

Gabriel Kenji Godoy Shimanuki

**Geração automática de casos críticos para  
avaliação de controle inteligente de veículos  
autônomos em ambiente simulado**

São Paulo, SP

2023

Gabriel Kenji Godoy Shimanuki

**Geração automática de casos críticos para avaliação de  
controle inteligente de veículos autônomos em ambiente  
simulado**

Trabalho de conclusão de curso apresentado  
ao Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro de Computação.

Universidade de São Paulo – USP

Escola Politécnica

Departamento de Engenharia de Computação e Sistemas Digitais (PCS)

Orientador: Prof. Dr. Paulo Sérgio Cugnasca

Coorientador: Alexandre Moreira Nascimento

São Paulo, SP

2023

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

#### Catálogo-na-publicação

Shimanuki, Gabriel Kenji Godoy

Geração automática de casos críticos para avaliação de controle inteligente de veículos autônomos em ambiente simulado / G. K. G. Shimanuki -- São Paulo, 2023.

97 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Veículo Autônomo 2.Simulação 3.Automação 4.Cenário Crítico 5.Geração I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

# Agradecimentos

Gostaria de expressar meus sinceros agradecimentos aos meus familiares, com especial reconhecimento aos meus dedicados pais, Roberto e Vanilda. A dedicação e apoio incondicionais foram cruciais em cada etapa da minha jornada, e manifesto profundo agradecimento pela presença contínua e apoio incansável.

Aos amigos que foram fundamentais em cada fase crucial do meu desenvolvimento.

A todos os professores que, de alguma forma, conseguiram instigar meu interesse e curiosidade.

Agradeço sinceramente ao Grupo de Análise de Segurança (GAS/PCS) pelo acolhimento, ao Professor Paulo Sérgio Cugnasca pela orientação ao longo dos últimos anos e aos mentores Lúcio Flávio Vismari e, de modo especial, Alexandre Moreira Nascimento, pela valiosa perspectiva de visão de mundo embasada em ciência.

Este trabalho foi realizado com o apoio do Itaú Unibanco S.A., por meio do Programa de Bolsas Itaú (PBI), vinculado ao Centro de Ciência de Dados da Escola Politécnica da Universidade de São Paulo



*"The meaning of world is the  
separation of wish and fact."  
– Kurt Friedrich Gödel*

# Resumo

O desenvolvimento de veículos autônomos busca atingir um grau de evolução onde a condução automatizada do veículo será, na média, mais segura do que a efetuada por um condutor humano. No entanto, tal objetivo depende, de muitos fatores como a criação de controles com nível de segurança crítica adequado. Isso envolve vários desafios técnicos, como a garantia de controles automáticos que suportem tal objetivo. O desenvolvimento desses controles requer muitos testes, que necessitam de um grande volume de dados que sejam representativos e promovam a cobertura adequada de todas as configurações possíveis de cenários de tráfego, incluindo quase acidentes que são evitados por manobras evasivas arriscadas. O trabalho de encontrar configurações de situações plausíveis para as quais um controle de veículo autônomo não esteja preparado para lidar não é uma tarefa fácil, mas é importante para que sejam identificados os casos críticos (*corner cases*) para serem incluídos tanto no treinamento como nos testes desses controles. A pesquisa proposta neste projeto tem por objetivo o aprimoramento da identificação e geração de casos críticos para possibilitar aos controles de veículos autônomos uma maior exposição aos *corner cases* durante o processo de treinamento, buscando um aumento de nível de confiabilidade desses sistemas críticos. Esse trabalho tem por objetivo construir uma infraestrutura de geração automatizada de simulações que maximize a ocorrência de situações de acidente e quase acidentes.

**Palavras-chave:** Veículo Autônomo. Simulação. Automação. Cenário Crítico. Geração.

# Abstract

The development of autonomous vehicles seeks to reach a level of evolution where the automated driving of the vehicle will be, on average, safer than that carried out by a human driver. However, this objective depends on many factors such as the creation of controls with an adequate level of critical security. This involves several technical challenges, such as ensuring automatic controls that support this objective. The development of these controls requires a lot of testing, which requires a large volume of data that is representative and promotes adequate coverage of all possible configurations of traffic scenarios, including near misses that are avoided by risky evasive maneuvers. The work of finding plausible situation configurations that an autonomous vehicle control is not prepared to deal with is not an easy task, but it is important so that critical cases (corner cases) can be identified to be included both in the training as well as testing these controls. The research proposed in this project aims to improve the identification and generation of critical cases to enable autonomous vehicle controls to have greater exposure to corner cases during the training process, seeking to increase the level of reliability of these critical systems. This work aims to build an infrastructure for automated simulation generation that maximizes the occurrence of accident and near-accident situations.

**Keywords:** Autonomous Vehicle. Simulation. Automation. Corner Case. Generation.

# Lista de ilustrações

Figura 1 – Componentes de um controle de veículo autônomo - adaptado de (RANJAN; SENTHAMILARASU, 2020) . . . . .	18
Figura 2 – Relação entre áreas de inteligência artificial - adaptado de (RANJAN; SENTHAMILARASU, 2020) . . . . .	19
Figura 3 – Composição de neurônio - adaptado de (SANTOS, 2020) . . . . .	20
Figura 4 – Mimetização neurônios - adaptado de (NIMBLEBOX.AI, 2022) . . . . .	20
Figura 5 – Exemplo de rede neural - adaptado de (SPOTFIRE, 2023) . . . . .	21
Figura 6 – Simulador <i>Webots</i> adaptado de (TUNCALI et al., 2020) . . . . .	25
Figura 7 – Algumas formas de representação dos indivíduos . . . . .	30
Figura 8 – Cruzamento entre indivíduos binários . . . . .	30
Figura 9 – Mutação de indivíduos binário . . . . .	30
Figura 10 – Fluxograma para a implementação do projeto . . . . .	32
Figura 11 – Arquitetura do Carla - Figura adaptado de (PIRRI et al., 2021) . . . . .	36
Figura 12 – Vista de simulação padrão em ambiente urbano . . . . .	37
Figura 13 – Simulações do Carla Simulator em diferentes condições climáticas . . . . .	37
Figura 14 – Visualização de diferentes fluxos de dados gerados pelo simulador (Profundo, RGB, Segmentação Semântica, LiDAR, Optical Flow, LiDAR Semântico) . . . . .	38
Figura 15 – Estrutura genérica da implementação do <i>Framework</i> . . . . .	40
Figura 16 – Vista superior (2D) dos mapas padrão - adaptado de (CARLA-TEAM, 2023a) . . . . .	41
Figura 17 – Imagem completa em (CARLA-TEAM, 2021) . . . . .	42
Figura 18 – Conjuntos de cenários de teste utilizados no estudo de caso de acordo com a Tabela 2 . . . . .	49
Figura 19 – Vetor de probabilidades de eventos para geração de novo indivíduo via AG . . . . .	56
Figura 20 – Comparação do grau de risco nos 6 cenários . . . . .	58
Figura 21 – Comparação da quantidade de colisões nos 6 cenários . . . . .	60
Figura 22 – Comparação da distância mínima considerando todas as simulações válidas nos 6 cenários . . . . .	62
Figura 23 – Comparação da distância mínima excluindo as simulações com colisões nos 6 cenários . . . . .	63
Figura 24 – Comparação da quantidade de simulações inválidas nos 6 cenários . . . . .	65
Figura 25 – Coluna esquerda - Cenário A (Colisão) / Coluna direita - Cenário B (Colisão) . . . . .	94

Figura 26 – Coluna esquerda - Cenário C (Colisão) / Coluna direita - Cenário D (Quase colisão) . . . . .	95
Figura 27 – Coluna esquerda - Cenário E (Seguro) / Coluna direita - Cenário F . .	96

# Lista de tabelas

Tabela 1 – Tabela dos principais sensores disponibilizados no Carla . . . . .	36
Tabela 2 – Descrição dos cenários utilizados no estudo de caso. Veículo A: controle, Veículo B: competidor. Vias $2x2$ representam vias de mão dupla com duas faixas de rodagem em cada sentido . . . . .	48
Tabela 3 – Descrição dos parâmetros utilizados na modelagem dos cenários de cruzamento . . . . .	51
Tabela 4 – Lista de hipóteses de métricas para criação de função objetivo - métricas em negrito e sublinhadas foram efetivamente utilizadas . . . . .	54
Tabela 5 – Risco associado a cada métrica . . . . .	55
Tabela 6 – Probabilidades utilizadas para geração dos experimentos do estudo de caso . . . . .	56
Tabela 7 – Risco de condução Médio - Todos os cenários . . . . .	59
Tabela 8 – Somatório de colisões - intervalos de 1 a 10 gerações; 11 a 20 gerações; 21 a 30 gerações . . . . .	61
Tabela 9 – Distância mínima média dos cenários válidos - intervalo de 1 a 10 gerações; intervalo de 11 a 20 gerações; intervalo de 21 a 30 gerações . . . . .	64
Tabela 10 – Distância mínima média dos cenários válidos que não ocorrem colisão - intervalo de 1 a 10 gerações; intervalo de 11 a 20 gerações; intervalo de 21 a 30 gerações . . . . .	64
Tabela 11 – Somatório de inválidos - intervalos de 1 a 10 gerações; 11 a 20 gerações; 21 a 30 gerações . . . . .	66
Tabela 12 – Risco de condução - Cenário A . . . . .	84
Tabela 13 – Risco de condução - Cenário B . . . . .	85
Tabela 14 – Risco de condução - Cenário C . . . . .	86
Tabela 15 – Risco de condução - Cenário D . . . . .	87
Tabela 16 – Risco de condução - Cenário E . . . . .	88
Tabela 17 – Risco de condução - Cenário F . . . . .	89
Tabela 18 – Quantidade de colisões por geração . . . . .	90
Tabela 19 – Distância mínima média dos cenários válidos por geração . . . . .	91
Tabela 20 – Distância mínima média dos cenários válidos que não ocorrem colisão por geração . . . . .	92
Tabela 21 – Quantidade de simulações inválidas por geração . . . . .	93

# Lista de abreviaturas e siglas

AG	Algoritmo Genético
IA	Inteligência Artificial
LIDAR	<i>Light Detection And Ranging</i>
NHTSA	<i>National Highway Traffic Safety Administration</i>
RADAR	<i>Radio Detection And Ranging</i>
RAG	Rede Adversarial Generativa
TCC	Trabalho de Conclusão de Curso
VA	Veículos Autônomos

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
1.1	Motivação	14
1.2	Objetivos	15
1.3	Justificativa	15
1.4	Organização do Trabalho	16
<b>2</b>	<b>ASPECTOS CONCEITUAIS</b>	<b>17</b>
2.1	Breve Contextualização Histórica	17
2.2	Veículos Autônomos	17
2.3	Controle Autônomo	19
2.4	Segurança Crítica	21
2.5	Riscos de Condução Veicular	23
2.6	Teste de Segurança Crítica	24
<b>3</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>26</b>
3.1	Técnicas de Geração de Cenários Críticos	26
3.2	Algoritmos Genéticos	29
<b>4</b>	<b>MÉTODO DO TRABALHO</b>	<b>32</b>
<b>4.1</b>	<b>Concepção do Projeto</b>	<b>32</b>
4.1.1	Mapeamento de Literatura	33
4.1.2	Testes de simuladores de ambiente de tráfego	33
4.1.3	Construção dos cenários de base e Implementação de controles de VAs	34
4.1.4	Experimentos de exploração de casos críticos	34
4.1.5	Avaliação dos resultados	34
4.1.6	Elaboração de materiais de divulgação dos resultados	34
<b>4.2</b>	<b>Tecnologias Utilizadas</b>	<b>35</b>
4.2.1	Python	35
4.2.2	Carla Simulator	35
4.2.3	Scenic	37
4.2.4	Bash Script	38
<b>4.3</b>	<b>A implementação</b>	<b>39</b>
4.3.1	Versões de Software	39
4.3.2	Principais elementos	39
4.3.3	Carla Simulator	39
4.3.3.1	Execução	40



4.3.3.2	Gravador . . . . .	41
4.3.3.3	Formato dos registros . . . . .	42
4.3.4	Scenic . . . . .	42
4.3.4.1	Descrição . . . . .	42
4.3.4.2	Funcionamento . . . . .	43
4.3.4.3	Execução Genérica . . . . .	45
4.3.4.4	Modificações da biblioteca para coleta de dados pelo Scenic . . . . .	45
4.3.4.5	Instalação da biblioteca modificada . . . . .	47
4.3.5	Geração dos Parâmetros . . . . .	48
4.3.5.1	Cenários de risco . . . . .	48
4.3.5.2	Descrição e geração dos cenários . . . . .	49
4.3.6	Algoritmo Genético . . . . .	51
4.3.6.1	Codificação do indivíduo . . . . .	51
4.3.6.2	Função Objetivo . . . . .	53
4.3.6.3	Parâmetros . . . . .	55
<b>5</b>	<b>RESULTADOS . . . . .</b>	<b>57</b>
<b>5.1</b>	<b>Estudos de Caso . . . . .</b>	<b>57</b>
5.1.1	Avaliação do grau de risco . . . . .	57
5.1.2	Quantidade de colisões . . . . .	59
5.1.3	Distância Mínima de Aproximação . . . . .	61
5.1.4	Quantidade de Inválidos . . . . .	64
<b>5.2</b>	<b>Limitações do Projeto . . . . .</b>	<b>66</b>
5.2.1	Geração de cenários do Scenic . . . . .	66
5.2.2	Algoritmo Genético . . . . .	67
5.2.3	Tipos de risco . . . . .	68
<b>6</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>69</b>
<b>6.1</b>	<b>Conclusões do Projeto de Formatura . . . . .</b>	<b>69</b>
<b>6.2</b>	<b>Contribuições . . . . .</b>	<b>70</b>
<b>6.3</b>	<b>Trabalhos Futuros . . . . .</b>	<b>71</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>72</b>
	<b>APÊNDICES . . . . .</b>	<b>79</b>
	<b>APÊNDICE A – ALGUNS CÓDIGOS DO PROJETO . . . . .</b>	<b>80</b>

<b>APÊNDICE B – TABELAS DE RESULTADOS DO ESTUDO DE CASO . . . . .</b>	<b>84</b>
<b>APÊNDICE C – EXEMPLOS DE SIMULAÇÕES GERADAS . . . . .</b>	<b>94</b>

# 1 Introdução

A pesquisa em Inteligência Artificial (IA) vem sendo desenvolvida nas últimas décadas, possibilitando a sua aplicação em diversos campos, incluindo a automação do processo de condução veicular a partir de algoritmos de controles inteligentes - embarcados no próprio veículo - que possibilitam a criação de Veículos Autônomos (VAs). O principal objetivo desse tipo de veículo é a redução das taxas de acidentes e riscos relacionados à condução insegura, que têm, em grande parte, associação às falhas humanas (NASCIMENTO *et al.*, 2020; SHIMANUKI, 2021; SHIMANUKI, 2022).

A aplicação de algoritmos de aprendizado de máquina, especialmente redes neurais, é um dos métodos mais frequentemente empregados na condução automatizada. Essa técnica é marcada pela necessidade de vastos conjuntos de dados e pela demanda significativa de capacidade computacional para treinar a rede com um propósito específico. Como VAs são sistemas de segurança crítica - ou seja, falhas podem levar a consequências graves, que incluem perda de vida, danos ao meio ambiente ou perdas financeiras significativas - é fundamental que a confiabilidade do algoritmo de controle seja garantida a partir de treinamento e testes adequados compatíveis com os níveis da importância desses sistemas.

## 1.1 Motivação

O desenvolvimento de VAs é um desafio complexo e desafiador, sobretudo na garantia da qualidade do sistema de controle. Isso decorre da disparidade entre o ambiente utilizado para o treinamento e o ambiente real em que esses veículos operam, podendo ocasionar comportamentos não confiáveis e comprometer a segurança física desses sistemas (XU; SOULY; BRAHMA, 2021; TUNCALI *et al.*, 2020).

Para garantir a segurança desses veículos durante a condução, é necessário que seus controles tenham acesso a grandes quantidades de dados e cobertura abrangente para treinamento e avaliação de segurança. A coleta de dados é essencial para garantir que os controles dos veículos autônomos sejam capazes de lidar com situações críticas de alto risco de acidentes. Existem duas alternativas mais comumente utilizadas para a obtenção desses dados: condução de veículos instrumentalizados com sensores e câmeras por diferentes cenários e situações, ou a utilização de simuladores que emulam essa atividade (KAUR *et al.*, 2021; ROSIQUE *et al.*, 2019). Apesar da existência de campos de teste como o *Mcity* da *Michigan University* (2015) (BRIEFS, 2015), ou *Castle* da Waymo, localizado em uma antiga base da força aérea no estado da americano da Califórnia (2018) (CERF, 2018; WAYMO, 2020), a utilização apenas da geração de dados física tende a apresentar uma cobertura insuficiente, pois os casos de quase acidentes, incluindo as manobras evasivas

que as evitaram são proporcionalmente raros. Além disso, a coleta forçada dessas situações críticas em ambientes reais e urbanos é contraprodutiva, custosa e perigosa, tendo em vista os potenciais riscos envolvidos (XU; SOULY; BRAHMA, 2021; GE et al., 2022; HOREL et al., 2022; NASCIMENTO et al., 2019b). A virtualização desse processo, mediante utilização de simuladores de tráfego de ambiente controlado (AMINI et al., 2022) tem se mostrado uma alternativa promissora, possibilitando maior segurança, economia e escalabilidade (NASCIMENTO et al., 2019a; NASCIMENTO et al., 2019b; NIU et al., 2021; ZHANG et al., 2023), sendo a grande maioria dos estudos da área suportadas por simuladores de tráfego.

## 1.2 Objetivos

O objetivo principal deste projeto é desenvolver uma metodologia para gerar automaticamente casos críticos, ou *corner cases*, com o intuito de contribuir para pesquisas futuras no desenvolvimento de sistemas de controle veicular mais robustos, baseados em cenários de condução de maior risco. Utilizando uma estrutura que extrai casos de condução arriscada de situações de tráfego comumente referenciadas na literatura, o projeto visa incrementar a geração de dados que representem situações de risco elevado, sendo desenvolvido em ambiente simulado. A principal abordagem utilizada envolve o emprego do Algoritmo Genético (AG) para otimizar a busca por cenários de risco. Juntamente com a implementação da infraestrutura de simulação, a pergunta de interesse a ser respondida por esse trabalho é:

Essa heurística é apropriada para a busca de configurações de cenários que sejam considerados *corner cases*?

Vale ressaltar que esse TCC foi desenvolvido no âmbito do programa de pré-mestrado do módulo de Pesquisa em Engenharia de Computação, desempenhando um papel fundamental para o aperfeiçoamento da futura tese de mestrado a ser desenvolvida no próximo ano.

## 1.3 Justificativa

A proposta de trabalho tem por objetivo explorar a utilização de algumas técnicas baseadas em heurísticas para a geração de dados sintéticos, particularmente aqueles de difícil obtenção. O intuito é contribuir para o aprimoramento do desenvolvimento de controles de veículos autônomos. O foco da pesquisa está em investigar a relevância dessas técnicas sob o ponto de vista da construção de um ambiente capaz de facilitar a geração simplificada de configurações de interesse para pesquisas. Este ambiente visa otimizar a execução de simulações que permitam a geração de dados sintéticos.

A importância dessa pesquisa reside no fato de que treinamentos e testes baseados em simulações virtuais exigem um grande número de casos críticos (GE et al., 2022; SUN et al., 2021), que raramente ocorrem em ambientes naturais de condução. Vale ressaltar que esses casos são raros em ambientes convencionais de condução. O uso de casos críticos pode ajudar a mitigar ou prevenir potenciais riscos de colisão, possibilitando a construção de *datasets* enriquecidos para treinamento de controles de veículos autônomos mais robustos, reduzindo, assim, a incidência de conduções com algum grau de risco que resultam em acidentes ou quase-acidentes. A diminuição de acidentes em escala mundial não apenas possui um impacto econômico significativo, mas também repercute de maneira relevante no âmbito social, contribuindo para a preservação da vida. A proposta de trabalho, por fim, é pertinente para a área de desenvolvimento de sistemas de segurança crítica aplicada em veículos autônomos.

## 1.4 Organização do Trabalho

Esse trabalho apresenta os principais conceitos para entendimento do funcionamento de VAs capítulo 2. A revisão bibliográfica sobre métodos de geração de casos críticos é apresentado no capítulo 3. No capítulo 4 é apresentada a especificação e metodologia do trabalho. No capítulo 5 são apresentados os principais resultados e limitações do trabalho. Finalmente, no capítulo 6 são feitas as considerações sobre o balanço geral do trabalho, as principais contribuições e oportunidades de pesquisas futuras.

## 2 Aspectos Conceituais

### 2.1 Breve Contextualização Histórica

A concepção de Veículo Autônomo (VA) vem sendo desenvolvido desde meados dos anos 80. Conforme (ANDERSON et al., 2016), a evolução dessa tecnologia pode ser entendida em três grandes blocos temporais. No primeiro, entre 1980 a 2003, foram realizados os primeiros estudos básicos de transportes autônomos a partir da parceria entre centros de pesquisas universitários, agências de transportes e empresas. O segundo bloco é marcado pelos desafios universitários de automação veicular da DARPA (*Defense Advanced Research Projects Agency*) (BEHRINGER et al., 2004; BUEHLER; IAGNEMMA; SINGH, 2009), entre 2003 a 2007, em que foram evidenciados tanto para a indústria, quanto para a academia a viabilidade técnica desse tipo de automação. Finalmente, o terceiro bloco de 2007 até os dias atuais, é o período de desenvolvimento comercial, marcado pelas parcerias entre os integrantes das equipes de destaque nos desafios da DARPA e algumas empresas como General Motors, Volkswagen e Google. Desde então, alguns estudos estimam o volume dos investimentos realizados para o desenvolvimento dessa tecnologia, sendo na ordem de US\$80 bilhões - entre 2014 a 2017 - (KERRY; KARSTEN, 2017), a mais de US\$100 bilhões na última década (SHETTY et al., 2021).

### 2.2 Veículos Autônomos

O veículo autônomo é um automóvel convencional que embarca uma variedade de tecnologias que o auxiliam na tomada de decisão em condições de tráfego urbano. Tipicamente, os veículos são equipados com sensores e câmeras para detecção do ambiente, podendo, em condições operacionais plenas, navegar sem nenhuma intervenção humana em tempo real. (MOBILEYE, 2023) é um exemplo de empresa que oferece sensores com soluções para prevenção de acidentes.

Atualmente, os projetos de veículos autônomos exploram algumas alternativas para o desenvolvimento de um sistema autônomo mais robusto. As duas abordagens mais bem sucedidas são abordagem robótica (SCHWARTING; ALONSO-MORA; RUS, 2018), (CHEBLY; TALJ; CHARARA, 2017) e abordagem de aprendizado profundo (KUUTTI et al., 2020), (GRIGORESCU et al., 2020)

Ambas as abordagens podem ser entendidas como complementares no desenvolvimento de sistemas autônomos. A abordagem robótica opera pela integração das saídas de um conjunto de sensores, analisando o ambiente no qual um veículo está inserido e

solicitando comandos de navegação coerentes. Isso é feito com o objetivo de seguir tanto o plano de rota quanto o plano de controle estabelecidos. Essa abordagem, apesar de funcional, não é efetiva, posto que a condução veicular pode apresentar uma infinidade de estados, dificilmente mapeáveis, inviabilizando a utilização dos veículos em diversas situações.

A abordagem por aprendizado profundo surge para auxiliar no processo de generalização de condução, por meio da imitação do comportamento de condução humana. Ou seja, ao invés da condução ser baseado em uma lista de descrições diretas que estabelecem regras fechadas para cada tipo de situação, a generalização proporcionada por algoritmos de aprendizagem abstrai a lista de regras que uma equipe de programadores deve construir. A inteligência do algoritmo, nesse sentido, surge pelo processo de abstração de um modelo computacional que mimetiza a arquitetura de um cérebro humano, com um conjunto de neurônios e conexões entre eles.

Genericamente, os cinco principais componentes de um veículo autônomo são visão computacional, fusão de sensores, localização, plano de trajetória e controle do veículo (KATO et al., 2015). No diagrama da figura 1 é mostrado como esses componentes são dispostos.

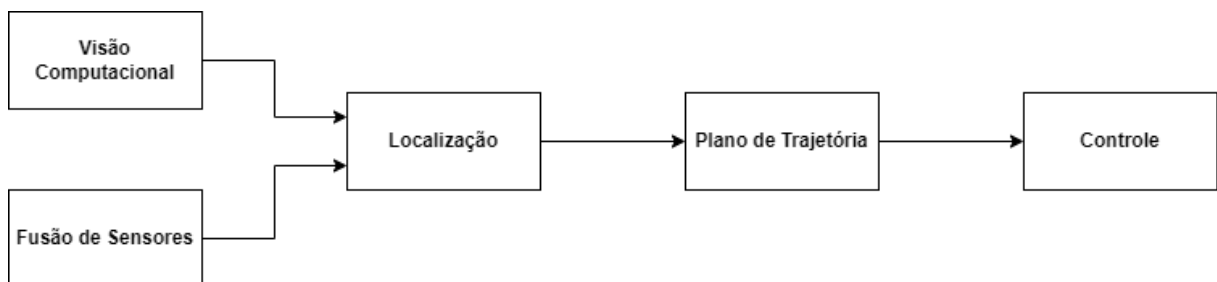


Figura 1 – Componentes de um controle de veículo autônomo - adaptado de (RANJAN; SENTHAMILARASU, 2020)

Esses componente são utilizados da seguinte maneira:

- Visão Computacional: técnica utilizada para que o veículo perceba o ambiente para que ajude na tomada de decisão do ponto de vista do reconhecimento de objetos, sinais de trânsito, pessoas e demais veículos
- Fusão de sensores: uma forma de incorporar dados de vários sensores como Radares, Lidares e Lasers, auxiliando no processo de reconhecimento de ambiente
- Localização: informação que posiciona o veículo em algum ponto do globo terrestre
- Plano de trajetória: o plano é construído para a execução da trajetória que o veículo deve seguir, baseado em sua localização inicial, sua localização final e os dados que descrevem as condições ambiente em que o veículo está inserido

- Controle do veículo: conjunto de atuadores que manipulam o volante e os pedais de aceleração e frenagem para a navegação do veículo.

## 2.3 Controle Autônomo

O aprendizado profundo é uma sub-área de Aprendizado de máquina baseado em Redes Neurais Artificiais. A figura 2, ilustra a relação entre as técnicas mencionadas. A técnica de aprendizado profundo mimetiza o cérebro humano, sendo inspirado na estrutura da função cerebral. O conceito de aprendizado profundo não é recente, no entanto foi explorado imensamente nos últimos anos. O motivo dessa popularização e o sucesso da técnica está relacionado com o desenvolvimento tecnológico das unidades de processamento, melhoria de CPUs e GPUs, bem como a enorme quantidade de dados disponíveis. Uma das razões para as redes neurais profundas apresentarem um bom desempenho se deve à identificação de relacionamentos complexos e a alta dimensão de dados (LECUN; BENGIO; HINTON, 2015).

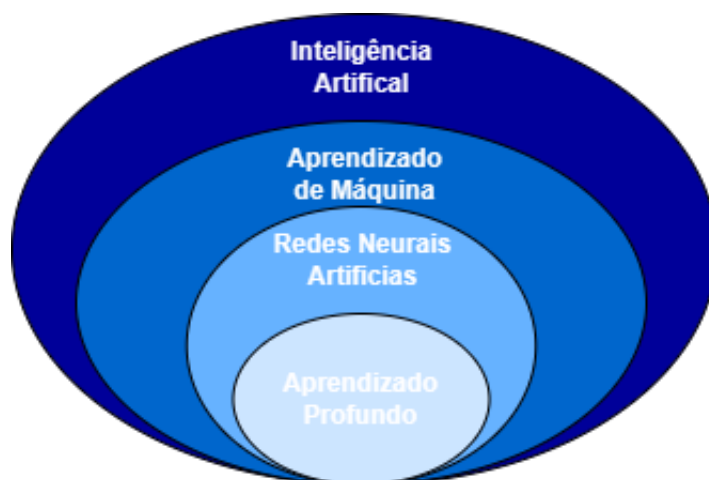


Figura 2 – Relação entre áreas de inteligência artificial - adaptado de (RANJAN; SENTHAMILARASU, 2020)

O funcionamento de redes neurais se dá pela relação entre neurônios. Neurônios são células que caracterizam sistemas nervosos centrais, responsáveis pela transmissão de pulsos elétricos. O funcionamento de um neurônio se dá por meio da conexão entre eles. O dendrito de um neurônio se conecta ao axônio de outros, para que o sinal seja transferido do axônio para o dendrito (DECO et al., 2008). Na figura 3 são ilustrados os principais elementos que compõem um neurônio.

Na área de inteligência artificial, os perceptrons são reconhecidos como os equivalentes matemáticos dos neurônios biológicos (BLOCK, 1962). O princípio fundamental do funcionamento de um perceptron assemelha-se ao de um neurônio biológico, onde o sinal elétrico (entrada do perceptron) é transmitido do dendrito para o corpo celular do neurônio, resultando em uma única saída ou sinal elétrico através de um único axônio. No



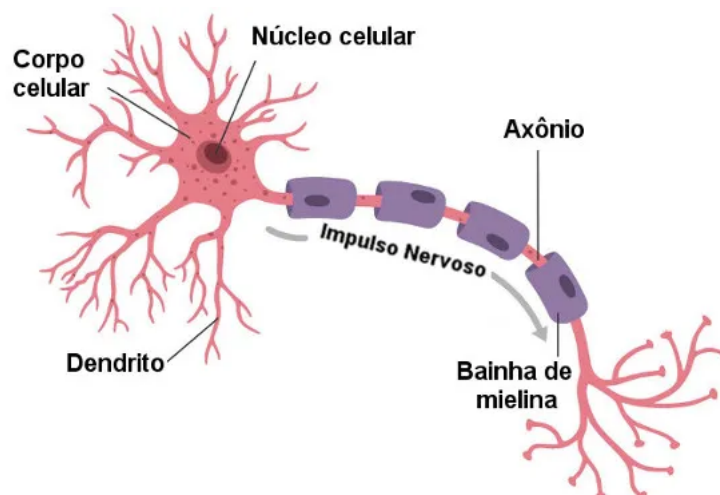


Figura 3 – Composição de neurônio - adaptado de (SANTOS, 2020)

contexto do neurônio computacional, uma operação é executada por meio de uma função de ativação. Essa função realiza uma transformação, geralmente, não linear, ou seja, a entrada do perceptron é processada e encaminhada para a saída, alimentando assim a entrada de outro perceptron. A figura 4 ilustra o funcionamento de um perceptron.

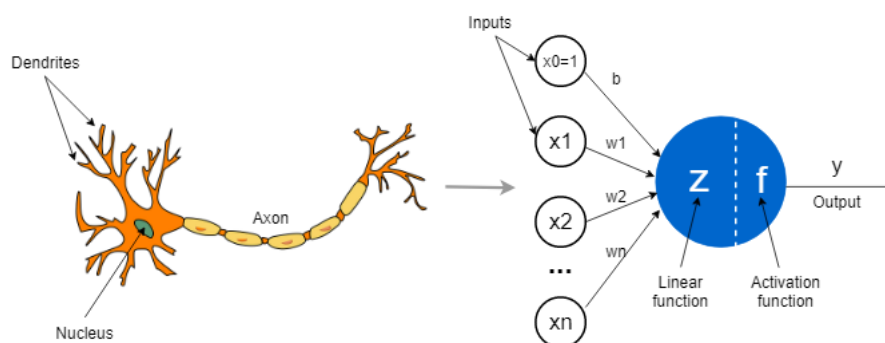


Figura 4 – Mimetização neurônios - adaptado de (NIMBLEBOX.AI, 2022)

As redes neurais artificiais são construídas como teias de perceptrons, como mostrado na figura 5

Ao analisar o diagrama, é possível identificar várias camadas de perceptrons interconectados por meio de suas entradas e saídas. A camada de entrada é representada em amarelo, enquanto as camadas verdes são denominadas camadas ocultas, e a camada vermelha constitui a saída da rede. As camadas de entrada consistem em valores reais dos dados. As camadas intermediárias situam-se entre as camadas de entrada e saída. Quando há três ou mais camadas ocultas presentes, a rede é considerada uma rede neural profunda. A última camada, a de saída, fornece a estimativa final para o tipo específico de saída que está sendo avaliado. Em um contexto de classificação de imagens, por exemplo, em um conjunto de animais domésticos, a saída pode indicar a classificação de que uma determinada imagem contém um cachorro.

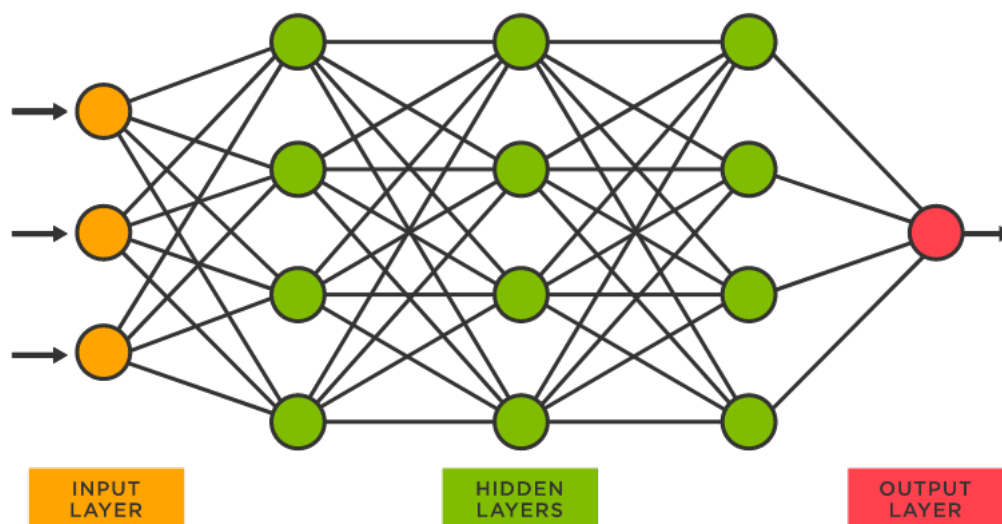


Figura 5 – Exemplo de rede neural - adaptado de (SPOTFIRE, 2023)

Para o contexto de VAs, o controle autônomo pode ser dividido em duas tarefas que são o controle lateral (manipulação do volante) e o controle longitudinal (manipulação de pedais).

Para o contexto de VAS, controles baseados em redes profundas são utilizadas nas duas principais tarefas que são o controle lateral (manipulação do volante) e o controle longitudinal (manipulação dos pedais) para um condução com o menor nível de risco possível (KUUTTI et al., 2020). Contudo, a grande limitação desse tipo de técnica é a incapacidade de generalização do controle nas situações de condução para as quais ele não foi treinado (PAREKH et al., 2022). Nesses casos, a exposição de VAs em situações reais de condução podem resultar em acidentes gerando a perda de vida e de bens materiais (YURTSEVER et al., 2020).

## 2.4 Segurança Crítica

Os sistemas de segurança crítica são sistemas cuja falha podem resultar em perda de vidas, danos materiais significativos e/ou meio ambiente, conforme (NASCIMENTO et al., 2020; KNIGHT, 2002). Como exemplo, existem sistemas críticos de segurança que envolvem algumas áreas de aplicação como, dispositivos médicos, controle de vôo de aeronaves, armas, sistemas nucleares e mais especificamente sistemas de controle de veículos autônomos. Os veículos autônomos podem ser considerados como sistemas de segurança crítica, posto que caso operem de maneira indesejável podem colocar em risco tanto a vida de pessoas, quanto o ambiente em que está inserido, como os próprios passageiros, pedestres, veículos ao seu redor e pessoas em outros veículos. Nesse sentido, o desenvolvimento efetivo de controles de veículos autônomos engloba, obrigatoriamente, a segurança de sua operação.

A segurança da operação dos veículos autônomos é um aspecto fundamental do

desenvolvimento tecnológico desse setor. Nesse sentido, foi adotada a escala de autonomia da *SAE - Society of Automotive Engineering's* (SAE... , 2023) para determinar a capacidade de autonomia de diversos níveis, ajudando a compreender os passos de desenvolvimento tecnológico desse setor. A seguir são apresentados brevemente as descrições dos seis níveis.

- Nível 0 - Carros Manuais

No nível 0 de autonomia, tanto o volante quanto os pedais do veículo são controlados pelo humano. Esse nível pode incluir alarmes de atenção para o condutor, mas o veículo em si, não realiza nenhuma ação.

- Nível 1 - Suporte de Direção

No nível 1, o condutor se encarrega de controlar a maioria dos recursos do carro. O suporte de direção passa a ter informações sobre ambientação do veículo, bem como pedais e volante. Nesse nível, o veículo passa a ter autonomia para utilização de freio automático de emergência.

- Nível 2 - Automação Parcial

No nível 2, o veículo é parcialmente automatizado. O veículo pode assumir a direção ou aceleração, reduzindo a necessidade do condutor realizar algumas tarefas. No entanto, o motorista ainda é necessário para monitorar as funções críticas de segurança e fatores ambientais.

- Nível 3 - Automação Condicional

No nível 3, o próprio veículo realiza todo monitoramento ambiental, por meio dos sensores. Nesse nível, o veículo pode conduzir de maneira autônoma em algumas situações, mas o motorista ainda deve estar alerta para assumir o controle caso o veículo seja exposto a alguma situação que possa resultar em alguma situação crítica de segurança.

- Nível 4 - Alta automação

No nível 4, o veículo pode controlar a direção e os pedais de aceleração e frenagem do veículo. O veículo é capaz de monitorar o ambiente, demais veículos, pedestres, objetos e sinalizações de trânsito. Nesse nível o veículo pode dirigir de maneira autônoma na maior parte do tempo, no entanto, o condutor ainda é necessário para assumir o controle em algumas situações de segurança crítica.

- Nível 5 - Automação completa

No nível 5, o veículo é totalmente autônomo. Nesse nível não é necessária a presença de um condutor humano. O veículo é capaz de controlar todas as tarefas críticas.

## 2.5 Riscos de Condução Veicular

A condução veicular tem uma relação íntima com uma lista de fatores que contribuem com a complexidade e dinâmica de tráfego urbano e rodoviário. Entre esses fatores, o impacto do risco emerge como uma preocupação central, influenciando significativamente na segurança das vias e nos projetos de desenvolvimento de veículos e equipamentos para a prevenção de acidentes. O risco durante a condução pode ser conceituado, de maneira análoga a outras áreas, como o produto qualitativo entre a probabilidade de ocorrência de eventos indesejados, como acidentes ou incidentes, e a gravidade de suas consequências. (EVANS, 2004) e (CHAND; JAYESH; BHASI, 2021) apresentam compreensões abrangentes dos diversos elementos que compõem essas fontes de risco é essencial para a implementação eficaz técnicas de prevenção e mitigação de acidentes.

As fontes de risco podem ser classificadas em diversas categorias. A seguir são listas algumas dessas fontes em condução:

### 1. **Risco Climático** - Impacto sobre as condições de dirigibilidade

- Condições meteorológicas extremas: Chuvas intensas, neve, neblina e ventos fortes que podem reduzir a visibilidade e, por consequência, aumentar a probabilidade de acidentes.
- Gelo e neve nas vias: Superfícies escorregadias colaboram para a perda de controle do veículo.

### 2. **Risco Humano** - Impacto sobre a segurança de tráfego causada por fatores humanos

- Comportamento do motorista: Distração, fadiga, excesso de velocidade e condução sob efeito de substâncias (álcool, drogas).
- Falta de conhecimento ou habilidade: Inexperiência, treinamento ou conhecimentos inadequados sobre as regras de trânsito.
- Condições de saúde do motorista: Problemas de saúde, como doenças cardiovasculares, visão debilitada e sonolência, podem afetar a capacidade do condutor.
- Interação com dispositivos eletrônicos: Uso de dispositivos eletrônicos durante a condução (celular).

### 3. **Condições da Via** - Impacto sobre o tempo de reação e habilidade dos condutores

- Má conservação da via: Buracos e manutenção insuficiente das vias.
- Tráfego congestionado: Alta densidade de veículos e pedestres.
- Construções e obstáculos na via: Obras, cones, animais entre outros obstáculos.

- Sinalização Inadequada: Falhas na sinalização de trânsito.

#### 4. Veículos e Equipamentos

- Manutenção inadequada: Problemas mecânicos devido à manutenção inadequada.
- Defeitos de fabricação: Veículos defeituosos.

#### 5. Interações com Outros usuários da Via

- Comportamento dos outros motoristas: Ações imprudentes ou arriscadas de outros condutores.
- Ciclistas e pedestres: Interação com usuários não motorizados, como ciclistas e pedestres.

## 2.6 Teste de Segurança Crítica

Em se tratando de sistema de segurança crítica, o desenvolvimento dos veículos autônomos estão amplamente vinculados com a realização de testes. Esses testes são pré-requisitos de validação de confiabilidade e segurança exigidos por órgãos reguladores como o NHTSA, bem como o público que consumirá essa classe de produto. No entanto, a realização de testes ineficientes, que não sejam capazes de descobrir potenciais falhas operacionais que podem induzir a um erro do sistema, podem ser letais e causar acidentes severos como mostra (SONG et al., 2022; GAMBI; MUELLER; FRASER, 2019). O acidente fatal envolvendo um veículo de teste da Uber, no Arizona, em 2018, ilustra uma falha crítica nos procedimentos de teste, o que resultou no falecimento de uma pessoa. A investigação posterior revelou que o VA falhou em identificar a presença da pedestre a tempo, devido a dificuldades em seu sistema de percepção para reconhecer objetos e pessoas na estrada. Além disso, a motorista de segurança no veículo não estava atenta à estrada no momento do acidente, contribuindo para a fatalidade. (CONGER, 2020).

Como mencionado em seções anteriores, controles fundamentados em técnicas de IA baseados no paradigma de aprendizado de máquina demandam a existência da grande quantidade de dados e uma grande abrangência dos mesmos para garantir uma adequada cobertura de cenários possíveis durante a condução do veículo, com o intuito de que seja possível atingir elevada eficiência e segurança crítica. A cobertura desses dados deve abranger, inclusive, os casos críticos que apresentam grande risco de acidentes, ou iminência de acidentes tanto para treino como para a avaliação da capacidade de segurança dos AVs. Existem duas alternativas mais comumente utilizadas para a obtenção desses dados: condução de veículos instrumentalizados com sensores e câmeras por diferentes cenários e situações, ou a utilização de simulações que emulam essa atividade. A utilização



Figura 6 – Simulador *Webots* adaptado de (TUNCALI et al., 2020)

apenas da geração de dados física tende a ter uma cobertura insuficiente, pois os casos de quase acidentes incluindo as manobras evasivas que possibilitaram o evitamento são proporcionalmente raros, além de que a coleta forçada dessas situações críticas em ambientes urbanos são contraproduativas, custosas e perigosas, tendo em vista os potenciais riscos envolvidos (XU; SOULY; BRAHMA, 2021; GE et al., 2022; HOREL et al., 2022; NASCIMENTO et al., 2020).

A virtualização desse processo, mediante utilização de simuladores de tráfego de ambiente controlado (AMINI et al., 2022) tem se mostrado uma alternativa promissora, possibilitando maior segurança, economia e escalabilidade (NIU et al., 2021; NASCIMENTO et al., 2020). A figura 6 mostra alguns exemplos de 2 simuladores utilizados para o treinamento de controles de VAs em ambiente simulado. Ainda que a virtualização traga benefícios nas reduções de acidentes, de exposição de protótipos em validação em tráfego real, e de custos, a definição, identificação e geração de dados provenientes de cenários críticos são uma tarefa difícil e importante para o desenvolvimento de controles de VAs (SUN et al., 2021; NIU et al., 2021).

Em suma, a realização de testes é essencial para a garantia de segurança na operação dos veículos autônomos. A complexidade do desenvolvimento desses sistemas e a necessidade de se considerar uma grande variedade de cenários demanda a realização de testes abrangentes, que incluam as situações críticas que possam culminar em acidente. Uma das formas mais comuns utilizadas para o desenvolvimento de testes é a simulação computacional em ambiente controlado.

## 3 Revisão Bibliográfica

Nesse capítulo são apresentados principais aspectos encontrados na literatura sobre a geração de casos de condução crítica de VAs e otimização via heurísticas evolutivas. Assim, são abordados trabalhos correlatos e técnicas relevantes para a comparação com o projeto. As seções a seguir correspondem aos métodos de geração e heurísticas evolutivas aplicadas ao contexto de ...

### 3.1 Técnicas de Geração de Cenários Críticos

Segundo a revisão de (DING et al., 2023), são categorizados três principais técnicas utilizadas para geração de *corner cases* em simulação que são:

1. **Geração Baseada em Dados** - As técnicas estão relacionadas com a análise dos dados de bancos de dados. Podem ser subdivididas em três categorias.
  - **Reprodução de Dados:** Envolve a replicação de registros de cenários, por meio de ferramentas de conversão especializadas para simulações computacionais. Essa abordagem permite uma análise mais aprofundada de cenários com repetição das condições reais.
  - **Agrupamento:** Essa abordagem categoriza os cenários coletados utilizando métodos de agrupamento não supervisionado. Essa categorização agrupa cenários similares proporcionando uma maneira eficiente de testar VAs em tipos específicos de situações. Essa técnica pode ser considerada como uma otimização do processo de teste, pois orienta os esforços para os conjuntos mais representativos de risco de condução.
  - **Perturbação Aleatória:** Com o intuito de diversificar os cenários, utiliza-se a técnica de perturbação aleatória, uma prática que incorpora aleatoriedade controlada nos cenários preexistentes. Essa abordagem tem como objetivo elevar consideravelmente a diversidade e complexidade dos testes, proporcionando um conjunto mais abrangente de situações que um VA pode enfrentar durante suas operações.
2. **Geração Adversarial** - Técnicas baseadas na criação de modelos competitivos. Essa categoria pode ser desdobrada em dois subtipos.
  - **Geração de Cenários Estáticos:** Consiste na produção das posições de objetos e na utilização de técnicas de renderização para alcançar a saída final



do cenário. Essa abordagem visa explorar desafios estáticos que podem afetar o desempenho do VA em situações de condução específicas.

- **Geração de Cenários Dinâmicos:** Concentra-se na avaliação dos módulos de planejamento e controle dos VAs, demandando a criação de cenários dinâmicos e sequenciais. Esse subtipo busca simular eventos em constante evolução, proporcionando um entendimento mais aprofundado sobre a capacidade do controle do VA de enfrentar desafios em ambientes de condução em constante mudança.

3. **Geração Baseada em Conhecimento** - Incorpora conhecimento externo do domínio no processo de geração de cenários. Essa categoria compreende:

- **Regras Pré-Definidas:** Elabora cenários mediante regras predeterminadas e condições específicas para desencadear eventos.
- **Aprendizado Orientado por Conhecimento:** Unifica métodos de aprendizado e conhecimento do domínio para criar cenários diversos, aplicando conhecimento explícito para orientar o processo de geração. Essa abordagem promove uma síntese eficaz entre métodos de aprendizado e a expertise do domínio, resultando na produção de cenários variados e alinhados com o conhecimento especializado.

Na prática, as principais linhas de pesquisa sobre a perspectiva de geração de cenários são a adversarial e a geração baseada em conhecimentos - majoritariamente com algoritmos de busca evolutivos que utilizam um conjunto de conhecimentos sobre o tipo o objetivo da geração dos cenários. A geração baseada em dados permeia as duas, porque as técnicas mencionadas anteriormente, no geral, utilizam informações de análises de agrupamento de dados de acidentes reais e são comparadas com geração aleatória de dados - referência comum de muitos trabalhos. Vale observar que a perturbação aleatória é a forma menos custosa de gerar cenários de teste, embora menos eficaz, conforme a maioria dos estudos que são detalhados posteriormente.

Para a categoria de Geração Adversarial, é importante observar trabalhos que fazem a geração de cenários críticos utilizando a técnica de Rede Adversarial Generativa (RAG). As RAGs são modelos de aprendizado profundo (*deep learning*) que podem gerar novos dados em vários campos, incluindo imagens e sons (GOODFELLOW et al., 2020). Esses modelos usam duas redes neurais que competem entre si. A rede geradora produz dados sintéticos, enquanto a rede discriminadora avalia a autenticidade do dado, ou seja, determina a probabilidade do novo dado sintético fazer parte do conjunto de dados utilizados no treinamento da rede geradora. Em aplicações de imagem, após o treinamento, o gerador pode produzir imagens tão realistas que o discriminador as confunde com imagens reais.



Para o contexto de tráfego, (XU; SOULY; BRAHMA, 2021) propõe o uso de RAGs para geração de dados realistas e aumento de casos críticos, com transformações de dados diurnos para noturnos. (SUN et al., 2021) criou um *framework* para cenários críticos baseado no processo de decisão de Markov e técnicas de aprendizado por reforço (*Deep Reinforcement Learning* - DeepRL). (AMINI et al., 2022) apontada lacunas de implementação de controles em ambiente físico causadas por inconsistência da distribuição de parâmetros e modelagem incorreta de simuladores, conhecidos como *Sim2Real gap*, com solução proposta de tornar treinamentos de VAs mais robustos utilizando RAGs com DeepRL.

(CHEN et al., 2021) foca na produção de cenários de mudança de faixa. A técnica de geração é do tipo adversarial, com os cenários gerados pro aprendizado por reforço e agrupamento Bayesiano não paramétrico.

A maioria dos estudos de geração baseada em conhecimento opta por empregar heurísticas evolutivas, com destaque para o uso predominante do Algoritmo Genético (AG). (LI et al., 2020) propõe o *framework* de teste - AV-FUZZER - para identificação de violações de segurança em tráfego de veículos. A técnica é baseada em duas principais etapas, sendo a primeira, a busca de cenários mais arriscados via AG. O AG realiza perturbações de parâmetros que descrevem manobras dos NPCs (veículos do cenário, exceto o VA de controle). Aqueles cenários de maior risco são explorados com uma técnica de difusão local, pois acredita-se que possa existir violações de segurança ao redor de casos de quase acidentes. Essa solução tem a particularidade de gerar casos aleatórios, ou seja, o *framework* não é capaz de gerar casos críticos a partir de descrições formais de manobras de condução. (KLÜCK et al., 2019a) realiza um estudo de comparação da adoção de heurística aleatória e AG para geração de casos críticos, que mostra que a produção de simulações via AG é mais efetiva. Entretanto, apesar de que é quase certo que o AG chegue em alguma simulação de caso crítico, a seleção aleatória é menos custosa. Os autores discutem que esse resultado se dá possivelmente pela falta de complexidade dos cenários testados - cenários de ultrapassagem em rodovias. Vale notar que o estudo utiliza o tempo para colisão (TPC) entre veículos como função objetivo - funcional para o caso em que dois veículos estão alinhados. A restrição de utilização de TPC é possivelmente insuficiente para situações mais genéricas de tráfego. (KLÜCK et al., 2019b) é uma extensão do estudo anterior, apenas comparando a técnicas de otimização AG e têmpera simulada, Novamente, o AG se mostrou mais efetivo para a seleção de parâmetros que resultam em colisões, no entanto os problemas apontados pelo estudo antecedente não foram abordados.

(TIAN et al., 2022) utiliza AG para modelar o comportamento de manobra dos veículos. Cada cromossomo, que define o comportamento dos veículos, relaciona uma ação (limitada em acelerar, frear, ou fazer a mudança de faixas) em um intervalo de tempo fixo. A sequência de ações nesses intervalos define a execução de uma simulação. Apesar

desse trabalho gerar diversos experimentos de violação de tráfego, ele também se limita apenas na geração de casos críticos de veículos trafegando no mesmo sentido. (ZHONG; KAISER; RAY, 2022) apresenta uma técnica de geração de casos baseado na busca de simulações utilizando uma gramática de descrição de cenários. A busca de parâmetros também é feita por AG, mas a grande diferença de outros trabalhos é que a operação de mutação é direcionada pelo acúmulo de conhecimento de casos de simulações anteriores. A cada nova geração uma rede neural é treinada para classificar os conjuntos de parâmetros que têm a maior probabilidade de ocorrer uma violação de tráfego.

## 3.2 Algoritmos Genéticos

O AG é um tipo de método de busca e otimização que é inspirado no domínio da evolução biológica (DARWIN, 2016), criado por John Holland na década de 1960 (HOLLAND, 1992a). Essa técnica replica o processo de seleção natural, priorizando os indivíduos com maior adaptação ao seu meio para a sobrevivência e reprodução (HOLLAND, 1992b). No contexto da computação, primeiramente é criada uma população de indivíduos (cromossomos, na biologia) de maneira aleatória. A forma mais comum de representação do indivíduo é em binário, sendo que, no caso mais simples, cada *bit* pode ser entendido como a representação de uma solução candidata para um problema de interesse (FORREST, 1996). No contexto geral, o indivíduo é uma solução de um problema específico que o AG tenta otimizar. A representação desses indivíduos é arbitrária e pode ser feita de acordo com o tipo de aplicação, (ROTHLAUF; ROTHLAUF, 2006), conforme o exemplo da Figura 7.

A partir da inicialização, o mecanismo de construção de novas gerações conta com quatro passos principais (KUMAR et al., 2010). O primeiro deles é a seleção, em que alguns indivíduos da seleção anterior são selecionados para compor a nova geração - uma forma de armazenar a informação genética (soluções) de gerações anteriores. Caso os indivíduos selecionados sejam os de melhor aptidão, essa estratégia é nomeada como elitismo (DU et al., 2018), sendo uma forma de mitigar a deriva genética, assegurando que os indivíduos mais aptos possam transmitir suas características para as gerações subsequentes (DUMITRESCU et al., 2000). O segundo passo é o cruzamento, em que ocorre uma mistura genética entre indivíduos de uma geração anterior para a produção de novos indivíduos da geração posterior, como mostrado na Figura 8. A terceira etapa é a mutação em que um dado indivíduo sofre uma mutação em um trecho de seu cromossomo, resultando em uma nova solução para teste, tal como ilustrado na Figura 9.

Por fim, o último passo é a operação de avaliação de cada indivíduo, por meio de uma função objetivo ou função de *fitness*. A função objetivo, que é a função a ser otimizada, serve como o instrumento para avaliar cada indivíduo. A amplitude de valores

<b>Binário</b>	1 0 0 1 0 1 1 1 1 1 0 0 1 0
<b>Inteiro</b>	7 1 3 10 5 6 9 9 0 2 7 3 1
<b>Real</b>	1 0.56 3.1 5.95 2.02 3 9.4

Figura 7 – Algumas formas de representação dos indivíduos



Figura 8 – Cruzamento entre indivíduos binários



Figura 9 – Mutaç o de indiv duos bin rio

da funç o objetivo varia entre diferentes problemas. Para garantir uniformidade em diversos contextos, utiliza-se a funç o normalizada numa faixa de 0 a 1. O valor normalizado da funç o objetivo reflete a aptid o do indiv duo, sendo o crit rio adotado pelo mecanismo de avaliaç o da geraç o (SRINIVAS; PATNAIK, 1994). Esse processo de criaç o de novas geraç es ocorre enquanto o crit rio de parada n o for atingido, sendo esse tamb m arbitr rio. Como apontado por (KUMAR et al., 2010), os crit rios de parada mais comuns s o a descoberta de uma soluç o que atinge um crit rio m nimo, um n mero fixo de geraç es, compromisso de alocaç o de recursos (custo/tempo), variaç o de aptid o entre geraç es, inspeç o manual, ou um conjunto das condiç es anteriores. A sequ ncia de passos do algoritmo   expressa pelo Algoritmo 1.

Do ponto de vista de aplicaç es, essa t cnica pode ser empregada em qualquer problema de busca ou otimizaç o, dependendo exclusivamente de uma modelagem de indiv duos que se ajustem ao problema em quest o.

---

**Algorithm 1**  $AG(N_G, T_P, \mu)$ 

---

```

1: // Número de Gerações:  $N_G$ , Tamanho População:  $T_P$ , Probabilidades
   (Seleção, Cruzamento, Mutação):  $\mu$  ( $\mu_s, \mu_c, \mu_m$ )
Require:  $N_G > 1; T_P > 1; \mu$ 
2: // Inicialização
3:  $n \leftarrow 0$ 
4:  $P_k \leftarrow Gera\_Populacao(T_P)$  // Geração população inicial
5:
6:  $Avalia\_Populacao(P_k)$  // Função avaliação
7:  $Ranking(P_k)$  // Ordenamento baseado na função objetivo
8: for  $n \leftarrow 1$  to  $N_G$  do
9:
10: // Criação da geração  $k + 1$ 
11: if  $n > 1$  then
12:    $P_k \leftarrow P_{k+1}$ 
13: end if
14:
15:  $a \leftarrow Seleção(P_k, \mu_s)$  // Cópia indivíduos mais adaptados geração anterior
16:  $b \leftarrow Cruzamento(P_k, \mu_c)$  // Cruzamento indivíduos geração anterior
17:  $c \leftarrow Mutação(P_k, \mu_m)$  // Mutação de indivíduos
18:
19:  $P_{k+1} \leftarrow Merge(a, b, c)$  // Fusão das 3 listas
20:
21:  $Avalia\_Populacao(P_k)$ 
22:  $Ranking(P_k)$ 
23: end for

```

---

## 4 Método do trabalho

Este trabalho busca atingir o objetivo proposto e responder a pergunta de pesquisa por meio da construção de *framework* que seja capaz de gerar de forma automatizada *datasets*, os quais apresentem uma maior incidência de *corner cases*, auxiliando em futuras pesquisas de VAs. Para a desenvolvimento e implementação do projeto, inicialmente foi elaborada uma especificação alto nível, apresentado na Seção 4.1. Na Seção 4.2 são apresentadas as tecnologias utilizadas, com as respectivas justificativas. Em seguida, a implementação do projeto é apresentada na Seção 4.3.

### 4.1 Concepção do Projeto

Para que o trabalho fosse executado, realizou-se um levantamento de requisitos e algumas estratégias de planejamento para que o objetivo primário pudesse ser atingido. Primeiramente, todo o trabalho deveria ser construído em ambiente simulado. Assim, fez-se um levantamento dos principais simuladores utilizados em pesquisas (ROSIQUE et al., 2019; ELMQUIST; SERBAN; NEGRUT, 2021). Após uma filtragem inicial, alguns foram selecionados para realização de testes e o *Carla Simulator* foi escolhido. A justificativa se dará na próxima seção.

Com base na definição do simulador, os dois passos seguintes são a modificação de cenários e desenvolvimento de controles de veículos, incluindo a capacidade de armazenar as informações de simulação. Esses requisitos são cruciais, posto que a partir deles, tornam-se viáveis experimentos de exploração e implementação de técnicas ou heurísticas para geração de cenários de teste. Concomitantemente, todos os experimentos devem ser avaliados para direcionar os esforços do trabalho. A especificação desses objetivos secundários possibilitou a consecução do objetivo primário, sendo ilustrado no diagrama da Figura 10.

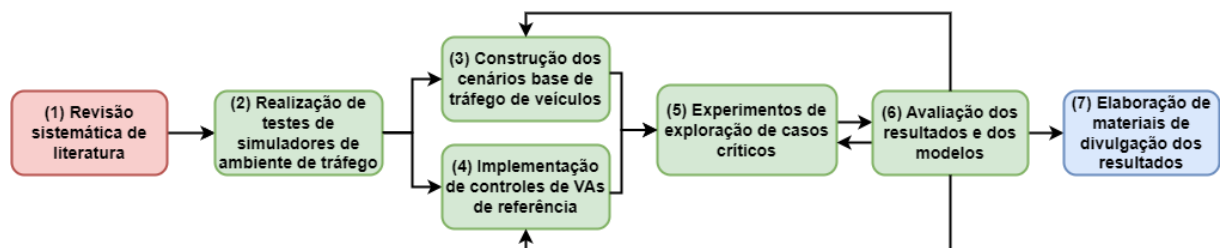


Figura 10 – Fluxograma para a implementação do projeto

Nas subseções descritas a seguir são explorados as etapas do procedimento metodológico desse projeto.

### 4.1.1 Mapeamento de Literatura

O propósito da revisão de literatura é realizar um mapeamento do estado atual de conhecimento sobre o tema, destacando as técnicas mais relevantes e suas possíveis limitações. Ao final da elaboração dessa análise introdutória, fora atingida uma cobertura considerada adequada, indicando os principais avanços e as lacunas existentes.

O mapeamento seguiu o procedimento de uma revisão sistemática de literatura:

1. Definição das perguntas de pesquisa;
2. Identificação de palavras-chave;
3. Pesquisa de trabalhos da área baseado nas palavras-chave;
4. Filtragem dos principais trabalhos;
5. Classificação de trabalhos relacionando com o tema de pesquisa.

A partir desse mapeamento, foi possível compreender a fronteira da área de pesquisa de geração de *corner cases* para veículos autônomos, dando início ao delineamento do tema de trabalho.

### 4.1.2 Testes de simuladores de ambiente de tráfego

Por conta de fatores de escalabilidade e custo, muitas pesquisas são realizadas em ambiente simulado. Para isso é necessário adotar um simulador amplamente utilizado e aceito pela indústria e comunidade acadêmica, favorecendo a comparação de estudos. Com base no mapeamento anterior, foram verificados os seguintes simuladores:

- *Carla Simulator*;
- Vista;
- Webots;
- Summit;
- LGSVL (descontinuado);
- Gazebo

Dentre todos, o Carla aparentou ser o mais indicado para a realização do projeto, porque:

1. é *open-source*, construído com a finalidade de pesquisa;
2. apresenta um alto grau de realismo;

3. Contém uma série de ferramentas que auxiliam em modificações tanto das simulações, quanto em implantação de controles de veículos e integração com diversos *softwares* relacionados.

### 4.1.3 Construção dos cenários de base e Implementação de controles de VAs

Em decorrência da escolha do simulador, foram realizados diversos testes e experimentos visando ter um domínio mais profundo sobre as possibilidades de implementação no simulador. Foram realizadas modificações nos cenários e implementações de controles de veículos utilizando desde descrições básicas, baseada na geometria dos mapas, até integrações com algumas bibliotecas que auxiliassem nesse processo.

A importância dessa etapa reside no fato de esclarecer os potenciais limites de desenvolvimento que poderiam ser atingidos durante o período de realização desse trabalho. O principal parâmetro para as tomadas de decisões mais críticas foi baseada no ponderamento de uma solução de compromisso que conseguisse balancear o nível de controle das implementações (ex.: controles baseados na geometria vs bibliotecas), e o grau de automação da solução. Nas próximas seções são justificadas as escolhas de projeto que foram definidas ao longo de seu desenvolvimento.

### 4.1.4 Experimentos de exploração de casos críticos

Os experimentos de exploração são o grande tópico desse projeto, do ponto de vista da geração de casos. Nessa etapa foram realizados experimentos utilizando como base algumas das métricas indicadas pelo mapeamento da literatura. Ao longo dos experimentos, o escopo do trabalho foi delineado convergindo para a adoção da técnica de AG para a otimização da busca de cenários de interesse.

### 4.1.5 Avaliação dos resultados

A avaliação e documentação foram ativamente realizadas durante o período de execução do trabalho, tanto para direcionar a evolução do trabalho, quanto facilitar a identificação de oportunidades de melhorias em todo o processo.

### 4.1.6 Elaboração de materiais de divulgação dos resultados

Durante a realização dos blocos experimentais, já foram sendo realizadas a elaboração dos materiais de divulgação, como esse texto de monografia. Ao final do processo, obteve-se também um *website*, um *press release* e um *banner* de apresentação.

## 4.2 Tecnologias Utilizadas

Nessa seção, exploram-se os aspectos tecnológicos empregados neste projeto, enfatizando as linguagens de programação e bibliotecas utilizadas.

### 4.2.1 Python

A linguagem *Python* (ROSSUM; DRAKE, 2009) é a tecnologia base para o desenvolvimento do projeto. A escolha foi fundamentada não apenas pela sua capacidade de integração com diversas bibliotecas, mas também pela integração com o simulador adotado para o desenvolvimento do projeto apresentado na subseção 4.2.2.

No contexto do simulador, o *Python* é utilizado como uma linguagem de *script* que facilita a criação e modificação de cenários, configurações e interações no simulador, permitindo implementar e modificar técnicas de inteligência artificial de controle de veículos de maneira eficiente. A versão utilizada no projeto é a 3.8.10, por conta da integração com as diversas bibliotecas dependentes.

### 4.2.2 Carla Simulator

O Carla Simulator (DOSOVITSKIY et al., 2017) é uma plataforma de simulação de código aberto projetada para pesquisa e desenvolvimento em veículos autônomos. Foi desenvolvido no Centro de Visão Computacional da Universitat Autònoma de Barcelona (UAB), permitindo a construção de cenários realistas de tráfego de veículos, bem como implementação de algoritmos de controle para VAs.

O Carla é constituído de uma arquitetura baseada em cliente-servidor. O servidor, considerado como o "*mundo*" da simulação, é responsável por todos os elementos que estão relacionados com a própria simulação em si, tais como computação dos modelos físicos, renderização dos sensores, atualização dos estados, entre outros (CARLA-TEAM, 2023b). O lado do cliente consiste em um conjunto de módulos de clientes responsáveis pelo controle lógico dos atores no cenário e pela definição das condições do mundo. Essa operação é realizada por meio da utilização da API CARLA, disponível em *Python* ou C++.

O simulador utiliza o motor gráfico *Unreal Engine 4* (Epic Games, 2019) para executar a simulação e usa o padrão OpenDrive 1.4 (AUTOMATION; SYSTEMS, 2023) para definir a estrutura das vias e ambientes urbanos. Por conta da arquitetura, é possível criar vários clientes concomitantes, no mesmo nó, ou em nós diferentes. A comunicação é feita por meio do protocolo TCP. A Figura 11 exemplifica a arquitetura do projeto.

O Carla permite a utilização de diversos tipos de sensores listados na Tabela 1:



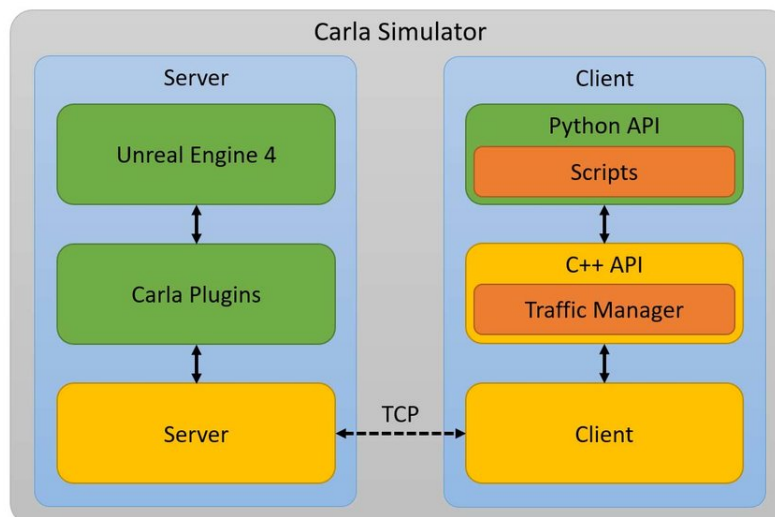


Figura 11 – Arquitetura do Carla - Figura adaptado de (PIRRI et al., 2021)

Sensores	Descrição / Tipo
Câmeras	RGB, profundidade, segmentação semântica, DVS, fluxo óptico
Radar	Deteção baseada no espectro de rádio
Lidar	Deteção baseada no espectro da luz
GNSS	Sensor de posicionamento
IMU	Sensor de medição inercial
Detector de Colisão	Indica eventos de colisão entre atores da simulação
Detector de invasão de faixa	Indica eventos em que veículos violam faixas
Lidar semântico	Lidar com segmentação semântica

Tabela 1 – Tabela dos principais sensores disponibilizados no Carla

Além disso, em diversas versões do Carla, já estão incluídas a implementação nativa de coleta de *logs* (registros binários) das simulações de tráfego, em que todos os principais elementos que compõem o cenário são descritos, como estado de cada ator (pedestres e veículos), estado dos semáforos, entre outros, ao longo de cada *frame* da simulação. Esses arquivos de registros são importantes por dois motivos. O primeiro é a coleta de métricas dos experimentos, em que é possível saber as principais informações de posição, velocidade e aceleração de cada veículo ao longo do tempo, como também a ocorrência de eventos especiais, como colisões. A segunda é que a partir desses registros é possível fazer repetições da simulação. Os registros servem como uma receita sequencial que descreve a dinâmica de cada elemento presente na simulação. Como vantagem adicional, durante essas repetições, é possível a integração dos sensores previamente mencionados, permitindo a coleta de dados suplementares. Devido a todos esses ganhos, este simulador tem sido extensivamente empregado em várias pesquisas acadêmicas e industriais, simplificando a condução de experimentos direcionados ao desenvolvimento de VAs, e, por conseguinte, eliminando a necessidade de lidar com a infraestrutura básica dos experimentos.

Nas Figuras 12 e 13 são mostradas algumas imagens de execuções do Carla Simulator.

Na primeira imagem é mostrado o carro de controle com as suas principais informações à esquerda. Na segunda, são expostas algumas simulações com diversas condições climáticas. Por fim, na Figura 14, é mostrado um exemplo dos tipos de dados gráficos gerados pelos sensores.



Figura 12 – Vista de simulação padrão em ambiente urbano



Figura 13 – Simulações do Carla Simulator em diferentes condições climáticas

#### 4.2.3 Scenic

O Scenic é uma linguagem de programação probabilística de domínio específico desenvolvida para modelar ambientes de sistemas ciberfísicos, como robôs e VAs (FREMONT et al., 2023). A lista de simuladores suportados é o Carla Simulator, GTA V (jogo de

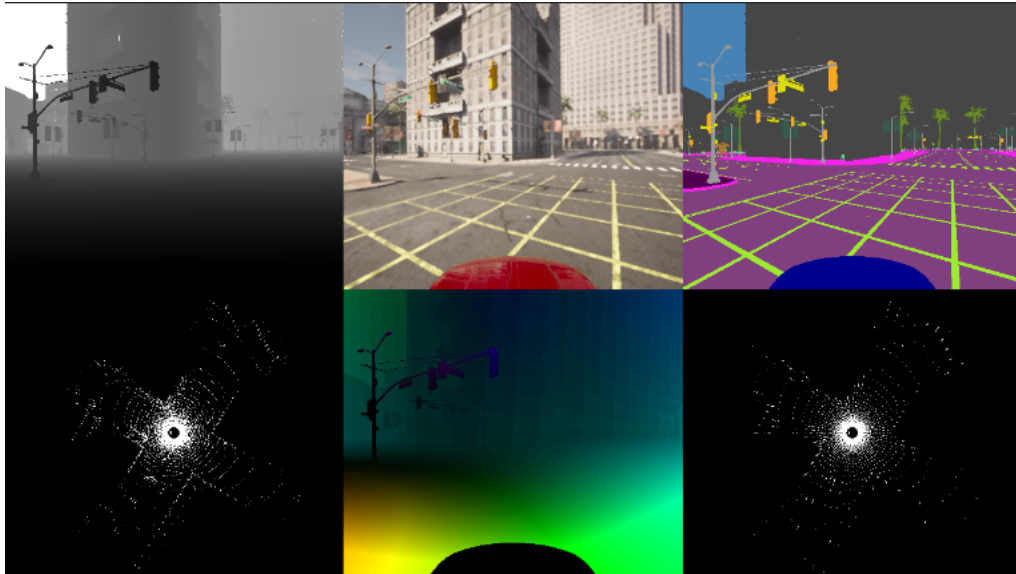


Figura 14 – Visualização de diferentes fluxos de dados gerados pelo simulador (Profundo, RGB, Segmentação Semântica, LiDAR, Optical Flow, LiDAR Semântico)

mundo aberto da *RockStar Games*), LGSVL (simulador de tráfego da LG - descontinuado), Webots (simulador de robótica) e X-Plane (simulador de voo). Conforme a documentação oficial ([SCENIC-TEAM, b](#)), um programa Scenic define uma distribuição entre cenas, configurações de objetos físicos e agentes. A amostragem dessa distribuição produz cenas concretas que podem ser simuladas para produzir dados de treinamento ou teste. O Scenic também pode definir políticas para agentes dinâmicos, permitindo modelar cenários em que os agentes realizam ações ao longo do tempo em resposta ao estado da simulação. Nesse sentido, a vantagem do Scenic é a possibilidade de descrição de cenários simulados com uma gramática quase textual, acelerando o desenvolvimento de simulações.

#### 4.2.4 Bash Script

O *Bash Script* refere-se tanto a uma linguagem de *script* quanto a um interpretador de comando associado, sendo um acrônimo para "*Bourne Again SHell*", ([GNU, 2023](#)). Essa linguagem é amplamente utilizada em sistemas operacionais Unix e Linux, incluindo distribuições como Ubuntu e Debian. o *Bash Script* tem a capacidade de automatizar tarefas e interagir com o sistema operacional por meio de linhas de comando, oferecendo uma ampla variedade de funcionalidades, tais como a execução de comandos do sistema, manipulação de arquivos, tomada de decisões lógicas, entre outros.

Essa linguagem é comumente utilizadas para a criação de *scripts* de automação para simplificar a execução de sequências de comandos de forma eficiente. Como o trabalho foi desenvolvido em ambiente Linux, o *Bash Script* foi utilizado para integração de todas as funções e automação da execução de simulações.

## 4.3 A implementação

Nessa seção, são abordados os principais conceitos da implementação realizada.

### 4.3.1 Versões de Software

O sistema operacional utilizado foi o Ubuntu 22.04, mas esse projeto também se mostrou compatível com versões mais antigas, como o 20.04, sendo o simulador a principal dependência do projeto. A versão do Carla Simulator escolhida é a 0.9.13, disponível em (CARLA-TEAM, 2021), em função da compatibilidade e estabilidade com os demais módulos dependentes. A versão do *Python* é a 3.8.10, também compatível com todas as bibliotecas necessárias. A versão do Scenic é a 2.1.0.

Por fim, vale notar que a escolha do sistema operacional se deu pela facilidade de integração dos módulos, bem como a execução sequencial de cada etapa do projeto. Testes realizados no sistema operacional *Windows* indicaram que o projeto seria menos eficiente, e, portanto, foi descartado como alternativa viável.

### 4.3.2 Principais elementos

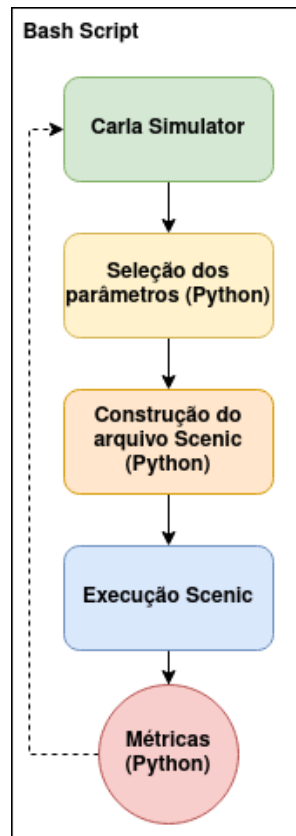
Como mencionado na subseção 4.2.2, o Carla funciona com uma arquitetura cliente-servidor. Assim, o funcionamento para qualquer experimento requer que pelo menos duas chamadas de função sejam executadas, nessa sequência, sendo primeiro a chamada do próprio simulador e, em seguida, algum *script Python* de controle do veículo. Vale notar que o simulador deve estar sempre ativo, portanto, a ordem das execuções é relevante, sendo a chamada da simulação prioritária. A chamada padrão do simulador é feita pela linha de comando.

Por conta da estrutura do Carla, todos os elementos que tem alguma interface com o Carla foram construídos em *scripts Python*. A orquestração de todos os passos foi realizada em *bash script*, devido a simplicidade de funcionamento. A sequência de passos genéricos da implementação é mostrada no diagrama da Figura 15.

Nas próximas seções são explicadas cada um dos elementos da estrutura genérica.

### 4.3.3 Carla Simulator

Nessa subseção são detalhados os principais elementos sobre a execução do Carla.

Figura 15 – Estrutura genérica da implementação do *Framework*

#### 4.3.3.1 Execução

A chamada padrão do simulador é feita a partir da raiz do Carla com o comando 1.

```
$ ./CarlaUe4.sh
```

#### Código 1 – Chamada padrão do Carla

A versão 0.9.13 do Carla contém 10 mapas padrões disponibilizados pela equipe do simulador. A inicialização padrão adota o mapa *Town01*, como mostrado na Figura 16.

A principal forma para mudar o mapa do servidor é através de um *script* de configuração. Para isso, é necessário abrir uma nova janela do terminal e ir para o diretório:

```
/PythonAPI/util
```

Nesse diretório, basta executar o comando:

```
$ python3 config.py --map TownXX
```



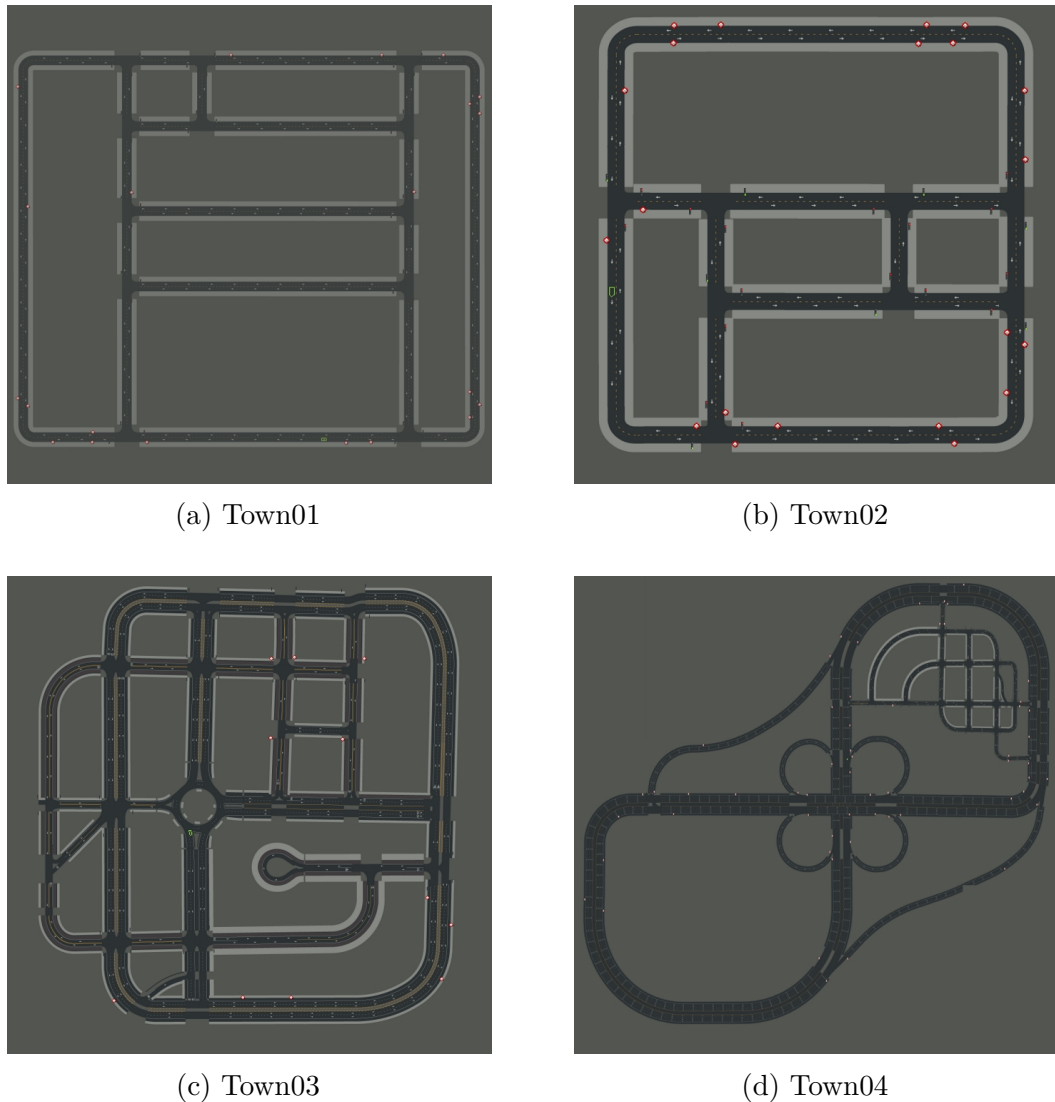


Figura 16 – Vista superior (2D) dos mapas padrão - adaptado de (CARLA-TEAM, 2023a)

em que  $XX$  é o número do mapa em questão.

#### 4.3.3.2 Gravador

No Carla é implementado uma classe de gravação de dados que descrevem a dinâmica da simulação por meio de registros (*logs*) em formato binário (CARLA-TEAM, 2021). Esses registros tem a capacidade de coletar informações sobre Atores, Semáforos, Veículos e Pedestres, permitindo que, posteriormente, a simulação possa ser reproduzidas novamente. Uma informação relevante como subproduto desse trabalho é que a partir desses arquivos de registros, durante a reprodução é possível fazer a inclusão dos sensores descritos na Tabela 1, possibilitando a coleta de dados desses sensores. Dado que o objetivo principal desse projeto é conseguir automatizar o processo de geração de cenários mais arriscados, a subsequente coleta de dados de sensores poderá sustentar pesquisas futuras que utilizem esse tipo de dado.

### 4.3.3.3 Formato dos registros

O sistema de gravação do simulador utiliza um padrão de arquivos binários ilustrados na Figura 17.

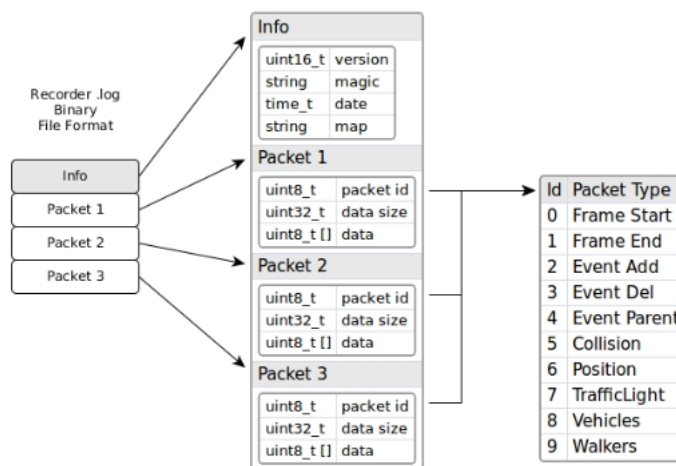


Figura 17 – Imagem completa em (CARLA-TEAM, 2021)

A estrutura de dados é concebida para fazer o registro sequencial dos dados em pacotes, sendo que cada pacote armazena os principais atributos relacionado ao estado de todos os atores. Para veículos, por exemplo, coleta-se dados de posição, velocidade instantânea, acelerações e a posição dos elementos de controle (pressão sobre pedais, ângulo do volante, entre outros).

Para processar os arquivos de registro, foi desenvolvido uma estrutura de processamento que organiza os dados de telemetria dos veículos, permitindo a criação de métricas de condução específicas para cada cenário de simulação.

### 4.3.4 Scenic

Nessa subsecção são discutidos alguns aspectos relevantes sobre o funcionamento do Scenic e a forma que foi integrado no projeto.

#### 4.3.4.1 Descrição

Conforme mencionado na subsecção 4.2.3, o Scenic é uma linguagem de programação probabilística, criado para especificar distribuições que caracterizam entradas diversas e complexas, para sistemas baseados em aprendizado de máquina. A sua principal aplicação se dá no contexto de simulação robótica. A linguagem se destaca por permitir que cenários sejam definidos com diversas entidades físicas e agentes autônomos de maneira clara e concisa. No contexto de tráfego de veículos, a depender do nível de configuração do simulador, podem ser definidos objetos e o comportamento dos veículos dentro de um

experimento de simulação. Ademais, o principal atributo desse projeto é a legibilidade da sintaxe durante a descrição de relacionamentos espaciais e temporais dentro de cada cenário, incluindo a definição de restrições que podem ser rígidas ou flexíveis, (FREMONT et al., 2023). Para o contexto do presente trabalho, a utilização do Scenic se deu pela facilidade de geração eficiente de cenários de simulação. A partir da descrição formal de um cenário, a biblioteca do Scenic gera, se viável, a simulação correspondente dentro do Carla.

A versão 2.1.0 do Scenic permite fazer a configuração de cenários no Carla utilizando veículos, motocicletas, bicicletas, caminhões, pedestres, latas ou cestos de lixo, cones, obstáculos, escombros, máquina de vendas automática, cadeira, ponto de ônibus, placas, mesas, barreiras de tráfego, plantas, caixa de correspondência, decoração de jardim, papelões, bancos, malas, guaritas, caixa eletrônico e placas de sinalização de tráfego.

#### 4.3.4.2 Funcionamento

Nessa subseção é explicada a sintaxe de um *script* Scenic a partir de um exemplo de descrição de um experimento adaptado do repositório oficial do projeto (SCENIC-TEAM, a).

O Código 2, representa o início da descrição em que são definidos tanto o mapa, quanto o tipo de simulador utilizado.

```
1 param map = localPath('caminho_mapa')
   ↪ # definição do endereço do arquivo do mapa
2 param carla_map = 'nome_mapa'
3 model scenic.simulators.carla.model # definição do simulador
```

#### Código 2 – Script Scenic para Carla - Descrição do mapa e dos modelos

No código 3, são definidas as principais constantes do cenário descrito. Nesse caso, é utilizado apenas um único veículo (o modelo MKZ de 2017 na marca Lincoln), bem como a sua velocidade máxima.

```
1 EGO_MODEL = "vehicle.lincoln.mkz2017"
2 EGO_SPEED = 10
```

#### Código 3 – Script Scenic para Carla - Descrição das constantes do experimento

A seguir, no Código 4 é descrito o comportamento do veículo de controle, que no caso é para seguir na faixa e frear após ultrapassar uma distância limite até o veículo da



frente.

```

1   behavior EgoBehavior(speed=10):
2       do FollowLaneBehavior(speed)

```

Código 4 – *Script Scenic* para Carla - Descrição do comportamento do veículo de controle

```

1   lane = Uniform(*network.lanes)
2
3   start = OrientedPoint on lane.centerline
4   ego = Car at start,
5       with blueprint EGO_MODEL,
6       with behavior EgoBehavior(EGO_SPEED)
7
8   debris1 = Debris following roadDirection for Range(10, 20)
9   ↪ # definição do objeto 1
10  debris2 = Debris following roadDirection from debris1 for Range(5,
11  ↪ 10) # definição do objeto 2
12  debris3 = Debris following roadDirection from debris2 for Range(5,
13  ↪ 10) # definição do objeto 3
14
15  require (distance to intersection) > 50 # condição inicial
16  terminate when (distance from debris3 to ego) > 10 and (distance to
17  ↪ start) > 50 # condição de parada

```

Código 5 – *Script Scenic* para Carla - Descrição das relações espaciais do experimento

Na sequência, o Código 5 define todas as relações espaciais da simulação, inclusive a condição de parada do experimento. Inicialmente são listadas todas as faixas existentes do mapa. Em uma das faixas, o veículo é inicializado no centro da faixa. Ao longo da via são colocados alguns objetos dentro de alguns intervalos. Por fim, a condição de parada é que a distância do veículo para o objeto 3 seja superior a 10 metros e que o veículo esteja pelo menos a 50 metros afastados do seu ponto de partida.

Com esses trechos de código, torna-se viável criar uma descrição formal de um cenário de testes, dispensando a necessidade de uma explicação geométrica dos problemas. O foco está exclusivamente na definição das restrições de alto nível do experimento em questão. Vale notar também que trechos apresentados no Código 4 `EgoBehavior()` e `FollowLaneBehavior()` são objetos definidos pela linguagem que podem ser encontrados no diretório `/Scenic/src/scenic/domains/driving/` da biblioteca. Finalmente, por

simplicidade, essa linguagem foi desenvolvida com uma sintaxe que se assemelha à do *Python*.

#### 4.3.4.3 Execução Genérica

Para executar o Scenic de maneira genérica, é necessário abrir dois terminais. No primeiro, entre na raiz do Carla e execute o Comando 1 para fazer a chamada do simulador. Em seguida, no segundo terminal, acesse o diretório que contém o script Scenic de interesse e execute o Comando 6. Caso não haja a definição do parâmetro `count`, são gerados inúmeras simulações seguindo a descrição do arquivo - nesse caso, a condição de parada passa a ser a interrupção no terminal.

```
$ scenic <nome_script>.scenic --simulate --count N
```

Código 6 – Chamada de execução de um *script* Scenic - N execuções

O acesso ao simulador se dá pelo endereço IP (127.0.0.1) e porta 2000, definidos como padrão. Do lado do Scenic, a implementação também utiliza o mesmo endereço e porta. No entanto, caso seja necessário, é possível passar como parâmetros os novos endereço e porta do simulador. Além disso, também é possível passar os seguintes elementos:

- `carla_map`: Escolha do mapa
- `timestep`: Intervalo de tempo das simulações
- `weather`: Condição climática do simulador
- `render`: Booleano que define a criação de janela mostrando o ponto de vista do veículo principal
- `record`: Se não vazio, é a variável de endereço que será salvo o log de cada simulação

Todos esses parâmetros podem ser definidos tanto dentro do *script*, quanto durante a chamada da biblioteca Scenic.

#### 4.3.4.4 Modificações da biblioteca para coleta de dados pelo Scenic

Como discutido nas subseções 4.3.3.2 e 4.3.4.3, tanto o Carla, quanto o Scenic permitem que os registros de cada simulação sejam salvos em arquivos binários. Entretanto, a implementação padrão do Scenic não habilita a coleta completa de registros, omitindo, por exemplo, informações de colisão durante as simulações. Tendo em vista essa limitação, foram necessárias realizar modificações na biblioteca padrão do Scenic 2.1.0 para que todas as informações de interesse fossem coletadas durante a execução dos cenários gerados pelo Scenic.

As modificações necessárias foram feitas no arquivo de diretório `/src/scenic-simulators/carla/simulator.py` do projeto do Scenic 2.1.0, disponível em ([SCENIC-TEAM, a](#)). A primeira modificação necessária é incluir na linha do Código 7, o parâmetro `True`, resultando no Código 8

```
self.client.start_recorder(name)
```

Código 7 – Carla `start_record()` - original

```
self.client.start_recorder(name, True)
```

Código 8 – Carla `start_record()` - modificado

A variável `True` está relacionada com o parâmetro `additional_data`, que habilita a gravação de todos os elementos presentes na simulação. Caso contrário, o gravador apenas armazena apenas as informações necessárias para a sua reprodução - excluindo dados de telemetria utilizados na geração de métricas de condução. Para que o gravador colete informações de colisão durante a simulação, é necessário fazer a inclusão do sensor de colisão nos veículos de interesse. Para isso, é necessário criar um sensor e associar a cada veículo do cenário de teste. No Código 12 (Apêndice) é mostrada a implementação da classe dos sensores.

No método `createObjectInSimulator(self, obj)` da classe `CarlaSimulation(DrivingSimulation)`, foi necessário incluir o trecho `self.collision_sensor_list.append(CollisionSensor(carlaActor))`, como no Código 9, grifado em verde. Também foi necessária a modificação, em amarelo, do tipo de transmissão, passando de manual para automática. Isso permite que seja possível atingir velocidades acima de  $15\text{km/h}$ .

```
1 if isinstance(carlaActor, carla.Vehicle):
2     obj.width = carlaActor.bounding_box.extent.y * 2
3     obj.length = carlaActor.bounding_box.extent.x * 2
4     carlaActor.apply_control(carla.VehicleControl(manual_gear_shift=
5         ↪ False))
6     self.CSL.append(SensorColisao(carlaActor))
```

Código 9 – Criação e atribuição de sensores de colisão a cada veículo de teste

É importante salientar que todos os elementos e atores gerados na simulação devem

ser armazenados em listas, de modo a desativá-los e destruí-los ao término da execução do teste. A necessidade desse procedimento decorre do fato de que uma automação que deixa resíduos de atores no simulador, além de potenciais interferências ao longo de novas simulações, pode resultar em perda de desempenho ao longo do tempo. Dado que esse processo já é implementado para veículos e a câmera de interface na versão original da biblioteca, é indispensável aplicar o mesmo procedimento simultaneamente a todos os sensores de colisão incorporados nos veículos, conforme exemplificado no Código 10 do método `destroy()`.

```
1 def destroy(self):
2     ...
3     if self.CSL is not None:
4         for sensor in self.CSL:
5             if isinstance(sensor.sensor, carla.Sensor):
6                 sensor.sensor.stop()
7                 sensor.sensor.destroy()
8
```

Código 10 – Inclusão do trecho de deleção dos sensores

#### 4.3.4.5 Instalação da biblioteca modificada

Para que todas as instalações sejam válidas, faz-se necessário a reinstalação da biblioteca, conforme apresentado em (SCENIC-TEAM, b). O projeto foi realizado utilizando *Conda Environments*, que são ambientes isolados e independentes suportados pelo gerenciador de pacotes Conda. Assim, para realizar a instalação, basta clonar o repositório modificado (SHIMANUKI, 2023), seguindo os comandos a seguir

```
1 $ conda activate <nome_environment> # ativar \textit{environment}
2 $ git clone https://github.com/GabrielShimanuki/scenic-2.1.0-mod
3 $ cd Scenic
4 $ python -m pip install -e .
```

Portanto, a partir de todos esse procedimentos é possível fazer com que os cenários de simulação gerados pelo Scenic armazenem dados de telemetria úteis para avaliação de risco de condução.

### 4.3.5 Geração dos Parâmetros

Por conta da característica de geração de cenários do Scenic, que é marcado pela utilização de parâmetros, é possível realizar uma integração de duas etapas, no qual a primeira faz a geração de arquivos de descrição Scenic baseado em um vetor de parâmetros e, posteriormente, executa-se a simulação, podendo inferir o nível de risco do experimento. Nas subseções seguintes são discutidas tanto a integração para automação de geração de cenários, quanto a avaliação de risco adotada.

#### 4.3.5.1 Cenários de risco

Com base no relatório de estatísticas de cenários de pré-colisão de veículos leves dos dados de acidentes dos Estados Unidos entre 2011 a 2015 (SWANSON et al., 2019), do *National Highway Traffic Safety Administration* (NHTSA) foram escolhidos os principais cenários com alto grau de incidência de acidentes. Vale notar que o NHTSA é um dos órgãos que ajuda a criar regulações de segurança de VAs nos Estados Unidos (NHTSA, ; ADMINISTRATION et al., 2017).

Foram escolhidos seis cenários de cruzamento que envolvem apenas dois veículos em movimento. Na Tabela 2 são listados os cenários do estudo.

Cenário	Número de Vias	Descrição
A	2 x 2	Veículo A efetua <b>cruzamento</b> Veículo B (na mesma via, sentido oposto) efetua <b>conversão à esquerda</b>
B	2 x 2	Veículo A efetua <b>conversão à esquerda</b> Veículo B (na via perpendicular) efetua <b>cruzamento</b>
C	2 x 2	Veículo A efetua <b>cruzamento</b> ou <b>conversão à esquerda</b> Veículo B (na via perpendicular) efetua <b>conversão à esquerda</b>
D	2 x 2	Veículo A efetua <b>conversão à direita</b> Veículo B (na mesma via, sentido oposto) efetua <b>conversão à esquerda</b>
E	2 x 2	Veículo A efetua <b>conversão à direita</b> Veículo B (na via perpendicular) efetua <b>cruzamento</b> no mesmo sentido de A
F	3	Veículo A efetua <b>conversão à esquerda</b> Veículo B (na via perpendicular) efetua <b>cruzamento</b> no mesmo sentido de A

Tabela 2 – Descrição dos cenários utilizados no estudo de caso. Veículo A: controle, Veículo B: competidor. Vias 2x2 representam vias de mão dupla com duas faixas de rodagem em cada sentido

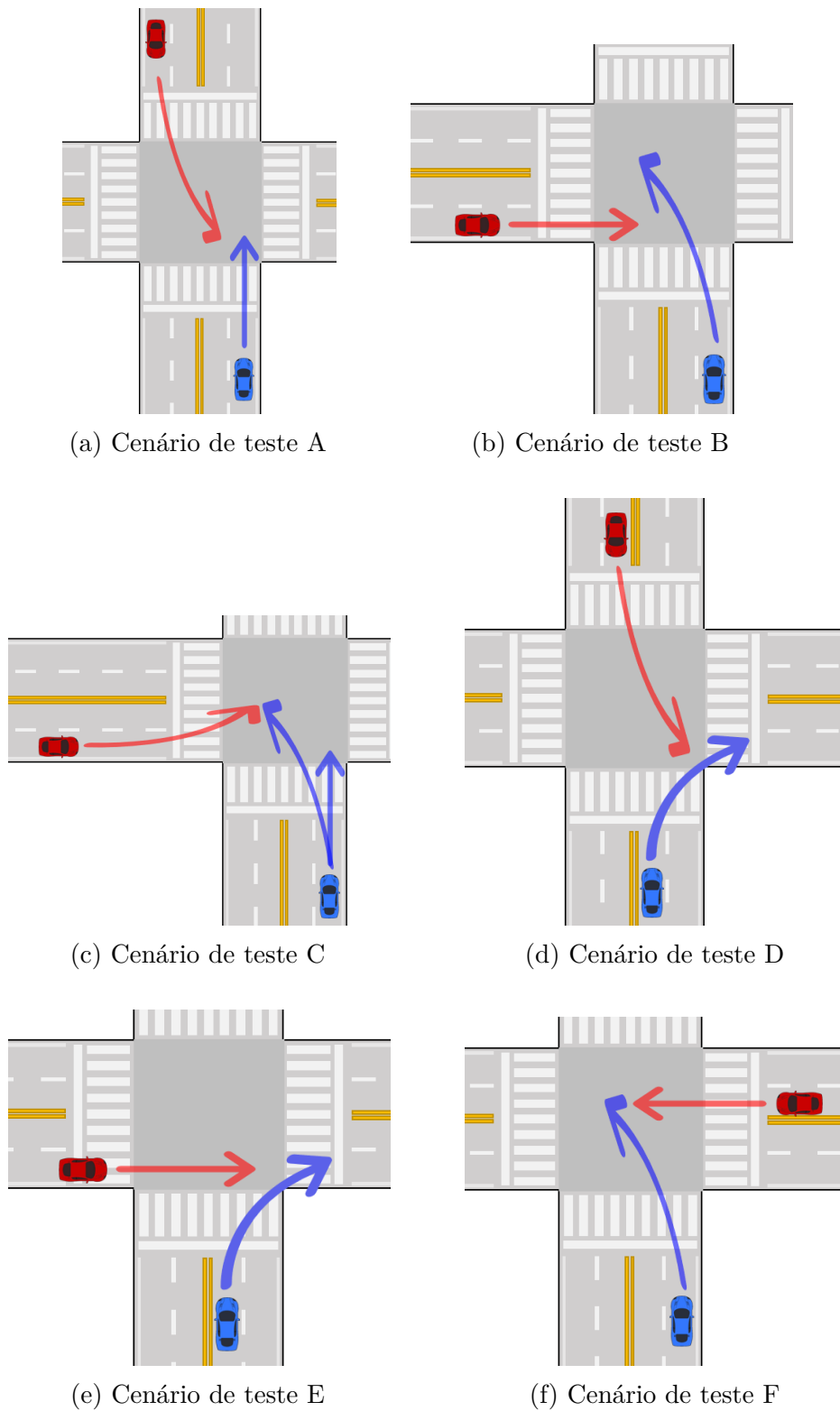


Figura 18 – Conjuntos de cenários de teste utilizados no estudo de caso de acordo com a Tabela 2

#### 4.3.5.2 Descrição e geração dos cenários

A seguir, no Código 11 recortado são mostrados os parâmetros definidos que podem definir um estudo do cenário A. O Código 13 encontra-se no apêndice.

```

1 #####
2 # CONSTANTES #
3 #####
4
5 MODEL = 'vehicle.lincoln.mkz_2017'
6
7 EGO_INIT_DIST = [10, 13] # intervalo da posição inicial do VA
8 param EGO_SPEED = VerifaiRange(60, 80)
9   ↪ # intervalo da velocidade máxima do VA
10 param EGO_BRAKE = VerifaiRange(0.5098506134136033, 0.5298506134136033)
11   ↪ # taxa de frenagem do VA
12
13 MODEL_ADV = 'vehicle.mercedes.coupe_2020'
14
15 ADV_INIT_DIST = [12, 15]
16   ↪ # intervalo da posição inicial do veículo adversário
17 param ADV_SPEED = VerifaiRange(18, 19)
18   ↪ # intervalo da velocidade máxima do veículo adversário
19
20 param SAFETY_DIST = VerifaiRange(9, 10) # distância considerada segura

```

Código 11 – Parâmetros utilizados para definição de teste no cenário A

Esses parâmetros são os únicos elementos que variam nesse *script*, viabilizando criar uma lógica de construção desses arquivos. Nessa lógica, apenas duas variáveis macro são relevantes, sendo a primeira o cenário de interesse, e a segunda o vetor de que armazena todos os parâmetros, que nesse caso são:

```
parametros = (10, 13, 60, 80, 0.5098506134136033, 0.5298506134136033, 12, 15, 18, 19, 9, 10)
```

Pela similaridade da descrição dos cenários, todos os 6 escolhidos para o estudo de caso utilizam o mesmo conjunto de parâmetros, sendo possível reaproveitar as mesmas funções de construção dos *scripts*. O *script* composto por seis blocos, Mapa e Modelo, Constantes, Comportamento dos Agentes, Relações Espaciais e Especificação do Cenário, apenas o bloco de descrição das relações espaciais varia conforme o tipo de cenário utilizado. A lógica foi implementada usando linguagem *Python* e o objetivo do *script* é gerar arquivos de descrição de cenários no padrão do Scenic, conforme o Pseudocódigo 2.

Nesse ponto, a automação da geração de cenários de testes já é viabilizada pela integração entre o gerador de parâmetros e a biblioteca modificada do Scenic, conforme detalhado na subseção 4.3.4.4.

**Algorithm 2** Gerador de *scripts* Scenic**Ensure:** Parâmetros (tipo\_cenário, vetor\_parâmetros, endereço\_destino)

- 1: escreve\_comentários()
- 2: escreve\_mapa\_modelo()
- 3: escreve\_constantes(vetor\_parâmetros)
- 4: escreve\_comportamento\_agentes()
- 5: escreve\_relações\_espaciais(tipo\_cenário)
- 6: escreve\_especificação\_cenário()
- 7: salva\_arquivo()

### 4.3.6 Algoritmo Genético

De acordo com a revisão 3.2, a utilização de AGs permite realizar uma forma de otimização inspirado no processo de seleção natural, tal como ocorre na evolução biológica. No contexto desse trabalho, o objetivo da implementação de um AG é para conseguir otimizar a buscas de parâmetros que são utilizados em *scripts* Scenic em descrições de cenários mais arriscados, envolvendo a situação de acidentes ou quase-acidentes. Os tópicos a seguir detalharão a forma como essa técnica foi implementada no projeto.

#### 4.3.6.1 Codificação do indivíduo

Os *scripts* Scenic, no geral, utilizam valores de parâmetros numéricos como ilustrado no exemplo do Código 13. A Tabela 3 lista o nome dos parâmetros, seus tipos e possíveis intervalos de valores válidos.

Parâmetros	Tipo	Intervalo Válido
EGO_INIT_DIST	numeric	$[0, +\infty[$
EGO_SPEED	numeric	$[5, 80]$
EGO_BRAKE	float	$[0, 1]$
ADV_INIT_DIST	numeric	$[0, +\infty[$
ADV_SPEED	float	$[5, 80]$
SAFETY_DIST	float	$[0, 20]$
CRASH_DIST	float	$[0, 5]$

Tabela 3 – Descrição dos parâmetros utilizados na modelagem dos cenários de cruzamento

Há várias maneiras de codificar a sequência genética de um indivíduo em um algoritmo genético. Para o contexto deste projeto, que envolve o uso de intervalos de números variados, podendo ser contínuos, a representação mais apropriada é a literal, utilizando o próprio valor das variáveis. Nessa abordagem, os valores dos parâmetros são listados na forma de um vetor, conforme exemplificado abaixo.

<EGO\_INIT\_DIST, EGO\_SPEED, EGO\_BRAKE, ADV\_INIT\_DIST, ADV\_SPEED, SAFETY\_SPEED, CRASH\_DIST>



A propósito de exemplo, atribuindo cada uma das variáveis acima a uma letra grega, podemos definir, a população de uma geração de 6 indivíduos, como mostrado a seguir.

Geração #1	
1	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$
2	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$
3	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$
4	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$
5	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$
6	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$

Observe que todos os genes de cada indivíduo dessa geração tem uma cor distinta para indicar que são os indivíduos iniciais únicos do algoritmo. Nesse exemplo, no ciclo evolutivo, cada nova geração será definida pela seleção de 1 indivíduo, o mais apto da geração anterior, 4 terão origem do cruzamento entre seus respectivos pais e 1 terá origem de um processo de mutação.

Sendo assim, o primeiro passo de criação da nova população é selecionar o indivíduo mais apto da geração anterior. Para o projeto, a adaptabilidade é medida em relação ao nível de risco de condução da simulação executada, considerando medidas de condução dos veículos e a ocorrência de colisão. Para esse exemplo ilustrativo, considere que o indivíduo 3 foi considerado como o mais apto da Geração #1.

Passo - Seleção - Geração #2	
1	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$

A seguir executa-se o cruzamento dos pais para a geração dos novos filhos. Nessa etapa, cada cruzamento gera dois indivíduos. Especificamente, os filhos 2 e 3, tem origem no cruzamento entre os pais **vermelho (1)** e **oliva (6)**, enquanto os filhos 4 e 5 são a mistura entre os pais **violeta (5)** e **verde-azulado (2)**.

## Passo - Cruzamento - Geração #2

2	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$
3	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$
4	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$
5	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$

Finalmente a mutação pode ocorrer de duas formas. A primeira é resultado da mutação de um gene da Geração #1, ao passo que a segunda tem origem na mutação dos filhos da Geração #2. Tomemos como exemplo a mutação de um indivíduo da Geração #1. Como resultado, o **indivíduo 4** foi sorteado na mutação, tendo o seu segundo gene ( $\beta$ ) modificado.

## Passo - Mutação - Geração #2

1	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$
---	----------------------------------------------------------------------------

Como resultado, a geração 2 correspondente é mostrada abaixo.

## Geração #1

1	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$
2	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$
3	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$
4	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$
5	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$
6	$\langle \alpha, \beta, \gamma, \delta, \epsilon, \epsilon, \zeta \rangle$

A partir desse modelo de codificação literal, é possível inserir a técnica de AG dentro do contexto de criação de cenários de condução crítica, utilizando a integração entre o Scenic e o Carla.

## 4.3.6.2 Função Objetivo

Das informações coletadas dos veículos foi gerada uma lista de hipóteses das métricas que poderiam ser utilizadas para compor a função objetivo, listados na Tabela 4. Com base em testes que destacassem a relevância de cada métrica, a partir da avaliação do grau de risco gerado por um procedimento aleatório, optou-se por utilizar apenas aquelas que contribuísem para discernir o risco associado a cada cenário simulado. Foram

consideradas as mesmas métricas para todos os 6 tipos de cenários de cruzamentos, listados na Tabela 2.

<b>Tipo</b>	<b>Descrição</b>
Evento	<b><u>Ocorrência de colisão</u></b>
Comando	Oscilação do volante
	Oscilação entre pedais (acelerador, freio)
Dinâmica	<b><u>Distância relativa Mínima entre os veículos (DM)</u></b>
	Velocidade relativa (de aproximação) dos veículos no ponto de DM
	Tempo para colisão dos veículos no ponto de DM
	<b><u>Distância relativa entre veículos no ponto de Velocidade Máxima (VM)</u></b>
	Velocidade relativa (de aproximação) dos veículos no ponto de VM
	<b><u>Tempo para colisão dos veículos no ponto de VM</u></b>

Tabela 4 – Lista de hipóteses de métricas para criação de função objetivo - métricas em negrito e sublinhadas foram efetivamente utilizadas

Das métricas utilizadas, foram geradas gradações que estão associadas ao nível de risco do cenário como mostrado na Tabela 5. O grau ou nível de risco de uma simulação é o somatório do *Score* de risco de cada variável (Colisão [C], Distância Mínima entre os veículos [DM], Distância dos veículos no instante de Velocidade Máxima de aproximação [D\_VM] e Tempo Para Colisão no instante de Velocidade Máxima de Aproximação [TCC\_VM]. Nesse sentido, um nível de risco 12 pode indicar superficialmente algumas possibilidades:

- Ocorrência de Colisão (10) + alguma combinação entre as demais variáveis totalizando 2
- *Score* de DM, D\_VM e TCC\_VM igual a 4, totalizando 12

Já para um caso de nível de risco 14, necessariamente houve alguma colisão, porque a não ocorrência de colisão alcança apenas um *score* de 12. Nesse caso, além da colisão, há também alguma combinação das demais variáveis totalizando 4.

Métrica	Intervalo	Score de Risco
C	Verdadeiro	10
	Falso	0
DM	[0, 820[	4
	[820, 1100[	3
	[1100, 1376[	2
	[1376, 1655[	1
	[1655, +∞[	0
D_VM	[0, 3780[	4
	[3780, 4255[	3
	[4020, 4255[	2
	[4255, 4490[	1
	[4490, +∞[	0
TTC_VM	[0, 359[	4
	[359, 394[	3
	[394, 429[	2
	[429, 464[	1
	[464, +∞[	0

Nota: C: Ocorrência de colisão, DM: Distância relativa mínima, D\_VM: Distância no instante de velocidade relativa máxima, TTC\_VM: Tempo para colisão no instante de velocidade relativa máxima

Tabela 5 – Risco associado a cada métrica

Em função da característica do Scenic (explicado no Capítulo 5), ocasionalmente, a busca por uma simulação válida a partir de um conjunto de parâmetros pode tornar-se inviável, resultando em um caso de teste não executável. Nessas situações, atribui-se ao indivíduo o valor -1 como *score* de risco. É relevante observar que as fases de geração de cenários por meio de AG (4.3.6.1) descarta a utilização de cenários não executados nas etapas de seleção, cruzamento e mutação, a fim otimizar o processo de busca de indivíduos válidos.

#### 4.3.6.3 Parâmetros

Por meio de experimentos de validação do modelo, foram definidos as probabilidades de ocorrência da seleção ( $\mu_s$ ), cruzamento ( $\mu_c$ ) e mutação ( $\mu_m$ ). Considerando que apenas esses três eventos de geração podem ocorrer, temos que:

$$\mu_s + \mu_c + \mu_m = 1$$

Nessa modelagem, enquanto a lista de novos indivíduos for menor que o tamanho da população, realiza-se o sorteio de um valor real via uma distribuição uniforme dentro do intervalo  $[0, 1]$ . Da Figura 19, se o valor estiver no intervalo  $[0, \mu_s[$  ocorre a seleção de um indivíduo da geração anterior. Para um valor no intervalo  $[\mu_s, \mu_s + \mu_c[$  ocorre um

cruzamento entre dois indivíduos da geração anterior, resultando em dois novos indivíduos. Por fim, se o valor estiver no intervalo  $[\mu_s + \mu_c, 1]$  sucede-se uma mutação.

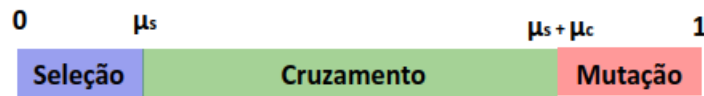


Figura 19 – Vetor de probabilidades de eventos para geração de novo indivíduo via AG

Uma abordagem alternativa de implementação consiste na contagem do número de indivíduos provenientes de cada uma das três fontes. Em vez de selecionar aleatoriamente um valor e verificar se ele está dentro do intervalo previamente determinado, fixa-se uma quantidade para cada tipo, que somas resultam no tamanho da população. Por exemplo, em uma população de 100 indivíduos, espera-se que 20 tenham origem na Seleção, 70 no Cruzamento e 10 na Mutaçāo. Para os objetivos deste projeto, não há diferença significativa entre as abordagens, permitindo a flexibilidade na escolha da implementação.

Para os experimentos realizados foram utilizados as probabilidades mostradas na Tabela 6. Esses valores foram definidos de maneira empírica. Apesar de estudos indicarem que  $\mu_m$  próximo de 1% podem atingir bons resultados de otimizaçāo (HESSER; MÄNNER, 1991; SCHAFFER et al., 1989), a utilizaçāo de uma probabilidade de 10% tem por objetivo proporcionar um espaço de busca com maior variabilidade (GREFENSTETTE, 1986), mas sem perder os bons resultados de indivíduos encontrados em geraçōes anteriores.

$\mu_s$	$\mu_c$	$\mu_m$
0.1	0.8	0.1

Tabela 6 – Probabilidades utilizadas para geraçāo dos experimentos do estudo de caso

## 5 Resultados

Nesse capítulo são apresentados os resultados dos estudos de caso obtidos por meio da técnica de geração de dados críticos baseados no refinamento dos parâmetros via AG.

### 5.1 Estudos de Caso

Conforme apresentado na Seção 4.3.5, o estudo de caso consiste na realização de experimentos envolvendo seis abordagens diferentes de cruzamento, detalhado na Tabela 2 e Figura 18. A hipótese desses experimentos é que a utilização de AG como heurística para otimização de parâmetros é significativamente efetiva quando comparado com a geração aleatória de cenários. Dessa forma, em todos os cenários de teste, foram executados 6000 simulações de cruzamento, das quais 3000 são gerados de maneira aleatória, enquanto as restantes foram geradas a partir do AG. No total, foram executados aproximadamente, 40.000 simulações válidas, exigindo cerca de 150 horas de execuções contínuas de experimentos.

Como critério de parada padrão para a execução do Algoritmo Genético (AG), estabeleceu-se a realização de 30 gerações de populações, cada uma composta por 100 indivíduos distintos. Para simplificar a comparação, optou-se por empregar a mesma estrutura de execução de simulações no caso da geração de parâmetros de forma aleatória. Assim, tornou-se viável realizar uma comparação padronizada, utilizando dois conjuntos de 100 cenários: um gerado de maneira aleatória e o outro direcionado pelo AG.

A apresentação dos resultados é dividida em quatro partes pelas métricas de nível de risco, quantidade de acidentes, distância mínima entre veículos e quantidade de cenários inválidos.

#### 5.1.1 Avaliação do grau de risco

Com base na função de risco que atribui um grau de risco no intervalo  $[-1, 22]$  a cada cenário gerado, foi elaborada a Figura 20. As tabelas correspondentes para cada cenário de teste 12, 13, 14, 15, 16, 17 estão disponíveis no apêndice. Em todas as tabelas, cada linha representa uma geração de 100 indivíduos do passo evolutivo do AG, sendo comparado com um conjunto de 100 cenários correspondentes que utilizam a heurística aleatória na geração dos parâmetros.

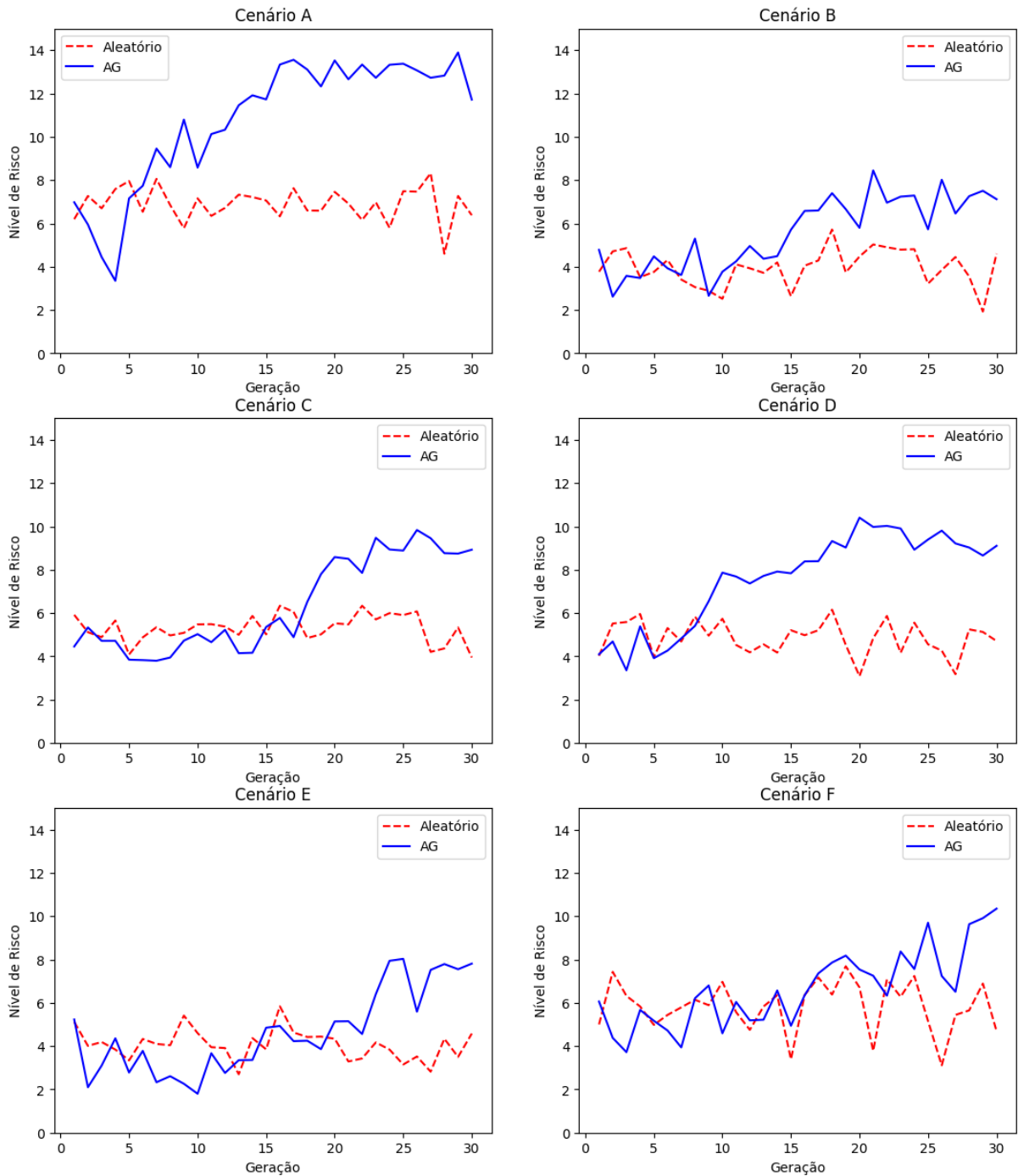


Figura 20 – Comparação do grau de risco nos 6 cenários

Como a métrica de risco é a própria função objetivo utilizada no AG, os diagramas da Figura 20 evidenciam uma característica da otimização via AG, que é a não monotonicidade da função objetivo. Analisando o resultado do cenário B (Tabela 13), é possível observar no intervalo das 10 primeiras gerações uma oscilação do resultado, tendo a quinta, sétima e oitava gerações do AG um ganho positivo em relação aos experimentos aleatórios, com as demais gerações com ganhos negativos. Esse mesmo comportamento é visto no cenário F (Tabela 17), com a ligeira diferença que esse comportamento se estendeu até a geração 16.

Um resultado importante para todos os cenários é que a partir de alguns ciclos de otimização, a busca consistentemente revela ganhos significativos do AG em comparação com a abordagem aleatória. Nesses casos, foi constatado que para todos os cenários, as últimas 10 gerações, em geral, o AG apresenta um resultado superior no que diz respeito ao nível de risco, tendo a grande maioria dos casos significância estatística relevante. Nos cenários A, B, C e D, a convergência ocorre pelo menos até a metade do experimento, enquanto nos cenários E e F, a convergência é mais evidente nas gerações finais. Para essas situações, pode ser indicado a execução de gerações adicionais para validar a hipótese de melhora da convergência. Notavelmente, nos experimentos, a utilização do AG nos cenários A e D apresentam os melhores desempenhos em termos de velocidade de convergência, enquanto que os cenários C e E apresentam maior resistência, possivelmente associado à inicialização dos parâmetros de busca.

A Tabela 7, é resultado de uma análise global de desempenho entre as técnicas. Em contraste com as tabelas anteriores, seu propósito é avaliar, na média, ao longo de todas as gerações, qual das duas técnicas em uma busca mais efetiva. Novamente, o AG demonstrou ser a heurística mais efetiva para todos os cenários, exibindo um alto nível de significância estatística, exceto para o caso F.

CENARIO	AG	ALEATÓRIO	DIFERENÇA	GANHO (%)
A	13,32 ± 6,68***	9,70 ± 7,69	3,62	37,30
B	7,48 ± 6,86***	6,07 ± 6,56	1,41	23,27
C	8,78 ± 6,82***	7,58 ± 6,56	1,20	15,83
D	9,84 ± 6,03***	7,53 ± 6,32	2,31	30,64
E	7,69 ± 6,71***	5,96 ± 6,18	1,74	29,15
F	10,08 ± 6,5.	9,68 ± 6,92	0,40	4,09

Tabela 7 – Risco de condução Médio - Todos os cenários

Nota: Os símbolos adotados nesta tabela indicam o nível de significância estatística. O símbolo "\*\*\*" indica um *p-value* menor ou igual a 0,001, "\*\*\*" denota um *p-value* menor ou igual a 0,01, "\*" denota um *p-value* menor ou igual a 0,05 e "." denota um *p-value* menor ou igual a 0,1.

### 5.1.2 Quantidade de colisões

Essa métrica visa simplificar a avaliação do nível de risco das simulações, uma vez que, no ambiente de controle desenvolvido, existe uma correlação positiva entre as simulações que ocorrem colisões e aquelas que quase as experimentam. Nesse contexto, a Figura 21 mostra, para cada geração, a quantidade de colisões observadas. Um padrão notável é que, no caso aleatório, a média de todos os cenários permanece semelhante ao longo de todas as gerações, enquanto nos casos de AG, as gerações iniciais apresentam quantidades de colisões inferiores à média, sendo seguidas por aumentos consideráveis nas gerações finais. A Tabela 18 com as contagens completas de colisões está disponível no apêndice.



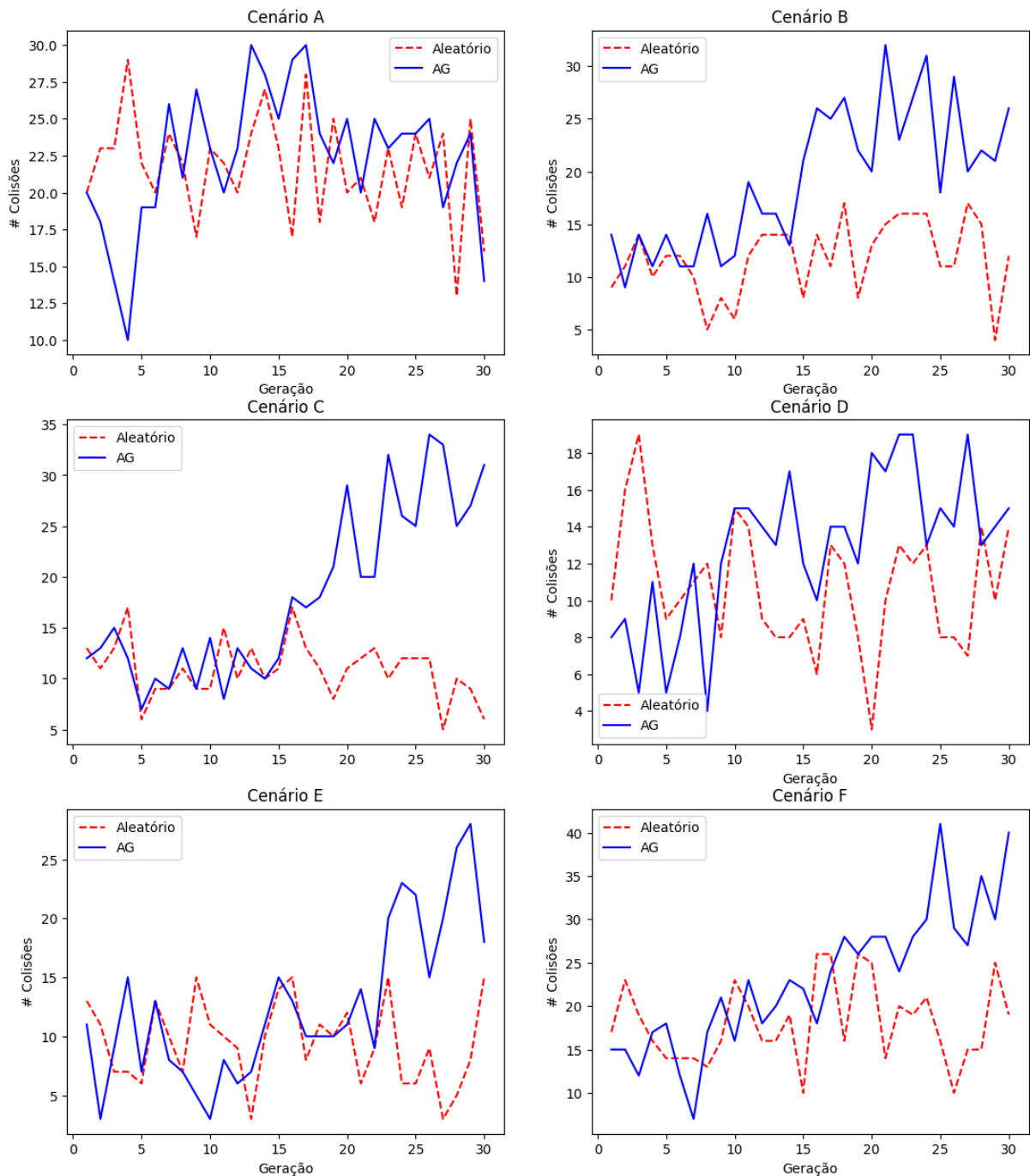


Figura 21 – Comparação da quantidade de colisões nos 6 cenários

Apesar desse padrão de comportamento dos resultados de cenários via AG, nota-se que não há diferença significativa na quantidade de colisões totais entre os cenários de teste A, D e E, como indicado na última linha da Tabela 18. Nesse sentido, a Tabela 8 foi elaborada para facilitar a análise do efeito evolutivo nos experimentos. Essa tabela fraciona a quantidade de colisões das 30 gerações em apenas três grupos equitativos - primeiro, segundo e último terço do total de gerações. Como a inicialização do AG utiliza a heurística aleatória, a dificuldade de convergência nas primeiras gerações é um resultado aceitável, observável nos cenários A, D, E e F. Entretanto, para todos os cenários, nota-se

que os casos gerados pelo AG no último terço de gerações apresentam consistentemente um grau de risco mais elevado, em conformidade com a hipótese proposta.

Somatório Gerações	A		B		C		D		E		F	
	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA
1 - 10	223	197	97	123	107	114	123	89	100	81	169	150
11 - 20	224	256	125	205	119	157	90	139	102	101	200	230
21 - 30	204	220	133	249	101	273	109	158	82	195	174	312

Tabela 8 – Somatório de colisões - intervalos de 1 a 10 gerações; 11 a 20 gerações; 21 a 30 gerações

Vale notar que dependendo do modelo gerado para função objetivo, as colisões podem não exercer um impacto substancial no grau de risco. Os experimentos apresentados têm uma alta correlação, porque, no intervalo do *score* de risco  $[-1, 22]$ , a ocorrência de colisão adiciona um valor 10 no nível de risco.

### 5.1.3 Distância Mínima de Aproximação

A distância mínima é uma métrica que tem por objetivo avaliar a proximidade entre os veículos para mensurar de forma indireta o nível de risco das simulações. A atribuição de risco para a distância mínima depende do contexto da ação dos veículos, posto que em condições normais de tráfego, os veículos se aproximam, mas não necessariamente de forma arriscada ou agressiva. Uma vez que o contexto dos experimentos forçam o confronto entre veículos na zona de conflito em ações de cruzamento e conversão, a distância mínima é uma forma de aferir a agressividade de aproximação entre eles. Nesse sentido, as Figuras 22, 23 mostram curvas médias das distâncias mínimas de aproximação entre os dois veículos avaliados, sendo a Figura 22 a média considerando todos os cenários válidos dos experimentos, enquanto a Figura 23 desconsidera as simulações que resultaram em colisões. O objetivo dessa diferenciação é conseguir avaliar as situações de quase acidentes, já que a ocorrência de colisão já são situações de aproximação mínima entre os veículos. As respectivas Tabelas 19 e 20 completas estão disponíveis no apêndice.

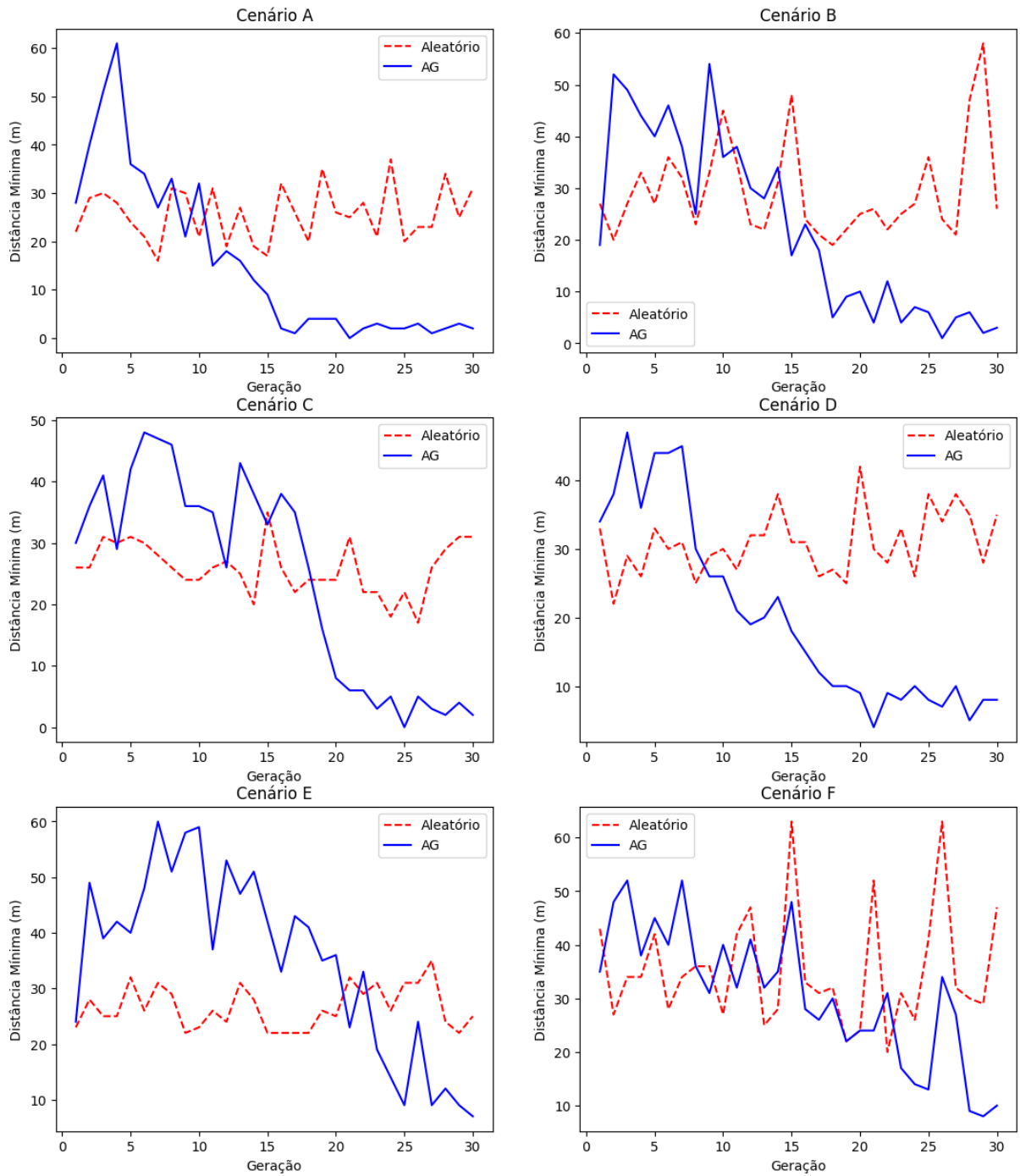


Figura 22 – Comparação da distância mínima considerando todas as simulações válidas nos 6 cenários

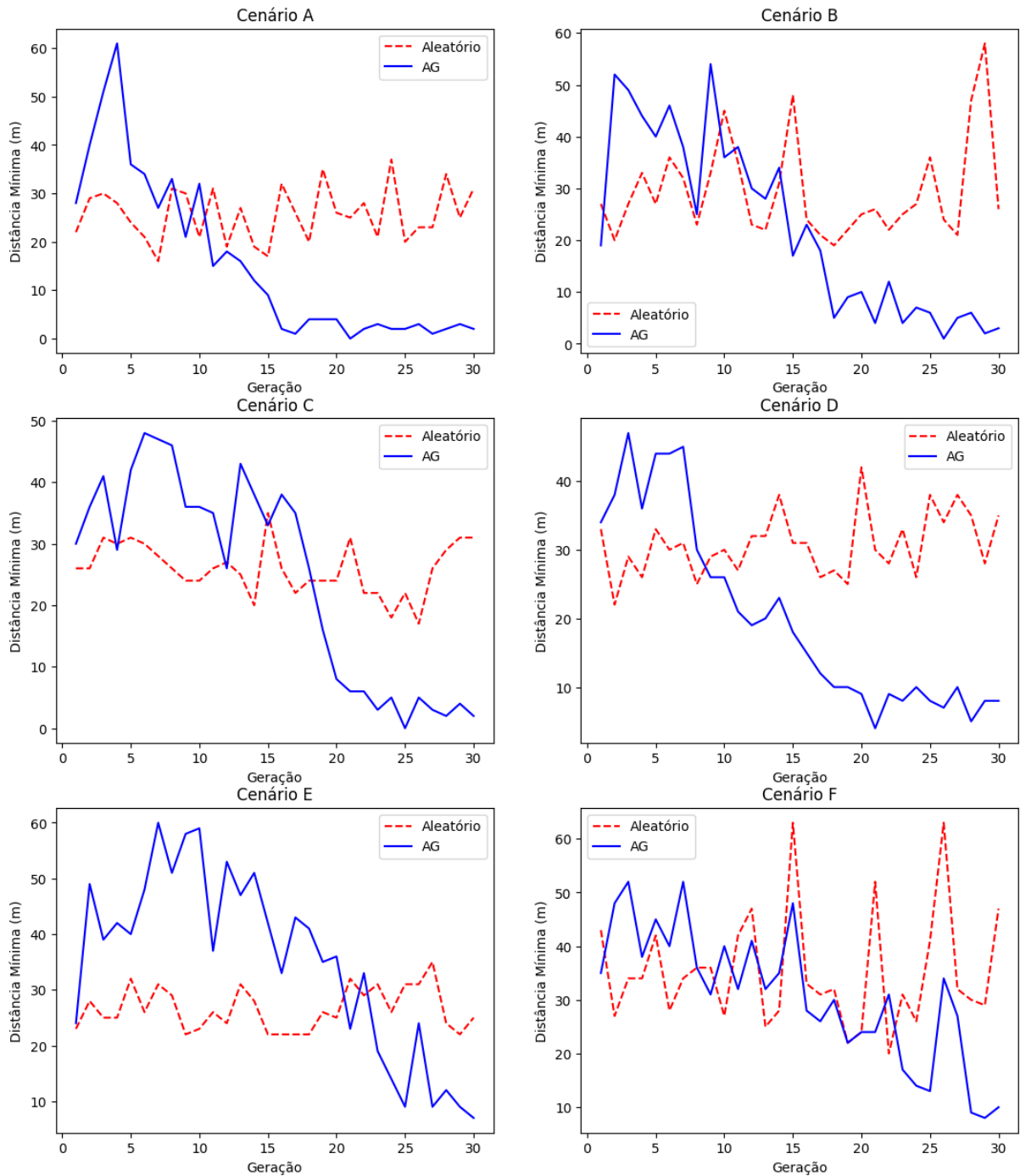


Figura 23 – Comparação da distância mínima excluindo as simulações com colisões nos 6 cenários

Como apresentado nas subseções anteriores, o padrão de comportamento de todos os cenários que utilizam AG evoluem para o aumento do risco, com a redução da distância mínima à medida que mais gerações são executadas. Em comparação, as médias de gerações que utilizam a heurística aleatória orbitam ao redor da média global de seus respectivos experimentos completos. Analogamente à análise anterior, a Tabelas 9 e 10 seccionam as 30 gerações em três grupos. A partir dessas tabelas, observa-se um resultado não tão positivo do ponto de vista de aumento de risco, em que, principalmente para o caso de

distância mínima sem colisão, a média dos cenários D e F apresentam um aumento da distância mínima no grupo das gerações finais. Esse resultado pode ser consequência da abordagem evolutiva de busca AG, em que conjuntos de indivíduos menos aptos podem ser criados. Finalmente, os cenários A, B, C e E apresentam um comportamento mais aderente com o objetivo do projeto.

Somatório Gerações	A		B		C		D		E		F	
	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA
1 - 10	1416,3	879,7	1869,3	1403,2	1684,0	1319,9	1755,6	1433,8	1955,3	1790,8	2202,5	1998,8
11- 20	1398,7	713,8	1924,3	1254,5	1618,3	1019,4	1812,8	1324,7	2057,3	1530,3	1981,2	1845,3
21 - 30	1389,1	693,6	1874,5	1214,9	1625,1	1014,4	1808,4	1445,7	2136,4	1438,6	2006,5	1667,8

Tabela 9 – Distância mínima média dos cenários válidos - intervalo de 1 a 10 gerações; intervalo de 11 a 20 gerações; intervalo de 21 a 30 gerações

Somatório Gerações	A		B		C		D		E		F	
	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA
1 - 10	1678,9	1035,2	2075,9	1591,3	1857,5	1458,2	1881,8	1538,0	2043,6	1861,4	2619,2	2289,3
11- 20	1667,2	810,3	2177,1	1452,0	1818,2	1183,9	1925,7	1449,7	2147,9	1629,4	2375,8	2144,1
21 - 30	1612,1	779,0	2176,1	1417,8	1782,2	1225,7	1957,0	1609,2	2246,2	1547,0	2360,7	1915,4

Tabela 10 – Distância mínima média dos cenários válidos que não ocorrem colisão - intervalo de 1 a 10 gerações; intervalo de 11 a 20 gerações; intervalo de 21 a 30 gerações

Nota: Os símbolos adotados nesta tabela indicam o nível de significância estatística. O símbolo "\*\*\*\*" indica um *p-value* menor ou igual a 0,001, "\*\*\*"denota um *p-value* menor ou igual a 0,01, "\*"denota um *p-value* menor ou igual a 0,05 e "."denota um *p-value* menor ou igual a 0,1.

#### 5.1.4 Quantidade de Inválidos

Devido às particularidades da técnica de geração de cenários, conforme delineadas na subseção 5.2.1, existe uma parcela significativa de cenários gerados que não resultam em uma simulação executável. Como uma medida adicional para avaliar a eficiência do AG, também foi construída uma análise que destaca a eficácia da heurística. Para esse trabalho, define-se eficácia como uma taxa de aproveitamento dos cenários testados, ou seja, quanto maior a quantidade de simulações válidas em relação a um número fixo de testes, maior é a eficácia.

Assim como nas análises de colisões, a Figura 24 resume a quantidade de cenários inválidos por simulação. A Tabela 21 que suporta a figura se encontra no apêndice. De maneira análoga, a heurística aleatória apresenta um comportamento próximo tanto de uma distribuição uniforme ao longo das gerações, quanto da média global. Todavia, existe uma tendência dos experimentos que utilizam AG de reduzir significativamente a quantidade de cenários inválidos conforme o avanço das gerações. Vale notar que no cenário de teste F não houve um ganho significativo, sugerindo que, potencialmente, a descrição formal desse cenário é mais restrita em relação ao mapa utilizado para a geração dos experimentos.

A confirmação dessa hipótese requer a realização de mais experimentos com a mesma descrição, tanto no mapa utilizado para esse estudo de caso, quanto em outros mapas.

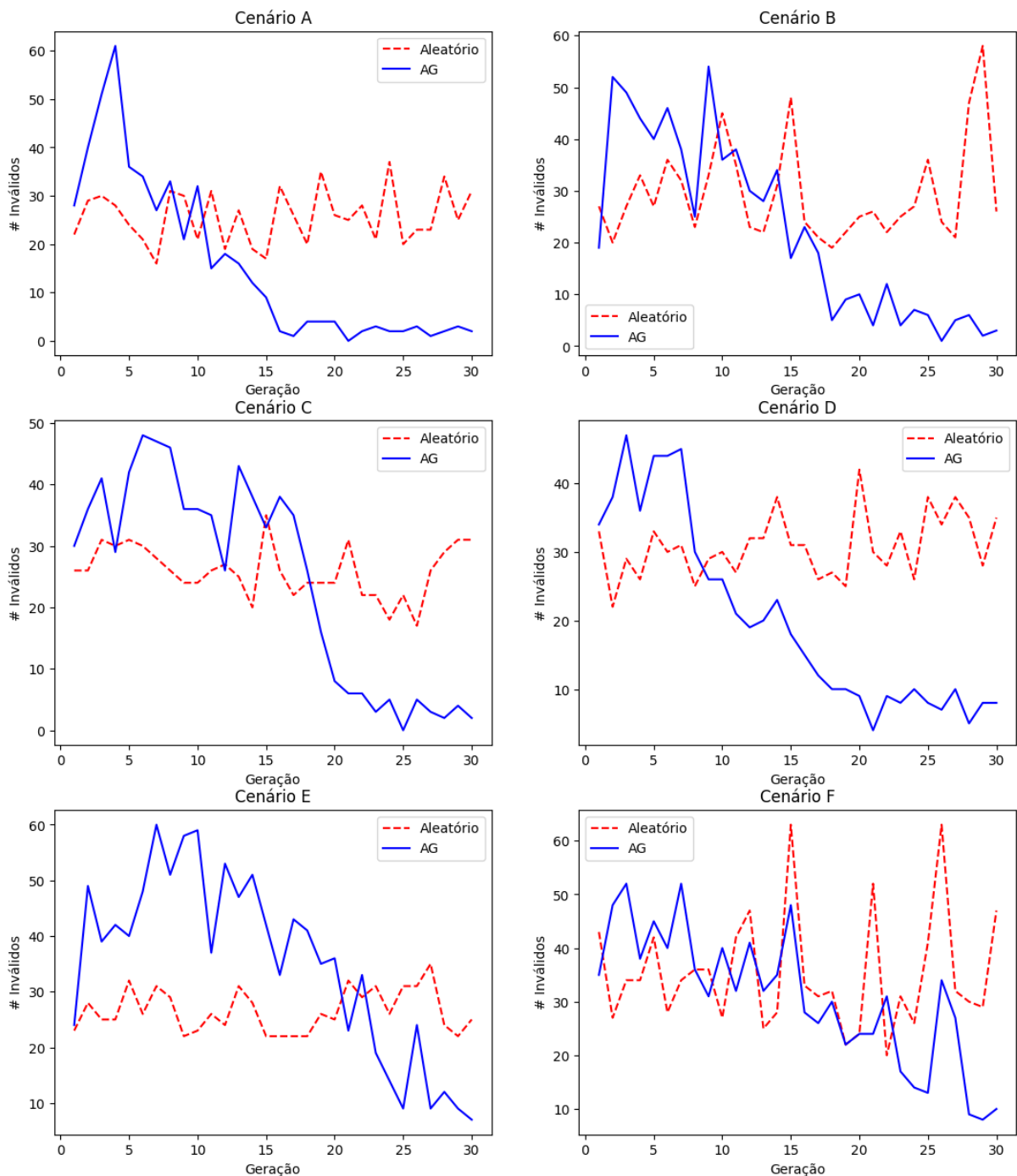


Figura 24 – Comparação da quantidade de simulações inválidas nos 6 cenários

A Tabela 11 reforça a análise de que à medida que o algoritmo evolui, melhor a eficácia do processo de geração. Em todos os cenários de teste deste estudo, observa-se uma queda significativa na quantidade de cenários inválidos no último terço das gerações.

Somatório Gerações	A		B		C		D		E		F	
	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA
1 - 10	252	363	303	403	276	391	288	370	264	470	341	417
11- 20	252	85	270	212	253	298	311	157	248	418	347	318
21 - 30	267	20	312	50	249	36	325	77	286	159	371	187

Tabela 11 – Somatório de inválidos - intervalos de 1 a 10 gerações; 11 a 20 gerações; 21 a 30 gerações

A utilização do AG, portanto, se mostrou efetiva na melhora da eficácia no processo de geração de cenários com maior grau de risco.

## 5.2 Limitações do Projeto

O projeto alcançou com sucesso o principal objetivo de gerar quantidades significativas de dados de cenários voltados para conduções críticas. No entanto, o método de geração apresenta algumas deficiências relacionadas ao consumo de energia computacional causado pela eficácia de geração de simulações válidas. Esta seção discute alguns obstáculos identificados na solução proposta.

### 5.2.1 Geração de cenários do Scenic

A linguagem Scenic foi concebida para reduzir o esforço necessário na especificação de ambientes complexos, como a definição e controle da trajetória de veículos em uma simulação. Ela possibilita a construção de cenários que podem ser compostos por restrições tanto fixas quanto flexíveis. Dentro do contexto do Carla, a geração do cenário é baseada na busca de simulações que esteja dentro do conjunto de restrições que define um cenário.

Das diversas formas de implementação dessa busca, a linguagem utiliza a abordagem mais direta de amostragem por rejeição. Nessa abordagem, o programa continua a amostrar cenários desconsiderando o conjunto de requisitos até encontrar a primeira amostra que o satisfaça (FREMONT, 2019). Esse método tem a capacidade de gerar resultados desejados, contanto que o conjunto de parâmetros resulte em uma simulação válida. No entanto, um aspecto que merece atenção é importante notar que esse processo pode se estender indefinidamente, a depender da dificuldade das restrições. Já no caso do conjunto de parâmetros serem inconsistentes ou tenham probabilidade zero (em função do mapa), a condição de parada do método nunca é atingida. Como forma de resolver esse problema de parada, como padrão, o algoritmo busca em até 2000 interações um cenário de simulação válido. Caso não seja encontrado, um `RejectionException` é acionado e nenhuma simulação é gerada.

Na perspectiva da automação, a escolha dos parâmetros pode apresentar desafios na criação de cenários válidos, especialmente quando o conjunto de parâmetros é excessivamente restrito. A abordagem adotada consistiu em aplicar um sutil relaxamento nas

variáveis, viabilizando a exploração de cenários em intervalos mais flexíveis. No caso de distâncias, por exemplo, a busca é realizada em um intervalo com variação de até 3 metros, equivalente ao comprimento de um veículo pequeno. Quanto à velocidade, estabeleceu-se um intervalo de variação de  $1m/s$ .

### 5.2.2 Algoritmo Genético

O AG é um algoritmo de otimização em que a sua busca é incompleta, porque pode não explorar exaustivamente todas as soluções devido à sua natureza estocástica na aplicação de operadores de seleção, cruzamento e mutação. A incompletude tem algumas implicações no que diz respeito a:

- Garantia de solução: Não há garantia de encontrar soluções para todos os problemas
- Soluções não ótimas: Não há garantia de encontrar soluções globais ou ótimas. A busca pela solução global dependendo das condições iniciais
- Condições iniciais: A eficiência, no geral, depende das condições iniciais. Inicializações ruins resultam em soluções insatisfatórias
- Critério de parada: A escolha arbitrária de um critério de parada pode impactar na qualidade da solução obtida

No contexto da aplicação na busca de configurações de cenários mais arriscados, os quatro itens da lista têm influência direta na qualidade das soluções obtidas. Primeiramente, a característica central do problema é o alto grau de complexidade. A parametrização para a busca dos cenários foi modelada na tentativa de reduzir o grau de complexidade. Mesmo assim, dependendo do cenário o conjunto de soluções geradas pode não ser viáveis conforme a subseção 5.2.1. Nos experimentos realizados, o cenário F foi aquele que apresentou menor diferença entre os tipos de seleção aleatória e evolutiva, possivelmente relacionado com a dificuldade da busca da situação de cruzamento.

Do ponto de vista de soluções sub-ótimas no AG há uma conexão com a forma como a busca é conduzida. Foi constatado que a partir de algumas interações evolutivas, o padrão da solução tende a se repetir na maioria dos indivíduos. Em termos práticos, isso significa que há um trecho genético (um conjunto de parâmetros) que se repete ou é muito semelhante para a maioria dos indivíduos. Vale ressaltar que a composição de novas gerações não utilizam soluções repetidas, mas, eventualmente, algumas simulações aparentam alto grau de semelhança. Essa convergência local está ligada com a inicialização da primeira geração.

No projeto, o critério de parada escolhido foi arbitrário. Como o objetivo é a geração de muitos conjuntos de parâmetros, ao invés de definir o critério de parada como um valor



para a taxa média de risco em uma geração, optou-se pela contagem de gerações, 30 como padrão. Os resultados indicam uma tendência de que quanto mais gerações, melhor a avaliação das métricas utilizadas nas análises, todavia, não é possível, antecipadamente, prever se as soluções obtidas são as possíveis melhores, tanto para o cenário de ação especificado, quanto para a inicialização do problema.

A função objetivo também é uma escolha arbitrária do AG. Nesse projeto, foi efetuada um estudo superficial das principais variáveis que podem indicar risco, além da atribuição de uma graduação de risco para cada uma delas. Pequenos ajustes na função objetivo podem, eventualmente, melhorar ou deteriorar as soluções obtidas. De maneira análoga, as escolhas das probabilidades de ocorrência de seleção, cruzamento e mutação no AG foram arbitrárias. Embora estejam alinhadas com a literatura, os estudos consultados são em aplicações de outras áreas.

Dessa forma, para ter um entendimento mais aprofundado sobre a utilização do AG como heurística de busca, é necessário conduzir um estudo mais aprofundado fazendo variações do critério de parada, função objetivo, probabilidades e padronização das condições iniciais.

### 5.2.3 Tipos de risco

O estudo de caso utilizou apenas um dos mapas disponibilizados pelo Carla. A quantidade de soluções inválidas está diretamente relacionado ao mapa de teste, posto que, quanto maior o mapa, maior a probabilidade da descrição de um cenário ser factível. Nesse sentido, em um estudo mais completo é relevante variar os mapas, considerando um critério pertinente na escolha de mapas os quais, notoriamente indicados pela literatura, apresentem, devido à sua geometria, a ocorrência de acidentes (por exemplo, em trechos de vias comuns com maior incidência de colisões).

A construção dos *dataset* de risco vão além do tipo de risco estudado no estudo de caso - situações de cruzamento entre dois veículos. Uma avaliação mais rigorosa da solução proposta deve envolver, além da ação a ser performada (cruzamentos, ultrapassagens, entre outros), as condições climáticas, presença de pedestres, animais e objetos na via. Assim, apesar dos resultados do estudo de caso indicarem que a proposta de AG é efetiva, ainda são necessárias construções de experimentos que façam a variação da origem do risco.

## 6 Considerações Finais

### 6.1 Conclusões do Projeto de Formatura

Esse projeto desenvolveu um método para geração de casos críticos de condução veicular. A infraestrutura foi baseada na integração entre um simulador e algumas bibliotecas e implementações para a produção dos cenários de interesse. O simulador utilizado foi o Carla Simulator, a principal biblioteca de interface para a construção dos cenários foi o Scenic. No estudo de caso, foi feita uma comparação entre técnicas de seleção de parâmetros para o aumento da probabilidade de criação de um cenário de risco.

A utilização de AG apresentou um ganho positivo no nível de risco variando entre 4% (cenário F) e 37% (cenário A), com um ganho médio geral de todos os cenários de aproximadamente 23%. Para o experimento completo, a abordagem aleatória resultou na coleta de 2482 colisões, enquanto via AG 3249, um acréscimo de 4,26% de acidentes no conjunto completo de simulações. Do ponto de vista da distância mínima média, a redução de distâncias, considerando todas as simulações válidas de todos os cenários, passou de 1,806m para 1,332m quando o AG foi utilizado. Já as distâncias mínimas válidas, excluindo as simulações que resultaram em colisão, passou de 2,022m para 1,496m indicando um aumento do risco de condução pela proximidade dos veículos. Finalmente, nota-se que o AG também apresentou um impacto positivo na redução de geração de simulações inválidas, totalizando 5215 da abordagem aleatória para 4431 - redução de 784 simulações.

O principal resultado geral desse trabalho foi a construção da infraestrutura de simulação que consegue gerar cenários a partir de uma descrição similar a textual. Essa arquitetura possibilita a realização de diversos estudos que vão desde o desempenho do controle do VA na simulação, até a construção de novos controles baseados nos dados dos sensores de cada cenário executado. Nesse contexto, o estudo de caso, tem por objetivo validar o funcionamento correto da implementação, ao permitir o refinamento de parâmetros para a coleta de dados de risco.

Inicialmente, a primeira ideia do projeto era utilizar Redes Adversariais Generativas (RAGs) para a construção de *datasets* orientados a acidentes. Essa técnica abstrai as principais características de um conjunto de dados para a geração de novos dados. Nesse sentido, a implementação da técnica requer que exista um conjunto de dados relevantes de acidentes para que, de fato, seja possível construir um modelo que gere novos dados. A primeira parte do projeto - geração abundante de simulações com o envolvimento de risco - então, já se tornou um grande desafio. De fato, devido a alta complexidade da dinâmica

de tráfego de veículos, a descrição e a enumeração de cenários de risco torna-se inviável. Os primeiros testes de implementação de cenários baseados na descrição geométrica da via indicaram ser uma alternativa inviável, uma vez que o projeto não seria escalável. um segundo aspecto relevante, é que o produto principal da estrutura eram os arquivos de *logs* de descrição dos cenários, pois, a partir deles, é possível fazer novas reproduções com a inclusão e coleta de dados dos sensores utilizados para o treinamento de controles inteligentes de VA. A utilização de um modelo de RAG limitaria imporia uma restrição do tipo de dados gerado. Por exemplo, um modelo de RAG de geração de imagens no padrão RGB, só consegue gerar novas imagens RGB. Além disso, essa abordagem possivelmente não permitiria a integração de diversos tipos de sensores, ou seja, modelos de geração de imagens, possivelmente não gerariam os respectivos dados complementares de outros sensores (lidares, radares, ou tipos de câmera). Por conta dessas restrições observadas, houve um direcionamento efetivo para o enfoque de otimização de parâmetros no próprio simulador. Essa decisão foi importante, porque entende-se que a solução atingida além de ser mais econômica, energética e computacionalmente, também é mais efetiva. Um banco de dados de *logs* reproduzíveis de simulação da ordem de 100k simulações ocupa aproximadamente 10GB de espaço, enquanto que um banco de dados de apenas câmeras das simulações desses *logs* ocuparia na ordem de dezenas a centenas de Terabytes, a depender da qualidade de resolução das imagens. Assim, pode-se entender que a coleta de *logs* é a forma compacta de gerar os respectivos dados de sensores.

Do estudo de caso, apesar de ser um teste inicial de produção de cenários, conclui-se que, de fato, é possível automatizar o processo de geração de dados orientados a situações críticas. A arquitetura ainda necessita de algumas otimizações para que outros pesquisadores da área possam contribuir caso o projeto seja disponibilizado em um repositório de código aberto.

## 6.2 Contribuições

A partir da integração entre o simulador, a biblioteca Scenic modificada e os demais módulos de seleção, metrificação e avaliação dos resultados, foi possível construir um sistema que, a partir de descrições mais genérica de situações, são gerados *logs* de simulações que realizam a descrição de entrada. Além disso, com o módulo de seleção dos parâmetros, é possível maximizar a probabilidade de ocorrência de situações críticas. A importância dessa implementação está relacionado com o grau de automação do processo de geração de dados de direção que poderão ser posteriormente utilizado para outras pesquisas sobre VAs.

Do ponto de vista acadêmico, esse trabalho tem o potencial de contribuir com o estado da arte na geração de cenários críticos que são utilizados na construção de controles

de VAs mais seguros. O impacto para a sociedade é construir controles efetivamente mais seguros do que comparado com um condutor humano. Indiretamente, caso essa tecnologia se torne mais segura, a adoção em escala é viabilizada. Como consequência de uma ampla adoção, tem-se redução de custos que envolvem tanto a prevenção de acidentes e a otimização do fluxo de tráfego urbano. Assim, gasta-se menos com os efeitos de acidentes, reduz-se o custo de operação dos veículos devido a falta de otimização do tráfego e o mais importante, preserva-se a vida.

### 6.3 Trabalhos Futuros

Vale novamente notar que esse projeto foi realizado dentro do programa de pré-mestrado da Escola Politécnica da USP, portanto, será utilizado durante a continuação do mestrado e demais pesquisas sobre o contexto de segurança de VAs.

A continuação do projeto futuro pode se dar em diversas frentes, sendo primeira delas é a realização de mais testes, variando os mapas do simulador (até utilizando trechos de mapas reais), as ações, com a execução de ultrapassagens, presença de pedestres animais e objetos e também a otimização dos parâmetros de controle do algoritmo genético e da função objetivo. A partir desse trabalho base, será realizado um estudo mais aprofundado sobre a influência de todos esses elementos. Possivelmente, com a geração massiva de *logs* de simulações, serão coletados e montados bancos com os dados dos sensores de cada simulação de interesse para a validação da hipótese de que a utilização dessa massa de dados mais arriscados melhora a eficiência, do ponto de vista da segurança do controle. A ideia anterior pode ser um tema interessante para o mestrado.

Outras implementações podem ser realizadas para permitir que estudos sobre o impacto das condições climáticas e de ambiente tem sobre a segurança de controles. O escopo principal dos estudos propostos anteriormente estão relacionados com a conduta dos motorista, excluído todos os elementos externos, que sabidamente influem sobre a qualidade dos dados dos sensores e da performance mecânica dos veículos.

Finalmente, com a geração massiva de *logs* de simulações e uma atribuição de risco, é possível realizar um estudo de *meta learning* para gerar um modelo que consiga exprimir a relação entre os parâmetros de descrição de cenários e o seu risco correspondente, sem a necessidade de testar a descrição no simulador. O *meta learning* é uma abordagem de aprendizado de máquina que se concentra na capacidade de um modelo aprender a partir de um conjunto limitado de dados. Para o contexto de simulação de VAs, a principal ideia é conseguir extrair os padrões de risco associado a um conjunto de parâmetros, e, a partir daí, inferir o risco de novos conjuntos de parâmetros de maneira mais eficaz e barata computacionalmente.

# Referências

- ADMINISTRATION, N. H. T. S. et al. Automated driving systems 2.0: A vision for safety. *Washington, DC: US Department of Transportation, DOT HS*, v. 812, p. 442, 2017. Citado na página 48.
- AMINI, A. et al. Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles. In: IEEE. *2022 International Conference on Robotics and Automation (ICRA)*. [S.l.], 2022. p. 2419–2426. Citado 3 vezes nas páginas 15, 25 e 28.
- ANDERSON, J. M. et al. *Autonomous Vehicle Technology: A Guide for Policymakers*. Santa Monica, CA: RAND Corporation, 2016. Citado na página 17.
- AUTOMATION, A. for Standardization of; SYSTEMS, M. *ASAM OpenDrive*. 2023. Disponível em: <<https://www.asam.net/standards/detail/opendrive/>>. Acesso em: 04 jul 2023. Citado na página 35.
- BEHRINGER, R. et al. The darpa grand challenge - development of an autonomous vehicle. In: *IEEE Intelligent Vehicles Symposium, 2004*. [S.l.: s.n.], 2004. p. 226–231. Citado na página 17.
- BLOCK, H.-D. The perceptron: A model for brain functioning. i. *Reviews of Modern Physics*, APS, v. 34, n. 1, p. 123, 1962. Citado na página 19.
- BRIEFS, U. Mcity grand opening. *Research Review*, v. 46, n. 3, 2015. Citado na página 14.
- BUEHLER, M.; IAGNEMMA, K.; SINGH, S. *The DARPA urban challenge: autonomous vehicles in city traffic*. [S.l.]: springer, 2009. v. 56. Citado na página 17.
- CARLA-TEAM. *Carla Recorder - Website*. 2021. Disponível em: <[https://carla.readthedocs.io/en/latest/adv\\_recorder/](https://carla.readthedocs.io/en/latest/adv_recorder/)>. Acesso em: 25 jul 2023. Citado na página 41.
- CARLA-TEAM. *GitHub Carla simulator - 0.9.13*. 2021. Disponível em: <<https://github.com/carla-simulator/carla/tree/0.9.13>>. Acesso em: 15 nov 2023. Citado na página 39.
- CARLA-TEAM. *Recorder Binary File Format*. 2021. Disponível em: <[https://carla.readthedocs.io/en/latest/ref\\_recorder\\_binary\\_file\\_format/](https://carla.readthedocs.io/en/latest/ref_recorder_binary_file_format/)>. Acesso em: 25 jul 2023. Citado 2 vezes nas páginas 7 e 42.
- CARLA-TEAM. *3rd. Maps and navigation*. 2023. Disponível em: <[https://carla.readthedocs.io/en/0.9.11/core\\_map/](https://carla.readthedocs.io/en/0.9.11/core_map/)>. Acesso em: 20 nov 2023. Citado 2 vezes nas páginas 7 e 41.
- CARLA-TEAM. *Carla Simulator Introduction*. 2023. Disponível em: <[https://carla.readthedocs.io/en/latest/start\\_introduction/](https://carla.readthedocs.io/en/latest/start_introduction/)>. Acesso em: 10 ago 2023. Citado na página 35.

- CERF, V. G. A comprehensive self-driving car test. *Communications of the ACM*, ACM New York, NY, USA, v. 61, n. 2, p. 7–7, 2018. Citado na página 14.
- CHAND, A.; JAYESH, S.; BHASI, A. Road traffic accidents: An overview of data sources, analysis techniques and contributing factors. *Materials Today: Proceedings*, Elsevier, v. 47, p. 5135–5141, 2021. Citado na página 23.
- CHEBLY, A.; TALJ, R.; CHARARA, A. Coupled longitudinal and lateral control for an autonomous vehicle dynamics modeled using a robotics formalism. *IFAC-PapersOnLine*, Elsevier, v. 50, n. 1, p. 12526–12532, 2017. Citado na página 17.
- CHEN, B. et al. Adversarial evaluation of autonomous vehicles in lane-change scenarios. *IEEE transactions on intelligent transportation systems*, IEEE, v. 23, n. 8, p. 10333–10342, 2021. Citado na página 28.
- CONGER, K. *Driver Charged in Uber’s Fatal 2018 Autonomous Car Crash*. New York Times, 2020. Disponível em: <<https://www.nytimes.com/2020/09/15/technology/uber-autonomous-crash-driver-charged.html>>. Citado na página 24.
- DARWIN, C. *On the origin of species, 1859*. [S.l.]: Routledge London, 2016. Citado na página 29.
- DECO, G. et al. The dynamic brain: from spiking neurons to neural masses and cortical fields. *PLoS computational biology*, Public Library of Science San Francisco, USA, v. 4, n. 8, p. e1000092, 2008. Citado na página 19.
- DING, W. et al. A survey on safety-critical driving scenario generation—a methodological perspective. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, 2023. Citado na página 26.
- DOSOVITSKIY, A. et al. CARLA: An open urban driving simulator. In: *Proceedings of the 1st Annual Conference on Robot Learning*. [S.l.: s.n.], 2017. p. 1–16. Citado na página 35.
- DU, H. et al. Elitism and distance strategy for selection of evolutionary algorithms. *IEEE Access*, IEEE, v. 6, p. 44531–44541, 2018. Citado na página 29.
- DUMITRESCU, D. et al. *Evolutionary computation*. [S.l.]: CRC press, 2000. Citado na página 29.
- ELMQUIST, A.; SERBAN, R.; NEGRUT, D. A sensor simulation framework for training and testing robots and autonomous vehicles. *Journal of Autonomous Vehicles and Systems*, American Society of Mechanical Engineers, v. 1, n. 2, p. 021001, 2021. Citado na página 32.
- Epic Games. *Unreal Engine*. 2019. Disponível em: <<https://www.unrealengine.com>>. Citado na página 35.
- EVANS, L. *Traffic safety*. [S.l.: s.n.], 2004. Citado na página 23.
- FORREST, S. Genetic algorithms. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 28, n. 1, p. 77–80, 1996. Citado na página 29.

- FREMONT, D. J. *Algorithmic improvisation*. [S.l.]: University of California, Berkeley, 2019. Citado na página 66.
- FREMONT, D. J. et al. Scenic: A language for scenario specification and data generation. *Machine Learning*, Springer, v. 112, n. 10, p. 3805–3849, 2023. Citado 2 vezes nas páginas 37 e 43.
- GAMBI, A.; MUELLER, M.; FRASER, G. Automatically testing self-driving cars with search-based procedural content generation. In: *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*. New York, NY, USA: Association for Computing Machinery, 2019. (ISSTA 2019), p. 318–328. ISBN 9781450362245. Disponível em: <<https://doi.org/10.1145/3293882.3330566>>. Citado na página 24.
- GE, J. et al. Heterogeneous driver modeling and corner scenarios sampling for automated vehicles testing. *Journal of advanced transportation*, Hindawi, v. 2022, 2022. Citado 3 vezes nas páginas 15, 16 e 25.
- GNU. *Bash Reference Manual*. 2023. Disponível em: <<https://www.gnu.org/software/bash/manual/bash.html>>. Acesso em: 15 nov 2023. Citado na página 38.
- GOODFELLOW, I. et al. Generative adversarial networks. *Communications of the ACM*, ACM New York, NY, USA, v. 63, n. 11, p. 139–144, 2020. Citado na página 27.
- GREFENSTETTE, J. J. Optimization of control parameters for genetic algorithms. *IEEE Transactions on systems, man, and cybernetics*, IEEE, v. 16, n. 1, p. 122–128, 1986. Citado na página 56.
- GRIGORESCU, S. et al. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, Wiley Online Library, v. 37, n. 3, p. 362–386, 2020. Citado na página 17.
- HESSER, J.; MÄNNER, R. Towards an optimal mutation probability for genetic algorithms. In: SPRINGER. *Parallel Problem Solving from Nature: 1st Workshop, PPSN I Dortmund, FRG, October 1–3, 1990 Proceedings 1*. [S.l.], 1991. p. 23–32. Citado na página 56.
- HOLLAND, J. H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. [S.l.]: MIT press, 1992. Citado na página 29.
- HOLLAND, J. H. Genetic algorithms. *Scientific american*, JSTOR, v. 267, n. 1, p. 66–73, 1992. Citado na página 29.
- HOREL, J.-B. et al. Using formal conformance testing to generate scenarios for autonomous vehicles. In: IEEE. *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. [S.l.], 2022. p. 532–537. Citado 2 vezes nas páginas 15 e 25.
- KATO, S. et al. An open approach to autonomous vehicles. *IEEE Micro*, IEEE, v. 35, n. 6, p. 60–68, 2015. Citado na página 18.
- KAUR, P. et al. A survey on simulators for testing self-driving cars. In: IEEE. *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*. [S.l.], 2021. p. 62–70. Citado na página 14.

- KERRY, C. F.; KARSTEN, J. Gauging investment in self-driving cars. *Brookings Institution, October*, v. 16, 2017. Citado na página 17.
- KLÜCK, F. et al. Genetic algorithm-based test parameter optimization for adas system testing. In: IEEE. *2019 IEEE 19th International Conference on Software Quality, Reliability and Security (QRS)*. [S.l.], 2019. p. 418–425. Citado na página 28.
- KLÜCK, F. et al. Performance comparison of two search-based testing strategies for adas system validation. In: SPRINGER. *Testing Software and Systems: 31st IFIP WG 6.1 International Conference, ICTSS 2019, Paris, France, October 15–17, 2019, Proceedings 31*. [S.l.], 2019. p. 140–156. Citado na página 28.
- KNIGHT, J. C. Safety critical systems: Challenges and directions. In: *Proceedings of the 24th International Conference on Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2002. (ICSE '02), p. 547–550. ISBN 158113472X. Disponível em: <<https://doi.org/10.1145/581339.581406>>. Citado na página 21.
- KUMAR, M. et al. Genetic algorithm: Review and application. *Available at SSRN 3529843*, 2010. Citado 2 vezes nas páginas 29 e 30.
- KUUTTI, S. et al. A survey of deep learning applications to autonomous vehicle control. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 22, n. 2, p. 712–733, 2020. Citado 2 vezes nas páginas 17 e 21.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015. Citado na página 19.
- LI, G. et al. Av-fuzzer: Finding safety violations in autonomous driving systems. In: IEEE. *2020 IEEE 31st international symposium on software reliability engineering (ISSRE)*. [S.l.], 2020. p. 25–36. Citado na página 28.
- MOBILEYE. *Mobileye*. 2023. Disponível em: <<https://www.mobileye.com/>>. Acesso em: 15 out 2023. Citado na página 17.
- NASCIMENTO, A. M. et al. The role of virtual reality in autonomous vehicles' safety. In: *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. [S.l.: s.n.], 2019. p. 50–507. Citado na página 15.
- NASCIMENTO, A. M. et al. A systematic literature review about the impact of artificial intelligence on autonomous vehicle safety. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 21, n. 12, p. 4928–4946, 2019. Citado na página 15.
- NASCIMENTO, A. M. et al. A systematic literature review about the impact of artificial intelligence on autonomous vehicle safety. *IEEE Transactions on Intelligent Transportation Systems*, v. 21, n. 12, p. 4928–4946, 2020. Citado 3 vezes nas páginas 14, 21 e 25.
- NHTSA. *Automated Vehicles for Safety*. Disponível em: <<https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety>>. Acesso em: 12 nov 2023. Citado na página 48.
- NIMBLEBOX.AI. *Mimetização Neurônio*. 2022. Disponível em: <<https://blog.nimblebox.ai/hyperparameter-tuning-machine-learning>>. Citado 2 vezes nas páginas 7 e 20.



- NIU, H. et al. Dr2l: Surfacing corner cases to robustify autonomous driving via domain randomization reinforcement learning. In: *Proceedings of the 5th International Conference on Computer Science and Application Engineering*. [S.l.: s.n.], 2021. p. 1–8. Citado 2 vezes nas páginas 15 e 25.
- PAREKH, D. et al. A review on autonomous vehicles: Progress, methods and challenges. *Electronics*, MDPI, v. 11, n. 14, p. 2162, 2022. Citado na página 21.
- PIRRI, P. et al. Towards cooperative maneuvering simulation: Tools and architecture. In: . [S.l.: s.n.], 2021. p. 1–6. Citado 2 vezes nas páginas 7 e 36.
- RANJAN, S.; SENTHAMILARASU, S. *Applied Deep Learning and Computer Vision for Self-Driving Cars: Build autonomous vehicles using deep neural networks and behavior-cloning techniques*. [S.l.]: Packt Publishing Ltd, 2020. Citado 3 vezes nas páginas 7, 18 e 19.
- ROSIQUE, F. et al. A systematic review of perception system and simulators for autonomous vehicles research. *Sensors*, mdpi, v. 19, n. 3, p. 648, 2019. Citado 2 vezes nas páginas 14 e 32.
- ROSSUM, G. V.; DRAKE, F. L. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697. Citado na página 35.
- ROTHLAUF, F.; ROTHLAUF, F. *Representations for genetic and evolutionary algorithms*. [S.l.]: Springer, 2006. Citado na página 29.
- SAE levels of Driving Automation™ refined for clarity and international audience. 2023. Disponível em: <<https://www.sae.org/blog/sae-j3016-update>>. Citado na página 22.
- SANTOS, V. S. d. *Neurônio: O que É, tipos, função, estrutura*. Mundo Educação, 2020. Disponível em: <<https://mundoeducacao.uol.com.br/biologia/neuronios.htm>>. Citado 2 vezes nas páginas 7 e 20.
- SCENIC-TEAM. *GitHub Scenic 2.1.0 Repository*. Disponível em: <<https://github.com/BerkeleyLearnVerify/Scenic/tree/v2.1.0>>. Acesso em: 01 nov 2023. Citado 2 vezes nas páginas 43 e 46.
- SCENIC-TEAM. *Scenic-Documentation*. Disponível em: <<https://scenic-lang.readthedocs.io/en/latest/quickstart.html>>. Acesso em: 07 ago 2023. Citado 2 vezes nas páginas 38 e 47.
- SCHAFFER, J. D. et al. A study of control parameters affecting online performance of genetic algorithms for function optimization. In: *Proceedings of the 3rd international conference on genetic algorithms*. [S.l.: s.n.], 1989. p. 51–60. Citado na página 56.
- SCHWARTING, W.; ALONSO-MORA, J.; RUS, D. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, Annual Reviews, v. 1, p. 187–210, 2018. Citado na página 17.
- SHETTY, A. et al. Safety challenges for autonomous vehicles in the absence of connectivity. *Transportation research part C: emerging technologies*, Elsevier, v. 128, p. 103133, 2021. Citado na página 17.

- SHIMANUKI, G. *GitHub - scenic-2.1.0-mod*. 2023. Disponível em: <<https://github.com/GabrielShimanuki/scenic-2.1.0-mod>>. Acesso em: 20 jul 2023. Citado na página 47.
- SHIMANUKI, G. K. G. Análise da influência da zona de segurança sobre a eficácia do algoritmo de controle de cruzamento de via para veículos autônomos. *SIICUSP*, v. 1, n. 1, 2021. Citado na página 14.
- SHIMANUKI, G. K. G. Análise da influência da zona de segurança sobre a eficácia do algoritmo de controle lateral para conversão de via em veículos autônomos. *SIICUSP*, v. 1, n. 1, 2022. Citado na página 14.
- SONG, Q. et al. *Critical scenario identification for realistic testing of autonomous driving systems - software quality journal*. Springer US, 2022. Disponível em: <<https://link.springer.com/article/10.1007/s11219-022-09604-2>>. Citado na página 24.
- SPOTFIRE. *What is a neural network?* 2023. Disponível em: <<https://www.spotfire.com/glossary/what-is-a-neural-network>>. Citado 2 vezes nas páginas 7 e 21.
- SRINIVAS, M.; PATNAIK, L. M. Genetic algorithms: A survey. *computer*, IEEE, v. 27, n. 6, p. 17–26, 1994. Citado na página 30.
- SUN, H. et al. Corner case generation and analysis for safety assessment of autonomous vehicles. *Transportation research record*, SAGE Publications Sage CA: Los Angeles, CA, v. 2675, n. 11, p. 587–600, 2021. Citado 3 vezes nas páginas 16, 25 e 28.
- SWANSON, E. D. et al. *Statistics of light-vehicle pre-crash scenarios based on 2011–2015 national crash data*. [S.l.], 2019. Citado na página 48.
- TIAN, H. et al. Mosat: finding safety violations of autonomous driving systems using multi-objective genetic algorithm. In: *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. [S.l.: s.n.], 2022. p. 94–106. Citado na página 28.
- TUNCALI, C. E. et al. Requirements-driven test generation for autonomous vehicles with machine learning components. *IEEE Transactions on Intelligent Vehicles*, v. 5, n. 2, p. 265–280, 2020. Citado 3 vezes nas páginas 7, 14 e 25.
- WAYMO. *The Waymo Driver’s training regimen: How structured testing prepares our self-driving technology for the real world*. Waymo, 2020. Disponível em: <<https://waymo.com/blog/2020/09/the-waymo-drivers-training-regime.html>>. Citado na página 14.
- XU, W.; SOULY, N.; BRAHMA, P. P. Reliability of gan generated data to train and validate perception systems for autonomous vehicles. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. [S.l.: s.n.], 2021. p. 171–180. Citado 4 vezes nas páginas 14, 15, 25 e 28.
- YURTSEVER, E. et al. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, IEEE, v. 8, p. 58443–58469, 2020. Citado na página 21.
- ZHANG, X. et al. Finding critical scenarios for automated driving systems: A systematic mapping study. *IEEE Transactions on Software Engineering*, v. 49, n. 3, p. 991–1026, March 2023. ISSN 1939-3520. Citado na página 15.

---

ZHONG, Z.; KAISER, G.; RAY, B. Neural network guided evolutionary fuzzing for finding traffic violations of autonomous vehicles. *IEEE Transactions on Software Engineering*, IEEE, 2022. Citado na página [29](#).

# Apêndices

# APÊNDICE A – Alguns códigos do projeto

No Código 12, é mostrado um trecho da implementação dos sensores para a coleta de eventos de colisão.

```
1 class SensorColisao(object):
2     def __init__(self, ator_responsavel):
3         self.sensor = None
4         self.historico = []
5         self._responsavel = ator_responsavel
6         world = self._responsavel.get_world()
7         bp = world.get_blueprint_library().find(
8             ↪ 'sensor.other.collission')
9         self.sensor = world.spawn_actor(bp, carla.Transform(),
10            ↪ attach_to=self._responsavel)
11
12         weak_self = weakref.ref(self)
13         self.sensor.listen(lambda event: SensorColisao.
14            ↪ _na_colisao(weak_self, event))
15
16     def get_historico_colisoos(self):
17         historico = collections.defaultdict(int)
18         for frame, intensidade in self.historico:
19             historico[frame] += intensidade
20         return historico
21
22     @staticmethod
23     def _na_colisao(weak_self, event):
24         self = weak_self()
25         if not self:
26             return
27         tipo_ator = get_actor_display_name(event.other_actor)
28         impulso = event.normal_impulse
29         intensidade = math.sqrt(impulso.x**2 + impulso.y**2 +
30            ↪ impulso.z**2)
31         self.historico.append((event.frame, intensidade))
```

## Código 12 – Implementação da classe de sensor de colisão

O Código 13 ilustra um exemplo de código completo que faz a descrição do cenário de teste A.

```
1 #####
2 # MAPA E MODELO #
3 #####
4
5 param map = localPath('.././.././../assets/maps/CARLA/Town05.xodr')
6 param carla_map = 'Town05'
7 model scenic.simulators.carla.model
8
9 param render = 0
10 param record = localPath('~././scenic/test_GA_30_100_2/gen_1/logs/temp')
11
12 #####
13 # CONSTANTES #
14 #####
15
16 MODEL = 'vehicle.lincoln.mkz_2017'
17
18 EGO_INIT_DIST = [10, 13]
19 param EGO_SPEED = VerifaiRange(60, 80)
20 param EGO_BRAKE = VerifaiRange(0.5098506134136033, 0.5298506134136033)
21
22 MODEL_ADV = 'vehicle.mercedes.coupe_2020'
23
24 ADV_INIT_DIST = [12, 15]
25 param ADV_SPEED = VerifaiRange(18, 19)
26
27 param SAFETY_DIST = VerifaiRange(9, 10)
28 CRASH_DIST = 0
29 TERM_DIST = 70
30 TERM_TIME = 20
31
32 #####
33 # COMPORTAMENTO DOS AGENTES #
34 #####
35
36 behavior EgoBehavior(trajectory):
37     try:
```

```

38     do FollowTrajectoryBehavior(target_speed=globalParameters.EGO_SPEED,
39     ↪ trajectory=trajectory)
40 interrupt when withinDistanceToAnyObjs(self, globalParameters.SAFETY_DIST)
41 ↪ :
42     take SetBrakeAction(globalParameters.EGO_BRAKE)
43 interrupt when withinDistanceToAnyObjs(self, CRASH_DIST):
44     terminate
45
46 #####
47 # RELAÇÕES ESPACIAIS #
48 #####
49
50 intersection = Uniform(*filter(lambda i: i.is4Way, network.intersections))
51
52 egoInitLane = Uniform(*intersection.incomingLanes)
53 egoManeuver = Uniform(*filter(lambda m: m.type is ManeuverType.STRAIGHT,
54 ↪ egoInitLane.maneuvers))
55 egoTrajectory = [egoInitLane, egoManeuver.connectingLane, egoManeuver.endLane
56 ↪ ]
57 egoSpawnPt = OrientedPoint in egoInitLane.centerline
58
59 advInitLane = Uniform(*filter(lambda m:
60     m.type is ManeuverType.STRAIGHT,
61     egoManeuver.reverseManeuvers)
62     ).startLane
63 advManeuver = Uniform(*filter(lambda m: m.type is ManeuverType.LEFT_TURN,
64 ↪ advInitLane.maneuvers))
65 advTrajectory = [advInitLane, advManeuver.connectingLane, advManeuver.endLane
66 ↪ ]
67 advSpawnPt = OrientedPoint in advInitLane.centerline
68
69 #####
70 # ESPECIFICAÇÃO DO CENÁRIO #
71 #####
72
73 ego = Car at egoSpawnPt,
74     with blueprint MODEL,
75     with behavior EgoBehavior(egoTrajectory)
76
77 adversary = Car at advSpawnPt,
78     with blueprint MODEL,

```

```
73   with behavior FollowTrajectoryBehavior(target_speed=globalParameters.  
    ↪   ADV_SPEED, trajectory=advTrajectory)  
74  
75 require EGO_INIT_DIST[0] <= (distance to intersection) <= EGO_INIT_DIST[1]  
76 require ADV_INIT_DIST[0] <= (distance to intersection) <= ADV_INIT_DIST[1]  
77 terminate after TERM_TIME seconds  
78 % \end{mdframed}
```

Código 13 – Exemplo de um possível script de descrição do cenário A



## APÊNDICE B – Tabelas de Resultados do Estudo de Caso

As tabelas 12, 13, 14, 15, 16, 17 mostram a média do nível de risco em cada geração para os 6 cenários de teste realizados.

GEN	AG	ALEATÓRIO	DIFERENÇA	GANHO (%)
1	6,98 ± 8,38	6,20 ± 7,94	0,78	12,6
2	5,95 ± 8,36	7,27 ± 8,35	-1,32	-18,1
3	4,45 ± 7,82.	6,70 ± 8,42	-2,25	-33,6
4	3,35 ± 7,41***	7,59 ± 8,62	-4,24	-55,9
5	7,16 ± 8,42	7,96 ± 8,45	-0,79	-10,0
6	7,74 ± 8,58	6,54 ± 7,34	1,20	18,4
7	9,46 ± 8,80	8,06 ± 7,81	1,40	17,4
8	8,60 ± 8,93	6,85 ± 8,22	1,76	25,7
9	10,80 ± 8,54***	5,78 ± 7,64	5,02	86,9
10	8,58 ± 8,92	7,16 ± 8,26	1,41	19,8
11	10,13 ± 7,75**	6,35 ± 8,47	3,78	59,5
12	10,33 ± 8,42**	6,72 ± 7,78	3,61	53,7
13	11,46 ± 8,64**	7,33 ± 7,83	4,13	56,3
14	11,92 ± 7,94***	7,22 ± 8,30	4,70	65,1
15	11,73 ± 7,65***	7,06 ± 8,16	4,67	66,1
16	13,34 ± 6,59***	6,33 ± 7,92	7,01	110,7
17	13,56 ± 6,94***	7,63 ± 8,56	5,93	77,7
18	13,11 ± 7,11***	6,60 ± 7,53	6,51	98,7
19	12,33 ± 6,98***	6,59 ± 8,90	5,75	87,3
20	13,53 ± 7,47***	7,46 ± 7,96	6,07	81,4
21	12,66 ± 6,08***	6,92 ± 8,06	5,74	83,0
22	13,34 ± 6,81***	6,16 ± 8,23	7,17	116,4
23	12,73 ± 6,79***	6,98 ± 7,89	5,75	82,3
24	13,33 ± 7,06***	5,80 ± 7,74	7,53	129,9
25	13,38 ± 6,54***	7,49 ± 7,85	5,89	78,6
26	13,07 ± 7,17***	7,47 ± 7,87	5,59	74,8
27	12,73 ± 6,10***	8,32 ± 8,27	4,41	53,0
28	12,83 ± 6,78***	4,60 ± 7,11	8,23	179,0
29	13,90 ± 7,53***	7,27 ± 8,68	6,63	91,2
30	11,72 ± 6,62***	6,38 ± 7,87	5,34	83,7

Tabela 12 – Risco de condução - Cenário A

Nota: Os símbolos adotados nesta tabela indicam o nível de significância estatística. O símbolo "\*\*\*" indica um *p-value* menor ou igual a 0,001, "\*" denota um *p-value* menor ou igual a 0,01, "\*\*" denota um *p-value* menor ou igual a 0,05 e "." denota um *p-value* menor ou igual a 0,1.

GEN	AG	ALEATÓRIO	DIFERENÇA	GANHO (%)
1	4,78 ± 6,72	3,77 ± 5,98	1,01	26,8
2	2,62 ± 6,17*	4,71 ± 6,47	-2,09	-44,4
3	3,58 ± 6,94	4,86 ± 6,81	-1,28	-26,4
4	3,48 ± 6,20	3,52 ± 6,08	-0,04	-1,0
5	4,48 ± 6,86	3,76 ± 5,86	0,72	19,1
6	3,93 ± 6,64	4,31 ± 6,79	-0,38	-8,8
7	3,62 ± 6,35	3,41 ± 5,88	0,21	6,2
8	5,30 ± 6,81**	3,06 ± 4,94	2,24	73,1
9	2,66 ± 6,29	2,89 ± 5,30	-0,22	-7,8
10	3,77 ± 6,10	2,52 ± 5,19	1,25	49,6
11	4,25 ± 6,97	4,11 ± 6,92	0,14	3,4
12	4,96 ± 6,87	3,93 ± 6,18	1,03	26,3
13	4,37 ± 6,41	3,72 ± 6,27	0,65	17,4
14	4,49 ± 6,88	4,20 ± 6,72	0,29	7,0
15	5,71 ± 6,87***	2,64 ± 5,82	3,07	116,2
16	6,58 ± 8,08*	4,05 ± 5,80	2,53	62,5
17	6,60 ± 6,91*	4,29 ± 6,14	2,31	53,9
18	7,40 ± 6,53.	5,72 ± 7,33	1,69	29,5
19	6,66 ± 7,22**	3,74 ± 5,13	2,92	77,9
20	5,80 ± 6,65	4,47 ± 6,33	1,33	29,8
21	8,45 ± 7,11***	5,03 ± 7,00	3,42	68,1
22	6,96 ± 7,80.	4,90 ± 6,81	2,06	42,0
23	7,24 ± 6,27*	4,79 ± 7,06	2,46	51,3
24	7,29 ± 7,84*	4,81 ± 7,08	2,48	51,6
25	5,73 ± 5,89**	3,22 ± 5,94	2,51	77,9
26	8,02 ± 7,17***	3,85 ± 5,97	4,17	108,4
27	6,46 ± 7,14.	4,45 ± 7,33	2,01	45,2
28	7,26 ± 7,02***	3,55 ± 7,59	3,71	104,4
29	7,51 ± 6,31***	1,93 ± 5,16	5,58	289,0
30	7,12 ± 6,55**	4,61 ± 6,52	2,51	54,4

Tabela 13 – Risco de condução - Cenário B

Nota: Os símbolos adotados nesta tabela indicam o nível de significância estatística. O símbolo "\*\*\*\*" indica um *p-value* menor ou igual a 0,001, "\*\*\*" denota um *p-value* menor ou igual a 0,01, "\*\*" denota um *p-value* menor ou igual a 0,05 e "." denota um *p-value* menor ou igual a 0,1.

GEN	AG	ALEATÓRIO	DIFERENÇA	GANHO (%)
1	4,46 ± 6,69	5,92 ± 6,68	-1,46	-24,6
2	5,34 ± 7,38	5,12 ± 6,38	0,22	4,3
3	4,72 ± 7,31	4,89 ± 6,83	-0,17	-3,5
4	4,72 ± 6,66	5,66 ± 7,58	-0,94	-16,6
5	3,85 ± 5,89	4,09 ± 6,24	-0,24	-5,9
6	3,83 ± 7,02	4,88 ± 6,52	-1,05	-21,5
7	3,80 ± 6,43	5,35 ± 6,56	-1,55	-29,0
8	3,95 ± 6,63	4,97 ± 6,44	-1,02	-20,5
9	4,73 ± 6,48	5,09 ± 6,43	-0,35	-6,9
10	5,03 ± 7,03	5,48 ± 6,97	-0,45	-8,3
11	4,66 ± 6,36	5,49 ± 6,70	-0,84	-15,3
12	5,23 ± 6,23	5,38 ± 6,65	-0,15	-2,7
13	4,15 ± 6,92	5,00 ± 6,50	-0,85	-16,9
14	4,17 ± 6,26.	5,87 ± 6,66	-1,69	-28,9
15	5,36 ± 7,02	5,02 ± 7,64	0,34	6,8
16	5,78 ± 7,75	6,34 ± 7,33	-0,56	-8,9
17	4,89 ± 7,28	6,06 ± 7,25	-1,17	-19,4
18	6,52 ± 7,24	4,85 ± 6,86	1,67	34,5
19	7,80 ± 7,14**	5,02 ± 6,31	2,78	55,4
20	8,59 ± 7,38**	5,53 ± 6,88	3,06	55,2
21	8,51 ± 6,94**	5,48 ± 7,44	3,02	55,1
22	7,86 ± 6,54	6,34 ± 6,74	1,51	23,8
23	9,48 ± 7,45***	5,71 ± 6,68	3,77	65,9
24	8,94 ± 7,05**	6,00 ± 6,76	2,94	49,0
25	8,89 ± 6,68**	5,91 ± 7,13	2,98	50,4
26	9,84 ± 7,68***	6,08 ± 7,28	3,75	61,7
27	9,46 ± 7,72***	4,21 ± 5,97	5,25	124,7
28	8,77 ± 7,32***	4,37 ± 6,23	4,40	100,5
29	8,75 ± 7,08***	5,34 ± 6,93	3,41	64,0
30	8,93 ± 6,54***	3,95 ± 5,49	4,98	126,1

Tabela 14 – Risco de condução - Cenário C

Nota: Os símbolos adotados nesta tabela indicam o nível de significância estatística. O símbolo "\*\*\*\*" indica um *p-value* menor ou igual a 0,001, "\*\*\*" denota um *p-value* menor ou igual a 0,01, "\*\*" denota um *p-value* menor ou igual a 0,05 e "." denota um *p-value* menor ou igual a 0,1.

GEN	AG	ALEATÓRIO	DIFERENÇA	GANHO (%)
1	4,1 ± 5,82	4,03 ± 6,46	0,07	1,7
2	4,69 ± 6,20	5,52 ± 6,98	-0,83	-15,1
3	3,36 ± 5,88*	5,59 ± 6,98	-2,23	-39,9
4	5,39 ± 6,92	5,96 ± 7,00	-0,57	-9,6
5	3,92 ± 6,15	3,97 ± 5,91	-0,05	-1,3
6	4,27 ± 6,66	5,31 ± 6,60	-1,04	-19,6
7	4,83 ± 7,40	4,69 ± 6,21	0,14	2,9
8	5,41 ± 6,10	5,84 ± 6,84	-0,43	-7,3
9	6,55 ± 6,97.	4,96 ± 5,92	1,59	32,0
10	7,87 ± 7,63*	5,74 ± 7,30	2,13	37,1
11	7,69 ± 6,92***	4,53 ± 6,24	3,16	69,8
12	7,37 ± 6,94**	4,19 ± 6,47	3,18	75,9
13	7,72 ± 7,07**	4,56 ± 6,50	3,16	69,4
14	7,92 ± 7,96***	4,18 ± 6,44	3,74	89,4
15	7,84 ± 6,72**	5,21 ± 6,62	2,63	50,4
16	8,39 ± 6,22***	4,98 ± 6,36	3,41	68,4
17	8,40 ± 6,60***	5,21 ± 6,51	3,19	61,1
18	9,33 ± 6,05***	6,16 ± 6,94	3,17	51,5
19	9,03 ± 6,23***	4,53 ± 5,87	4,50	99,3
20	10,41 ± 6,25***	3,09 ± 5,48	7,32	236,5
21	9,98 ± 6,17***	4,84 ± 6,78	5,14	106,3
22	10,03 ± 6,90***	5,87 ± 6,42	4,16	70,9
23	9,91 ± 6,55***	4,17 ± 6,65	5,74	137,6
24	8,93 ± 6,21***	5,56 ± 6,93	3,37	60,6
25	9,40 ± 6,35***	4,56 ± 6,45	4,84	106,2
26	9,81 ± 5,73***	4,27 ± 6,32	5,54	129,7
27	9,22 ± 7,05***	3,18 ± 5,74	6,04	189,8
28	9,03 ± 6,42***	5,25 ± 7,38	3,78	71,9
29	8,66 ± 6,75***	5,13 ± 6,45	3,53	68,7
30	9,11 ± 6,86***	4,71 ± 6,45	4,40	93,3

Tabela 15 – Risco de condução - Cenário D

Nota: Os símbolos adotados nesta tabela indicam o nível de significância estatística. O símbolo "\*\*\*\*" indica um *p-value* menor ou igual a 0,001, "\*\*\*" denota um *p-value* menor ou igual a 0,01, "\*\*" denota um *p-value* menor ou igual a 0,05 e "." denota um *p-value* menor ou igual a 0,1.

GEN	AG	ALEATÓRIO	DIFERENÇA	GANHO (%)
1	5,23 ± 6,62	5,12 ± 6,78	0,11	2,1
2	2,10 ± 4,82*	4,01 ± 6,31	-1,91	-47,6
3	3,10 ± 5,66	4,19 ± 6,12	-1,09	-26,0
4	4,36 ± 6,80	3,84 ± 5,62	0,52	13,7
5	2,78 ± 5,46	3,34 ± 5,25	-0,56	-16,8
6	3,78 ± 7,01	4,33 ± 6,22	-0,55	-12,8
7	2,33 ± 6,03.	4,10 ± 6,78	-1,77	-43,1
8	2,61 ± 5,45.	4,04 ± 5,71	-1,43	-35,4
9	2,26 ± 5,58***	5,41 ± 6,51	-3,15	-58,3
10	1,80 ± 4,75***	4,60 ± 6,56	-2,79	-60,8
11	3,67 ± 5,89	3,95 ± 5,58	-0,28	-7,0
12	2,76 ± 5,96	3,91 ± 5,66	-1,15	-29,4
13	3,35 ± 6,44	2,70 ± 4,93	0,66	24,3
14	3,36 ± 6,68	4,37 ± 6,27	-1,01	-23,0
15	4,85 ± 7,53	3,85 ± 6,62	1,00	26,1
16	4,93 ± 7,07	5,84 ± 7,66	-0,91	-15,5
17	4,23 ± 6,92	4,63 ± 5,91	-0,40	-8,6
18	4,25 ± 6,65	4,42 ± 6,20	-0,17	-3,8
19	3,86 ± 5,77	4,44 ± 6,35	-0,58	-13,1
20	5,14 ± 7,13	4,34 ± 6,21	0,80	18,4
21	5,15 ± 6,47*	3,29 ± 5,63	1,86	56,6
22	4,56 ± 6,43	3,43 ± 6,19	1,13	33,0
23	6,40 ± 7,08*	4,17 ± 6,30	2,23	53,5
24	7,94 ± 7,54***	3,84 ± 5,64	4,10	106,9
25	8,03 ± 7,50***	3,15 ± 5,86	4,88	155,3
26	5,59 ± 6,39*	3,52 ± 6,00	2,07	58,8
27	7,52 ± 7,45***	2,82 ± 5,08	4,70	166,2
28	7,79 ± 7,41***	4,34 ± 5,84	3,45	79,6
29	7,55 ± 7,17***	3,50 ± 4,97	4,05	115,7
30	7,81 ± 7,08**	4,58 ± 6,72	3,23	70,6

Tabela 16 – Risco de condução - Cenário E

Nota: Os símbolos adotados nesta tabela indicam o nível de significância estatística. O símbolo "\*\*\*\*" indica um *p-value* menor ou igual a 0,001, "\*\*\*" denota um *p-value* menor ou igual a 0,01, "\*\*" denota um *p-value* menor ou igual a 0,05 e "." denota um *p-value* menor ou igual a 0,1.

GEN	AG	ALEATÓRIO	DIFERENÇA	GANHO (%)
1	6,06 ± 7,44	5,00 ± 7,62	1,06	21,2
2	4,38 ± 7,10**	7,43 ± 8,04	-3,05	-41,1
3	3,72 ± 6,71*	6,32 ± 8,19	-2,60	-41,2
4	5,67 ± 6,86	5,83 ± 7,50	-0,16	-2,8
5	5,17 ± 7,71	4,98 ± 7,60	0,19	3,9
6	4,72 ± 6,83	5,44 ± 6,96	-0,72	-13,2
7	3,94 ± 6,30.	5,79 ± 7,64	-1,85	-32,0
8	6,21 ± 7,84	6,14 ± 7,57	0,07	1,1
9	6,80 ± 7,59	5,89 ± 7,76	0,91	15,5
10	4,59 ± 6,19*	6,97 ± 7,80	-2,38	-34,2
11	6,04 ± 7,41	5,57 ± 7,92	0,47	8,5
12	5,19 ± 7,26	4,75 ± 7,33	0,45	9,4
13	5,22 ± 6,52	5,83 ± 6,93	-0,61	-10,4
14	6,57 ± 7,60	6,35 ± 6,77	0,22	3,4
15	4,94 ± 7,73	3,39 ± 7,01	1,55	45,5
16	6,35 ± 6,77	6,36 ± 7,91	-0,01	-0,2
17	7,36 ± 7,69	7,17 ± 8,26	0,19	2,7
18	7,86 ± 7,89	6,38 ± 7,57	1,48	23,2
19	8,18 ± 7,41	7,69 ± 7,50	0,49	6,4
20	7,54 ± 7,55	6,71 ± 7,05	0,83	12,3
21	7,25 ± 7,18***	3,78 ± 6,77	3,47	92,0
22	6,33 ± 6,93	7,08 ± 7,30	-0,75	-10,6
23	8,37 ± 7,15*	6,28 ± 7,50	2,09	33,2
24	7,56 ± 7,070	7,24 ± 7,61	0,32	4,3
25	9,70 ± 6,79***	5,16 ± 7,15	4,53	87,7
26	7,24 ± 7,85***	3,11 ± 7,17	4,13	132,6
27	6,51 ± 7,59	5,44 ± 6,95	1,07	19,6
28	9,63 ± 6,80***	5,65 ± 6,94	3,98	70,6
29	9,91 ± 6,87**	6,89 ± 7,72	3,02	43,9
30	10,35 ± 7,96***	4,69 ± 7,39	5,65	120,4

Tabela 17 – Risco de condução - Cenário F

Nota: Os símbolos adotados nesta tabela indicam o nível de significância estatística. O símbolo "\*\*\*" indica um *p-value* menor ou igual a 0,001, "\*\*" denota um *p-value* menor ou igual a 0,01, "\*" denota um *p-value* menor ou igual a 0,05 e "." denota um *p-value* menor ou igual a 0,1.

A Tabela 18 apresenta a contagem total do número de colisões para cada geração nos 6 cenários de teste. Já as Tabelas 19 e 20 mostram a distância média entre os dois veículos nos 6 cenários de teste. A Tabela 19 considera todos os cenários válidos, enquanto a Tabela 9 considera apenas simulações válidas em que não houve colisão. Por fim, a Tabela 18 apresenta a contagem total do número de simulações inválidas para cada geração nos 6 cenários de teste.

Geração	A		B		C		D		E		F	
	Aleatório	AG	Aleatório	AG	Aleatório	AG	Aleatório	AG	Aleatório	AG	Aleatório	AG
1	20	20	9	14	13	12	10	8	13	11	17	15
2	23	18	11	9	11	13	16	9	11	3	23	15
3	23	14	14	14	13	15	19	5	7	9	19	12
4	29	10	10	11	17	12	13	11	7	15	16	17
5	22	19	12	14	6	7	9	5	6	7	14	18
6	20	19	12	11	9	10	10	8	13	13	14	12
7	24	26	10	11	9	9	11	12	10	8	14	7
8	22	21	5	16	11	13	12	4	7	7	13	17
9	17	27	8	11	9	9	8	12	15	5	16	21
10	23	23	6	12	9	14	15	15	11	3	23	16
11	22	20	12	19	15	8	14	15	10	8	20	23
12	20	23	14	16	10	13	9	14	9	6	16	18
13	24	30	14	16	13	11	8	13	3	7	16	20
14	27	28	14	13	10	10	8	17	10	11	19	23
15	23	25	8	21	11	12	9	12	14	15	10	22
16	17	29	14	26	17	18	6	10	15	13	26	18
17	28	30	11	25	13	17	13	14	8	10	26	24
18	18	24	17	27	11	18	12	14	11	10	16	28
19	25	22	8	22	8	21	8	12	10	10	26	26
20	20	25	13	20	11	29	3	18	12	11	25	28
21	21	20	15	32	12	20	10	17	6	14	14	28
22	18	25	16	23	13	20	13	19	9	9	20	24
23	23	23	16	27	10	32	12	19	15	20	19	28
24	19	24	16	31	12	26	13	13	6	23	21	30
25	24	24	11	18	12	25	8	15	6	22	16	41
26	21	25	11	29	12	34	8	14	9	15	10	29
27	24	19	17	20	5	33	7	19	3	20	15	27
28	13	22	15	22	10	25	14	13	5	26	15	35
29	25	24	4	21	9	27	10	14	8	28	25	30
30	16	14	12	26	6	31	14	15	15	18	19	40
<b>Média</b>	$21,7 \pm 3,6$	$22,4 \pm 4,7$	$11,8 \pm 3,5$	$19,2 \pm 6,6$	$10,9 \pm 2,8$	$18,1 \pm 8,3$	$10,7 \pm 3,3$	$12,9 \pm 4,1$	$9,5 \pm 3,5$	$12,6 \pm 6,6$	$18,1 \pm 4,6$	$23,1 \pm 8$

Tabela 18 – Quantidade de colisões por geração

Geração	A		B		C		D		E		F	
	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA
1	1589,3	1375,6	1690,5	1912	1446,8	1759,8	2097,5	1852,1	1889,9	1910,6	2468,8	1792,6
2	1221,4	1242,9	1872,9	1729,1	1726,0	1393,5	1728,4	1377,7 *	2066,3	2315,1	2181,8	2011,2
3	1394,4	1214,4	1693,1	1369,9	1826,5	1474,5	1821,6	1547,2	1766,3	1968,6	2127,7	2651,9
4	1379,9	812,7 *	1894,7	1429,9	1448,5	1592	1499,8	1409,1	2150,4	1696,3	2708,8	1752,4
5	1197,1	818,7 *	2049,4	1170,6 ***	1852,0	1366,9	1835,6	1369,9 *	1903,7	1957,3	2048,9	1546,6
6	1500,8	734,7 ***	1701,8	1057 **	1519,7	1190,1	1573,1	1413,4	2017,4	1520,1 *	1990,9	1873,2
7	1422,4	674,7 ***	1853,8	1363,3 *	1626,0	980,3 **	1850,4	1403,2 *	2106,8	1726,9	2360,8	1735
8	1373,9	650,5 ***	1940,4	1312 **	1845,1	1296,8	1527,5	1420,8	1934,9	1673,4	1639,5	2388,5
9	1418,0	625,1 ***	2082,9	1292,5 **	1851,2	1134,1 **	1857,5	1465,3	1837,8	1627,4	1955,6	1955,6
10	1665,9	647,4 ***	1913,5	1395,9 *	1698,5	1010,9 **	1764,1	1079,3 **	1879,8	1511,8	2327,5	2281,4
11	1527,2	739,4 ***	1507,7	1345,8	1412,1	1161,7	2278,0	1341,1 ***	2071,6	1667,9	1763,1	1816,4
12	1539,7	643,3 ***	2046,2	1327,3 **	1377,8	1050	1735,7	1411,5	2434,2	1731,2 *	2034,8	1906,1
13	1158,3	704,4 ***	2101,1	1246,1 ***	1905,2	997,4 **	1834,2	1369,6 *	1988,4	1777,4	2746,6	2055,1
14	1578,1	680,4 ***	1832,7	1208 **	1533,1	1158,8	1875,8	1486,4	2158,0	1564,6 *	1710,3	1726,8
15	1506,0	778,7 ***	1886,7	1259,1 **	1963,5	831,2 ***	1673,4	1390,6	2363,8	1320,2 ***	1465,0	2147,6
16	1211,9	718,7 ***	1869,1	1317 *	1455,4	947,9 *	1627,4	1297,2	2056,7	1409,4 **	2415,0	2240,1
17	1404,6	774,3 ***	2022,5	1045,5 ***	1535,2	1266,9	1930,5	1480,5 *	1746,2	1645,3	1563,4	2177,9
18	1625,7	714,5 ***	1795,9	1245,6 **	1863,3	983,3 ***	1312,6	1124,9	2039,2	1376,7 **	2089,6	1377,4
19	1338,6	733,3 ***	1951,8	1259,6 **	1664,6	988,8 **	1915,7	1265,1 **	1809,6	1556,3	1629,7	1468,8
20	1097,3	650,5 ***	2229,6	1291,3 ***	1472,5	807,6 **	1924,4	1080 ***	1905,6	1254,3 ***	2394,0	1536,7
21	1500,4	776,3 ***	1762,1	1249,7 **	1502,2	982,5 *	1982,6	1202,3 ***	1957,4	1742,4	2353,4	1791,1
22	1612,4	685,6 ***	1738,5	1142,2 **	1479,1	1001,9 *	1593,0	1309,2	2171,0	1415,1 ***	2153,3	1862,4
23	1376,1	759,6 ***	1889,7	1107,3 ***	1735,1	1181,9 *	2032,9	1328,1 **	2288,2	1442,9 ***	2034,3	1566,4
24	1412,2	726,2 ***	1583,1	1321	1587,3	905,7 **	1787,7	1525,3	1943,9	1299,3 ***	1717,0	2304,9
25	1322,1	608,7 ***	1911,2	1340,6 *	1590,9	940,5 **	1533,6	1335,9	2079,3	1447,1 **	2108,2	1647,7
26	1272,0	618 ***	2066,7	1299,6 **	1761,5	1049,2 **	1825,4	1224,9 **	2320,1	1625,6 **	1973,2	1452,2
27	1136,4	731,5 **	1989,2	1089 ***	1662,9	1176,2 *	2217,5	1486,7 **	2089,3	1352,8 ***	2118,6	1642,5
28	1713,3	691,1 ***	1906,2	1174 **	1728,2	1069,8 *	1661,5	1739,6	2124,5	1462,2 **	2154,4	1586,5
29	1299,8	665,1 ***	1909,6	1268,9 *	1276,8	963,8	1906,5	1702,9	2329,7	1415,3 ***	1509,1	1311,9
30	1246,5	673,5 ***	1988,7	1156,8 ***	1926,8	872,8 ***	1543,3	1602,6	2060,7	1183,4 ***	1943,9	1512,1
Média	1401,4 ± 165,1	762,3 ± 184,6	1889,4 ± 158,8	1290,9 ± 176,3	1642,5 ± 185	1117,9 ± 224,7	1792,2 ± 216,9	1401,4 ± 178	2049,7 ± 177,4	1586,6 ± 244,6	2063,4 ± 333,3	1837,3 ± 329,7

Tabela 19 – Distância mínima média dos cenários válidos por geração

Nota: Os símbolos adotados nesta tabela indicam o nível de significância estatística. O símbolo "\*\*\*" indica um *p-value* menor ou igual a 0,001, "\*\*" denota um *p-value* menor ou igual a 0,01, "\*" denota um *p-value* menor ou igual a 0,05 e "." denota um *p-value* menor ou igual a 0,1.



Geração	A		B		C		D		E		F	
	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA
1	1866,5	1716,7	1885,3	2203,7	1521,2	1827,1	2270,7	1960,8	2017,5	1981	3129,9	2040,2
2	1458,1	1548,1	2108,0	1946,7	1842,0	1637,2	1872,8	1428,8 *	2178,0	2301,2	2693,0	2350,8
3	1715,5	1401,9	1983,5	1537,8	2066,2	1809,9	2022,5	1630,5	1841,7	2001,3	2617,3	3150
4	1789,9	909,9 *	2151,1	1535,3 *	1753,5	1717	1598,6	1537,8	2232,1	1923,2	3192,9	1867,8 .
5	1453,8	1008,9 .	2277,4	1420,5 ***	1997,6	1393,5 *	1911,2	1433,7 .	1971,4	1991,3	2439,0	1829,6
6	1636,8	859,7 ***	1958,5	1203,9 **	1678,0	1364,3	1648,7	1484,4	1998,6	1648,4	2187,0	2066,6
7	1632,6	700,9 ***	2041,0	1521,2 *	1792,5	1112,3 *	1943,7	1608,1	2318,0	1894	2804,2	1804,5
8	1624,3	716 ***	2039,4	1529 *	2042,9	1409,5 .	1654,8	1479,3	1944,6	1609,6	1894,4	2805,4
9	1582,1	741,9 ***	2245,7	1510,6 *	2009,7	1209,2 **	1924,7	1592,5	1903,2	1736,5	2547,4	2425,5
10	2029,7	747,8 ***	2068,6	1503,9 *	1871,3	1101,5 **	1970,2	1224 **	2031,2	1527,7 *	2086,8	2552,1
11	1880,0	833,6 ***	1790,5	1588	1573,7	1280,5	2455,6	1416,9 ***	2014,5	1708,3	2218,5	2282,9
12	1745,7	742,8 ***	2310,5	1513,5 **	1460,9	1154,5	1837,2	1558,1	2387,2	1859,3 .	2454,5	2152
13	1291,9	835,8 **	2306,4	1427,8 ***	2118,2	1168,7 **	1958,2	1520,2 .	2010,9	1904,3	3268,2	2356,6
14	1913,5	728,2 ***	2130,5	1361,3 ***	1677,6	1245,1 .	1979,4	1748,2	2292,6	1663,5 *	1742,1	2024,1
15	1843,0	837,7 ***	2068,1	1424,2 *	2310,4	931,1 ***	1765,1	1510,6	2651,3	1460,6 ***	1650,7	2706,2
16	1358,5	776,5 ***	2095,9	1623,3 .	1747,2	1152,8 *	1706,0	1379	2328,9	1507,5 ***	3108,7	2459,1
17	1747,9	943,1 ***	2279,7	1205,6 ***	1743,9	1590,2	2163,8	1591,7 *	1760,5	1796,9	1985,4	2573,7
18	1810,0	829,3 ***	2133,9	1481,2 **	2118,6	1158,6 ***	1378,8	1224,3	2099,4	1492,6 **	2475,9	1595,7
19	1811,2	825,4 ***	2084,7	1456,2 **	1821,7	1196,5 *	2025,7	1362,1 **	1915,6	1560	1771,5	1645
20	1270,3	750,8 **	2570,8	1439,3 ***	1609,3	960,6 *	1986,7	1185,9 ***	2017,7	1341,2 ***	3082,5	1645,9 *
21	1814,3	844,9 ***	2016,7	1489,5 *	1682,3	1114,8 *	2179,5	1341,9 ***	2064,8	1826,9	2664,8	1949,3
22	1899,6	718,4 ***	2022,8	1307,6 **	1623,9	1166,1 .	1699,4	1481,8	2373,9	1460,7 ***	2525,8	2015,2
23	1552,2	822,9 ***	2308,3	1341,1 ***	1879,4	1533	2228,9	1501 **	2361,6	1494,4 **	2389,3	1701,4
24	1584,5	830,5 **	1883,0	1586	1785,0	1058,8 **	1927,6	1622,2	2027,0	1454,2 **	2129,4	2842,8
25	1504,8	695,2 ***	2146,7	1510,4 *	1768,8	1048,9 **	1611,3	1498,8	2239,0	1649,7 *	2507,0	1857,4
26	1450,8	713,7 ***	2344,6	1546,3 **	2007,7	1368,4 *	1987,4	1274,8 ***	2498,1	1684,7 **	2487,4	1761,4
27	1312,4	793,9 *	2393,5	1257,1 ***	1740,8	1488,4	2289,7	1715 *	2153,0	1517,9 **	2238,0	1879,9
28	1920,9	781 ***	2353,6	1357,2 **	1913,4	1282,2 *	1870,4	1906,8	2213,0	1640,5 *	2414,1	1881,1
29	1668,0	853,9 ***	2083,1	1466,5 *	1400,5	1133,9	2053,7	1912,9	2292,3	1467,8 ***	1879,5	1462,8
30	1413,1	735,1 ***	2208,6	1316,5 ***	2020,1	1062 ***	1722,5	1836,9	2239,3	1273,1 ***	2371,8	1802,4
Média	1652,7 ± 211	874,8 ± 245,6	2143 ± 171,8	1487 ± 196,5	1819,3 ± 212,6	1289,2 ± 243	1921,5 ± 237,1	1532,3 ± 201,9	2145,9 ± 205,3	1679,3 ± 231	2451,9 ± 438,8	2116,2 ± 422,9

Tabela 20 – Distância mínima média dos cenários válidos que não ocorrem colisão por geração

Nota: Os símbolos adotados nesta tabela indicam o nível de significância estatística. O símbolo "\*\*\*" indica um *p-value* menor ou igual a 0,001, "\*" denota um *p-value* menor ou igual a 0,01, "\*\*" denota um *p-value* menor ou igual a 0,05 e "." denota um *p-value* menor ou igual a 0,1.

Geração	A		B		C		D		E		F	
	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA	Aleatório	GA
1	22	28	27	19	26	30	33	34	23	24	43	35
2	29	40	20	52	26	36	22	38	28	49	27	48
3	30	51	27	49	31	41	29	47	25	39	34	52
4	28	61	33	44	30	29	26	36	25	42	34	38
5	24	36	27	40	31	42	33	44	32	40	42	45
6	21	34	36	46	30	48	30	44	26	48	28	40
7	16	27	32	38	28	47	31	45	31	60	34	52
8	31	33	23	25	26	46	25	30	29	51	36	36
9	30	21	33	54	24	36	29	26	22	58	36	31
10	21	32	45	36	24	36	30	26	23	59	27	40
11	31	15	35	38	26	35	27	21	26	37	42	32
12	19	18	23	30	27	26	32	19	24	47	41	41
13	27	16	22	28	25	43	32	20	31	53	25	32
14	19	12	31	34	20	38	38	23	28	51	28	35
15	17	9	48	17	35	33	31	18	22	42	63	48
16	32	2	24	23	26	38	31	15	22	33	33	28
17	26	1	21	18	22	35	26	12	22	43	31	26
18	20	4	19	5	24	26	27	10	22	41	32	30
19	35	4	22	9	24	16	25	10	26	35	22	22
20	26	4	25	10	24	8	42	9	25	36	24	24
21	25	0	26	4	31	6	30	4	32	23	52	24
22	28	2	22	12	22	6	28	9	29	33	20	31
23	21	3	25	4	22	3	33	8	31	19	31	17
24	37	2	27	7	18	5	26	10	26	14	26	14
25	20	2	36	6	22	0	38	8	31	9	41	13
26	23	3	24	1	17	5	34	7	31	24	63	34
27	23	1	21	5	26	3	38	10	35	9	32	27
28	34	2	47	6	29	2	35	5	24	12	30	9
29	25	3	58	2	31	4	28	8	22	9	29	8
30	31	2	26	3	31	2	35	8	25	7	47	10
<b>Média</b>	$25,7 \pm 5,5$	$15,6 \pm 16,9$	$29,5 \pm 9,5$	$22,2 \pm 17,2$	$25,9 \pm 4,2$	$24,2 \pm 16,9$	$30,8 \pm 4,6$	$20,1 \pm 13,7$	$26,6 \pm 3,8$	$34,9 \pm 16,3$	$35,3 \pm 10,8$	$30,7 \pm 12,5$

Tabela 21 – Quantidade de simulações inválidas por geração

## APÊNDICE C – Exemplos de Simulações Geradas

As imagens da Figura 25 são exemplos de simulações dos cenários A e B gerados pela automação. Em todas as figuras, cada coluna representa um cenário.



(a) A - 1



(b) B - 1



(c) A - 2



(d) B - 2



(e) A - 3



(f) B - 3

Figura 25 – Coluna esquerda - Cenário A (Colisão) / Coluna direita - Cenário B (Colisão)

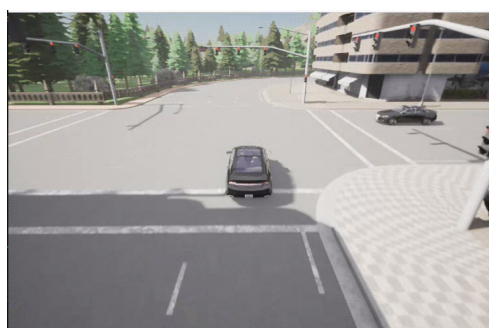
Já as imagens da Figura 26 são exemplos de simulações dos cenários A e B gerados pela automação. Em todas as figuras, cada coluna representa um cenário.



(a) C - 1



(b) D - 1



(c) C - 2



(d) D - 2



(e) C - 3



(f) D - 3

Figura 26 – Coluna esquerda - Cenário C (Colisão) / Coluna direita - Cenário D (Quase colisão)

Por fim, as imagens da Figura 27 são exemplos de simulações dos cenários A e B gerados pela automação. Em todas as figuras, cada coluna representa um cenário.



(a) E - 1



(b) F - 1



(c) E - 2



(d) F - 2



(e) E - 3



(f) F - 3

Figura 27 – Coluna esquerda - Cenário E (Seguro) / Coluna direita - Cenário F