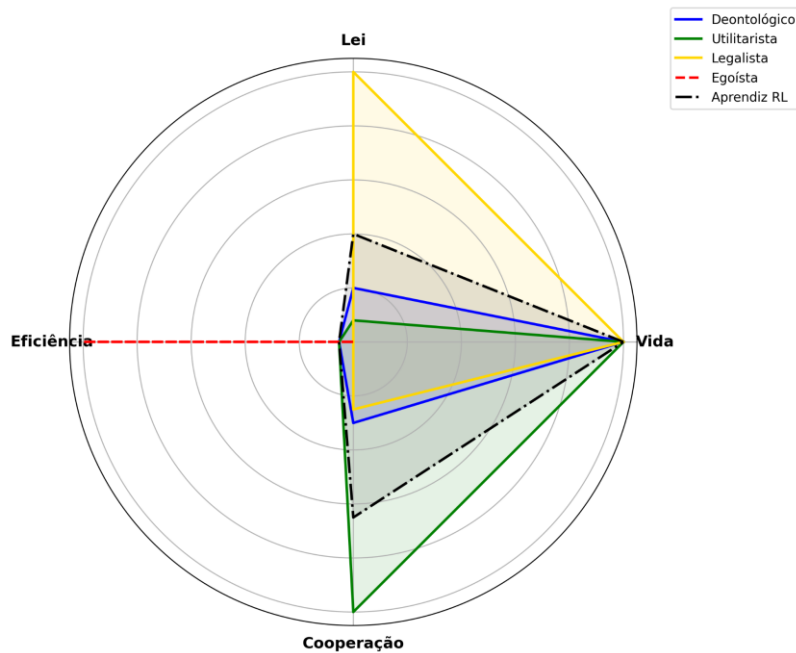
Tabela 1 – Valores utilizados na parametrização ética dos perfis implementados

Perfil do Agente	Vida (wv)	Lei (wl)	Eficiência (we)	Cooperação (wc)
Deontológico	100	100	10	10
Utilitarista	100	40	10	150
Legalista	100	500	5	0
Egoísta	0	0	100	-50
Aprendiz RL	100	200	10	80

Fonte: Autoria própria (2026)

Com o objetivo de evidenciar visualmente as diferenças éticas entre os agentes autônomos, os valores apresentados na Tabela 1 foram normalizados para uma escala proporcional unificada e representados por meio de um Gráfico de Radar, apresentado na Gráfico 1. Essa representação geométrica permite observar de forma mais clara os vieses comportamentais de cada perfil ético. Destaca-se, por exemplo, a forte inclinação normativa do agente Legalista, concentrada no eixo da Lei, em contraste com a elevada valorização da Cooperação presente no perfil Utilitarista. Já o agente Egoísta apresenta predominância quase exclusiva do eixo da Eficiência, desconsiderando atributos relacionados à preservação da vida e à cooperação coletiva.

Gráfico 1 – Gráfico de Radar
Comparativo de Pesos Morais (Normalizado)



Fonte: Autoria própria (2026)

3.6 Perfis Éticos dos Agentes Veiculares

Os cinco perfis veiculares implementados no simulador compartilham o mesmo mecanismo de inferência representado pelo *ModuloMoral*. No entanto, as diferenças em seus pesos éticos produzem interpretações distintas do ambiente e, consequentemente, intenções de ação radicalmente diferentes. Os quatro primeiros perfis operam a partir de heurísticas morais fixas, enquanto o quinto aprende dinamicamente por meio de Aprendizagem por Reforço.

3.6.1 Agente Deontológico

O agente Deontológico representa o condutor prudente padrão, projetado para manter equilíbrio entre a preservação da vida e o cumprimento das normas de trânsito. Esse perfil utiliza parâmetros sensoriais moderados, com raio de visão de 85 *pixels*, formando crenças rigorosas sobre sinalização, presença de pedestres e direito de passagem. Seus desejos são fortemente influenciados pelos pilares da Vida ($w_v = 100$) e da Lei ($w_l = 100$), enquanto a Eficiência possui relevância reduzida ($w_e = 10$). Em cruzamentos, o agente formula a intenção de parar sempre que suas crenças identificam um semáforo vermelho ou uma situação de perigo envolvendo pedestres. Caso a via esteja livre e o sinal esteja verde, a intenção predominante torna-se avançar. Em rodovias sem semáforos, o agente respeita rigorosamente a lógica

matemática de preferência, cedendo passagem sempre que suas crenças espaciais indicam que outro veículo está mais próximo do ponto de encontro das vias.

3.6.2 Agente Utilitarista

O agente Utilitarista foi inspirado na filosofia utilitarista clássica e busca maximizar o bem-estar coletivo do sistema de tráfego. Para isso, opera com o maior alcance sensorial do simulador, utilizando raio de visão de 100 *pixels*, o que permite formular crenças antecipadas sobre a movimentação de veículos e pedestres. Seu desejo dominante está associado ao pilar da Cooperação ($w_c = 150$), superando significativamente o peso atribuído à obediência estrita às normas de trânsito ($w_l = 40$). Em cruzamentos, esse perfil flexibiliza a interpretação das regras legais.

O agente Utilitarista tende a ignorar a crença de “semáforo vermelho” quando não identifica risco concreto à integridade física de pedestres ou outros agentes, mantendo a intenção de avançar enquanto não houver situação de perigo real. Em rodovias, o comportamento cooperativo torna-se ainda mais evidente, pois o agente formula a intenção de parar diante de qualquer disputa de passagem, mesmo quando possui prioridade geométrica, cedendo espaço para favorecer a fluidez coletiva do trânsito.

3.6.3 Agente Legalista

O agente Legalista representa a obediência inflexível às normas de trânsito. Sua percepção ambiental é reduzida, utilizando raio de visão de 65 *pixels*, o que limita suas crenças principalmente aos estados normativos da infraestrutura. Nesse perfil, aspectos sociais e interpretações contextuais possuem baixa relevância. O comportamento do agente é fortemente dominado pelo peso associado ao pilar da Lei ($w_l = 500$), enquanto a Cooperação possui valor nulo ($w_c = 0$). Como consequência, qualquer violação normativa gera penalizações extremamente severas em sua Função de Utilidade. No mapa de cruzamentos, o agente apresenta comportamento semelhante ao perfil Deontológico, parando obrigatoriamente diante de semáforos vermelhos. Entretanto, no mapa de rodovias e em situações de disputa de preferência, sua rigidez torna-se mais evidente. Sempre que não possui prioridade geométrica, o agente formula a intenção irrevogável de parar e permanecer aguardando até que a via esteja completamente livre, comportamento que frequentemente contribui para a formação de congestionamentos.

3.6.4 Agente Egoísta

O agente Egoísta atua como o elemento caótico da simulação, representando um condutor imprudente orientado exclusivamente pela redução do próprio tempo de deslocamento. Sua percepção ambiental é limitada ao mínimo necessário para evitar a interrupção do próprio movimento. Diferentemente dos demais perfis, o agente não atribui valor à preservação da vida ($w_v = 0$) nem ao cumprimento das normas legais ($w_l = 0$). Toda a sua motivação está concentrada no pilar da Eficiência ($w_e = 100$), além de apresentar um peso negativo para Cooperação ($w_c = (-50)$), demonstrando resistência ativa a ceder vantagem aos demais agentes. No mapa do cruzamento, o processo decisório ignora completamente as condições do ambiente, mantendo permanentemente a intenção de avançar, independentemente da sinalização ou da presença de pedestres. Como consequência, esse perfil produz infrações semafóricas constantes e situações recorrentes de atropelamento. No mapa da rodovia, o agente nunca cede passagem em disputas de preferência, formulando a intenção de parar apenas em situações extremas de risco imediato de colisão, quando o obstáculo atinge um estado crítico de perigo mecânico.

3.6.5 Agente Aprendiz RL

Diferentemente dos quatro perfis éticos baseados em regras fixas apresentados anteriormente, o quinto agente implementado na simulação, denominado Aprendiz RL, não possui estruturas condicionais estáticas previamente programadas para definir suas decisões. Em vez disso, o agente constrói suas próprias intenções dinamicamente por meio de um processo contínuo de tentativa e erro, utilizando o algoritmo de aprendizado por reforço tabular conhecido como *Q-Learning*.

A principal contribuição metodológica desse perfil está relacionada à origem do sinal de aprendizagem utilizado durante o treinamento. Em abordagens tradicionais de veículos autônomos, os mecanismos de recompensa geralmente são limitados à prevenção de colisões e à conclusão eficiente do trajeto. Neste trabalho, entretanto, o algoritmo de aprendizagem foi integrado diretamente ao *ModuloMoral*, fazendo com que a Função de Utilidade Moral seja utilizada como sinal de recompensa responsável por retroalimentar o processo de aprendizagem do agente.

Dessa maneira, o agente Aprendiz adquire padrões de condução buscando maximizar valores éticos definidos em sua configuração interna, compostos pelos pesos de Vida (100), Lei (200), Eficiência (10) e Cooperação (80). Assim, o comportamento emergente do agente não deriva de regras explícitas previamente definidas, mas da consolidação gradual de experiências acumuladas durante a interação com o ambiente de simulação.

3.6.5.1 Mapeamento de Estados e Política de Exploração

A inteligência do agente Aprendiz é armazenada em uma estrutura de memória contínua denominada *Q-Table*, responsável por mapear o valor esperado de utilidade associado a cada ação possível em diferentes estados do ambiente. Os estados são definidos com base nas crenças perceptivas formadas pelo agente. Nos cenários de cruzamento, o estado S é definido pela combinação entre a percepção da sinalização e a identificação de obstáculos na faixa de travessia, conforme representado pela Fórmula 6.

$$S = (cor_consulta, obstaculo) \quad (6)$$

Nos cenários de rodovia, o estado passa a considerar interações dinâmicas relacionadas à disputa de preferência entre veículos, conforme é apresentado na Fórmula 7.

$$S = (obstaculo_rodovia, tem_preferencia) \quad (7)$$

Para selecionar a ação mais adequada, limitada às opções de “andar” ou “parar”, o agente utiliza a política ϵ -greedy (épsilon-gulosa). Essa estratégia controla matematicamente o equilíbrio entre exploração e exploração do conhecimento adquirido.

Durante a fase de exploração, o algoritmo gera um valor aleatório a cada ciclo decisório. Caso esse valor seja inferior ao limiar ϵ , inicialmente configurado como 1.0, o agente executa uma ação aleatória. Esse mecanismo permite que o sistema descubra consequências ainda não mapeadas, como atravessar um sinal vermelho ou entrar em situações de risco envolvendo pedestres e outros veículos.

Com o avanço dos ciclos de aprendizagem, o valor de ϵ sofre um decaimento multiplicativo parametrizado por $\epsilon_{decay} = 0.99$. Esse processo reduz

gradualmente o comportamento exploratório e faz com que o agente passe a consultar sua Q_table com maior frequência, selecionando intencionalmente as ações que apresentam os maiores valores morais acumulados durante o treinamento.

3.6.5.2 Atualização da Q-Table

A cada transição temporal provocada por uma decisão, o agente observa as consequências da ação anteriormente executada e atualiza o conhecimento armazenado em sua Q_Table . Esse processo ocorre por meio da Equação de Bellman (Mousavi *et al.*, 2025), responsável por recalcular iterativamente a utilidade esperada das ações com base nas recompensas observadas e nas estimativas futuras do ambiente. A fórmula matemática implementada no simulador é apresentada na Fórmula 8.

$$Q(S_{t-1}, A_{t-1}) \leftarrow Q(S_{t-1}, A_{t-1}) + \alpha \cdot [R_t + \gamma \cdot \max_a Q(S_t, a) - Q(S_{t-1}, A_{t-1})] \quad (8)$$

Nessa equação, $Q(S_{t-1}, A_{t-1})$ representa o valor previamente armazenado para a ação executada no estado anterior. O parâmetro α , definido como 0.2, corresponde à taxa de aprendizado responsável por controlar o impacto das novas experiências sobre os conhecimentos já consolidados na Q_table .

O termo R_t representa a recompensa moral fornecida pelo *ModuloMoral*, funcionando como mecanismo de *feedback* ético em resposta à ação executada pelo agente. Já o fator de desconto γ , parametrizado em 0.9, determina a relevância das recompensas futuras no processo decisório, incentivando o agente a considerar consequências de longo prazo em vez de priorizar apenas ganhos imediatos.

Por fim, o termo $\max_a Q(S_t, a)$ corresponde à estimativa do melhor valor possível que pode ser alcançado no novo estado percebido pelo agente, completando o ciclo de otimização iterativa da aprendizagem por reforço. A integração entre a Equação de Bellman e o sistema ético implementado no *ModuloMoral* ocorre continuamente durante toda a execução da simulação, permitindo que o agente refine progressivamente suas estratégias de condução com base em experiências acumuladas e avaliações morais do ambiente.

Algoritmo 5 – Atualização *Q-Table* e política ϵ -greedy

```

if self.tipo == "aprendiz_rl":
    if estado_atual not in self.q_table:
        self.q_table[estado_atual] = {
            self.str_andar: 0.0, self.str_parar: 0.0}

    # Atualização pela Equação de Bellman
    if self.estado_anterior is not None:
        acao_base = ("andar" if self.str_andar in
self.acao_anterior
                        else "parar")
        recompensa = ModuloMoral.calcular_utilidade(
            self.tipo, acao_base,
            self.estado_anterior[0], self.estado_anterior[1],
cenario)
        max_q_futuro = max(self.q_table[estado_atual].values())
        q_antigo =
self.q_table[self.estado_anterior][self.acao_anterior]
        self.q_table[self.estado_anterior][self.acao_anterior] =
(
            q_antigo + self.alpha
            * (recompensa + self.gamma * max_q_futuro -
q_antigo))

    # Política  $\epsilon$ -greedy
    if random.random() < self.epsilon:
        acao = random.choice([self.str_andar, self.str_parar])
    else:
        acao = max(self.q_table[estado_atual],
                    key=self.q_table[estado_atual].get)

    self.estado_anterior, self.acao_anterior = estado_atual,
acao

    if self.epsilon > self.epsilon_min:
        self.epsilon *= self.epsilon_decay

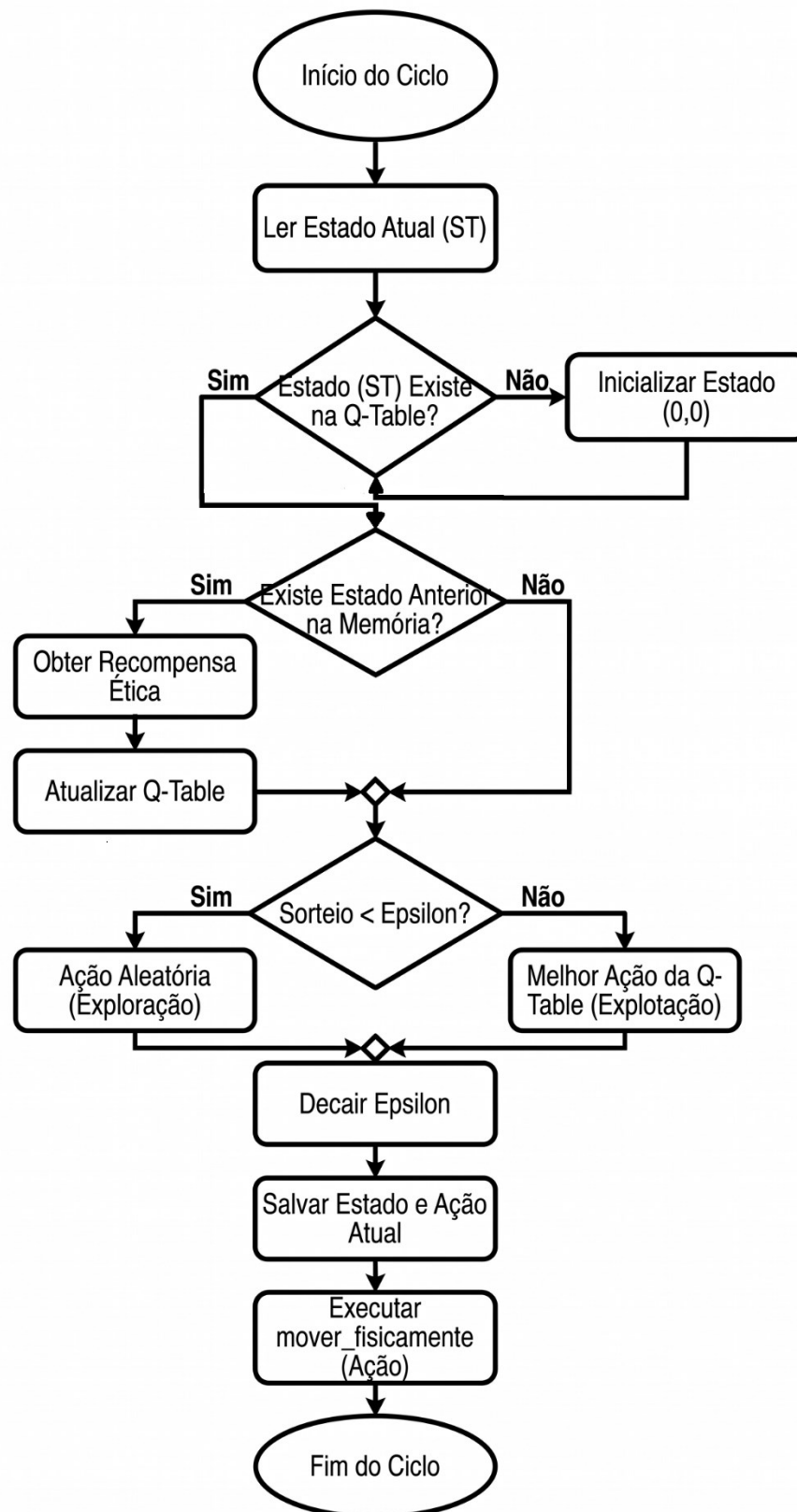
```

Fonte: Autoria própria (2026)

A atualização da *Q-Table* e a seleção de ações do agente Aprendiz são implementadas dentro do mesmo método `_passo_decisao`, no bloco exclusivo do perfil "aprendiz_rl", apresentado no Algoritmo 5. Quando o *estado_atual* ainda não existe na *Q-Table*, ele é inserido com valor zero para ambas as ações, permitindo que a tabela cresça dinamicamente conforme novos estados são encontrados durante a simulação. A atualização pela Equação de Bellman utiliza os atributos *estado_anterior* e *acao_anterior*, salvos no ciclo imediatamente anterior, para calcular a recompensa retroativa: o método *calcular_utilidade* que avalia a ação que foi tomada no ciclo passado e que retorna o valor moral dessa decisão como sinal de aprendizado. As

variáveis q_{antigo} e max_q_{futuro} correspondem, respectivamente, ao valor armazenado na *Q-Table* para o par estado-ação anterior e à maior estimativa de utilidade disponível para o estado atual, correspondendo aos dois termos que compõem a Equação de Bellman apresentada anteriormente. A política ϵ -greedy é aplicada logo em seguida: se o valor gerado por *random.random()* for menor que ϵ , uma ação aleatória é sorteada, caracterizando a fase de exploração; caso contrário, a ação com maior valor na *Q-Table* do estado atual é selecionada, caracterizando a fase de exploração. Ao final do bloco, *estado_atual* e *acao* são salvos em *estado_anterior* e *acao_anterior* para uso no próximo ciclo, e ϵ é multiplicado por ϵ_{decay} , reduzindo gradualmente a taxa de exploração ao longo da simulação.

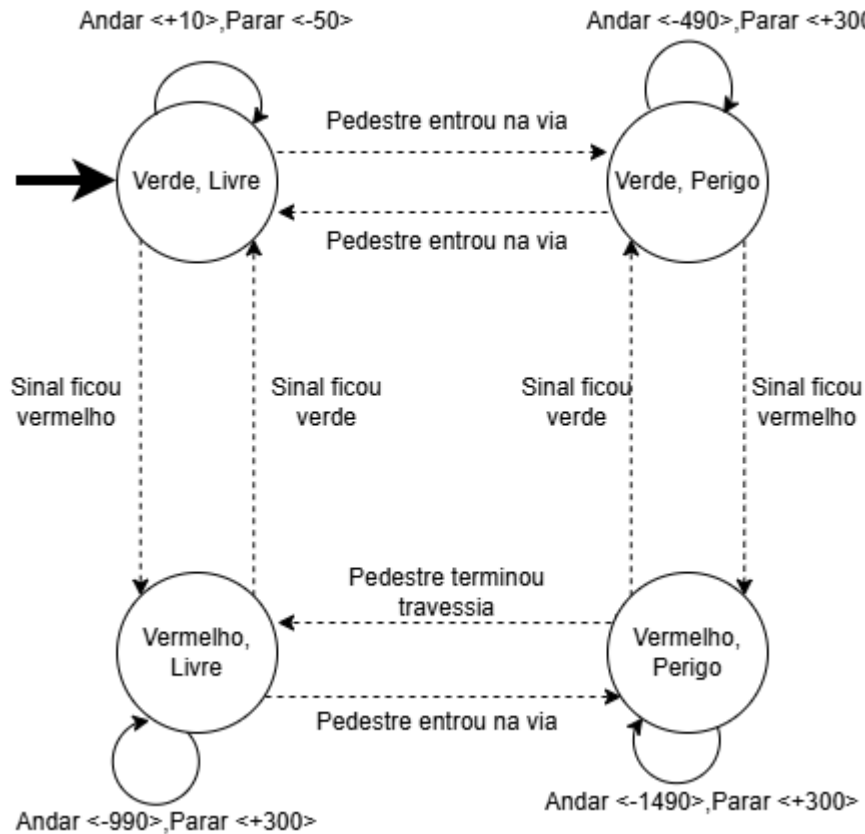
Figura 11 – Fluxograma do ciclo de raciocínio do agente com aprendizado por reforço



Fonte: Autoria própria (2026)

A Figura 11 apresenta o fluxo de controle que conecta o raciocínio do agente aprendiz ao mecanismo de Inteligência Artificial responsável pela atualização da *Q-Table*.

Figura 12 – Diagrama de transição de estados e recompensas do agente aprendiz.



Fonte: Autoria própria (2026)

Complementando o fluxo de decisão do sistema, a Figura 12 apresenta a modelagem das recompensas por meio de um autômato de transição de estados no cenário de cruzamento. Nesse modelo, as transições contínuas representam as ações realizadas pelo agente e os respectivos valores de utilidade calculados pelo Módulo Moral para cada situação. Já as transições tracejadas representam mudanças externas do ambiente, como a troca dos sinais do semáforo e o aparecimento de pedestres, demonstrando o comportamento dinâmico e imprevisível do trânsito.

3.6.6 Quadro Comparativo dos Perfis

O Quadro 1 consolida as principais características de cada agente, relacionando seus desejos predominantes às diretrizes práticas adotadas nos cenários de cruzamento semaforizado e rodovia. A coexistência simultânea dos cinco perfis éticos em um mesmo ambiente digital é responsável pela geração dos conflitos

dinâmicos observados durante a simulação, como disputas de preferência, impasses temporários, quase-colisões e diferentes padrões de cooperação entre veículos.

Quadro 1 – Comparação das principais características de cada agente

Perfil do Agente	Heurística Principal (Desejo)	Diretriz no Cruzamento	Diretriz na Rodovia
Deontológico	Equilíbrio entre Vida e Lei	Para diante de semáforo vermelho e pedestres na faixa.	Respeita a preferência e atua com prudência.
Utilitarista	Cooperação e Bem-Estar Coletivo	Pode avançar no sinal vermelho sem risco de atropelamento.	Cede a preferência para evitar congestionamentos.
Legalista	Obediência Estrita às Normas	Mantém parada obrigatória no sinal vermelho.	Exige rigorosamente sua preferência e evita comportamentos cooperativos.
Egoísta	Maximização da Eficiência Individual	Mantém avanço contínuo, ignorando sinalizações e elevando os índices de infração.	Nunca cede passagem e utiliza frenagem apenas em situações críticas de colisão.
Aprendiz RL	Maximização da Utilidade	Adapta o comportamento de forma dinâmica a partir da exploração do ambiente.	Aprende iterativamente estratégias que equilibram segurança e fluidez do trânsito.

Fonte: Autoria própria (2026).

3.7 Agente Pedestre

Embora o foco principal da simulação esteja concentrado no comportamento dos veículos autônomos, o *Agente Pedestre* também foi modelado segundo o paradigma *Belief-Desire-Intention* (BDI). Entretanto, sua arquitetura cognitiva apresenta menor complexidade quando comparada à dos agentes veiculares, uma vez que não depende da Função de Utilidade implementada no *ModuloMoral*. O ciclo

decisório do pedestre é orientado prioritariamente pela segurança da travessia e pela continuidade do deslocamento nas calçadas do ambiente simulado.

As crenças do agente pedestre são construídas a partir de uma percepção ambiental altamente especializada e limitada ao espaço imediatamente próximo. Para isso, o agente utiliza o método *verificar_carro_no_caminho*, responsável por detectar a presença de objetos da classe Carro em um raio de proximidade de 30 *pixels* e margem lateral de 20 *pixels*. A verificação ocorre por meio de cálculos geométricos baseados em Produto Escalar, permitindo que o pedestre formule crenças binárias relacionadas ao estado da travessia. Assim, o agente passa a acreditar que a via está “livre” ou que existe um “veículo no caminho”, condicionando diretamente o seu comportamento.

Os desejos do pedestre são mais simples e objetivos. O comportamento do agente é guiado por um único desejo operacional, instanciado por meio da meta *Goal("andar_e_treinar")*. Esse desejo representa a intenção contínua de selecionar um destino no grafo de caminhos dos agentes pedestres e concluir o percurso de travessia até alcançar o ponto final estabelecido.

As intenções do pedestre derivam diretamente das crenças formadas durante o processo perceptivo. Em condições normais, a intenção predominante consiste em continuar o deslocamento em direção ao destino. Entretanto, quando o agente pedestre valida a crença de que existe um veículo interceptando sua trajetória, o plano de ação é imediatamente alterado para o estado de espera. Nessa situação, a atualização do vetor de posição é interrompida temporariamente, simulando o comportamento cauteloso de um pedestre diante de uma possível situação de risco.

Para aumentar o realismo da simulação e evitar que o agente permaneça indefinidamente bloqueado na calçada, foi implementada uma heurística de hesitação e impaciência. Sempre que o pedestre entra em estado de espera, o sistema registra o instante exato do bloqueio por meio da variável *self.tempo_bloqueado*. Caso a via permaneça ocupada por mais de 1,5 segundos, calculados pela expressão *time.time() - self.tempo_bloqueado > 1.5*, o agente abandona a intenção de espera e retoma forçadamente o movimento. Essa estratégia busca reproduzir comportamentos humanos comuns em ambientes urbanos, nos quais pedestres eventualmente decidem atravessar mesmo diante de situações parcialmente inseguras após longos períodos de espera.

Após concluir a travessia, o agente registra o tempo total gasto no percurso, permitindo a coleta posterior de métricas relacionadas à mobilidade, tempo de deslocamento e interação entre pedestres e veículos dentro do ambiente simulado.

3.8 Coleta e Processamento de Dados

Para possibilitar a análise quantitativa dos comportamentos produzidos pelo ModuloMoral, o simulador passou a contar com rotinas de monitoramento em memória assíncrona, responsáveis pela coleta contínua dos eventos físicos e lógicos ocorridos durante a simulação.

Considerando a natureza concorrente da integração entre a Pygame e o MASPY, na qual múltiplos agentes veiculares operam simultaneamente por meio de *threads*, a coleta de dados exigiu mecanismos de sincronização para evitar inconsistências de escrita e condições de corrida. Para isso, o sistema utiliza estruturas de bloqueio, garantindo que o acesso às métricas globais ocorra de maneira segura e controlada.

Os eventos gerados durante a simulação alimentam continuamente o dicionário global métricas, responsável por armazenar indicadores relacionados a quatro eixos principais de desempenho do sistema.

O primeiro eixo corresponde aos tempos de viagem dos veículos, utilizados para mensurar a eficiência global do tráfego. O cronômetro é iniciado no instante em que o agente veicular é instanciado no ambiente, sendo finalizado quando o veículo alcança o último nó definido em seu vetor de roteamento geográfico. A diferença temporal entre esses dois eventos representa o tempo total de deslocamento do agente.

O segundo eixo está relacionado aos tempos de espera, métrica utilizada para quantificar os impactos de congestionamentos, retenções e situações de *deadlock*. Sempre que a intenção formulada pelo agente resulta em velocidade escalar nula, o sistema inicia a contagem do tempo de retenção, acumulando esse valor de acordo com o perfil ético correspondente ao veículo analisado.

A terceira métrica monitorada corresponde à quantidade de infrações semafóricas, utilizada como indicador de segurança no trânsito. Essa contagem é incrementada sempre que o ciclo decisório BDI identifica a ação de “andar” em uma região próxima aos pontos de verificação enquanto o estado perceptivo do semáforo permanece registrado como vermelho. Para validar a infração, o sistema considera

uma margem espacial aproximada de 5 *pixels* em relação ao ponto de controle da sinalização.

O quarto eixo de análise corresponde à eficiência da mobilidade ativa, mensurada por meio dos tempos de travessia dos pedestres. Essa métrica registra o intervalo temporal entre o instante em que o pedestre manifesta a intenção de atravessar a via e o momento em que alcança o nodo correspondente à calçada oposta. Dessa forma, torna-se possível avaliar como o comportamento dos veículos impacta diretamente a fluidez e a segurança dos pedestres.

Ao término do laço principal da simulação, os dados armazenados na memória compartilhada são consolidados e submetidos a um processo de serialização, sendo exportados automaticamente para o arquivo estruturado *metricas_tcc.json*. Esse arquivo funciona como a base de dados definitiva do estudo, permitindo o processamento estatístico posterior por meio de bibliotecas de Ciência de Dados, como Pandas, Matplotlib e Seaborn. A partir dessas ferramentas, são gerados gráficos de barras, linhas de tendência e mapas de calor analíticos utilizados no tópico de Resultados para interpretação do comportamento emergente dos agentes e avaliação das métricas de mobilidade urbana sustentável.

3.9 Interfaces do Sistema

Na sequência, serão apresentadas todas as interfaces da simulação, bem como o funcionamento de cada tela.

3.9.1 Tela de Menu

A tela de Menu é o ponto de entrada da aplicação. O usuário dispõe de quatro opções: 'Iniciar Simulação', que direciona para o ambiente de execução; 'Opções', onde é possível configurar os parâmetros da simulação; 'Sobre', que apresenta uma breve descrição do funcionamento e dos objetivos do simulador; e 'Sair', que encerra a aplicação.

Figura 13 – Tela de Menu (MenuScreen)



Fonte: Autoria própria (2026)

A Figura 13 apresenta a tela inicial da simulação, responsável por fornecer acesso às funcionalidades e configurações necessárias para a execução do ambiente de testes.

3.9.2 Tela de Opções

Na tela de Opções, o usuário pode configurar os parâmetros da simulação, incluindo a quantidade de agentes veículos e de agentes pedestres. Nessa tela, também é possível selecionar o mapa no qual a simulação será executada. Ao clicar no botão “Mapa: Cruzamento”, ele será alterado para “Mapa: Rodovia”. Caso deseje retornar ao mapa anterior, basta clicar novamente no botão. Os agentes são representados por cores e números (como é representado na Figura 16), de acordo com o tipo de comportamento adotado: o agente 1, na cor azul, representa o perfil deontológico; o agente 2, na cor verde, representa o perfil utilitarista; o agente 3, na cor amarela, representa o perfil legalista; o agente 4, na cor vermelha, representa o perfil egoísta; e o agente 5, na cor preta, representa o agente aprendiz.

Figura 14 – Tela de Opções com Agentes Éticos

Agentes Éticos

1 - Deontológicos (Azul):	-	1	+
2 - Utilitaristas (Verde):	-	1	+
3 - Legalista (Amarelo):	-	1	+
4 - Egoístas (Vermelho):	-	1	+
5 - Aprendiz RL (Preto):	-	1	+
Quantidade de Pedestres:	-	3	+

Mapa: Cruzamento

Voltar

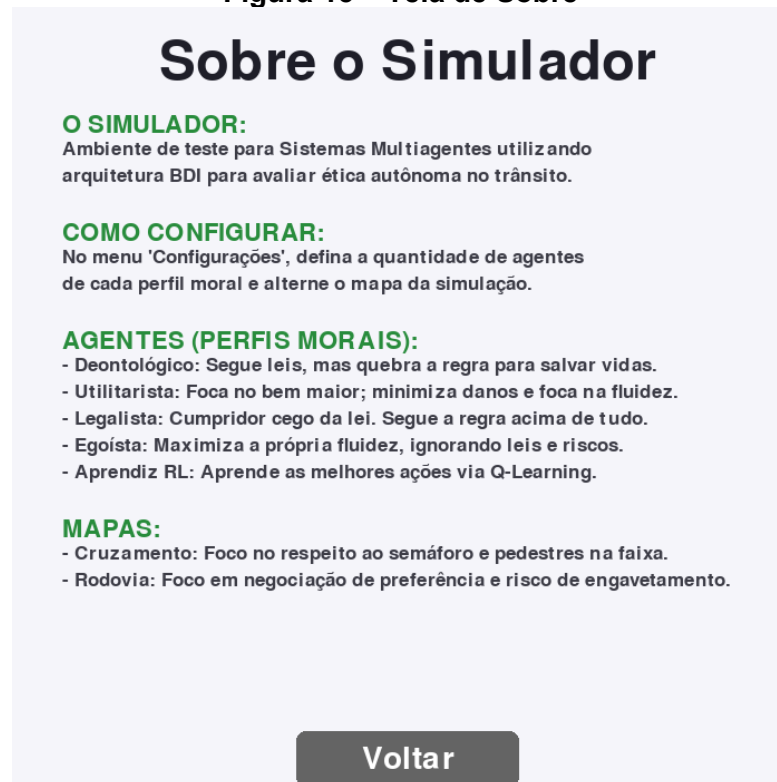
Fonte: Autoria própria (2026)

A Figura 14 apresenta a tela de configurações do simulador, na qual podem ser definidos os parâmetros necessários para a execução dos experimentos.

3.9.3 Tela de Sobre

Na tela de Sobre, é apresentado um breve resumo sobre o funcionamento do simulador, incluindo instruções de configuração, uma explicação resumida de cada perfil de agente e uma descrição dos mapas disponíveis na simulação.

Figura 15 – Tela de Sobre



Fonte: Autoria própria (2026)

A Figura 15 ilustra a tela "Sobre", na qual é apresentada uma descrição geral do simulador e de suas principais funcionalidades.

3.9.4 Tela de Simulação

A tela de Simulação é o ambiente de execução onde o sistema multiagentes é processado em tempo real. Esta tela gerencia a renderização dos ativos visuais, o ciclo de vida dos agentes BDI via MASPYPY e a detecção de colisões por meio da biblioteca Pygame. É durante a execução desta tela que o *SharedState* realiza a contagem das métricas éticas.

Figura 16 – Tela de Simulação (SimulationScreen)

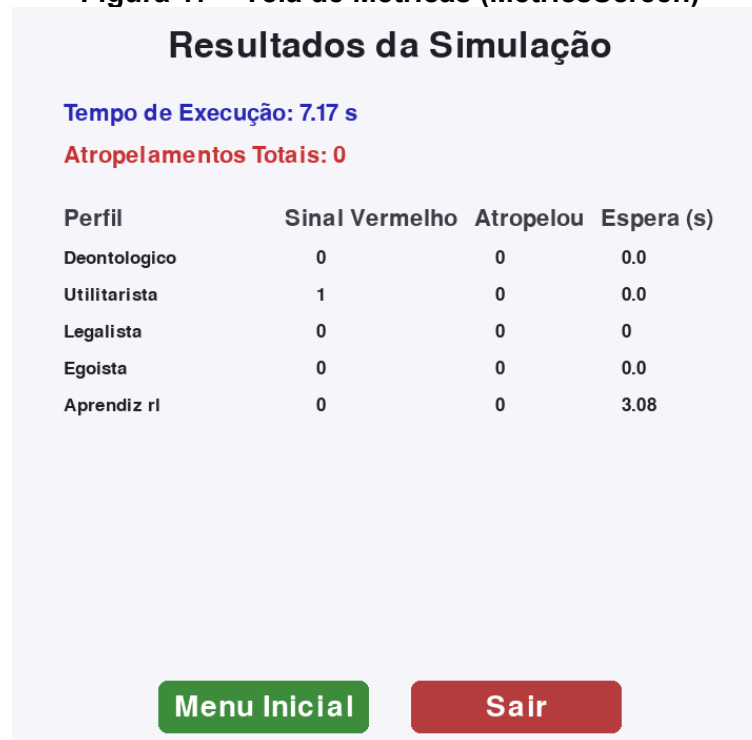


Fonte: Autoria própria (2026)

A Figura 16 apresenta o ambiente de simulação em execução. Nela, é possível visualizar os agentes identificados por cores distintas e por numeração específica, facilitando a diferenciação entre os diferentes perfis éticos adotados por cada agente.

3.9.5 Tela de Métricas

A tela de Métricas é responsável pela apresentação dos dados consolidados após o encerramento de uma simulação. Ela carrega as informações do arquivo *metricas_tcc.json* e apresenta tabelas comparativas que relacionam o perfil ético dos veículos com o desempenho operacional e a segurança viária, permitindo a análise imediata dos resultados obtidos.

Figura 17 – Tela de Métricas (MetricsScreen)

Fonte: Autoria própria (2026)

A Figura 17 ilustra a tela de métricas apresentada ao final da simulação. Ressalta-se que a maioria das variáveis apresenta valor igual a zero em razão do curto período de execução da simulação, que teve duração de apenas 7,17 segundos.

4 RESULTADOS E DISCUSSÃO

Este capítulo apresenta a análise dos dados coletados durante as execuções do simulador, avaliando o comportamento dos agentes autônomos sob diferentes dilemas éticos. A discussão consiste na comparação entre os perfis definidos na classe *ModuloMoral* e na capacidade de adaptação do agente de Aprendizagem por Reforço em relação aos agentes baseados em regras fixas.

4.1 Configurações e Ambiente de Teste

Com o objetivo de avaliar de forma abrangente a arquitetura BDI e a Função de Utilidade Ponderada implementadas no simulador, foram definidas dez baterias de testes distintas, distribuídas igualmente entre os dois cenários, com cinco baterias em cada um. Essa separação metodológica possibilita o isolamento de variáveis específicas associadas a diferentes topologias viárias, permitindo analisar tanto o comportamento dos agentes diante da sinalização urbana quanto a resolução de conflitos de preferência em cenários de fluxo contínuo.

O primeiro ambiente de teste corresponde ao cenário de Cruzamento Urbano, desenvolvido para analisar a interação dos agentes com semáforos, faixas de pedestres e situações de travessia compartilhada. Nesse ambiente, os veículos precisam interpretar estados de sinalização, detectar pedestres no cone de visão e formular intenções de movimento em conformidade com suas heurísticas morais e mecanismos de tomada de decisão.

O segundo ambiente corresponde ao cenário de Rodovia, estruturado para avaliar a eficiência do fluxo veicular e os comportamentos cooperativos em regiões de junção e interseções sem controle semaforico. Nesse contexto, os agentes precisam resolver disputas de preferência utilizando crenças espaciais e estratégias de cooperação, permitindo observar diferenças significativas entre os perfis éticos implementados.

Para garantir imparcialidade na coleta das métricas e permitir comparações diretas entre os cenários, ambas as simulações utilizaram configurações idênticas de carga de tráfego e tempo de amostragem. O período de execução de 5 minutos foi definido por possibilitar que todos os agentes realizassem múltiplos ciclos completos de deslocamento, incluindo processos de reaparecimento, produzindo volume

estatístico suficiente para a análise de tendências comportamentais. Os dados das configurações dos testes estão representados na Tabela 2.

Tabela 2 – Parâmetros globais do ambiente de simulação

Parâmetro	Valor	Descrição
Tempo Total de Execução	5 minutos (300 segundos)	Duração do tempo de execução.
Número Total de Execuções	10 baterias de testes	5 por cenário × 2 cenários
Agentes Veiculares	15 instâncias (3 de cada perfil)	Distribuição homogênea entre os perfis Deontológico, Utilitarista, Legalista, Egoísta e Aprendiz RL.
Agentes Pedestres	6 instâncias	Volume de pedestres transitando simultaneamente.

Fonte: Autoria própria (2026)

4.2 Método Estatístico Adotado

Para validar os resultados obtidos nas simulações e verificar se as diferenças observadas entre os perfis éticos são estatisticamente significativas, ou seja, se não poderiam ser explicadas pela aleatoriedade das execuções, foram aplicados testes estatísticos não-paramétricos.

4.2.1 Teste de Kruskal-Wallis

O teste de Kruskal-Wallis é utilizado para comparar simultaneamente três ou mais grupos independentes e verificar se pelo menos um deles apresenta distribuição diferente dos demais. Ele é uma alternativa não-paramétrica à ANOVA (Análise de Variância), o que significa que não exige que os dados sigam uma distribuição normal, característica adequada para dados de simulação computacional, cujo comportamento estatístico não é conhecido previamente (Field, 2009).

Inicialmente, todos os valores dos grupos analisados são ordenados conjuntamente e substituídos pelos seus respectivos postos (*ranks*). Dessa forma, o menor valor recebe o posto 1, o segundo menor recebe o posto 2 e assim sucessivamente, independentemente do grupo ao qual pertence.

Após a atribuição dos postos, o teste verifica se estes se distribuem de maneira equilibrada entre os grupos. Caso todos os grupos sejam estatisticamente equivalentes, espera-se que a soma dos postos seja proporcional ao tamanho de cada grupo. Quanto maior for o desequilíbrio observado nessa distribuição, maior será o valor da estatística H calculada pelo teste.

Por fim, a estatística H é utilizada para determinar o *p-value*, que representa a probabilidade de se observar uma diferença tão grande quanto a encontrada assumindo que todos os grupos sejam equivalentes, isto é, considerando verdadeira a hipótese nula. Neste trabalho, adota-se o nível de significância de 5% ($p < 0,05$), de modo que valores inferiores a esse limiar indicam evidências suficientes para rejeitar a hipótese nula e concluir que existe diferença estatisticamente significativa entre pelo menos um dos grupos analisados.

Uma limitação do teste de Kruskal-Wallis é que, quando o resultado é significativo, ele apenas confirma que algum grupo difere dos demais, mas não identifica quais pares específicos são diferentes. Para isso, é necessário aplicar um teste de comparação pareada, descrito na seção seguinte.

4.2.2 Teste de Mann-Whitney com Correção de Bonferroni

O teste de Mann-Whitney é aplicado para comparar dois grupos de cada vez, verificando se um tende a apresentar valores sistematicamente maiores ou menores que o outro. Assim como o Kruskal-Wallis, ele trabalha com postos (*rankings*) e não exige normalidade dos dados. Neste trabalho, ele foi aplicado como análise *post-hoc*, ou seja, após o Kruskal-Wallis identificar que existe ao menos uma diferença significativa, o Mann-Whitney é usado para descobrir exatamente entre quais pares de perfis essa diferença ocorre (Field, 2009).

Como são cinco perfis éticos, existem dez pares possíveis de comparação (Deontológico vs. Utilitarista, Deontológico vs. Legalista, e assim por diante). Ao realizar dez testes simultâneos, o risco acumulado de detectar uma diferença falsa, denominado erro do Tipo I, aumenta significativamente. Para controlar esse risco, é aplicada a correção de Bonferroni: o limiar de significância é dividido pelo número de comparações realizadas. Com dez pares e limiar original de 0,05, cada comparação individual precisa apresentar $p < 0,005$ para ser considerada significativa após a correção. Esse ajuste garante que os resultados identificados como significativos não sejam falsos positivos gerados pela quantidade de testes realizados.

4.2.3 Estrutura das Métricas Analisadas

As métricas coletadas pelo simulador se dividem em dois tipos, que exigem abordagens distintas:

- Métricas contínuas (tempos de viagem individuais): cada trajeto concluído por qualquer veículo gera um registro de tempo. Nas cinco execuções, cada perfil acumulou entre 273 e 340 observações. O Kruskal-Wallis e o Mann-Whitney são aplicados diretamente sobre esses valores individuais.
- Métricas de contagem (atropelamentos, infrações, colisões): registradas como um total por execução. Com cinco execuções, cada perfil tem cinco observações para cada métrica de contagem. Os testes são aplicados sobre esses cinco valores por grupo. O desvio padrão (DP) dessas contagens indica se o comportamento do perfil foi consistente entre as execuções: um DP baixo significa que o perfil repetiu um padrão semelhante em todas as rodadas; um DP alto indica variabilidade.

4.3 Cenário 1: Cruzamento Urbano Semaforizado

A avaliação do cenário de cruzamento urbano concentra-se na interação dos agentes com normas de trânsito urbanas, especialmente no respeito à sinalização semafórica e à prioridade de travessia dos pedestres. Esse ambiente foi projetado para analisar como os diferentes perfis éticos equilibram seus objetivos de eficiência com as penalizações associadas a infrações legais e riscos à integridade física dos pedestres.

4.3.1 Análise de Eficiência e Fluidez

A Tabela 3 consolida as estatísticas descritivas dos tempos de viagem individuais registrados no cenário de cruzamento. Cada linha corresponde a um perfil ético e apresenta medidas calculadas a partir do conjunto de todos os trajetos concluídos nas cinco execuções independentes. O número de observações (N obs.) representa a quantidade total de trajetos registrados para cada perfil, de modo que valores mais elevados tornam as estimativas estatísticas mais representativas.

A média (s) corresponde ao tempo médio gasto pelos veículos para concluir um trajeto, sendo a principal medida de eficiência considerada neste estudo. Complementarmente, a mediana (s) representa o valor central da distribuição dos

tempos de viagem, dividindo os resultados em duas metades. A comparação entre média e mediana auxilia na identificação de assimetrias na distribuição dos dados: valores próximos sugerem uma distribuição equilibrada, enquanto diferenças mais acentuadas podem indicar a presença de trajetos excepcionalmente longos ou curtos.

A dispersão dos resultados é avaliada por meio do desvio padrão (DP), que quantifica o grau de variação dos tempos de viagem em torno da média. Valores elevados de DP indicam maior heterogeneidade nos resultados, enquanto valores reduzidos sugerem maior consistência entre os trajetos observados. O erro padrão da média (EP) fornece uma medida da precisão da média amostral como estimativa do valor verdadeiro da população, sendo calculado a partir do desvio padrão e do número de observações. Quanto menor o EP, maior a confiabilidade da estimativa obtida.

Além disso, a Tabela 3 apresenta os limites inferior e superior do intervalo de confiança de 95% (IC 95%), que delimitam a faixa dentro da qual se espera que esteja a média real do perfil com 95% de confiança, caso o experimento fosse repetido inúmeras vezes. Por exemplo, um IC de [14,93; 15,89] para o Deontológico significa que a média verdadeira desse perfil está quase certamente entre esses dois valores. Por fim, os valores mínimo (Mín.) e máximo (Máx.) indicam, respectivamente, os menores e maiores tempos de viagem observados em qualquer trajeto das cinco execuções, permitindo identificar a amplitude dos resultados registrados.

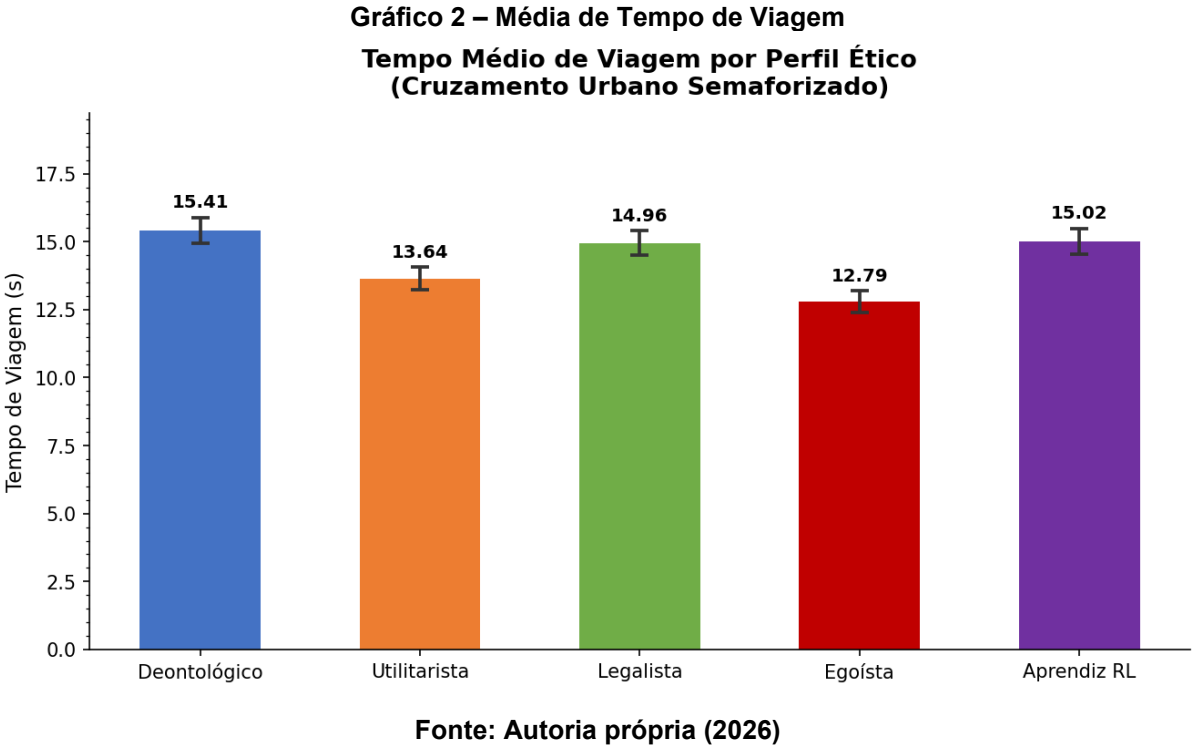
Tabela 3 – Estatísticas do tempo de viagem por perfil ético

Perfil Ético	N obs.	Média (s)	Mediana (s)	DP (s)	EP (s)	IC 95% inf.	IC 95% sup.	Mín (s)	Máx (s)
Deontológico	273	15,41	15,07	4,03	0,24	14,93	15,89	7,71	30,88
Utilitarista	313	13,64	12,84	3,74	0,21	13,23	14,06	6,96	28,17
Legalista	282	14,96	14,42	3,82	0,23	14,51	15,40	6,86	29,82
Egoísta	331	12,79	12,79	3,71	0,20	12,39	13,19	6,37	27,27
Aprendiz RL	284	15,02	14,51	4,01	0,24	14,56	15,49	7,36	30,88

Fonte: Autoria própria (2026)

Analisando os dados da Tabela 3, o perfil Egoísta apresentou o menor tempo médio de viagem (12,79 s), seguido pelo Utilitarista (13,64 s). Em contrapartida, o Deontológico registrou a maior média (15,41 s), resultado esperado dado o elevado peso atribuído ao cumprimento das normas semaforicas em sua função de utilidade moral, o que implica paradas frequentes mesmo quando não há conflito imediato. Os intervalos de confiança de 95% dos cinco perfis não se sobrepõem completamente, o que já sinaliza a existência de diferenças reais entre eles.

Para verificar formalmente se essas diferenças são estatisticamente significativas, foi aplicado o teste de Kruskal-Wallis (descrito na Seção 4.2.1), seguido de comparações pareadas por Mann-Whitney com correção de Bonferroni (Seção 4.2.2). O resultado indicou diferença significativa entre os perfis ($H = 104,14$, $p < 0,001$), confirmando que as diferenças não são produto do acaso. A análise *post-hoc* identificou que os perfis Egoísta e Utilitarista diferem significativamente do Deontológico, do Legalista e do Aprendiz RL em relação ao tempo de viagem, refletindo a consequência direta de suas políticas de menor respeito às normas de parada.



O Gráfico 2 apresenta o tempo médio de viagem obtido por cada perfil ético ao longo das cinco execuções realizadas no cenário de cruzamento.

4.3.2 Segurança e Conformidade Normativa

Esta seção analisa três métricas de segurança coletadas no cenário de cruzamento: infrações semafóricas, atropelamentos de pedestres e tempo de espera acumulado. Cada uma dessas métricas é registrada como um valor total por execução. Por esse motivo, a estrutura das tabelas apresentadas nesta seção difere da Tabela 3. Em vez de estatísticas calculadas sobre observações individuais, as colunas apresentam os valores acumulados e as medidas estatísticas obtidas a partir dos resultados das cinco execuções. As colunas das Tabelas 4, 5 e 6 seguem a lógica descrita abaixo.

A coluna N exec. indica o número de execuções realizadas para cada perfil, que neste trabalho corresponde a cinco. A coluna Total acumulado representa a soma dos valores registrados em todas as execuções. Assim, por exemplo, um total de 130 infrações significa que, ao longo das cinco simulações realizadas, determinado perfil cometeu 130 infrações semafóricas.

A coluna Média por execução corresponde ao total acumulado dividido pelo número de execuções, fornecendo uma estimativa do comportamento típico do perfil em uma única simulação. Já o desvio padrão (DP) é calculado a partir dos cinco

valores individuais observados e mede a variabilidade dos resultados entre as execuções. Valores reduzidos de DP indicam um comportamento mais consistente, enquanto valores elevados sugerem maior variação entre as simulações.

Por fim, as colunas Mín. e Máx. apresentam, respectivamente, os menores e maiores valores observados em uma única execução. Essas medidas permitem identificar os casos extremos registrados durante os experimentos e complementar a análise da variabilidade dos resultados.

Tabela 4 – Infrações semafóricas por perfil ético

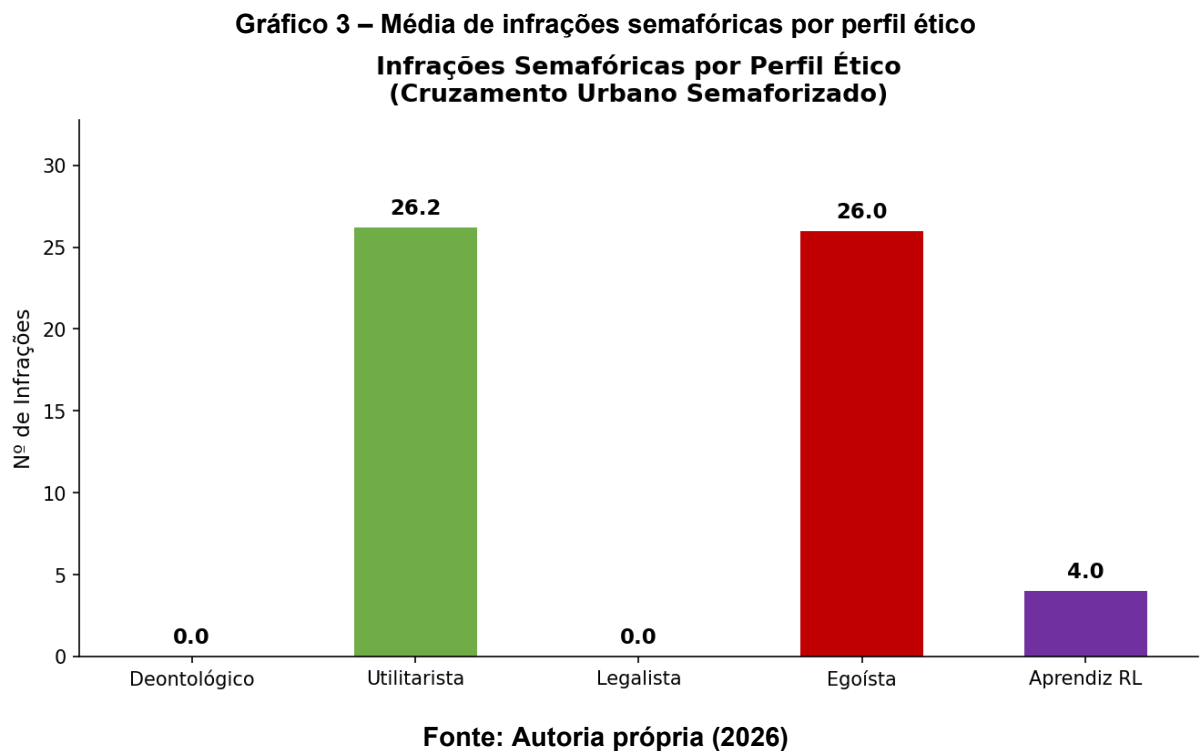
Perfil Ético	N exec.	Total acumulado	Média/exec.	DP	Mín/exec.	Máx/exec.
Deontológico	5	0	0,00	0,00	0	0
Utilitarista	5	131	26,20	5,17	20	33
Legalista	5	0	0,00	0,00	0	0
Egoísta	5	130	26,00	2,55	23	29
Aprendiz RL	5	20	4,00	0,00	4	4

Fonte: Autoria própria (2026)

A Tabela 4 evidencia que os perfis Deontológico e Legalista não cometeram nenhuma infração semafórica ao longo de todas as cinco execuções (total = 0, DP = 0,00), resultado que reflete diretamente o elevado peso atribuído ao cumprimento de normas em suas funções de utilidade moral. No extremo oposto, o Egoísta cometeu em média 26,00 infrações por execução, com baixo desvio padrão (DP = 2,55), evidenciando que esse comportamento de descumprimento das normas é consistente e recorrente, e não um fenômeno pontual. O Aprendiz RL registrou exatamente 4 infrações em todas as cinco execuções (DP = 0,00), indicando que o agente convergiu para uma política estável dentro da janela de 5 minutos, mesmo sem regras fixas programadas.

O teste de Kruskal-Wallis (Seção 4.2.1) confirmou que as diferenças entre os perfis são estatisticamente significativas ($H = 22,40$, $p < 0,001$). A análise post-hoc

identificou que o Aprendiz RL difere significativamente dos perfis Deontológico e Legalista em relação ao número de infrações semafóricas, refletindo que, mesmo apresentando poucas infrações por execução, seu comportamento aprendido ainda não atingiu o mesmo nível de conformidade normativa dos perfis baseados em regras fixas. Os demais pares não apresentaram diferença significativa, especialmente a comparação entre Utilitarista e Egoísta, resultado esperado, uma vez que ambos cometem infrações com frequência muito semelhante (26,20 e 26,00 por execução, respectivamente).



O Gráfico 3 apresenta a média de infrações semafóricas cometidas por cada perfil ético nas cinco baterias de testes realizadas no cenário de cruzamento.

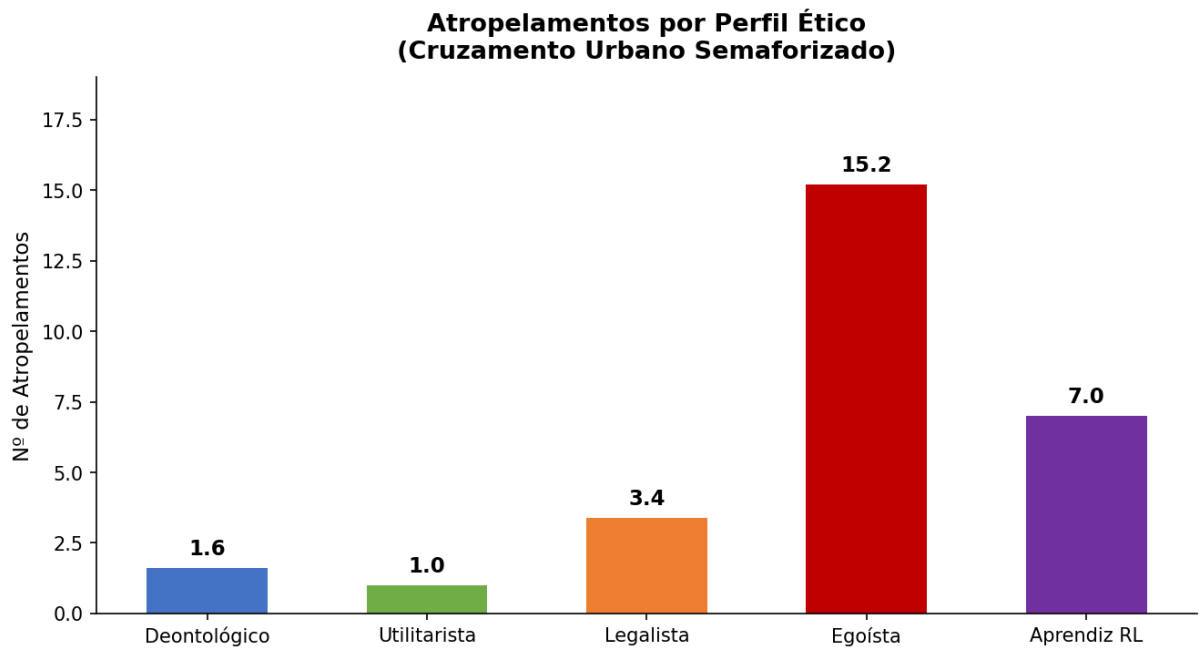
Tabela 5 – Atropelamentos de Pedestres por Perfil Ético

Perfil Ético	N exec.	Total acumulado	Média/exec.	DP	Mín/exec.	Máx/exec.
Deontológico	5	8	1,60	0,55	1	2
Utilitarista	5	5	1,00	0,71	0	2
Legalista	5	17	3,40	2,07	1	6
Egoísta	5	76	15,20	0,84	14	16
Aprendiz RL	5	35	7,00	3,32	3	11

Fonte: Autoria própria (2026)

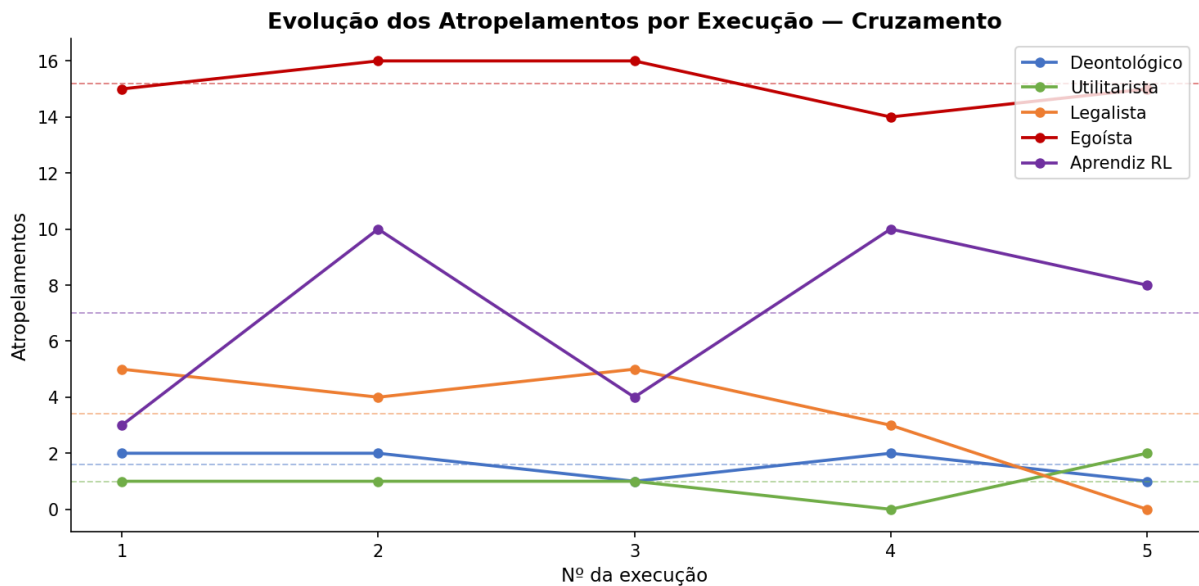
A Tabela 5 mostra que o perfil Egoísta foi responsável por 76 dos 141 atropelamentos totais registrados no cruzamento, mais da metade de todas as ocorrências, com uma média de 15,20 por execução e desvio padrão de apenas 0,84, reforçando a consistência desse comportamento de risco. O Utilitarista apresentou o menor número de atropelamentos (média = 1,00 por execução), resultado coerente com sua função de utilidade orientada ao bem coletivo, que penaliza fortemente danos a pedestres. O Aprendiz RL registrou média de 7,00 atropelamentos por execução, porém com DP = 3,32, a maior variabilidade do conjunto, indicando que sua política ainda estava em fase de aprendizado e oscilou entre execuções.

O teste de Kruskal-Wallis (Seção 4.2.1) confirmou diferença significativa entre os perfis ($H = 19,09$, $p < 0,001$). A análise *post-hoc* não identificou nenhum par individualmente significativo após a correção de Bonferroni. Esse resultado não significa que os perfis se comportam de forma igual, diferenças absolutas como os 15,20 atropelamentos de média do Egoísta frente a média 1,00 do Utilitarista são evidentes, mas sim que cinco execuções por perfil não foram suficientes para confirmá-las estatisticamente. Com um número maior de repetições, essas diferenças tendem a se tornar formalmente distinguíveis.

Gráfico 4 – Média de atropelamentos por perfil ético

Fonte: Autoria própria (2026)

O Gráfico 4 apresenta a média de atropelamentos registrados para cada perfil ético ao longo das cinco baterias de testes realizadas.

Gráfico 5 – Evolução dos atropelamentos por perfil ético

Fonte: Autoria própria (2026)

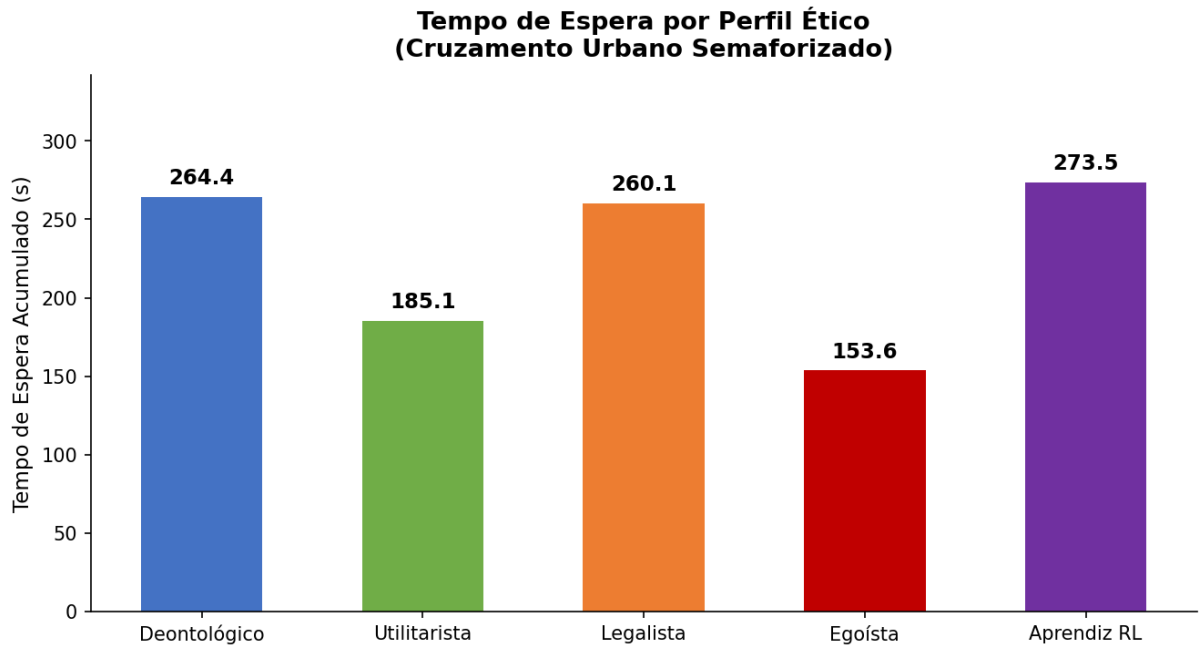
O Gráfico 5 apresenta a evolução do número de atropelamentos ao longo das cinco baterias de testes. Observa-se que os agentes baseados em regras fixas apresentaram um comportamento mais estável entre as execuções, enquanto o agente Aprendiz RL registrou a maior variação nos resultados.

Tabela 6 – Tempo de espera por perfil ético

Perfil Ético	N exec.	Total acumulado (s)	Média/exec. (s)	DP (s)	Mín/exec. (s)	Máx/exec. (s)
Deontológico	5	1.321	264,40	44,01	210	318
Utilitarista	5	925	185,05	20,31	158	210
Legalista	5	1.300	260,10	25,96	228	291
Egoísta	5	768	153,60	49,95	89	210
Aprendiz RL	5	1.367	273,49	60,09	188	342

Fonte: Autoria própria (2026)

A Tabela 6 revela um padrão oposto ao das infrações, o agente Egoísta que mais cometeu infrações, registrou o menor tempo médio de espera (153,60 s por execução), pois seu avanço contínuo sem paradas reduz naturalmente o tempo parado. O Aprendiz RL apresentou o maior tempo médio de espera (273,49 s), com o maior desvio padrão do conjunto (DP = 60,09 s), indicando que sua estratégia de cessão de passagem, ainda em aprendizado, oscilou bastante entre as execuções. Esse contraste entre as Tabelas 4 e 6 ilustra o *trade-off* central dos perfis éticos: agentes que menos tiveram infrações tendem a esperar mais, pois respeitam as normas de parada mesmo quando poderiam avançar.

Gráfico 6 – Tempo de espera acumulado por perfil ético

Fonte: Autoria própria (2026)

O Gráfico 6 apresenta o tempo médio de espera registrado para cada perfil ético, considerando os resultados obtidos nas cinco baterias de testes realizadas.

4.4 Cenário 2: Rodovia e Interseções de Fluxo Contínuo

Diferentemente do cenário urbano controlado por semáforos, o ambiente rodoviário exige que os agentes resolvam conflitos de preferência de maneira descentralizada. Em interseções e entroncamentos de fluxo contínuo, não existe um mecanismo central de coordenação responsável por determinar qual veículo deve avançar ou interromper o movimento. Nesse contexto, os agentes precisam interpretar o ambiente, formular crenças espaciais e decidir autonomamente suas intenções de deslocamento. O principal objetivo deste cenário consiste em avaliar os níveis de cooperação espontânea entre os agentes, a ocorrência de impasses geométricos e a eficácia do fluxo de trânsito em condições de mobilidade elevada.

4.4.1 Eficiência em Vias Expressas

A Tabela 7 consolida as estatísticas descritivas dos tempos de viagem na rodovia, essa tabela segue o mesmo padrão da tabela 3 (Explicada na seção 4.3.1 Análise de Eficiência e Fluidez). Ao contrário do cenário do cruzamento, as médias dos cinco perfis são muito próximas entre si, variando de 12,38 s (Aprendiz RL) a 12,98 s (Utilitarista).

Tabela 7 – Tempo de viagem por perfil ético (rodovia)

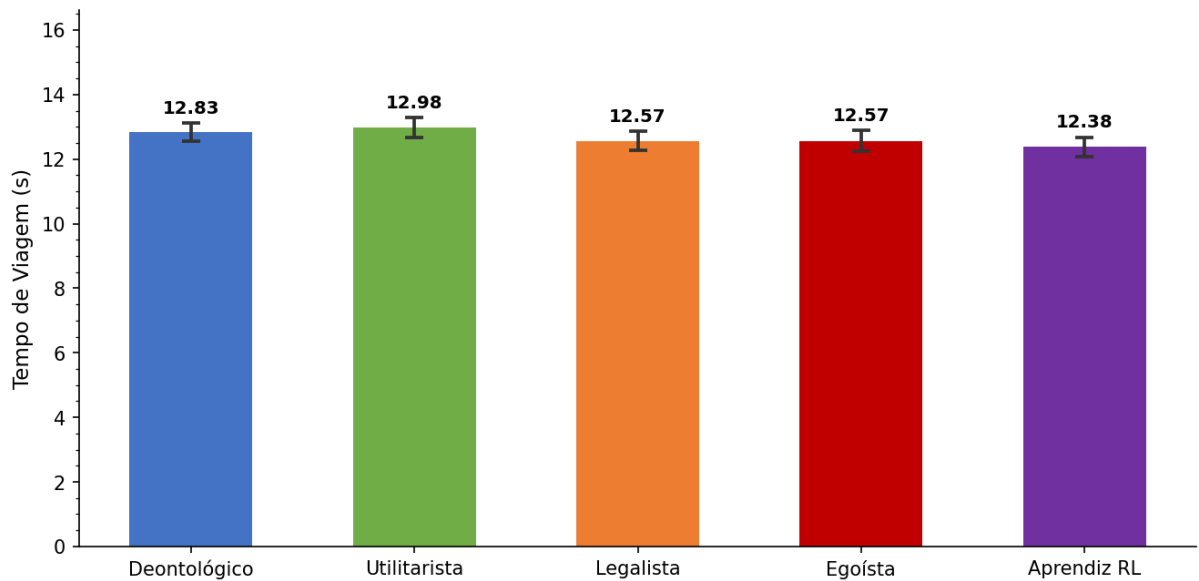
Perfil Ético	N obs.	Média (s)	Mediana (s)	DP (s)	EP (s)	IC 95% inf.	IC 95% sup.	Mín (s)	Máx (s)
Deontológico	327	12,83	12,41	2,55	0,14	12,55	13,11	8,35	23,56
Utilitarista	325	12,98	12,72	2,88	0,16	12,66	13,29	7,48	26,71
Legalista	336	12,57	12,25	2,71	0,15	12,28	12,86	7,41	24,83
Egoísta	336	12,57	12,04	2,98	0,16	12,25	12,88	7,40	25,19
Aprendiz RL	340	12,38	11,96	2,85	0,15	12,07	12,68	7,60	26,57

Fonte: Autoria própria (2026)

A Tabela 7 mostra que, diferentemente do cruzamento, as médias dos cinco perfis na rodovia são muito próximas entre si, variando apenas entre 12,38 s (Aprendiz RL) e 12,98 s (Utilitarista). O Legalista e o Egoísta apresentaram médias idênticas (12,57 s), embora com perfis de variabilidade distintos, o Legalista teve DP de 2,71 s enquanto o Egoísta apresentou DP de 2,98 s, indicando maior oscilação entre trajetos. Os intervalos de confiança de 95% dos perfis se sobrepõem quase completamente, sinalizando que as diferenças de tempo são pequenas.

O teste de Kruskal-Wallis (Seção 4.2.1) mostrou que existem diferenças estatisticamente significativas entre os perfis em relação ao tempo de viagem na rodovia ($H = 13,47$, $p = 0,009$). No entanto, embora a diferença seja estatisticamente detectável, ela é muito pequena na prática. Enquanto no cenário de cruzamento o valor de H foi de 104,14, na rodovia ele foi consideravelmente menor. Além disso, a diferença entre a maior e a menor média de tempo de viagem foi de apenas 0,60 s, um valor que dificilmente seria percebido em uma situação real de tráfego. Esse resultado sugere que a ausência de semáforos reduz as diferenças de desempenho entre os perfis éticos em termos de tempo de viagem.

Gráfico 7 – Média de Tempo de Viagem na Rodovia
Tempo Médio de Viagem por Perfil Ético
(Rodovia – Fluxo Contínuo)



Fonte: Autoria própria (2026)

O Gráfico 7 apresenta o tempo médio de viagem dos perfis éticos no cenário de rodovia, considerando os resultados obtidos nas cinco baterias de testes realizadas.

4.4.2 Tempos de Retenção e Resolução de *Deadlocks*

As colunas da Tabela 8 seguem a mesma lógica descrita para as Tabelas 5, 6 e 7: total acumulado nas cinco execuções, média por execução, desvio padrão e valores extremos.

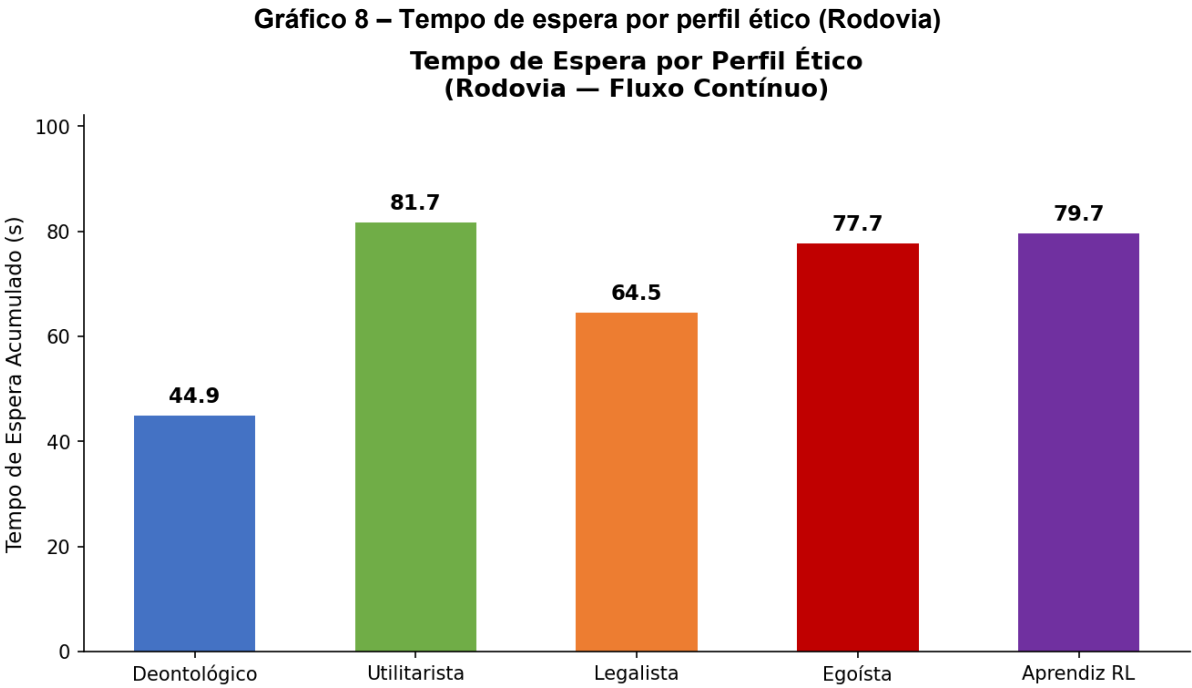
Tabela 8 – Tempo de espera por perfil ético (rodovia)

Perfil Ético	N exec.	Total acumulado (s)	Média/exec. (s)	DP (s)	Mín/exec. (s)	Máx/exec. (s)
Deontológico	5	224	44,91	18,04	24	68
Utilitarista	5	408	81,65	36,18	42	128
Legalista	5	322	64,48	13,92	48	82
Egoísta	5	388	77,71	28,71	42	112
Aprendiz RL	5	398	79,66	35,45	38	122

Fonte: Autoria própria (2026)

A Tabela 8 revela que o Utilitarista acumulou o maior tempo de espera na rodovia (408 s no total, média de 81,65 s por execução), comportamento coerente com sua função de utilidade orientada à cooperação coletiva, que o leva a ceder passagem com frequência mesmo sem obrigatoriedade legal. O Deontológico registrou o menor tempo de espera (224 s, média de 44,91 s por execução), pois no cruzamento esse mesmo perfil tinha alta retenção por respeitar semáforos. Na rodovia, sem semáforos para respeitar, seu comportamento normativo não impõe as mesmas penalizações de parada. Os desvios padrão elevados do Utilitarista (DP = 36,18 s) e do Aprendiz RL (DP = 35,45 s) indicam que os tempos de espera desses perfis variaram bastante entre as cinco execuções.

No cenário da rodovia, o perfil Aprendiz RL apresentou o melhor desempenho em termos de segurança, registrando a menor ocorrência de colisões entre os perfis avaliados. Esse resultado sugere que a combinação do algoritmo de aprendizado por reforço com a Função de Utilidade Moral foi capaz de desenvolver estratégias de navegação mais seguras, mesmo na ausência de regras explícitas de cedência de passagem. Em contraste, o perfil Egoísta apresentou o maior número de colisões, comportamento compatível com sua política de priorizar o próprio deslocamento em detrimento da consideração pelos demais veículos presentes no ambiente.



Fonte: Autoria própria (2026)

O Gráfico 8 apresenta o tempo médio de espera registrado para cada perfil ético ao longo das cinco baterias de testes realizadas.

4.5 Discussão Geral dos Resultados

A análise integrada dos dois cenários permite avaliar em que medida cada perfil ético implementado reflete os valores morais que o fundamenta. A Tabela 9 sintetiza as métricas mais relevantes de ambos os cenários em uma visão única, possibilitando uma comparação direta entre os cinco perfis. As colunas de tempo de viagem apresentam os valores no formato média $\pm$ DP calculados sobre todos os trajetos registrados nas cinco execuções, seguindo a mesma estrutura das Tabelas 3 e 7. As colunas de infrações e atropelamentos expressam a média por execução $\pm$ DP, seguindo a mesma lógica das Tabelas 5 e 6.

Tabela 9 – Síntese comparativa global das métricas de desempenho de ambos os cenários

Perfil Ético	T. Viagem Cruz. média ± DP (s)	T. Viagem Rod. média ± DP (s)	Infrações média/exec. ± DP	Atropel. média/exec. ± DP
Deontológico	15,41 ± 4,03	12,83 ± 2,55	0,00 ± 0,00	1,61 ± 0,55
Utilitarista	13,64 ± 3,74	12,98 ± 2,88	26,20 ± 5,17	1,00 ± 0,71
Legalista	14,96 ± 3,82	12,57 ± 2,71	0,00 ± 0,00	3,40 ± 2,07
Egoísta	12,79 ± 3,71	12,57 ± 2,98	26,00 ± 2,55	15,20 ± 0,84
Aprendiz RL	15,02 ± 4,01	12,38 ± 2,85	4,00 ± 0,00	7,00 ± 3,32

Fonte: Autoria própria (2026)

A Tabela 9 permite identificar dois padrões centrais que atravessam ambos os cenários. O primeiro é a consistência normativa dos perfis baseados em regras fixas, o agente Deontológico e o agente Legalista são os únicos a registrar zero infrações semaforicas nas cinco execuções, confirmando que o elevado peso atribuído ao cumprimento de normas em suas funções de utilidade moral se traduz em comportamento reproduzível e previsível independentemente das condições da simulação. Essa conformidade, porém, tem um custo mensurável, ambos apresentam os maiores tempos médios de viagem no cruzamento (15,41s e 14,96s respectivamente) pois respeitam semáforos e cedem passagem mesmo em situações sem conflito imediato.

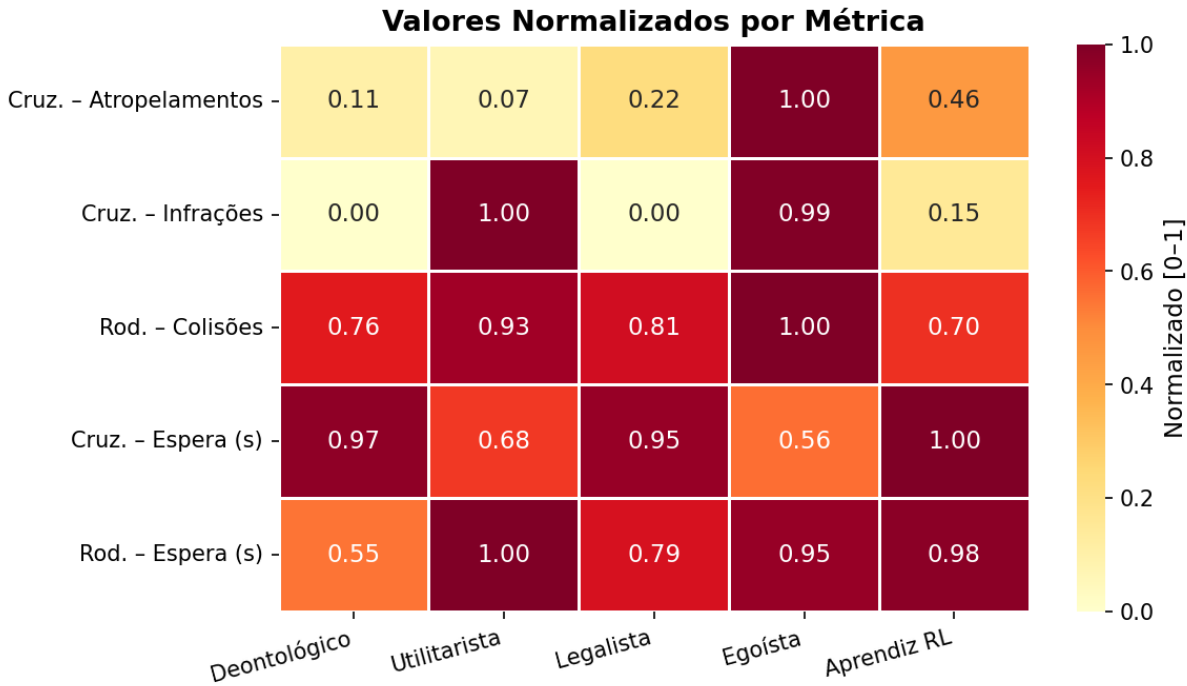
O segundo padrão é o trade-off entre eficiência individual e segurança coletiva, ilustrado com clareza pelo perfil Egoísta. Com o menor tempo médio de viagem no cruzamento (12,79 ± 3,71 s) e desempenho equivalente na rodovia (12,57 ± 2,98 s), o agente Egoísta obtém ganhos de eficiência à custa de consequências severas para os demais participantes do tráfego, sendo uma média de infrações semaforicas de 26,00 com um DP de 2,55 e uma média de atropelamentos por execução 15,20 com um DP de 0,84. O desvio padrão baixo em ambas as métricas de segurança indica que esse comportamento não é pontual, é sistemático e se reproduz de forma consistente a cada execução.

O perfil Utilitarista apresenta um resultado aparentemente contraditório: embora cometa, em média, 26,20 infrações por execução, valor próximo ao observado para o perfil Egoísta, registra apenas 1,00 atropelamento por execução, o menor entre todos os perfis avaliados. Esse resultado pode ser explicado pelo fato de que sua função de utilidade atribui uma penalização elevada aos danos causados a pedestres. O perfil Legalista, por sua vez, apresenta o comportamento oposto: não comete infrações semafóricas, mas registra, em média, 3,40 atropelamentos por execução. Esse resultado evidencia uma limitação da implementação adotada, uma vez que o agente respeita rigorosamente a sinalização semafórica, porém sua lógica de detecção de pedestres em movimento apresentou falhas em determinadas configurações do cenário.

O agente Aprendiz RL destacou-se por apresentar um desempenho equilibrado entre as diferentes métricas analisadas. Sem depender de regras fixas, o agente desenvolveu, ao longo das execuções, uma política adaptativa capaz de conciliar eficiência e segurança. No cenário de cruzamento, registrou um baixo número de infrações semafóricas e apresentou comportamento consistente entre as execuções, evidenciado pela ausência de variação nos resultados observados. Além disso, manteve níveis intermediários de atropelamentos e alcançou o menor tempo médio de viagem no cenário de rodovia.

Outro aspecto relevante é que o Aprendiz RL não apresentou o pior desempenho em nenhuma das métricas avaliadas. Esse resultado sugere que a combinação entre o algoritmo Q-Learning e a Função de Utilidade Moral permitiu ao agente desenvolver estratégias capazes de equilibrar objetivos potencialmente conflitantes. Em contraste, os perfis baseados em regras fixas tendem a priorizar determinados princípios ou objetivos específicos, o que pode favorecer o desempenho em algumas métricas, mas gerar limitações em outras.

Gráfico 9 – Mapa de Calor Global dos Perfis Éticos
Mapa de Calor Global — Desempenho dos Perfis Éticos



Fonte: Autoria própria (2026)

O Gráfico 9 apresenta o mapa de calor global com os valores normalizados de cada métrica, permitindo visualizar de forma consolidada os padrões discutidos. Tons mais escuros indicam valores mais elevados, o que representa maior risco nas métricas de segurança e maior tempo nas métricas de eficiência. A visualização confirma os dois padrões identificados na Tabela 9.

4.6 Limitações do Trabalho e Testes de Estresse

Apesar de a ferramenta atender aos objetivos propostos, este trabalho apresenta limitações metodológicas e técnicas. Em termos de simulação, o ambiente assume que os agentes possuem sensores perfeitos, ignorando ruídos do mundo real como oclusões visuais ou falhas de detecção. Além disso, a física do simulador 2D é puramente cinemática; ou seja, desconsidera variáveis automotivas dinâmicas como inércia, atrito do asfalto e tempo mecânico de frenagem. Outra limitação teórica encontra-se no perfil de Aprendizado por Reforço, que utiliza uma abordagem tabular clássica (*Q-Table*), o que dificulta a generalização para cenários com variáveis contínuas complexas.

Para avaliar os limites práticos do simulador, foram realizados testes de estresse computacional. O principal gargalo de desempenho da ferramenta é a verificação iterativa de colisões (com complexidade próxima a $O(N^2)$).

Durante os testes, o simulador conseguiu manter uma taxa de atualização ideal (60 FPS) e fluida com até 30 agentes simultâneos no mapa. Ao elevar a carga para cerca de 60 agentes, observou-se uma queda de desempenho (média de 25 FPS). O limite prático da ferramenta, onde o atraso começa a comprometer o ciclo de decisão dos agentes e a detecção de colisões, ocorre na casa de 65 agentes simultâneos.

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento e a análise de um sistema multiagente aplicado à simulação de tráfego urbano, fundamentado na arquitetura *Belief-Desire-Intention* (BDI) e orientado por diferentes perfis éticos de tomada de decisão. A proposta investigou como veículos autônomos podem utilizar mecanismos de avaliação moral para lidar com dilemas complexos de mobilidade, equilibrando fatores relacionados à eficiência do deslocamento, cumprimento das normas de trânsito, cooperação coletiva e preservação da vida.

Os resultados obtidos ao longo das simulações demonstraram que a utilização de uma Função de Utilidade Ponderada possibilita a construção de agentes capazes de adaptar suas decisões conforme o contexto do ambiente, superando limitações presentes em sistemas rigidamente baseados em regras estáticas. A integração entre crenças perceptivas, desejos axiológicos e intenções dinâmicas permitiu que os agentes formassem comportamentos distintos em cenários urbanos e rodoviários, evidenciando o impacto direto das diretrizes morais sobre a segurança e a fluidez do trânsito.

As análises empíricas realizadas nos cenários de cruzamento semaforizado e rodovia revelaram que abordagens extremistas apresentam limitações significativas para aplicações reais de mobilidade urbana sustentável. O perfil Egoísta, embora tenha alcançado os menores tempos médios de viagem, demonstrou comportamento altamente inseguro, sendo responsável pela maior parte dos atropelamentos e colisões registradas durante as simulações. Em contrapartida, os perfis Deontológico, Legalista e Utilitarista evidenciaram que modelos excessivamente rígidos, seja pela obediência absoluta às normas ou pelo altruísmo extremo, também produzem efeitos negativos, como aumento de retenções, formação de gargalos e dificuldades na resolução de conflitos descentralizados.

O principal resultado desta pesquisa foi observado no desempenho do agente Aprendiz. A integração entre aprendizado por reforço, utilizando o algoritmo *Q-Learning*, e o Módulo Moral baseado em utilidade ponderada demonstrou elevada capacidade de adaptação a diferentes cenários de tráfego. Diferentemente dos perfis programados por regras fixas, o agente Aprendiz foi capaz de aprender progressivamente comportamentos seguros e eficientes a partir da interação contínua com o ambiente e das penalizações associadas a infrações, colisões e riscos à vida.

Os resultados mostraram que o agente baseado em aprendizado por reforço alcançou o melhor equilíbrio global entre segurança, eficiência e cooperação, encerrando as simulações rodoviárias com o menor número de colisões entre todos os perfis avaliados. Esse comportamento evidencia que sistemas autônomos guiados por aprendizado adaptativo e avaliação moral dinâmica possuem maior potencial para operar em ambientes urbanos complexos e imprevisíveis.

Dessa forma, conclui-se que a implementação de ética computacional em veículos autônomos não deve se limitar à codificação rígida de regras absolutas de trânsito. Os experimentos demonstraram que abordagens baseadas em mecanismos dinâmicos de recompensa e aprendizagem apresentam maior capacidade de adaptação contextual, permitindo que agentes artificiais desenvolvam estratégias de convivência mais seguras, cooperativas e eficientes em tempo real.

Como trabalhos futuros, sugere-se a ampliação do ambiente de simulação para cenários mais complexos, incluindo múltiplas faixas de circulação, diferentes condições climáticas, comunicação entre veículos e integração com modelos de aprendizado profundo. Além disso, a incorporação de novos perfis morais e variáveis sociais poderá contribuir para investigações mais aprofundadas sobre o impacto ético da Inteligência Artificial aplicada à mobilidade urbana autônoma.

REFERÊNCIAS

- ALVES, G. V.; DENNIS, L.; FISHER, M. A Double-Level Model Checking Approach for an Agent-Based Autonomous Vehicle and Road Junction Regulations. **Journal of Sensor and Actuator Networks**, v. 10, n. 3, p. 41, 2021. DOI: <https://doi.org/10.3390/jsan10030041>.
- ANTON, Howard; RORRES, Chris. **Álgebra linear com aplicações**. 10. ed. Porto Alegre: Bookman, 2012.
- ARNOLD, T.; SCHEUTZ, M. Understanding the spirit of a norm: Challenges for norm-learning agents. **AI Magazine**, v. 44, p. 524–536, 2023. DOI: <https://doi.org/10.1002/aaai.12138>.
- AWAD, E. et al. The Moral Machine experiment. **Nature**, v. 563, n. 7729, p. 59–64, 2018. DOI: <https://doi.org/10.1038/s41586-018-0637-6>.
- BONNEFON, J.-F.; SHARIFF, A.; RAHWAN, I. The social dilemma of autonomous vehicles. **Science**, v. 352, n. 6293, p. 1573–1576, 2016. DOI: <https://doi.org/10.1126/science.aaf2654>.
- BOYLES, R. J. M. A Case for Machine Ethics in Modeling. **Kritike: An Online Journal of Philosophy**, v. 12, n. 1, p. 191-209, jun. 2018. DOI: <https://doi.org/10.25138/12.1.a9>.
- CHAPUT, R. et al. Learning to identify and settle dilemmas through contextual user preferences. In: **2023 IEEE 35th International Conference on Tools with Artificial Intelligence (ICTAI)**, p. 474-479, nov. 2023. DOI: <https://doi.org/10.1109/ICTAI59109.2023.00075>.
- DE SIO, F. S. Killing by autonomous vehicles and the legal doctrine of necessity. **Ethical Theory and Moral Practice**, v. 20, n. 2, p. 411–429, 2017. DOI: <https://doi.org/10.1007/s10677-017-9780-7>.
- DENNIS, L. A.; FISHER, M. **Verifiable Autonomous Systems: Using Rational Agents to Provide Assurance about Decisions Made by Machines**. Cambridge: Cambridge University Press, 2023.
- FOLEY, James D. et al. **Computer graphics: principles and practice**. 2. ed. Reading: Addison-Wesley, 1996.
- FIELD, A. **Descobrimos a estatística usando o SPSS**. 2. ed. Porto Alegre: Artmed, 2009.
- KUIPERS, B. How Can We Trust A Robot?. **Communications of the ACM**, v. 61, n. 3, p. 86–95, 2018. DOI: <https://doi.org/10.1145/3173087>.
- LI, L. et al. A Study of Common Principles for Decision-Making in Moral Dilemmas for Autonomous Vehicles. **Behavioral Sciences**, v. 12, n. 9, p. 344, 2022. DOI: <https://doi.org/10.3390/bs12090344>.
- LORINI, E. A logic for reasoning about moral agents. **Logique & Analyse**, v. 230, p. 177-218, 2015. DOI: <https://doi.org/10.2143/LEA.230.0.3141808>.

MA, Xiaohan et al. Law compliance decision making for autonomous vehicles on highways. **Accident Analysis & Prevention**, v. 201, p. 107648, 2024. DOI: 10.1016/j.aap.2024.107648.

MARKOVICZ, J. V.; ALVES, G. V. Modelagem formal de abordagens éticas para comportamento de agentes. **Departamento Acadêmico de Informática**, Universidade Tecnológica Federal do Paraná (UTFPR), Ponta Grossa, 2023.

MASHAYEKHI, M. et al. Prosocial Norm Emergence in Multi-agent Systems. **ACM Transactions on Autonomous and Adaptive Systems**, v. 17, n. 1-2, art. 3, p. 1-24, set. 2022. DOI: <https://doi.org/10.1145/3540202>.

MELLADO, A. L. L. et al. MASPY: Towards the Creation of BDI Multi-Agent Systems. In: WORKSHOP-ESCOLA DE SISTEMAS DE AGENTES, SEUS AMBIENTES E APLICAÇÕES (WESAAC), 17., 2023. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2023. Disponível em: <https://sol.sbc.org.br/index.php/wesaac/article/view/33440>. Acesso em: 15 jun. 2025.

MOUSAVI, Seyyed Morteza et al. Optimizing autonomous vehicle navigation through reinforcement learning in dynamic urban environments. **World Electric Vehicle Journal**, v. 16, n. 8, p. 472, 2025. DOI: 10.3390/wevj16080472.

PRAKKEN, Henry. On the problem of making autonomous vehicles conform to traffic law. **Artificial Intelligence and Law**, v. 25, n. 3, p. 341-363, 2017. DOI: 10.1007/s10506-017-9210-0.

REZAEI, Mohsen et al. Data-aware path planning for autonomous vehicles using reinforcement learning. **Applied Sciences**, v. 15, n. 11, p. 6099, 2025. DOI: 10.3390/app15116099.

ROBINSON, Jonathan; SMYTH, Joseph; WOODMAN, Roger; DONZELLA, Valentina. Ethical considerations and moral implications of autonomous vehicles and unavoidable collisions. **Theoretical Issues in Ergonomics Science**, v. 23, n. 4, p. 435-452, 2022. DOI: 10.1080/1463922X.2021.1978013.

RHIM, J.; LEE, J.-H.; CHEN, M.; LIM, A. A Deeper Look at Autonomous Vehicle Ethics: An Integrative Ethical Decision-Making Framework to Explain Moral Pluralism. **Frontiers in Robotics and AI**, v. 8, art. 632394, 2021. DOI: <https://doi.org/10.3389/frobt.2021.632394>.

RÜB, I.; DUNIN-KEPLICZ, B. BASTA: BDI-based architecture of simulated traffic agents. **Journal of Information and Telecommunication**, v. 4, n. 4, p. 440-460, 2020. DOI: <https://doi.org/10.1080/24751839.2020.1755529>.

TENNANT, E. et al. Dynamics of Moral Behavior in Heterogeneous Populations of Learning Agents. **AAAI Conference on Artificial Intelligence**, Association for the Advancement of Artificial Intelligence, 2024. Disponível em: arXiv:2403.04202. DOI: <https://doi.org/10.48550/arXiv.2403.04202>.

TOGHI, Behrad; VALIENTE, Rodolfo; SADIGH, Dorsa; PEDARSANI, Ramtin; FALLAH, Yaser P. Social coordination and altruism in autonomous driving. **IEEE Transactions on Intelligent Transportation Systems**, v. 23, p. 24791-24804, 2022. DOI: 10.1109/TITS.2022.3208136.

THORNTON, S. M. et al. Incorporating ethical considerations into automated vehicle control. **IEEE Transactions on Intelligent Transportation Systems**, v. 18, n. 6, p. 1429–1439, 2016. DOI: <https://doi.org/10.1109/TITS.2016.2609339>.

VISHWANATH, A. et al. Reinforcement Learning and Machine ethics: a systematic review. **arXiv preprint arXiv:2401.01804**, 2024. DOI: <https://doi.org/10.48550/arXiv.2407.02425>.

WOOLDRIDGE, Michael. *An Introduction to MultiAgent Systems*. Chichester: John Wiley & Sons, 2001.

WINTERLE, Paulo. **Vetores e geometria analítica**. 2. ed. São Paulo: Pearson, 2014.