



PUC Minas

# A B A K Ó S

Instituto de Ciências Exatas e Informática



Licença Creative Commons Attribution 4.0 International

## Aprendizado por Reforço em Veículos Autônomos para Detecção de Faixas em Condições de Baixa Visibilidade\*

Reinforcement Learning in Autonomous Vehicles for Lane End Detection in Low Visibility Conditions

Ana Vitória Menezes Vieira Barbosa<sup>1</sup>

Arthur Vinícius Firmiano Pereira<sup>2</sup>

Fernando Luiz Lessa de Araujo<sup>3</sup>

Gleydiston Braganca Martins Dos Santos<sup>4</sup>

Pedro Augusto Vaz De Souza E Silva<sup>5</sup>

Felipe Augusto Lara Soares<sup>6</sup>

### Resumo

Para que haja difusão dos veículos autônomos, é necessário que eles sejam capazes de tomar decisões em cenários inesperados de forma rápida e assertiva e, por isso, é importante que o passageiro possa confiar na inteligência artificial para tomada de decisões, visto que muitas vezes o poder computacional ultrapassa o tempo de reação de quem está dirigindo. Considerando os escopos adversos que dependem da parte sensorial do veículo, neste artigo é trabalhado o cenário de baixa visibilidade, dado principalmente por chuvas fortes, lombadas e nevadas. Utilizando o simulador CARLA junto ao *framework* ANTI CARLA para emular esses cenários, recorrendo aos sensores embutidos no programa e algoritmos de aprendizado por reforço (do inglês, *reinforcement learning*), foi desenvolvida uma solução para que o veículo autônomo verifique faixas na estrada, condição que se torna precária em cenários de baixa visibilidade. Os experimentos iniciais trouxeram resultados positivos com boas taxas de sucesso nos casos de teste, dados os conceitos descritos.

**Palavras-chave:** Veículos Autônomos. CARLA. Simulação. Inteligência Artificial.

\*Artigo apresentado à Revista Abakos

<sup>1</sup>E-mail: anavick.sk@gmail.com

Graduação em Engenharia de Computação da PUC Minas, Brasil

<sup>2</sup>E-mail: arthrvinicioiusbarbosa@gmail.com

Graduação em Engenharia de Computação da PUC Minas, Brasil

<sup>3</sup>E-mail: feluizlessa@gmail.com

Graduação em Engenharia de Computação da PUC Minas, Brasil

<sup>4</sup>E-mail: gleydiston35@gmail.com

Graduação em Engenharia de Computação da PUC Minas, Brasil

<sup>5</sup>E-mail: pavssilva@sga.pucminas.br

Graduação em Engenharia de Computação da PUC Minas, Brasil

<sup>6</sup>OrientadorE-mail: felipesoares@pucminas.br

Engenharia de Computação, Brasil

## Abstract

For the best adoption of autonomous vehicles, it seems necessary for them to be capable of handling quick and assertive decisions in unexpected scenarios, and for it, is important that the driver can trust in artificial intelligence for decision making, as the computational power surpasses the driver reaction time. Considering the unfavorable scopes that depends on the sensorial part of the vehicle, this article works in the low visibility, mainly given by storms, speed bumps and snow storms. Utilizing CARLA simulator with the ANTI CARLA framework to emulate this scenarios, recurring to embued sensors at the software and reinforcement learning algorithms, we developed a solution that makes the autonomous vehicle verify road strips, which becomes precarious in low visibility scenarios. The initial experiments brung positive results with good success rates in test cases, given the described concepts.

**Keywords:** Autonomous vehicles. CARLA. Simulation. Artificial Intelligence.

## 1 INTRODUÇÃO

Al-Ogaili et al. (2019), ao analisarem a literatura, concluíram que os veículos autônomos são uma ferramenta bastante útil para a evolução das cidades modernas e podem trazer benefícios e melhorias para toda a humanidade. Diminuição de colisões e acidentes arriscados causados por imprudência humana, redução das filas quilométricas de trânsito nas cidades e aumento na eficiência energética dos veículos, e consequentemente diminuição de gases de efeito estufa, são pontos importantes com os quais os carros autônomos podem contribuir para um futuro sustentável.

De acordo com Detran-BA (2023), estima-se que 90% dos acidentes de trânsito no Brasil são causados por falhas humanas, consequentemente, a introdução dos veículos autônomos poderia evitar pelo menos 585 mil mortes na década que começa em 2035 (REDMON; FARHADI, 2018). Isso direciona a evolução dos veículos *self driving* a uma necessidade de tomada de decisão para qualquer condição ou risco que o trânsito possa oferecer. Como relatado por Mehra et al. (2021), existe uma dificuldade associada ao tratamento de condições climáticas adversas; por isso, faz-se necessário contribuições mais robustas em relação ao comportamento do veículo nesse tipo de situação.

De acordo com Bonon (2019), um carro autônomo pode ser descrito como a combinação de tecnologias como Internet das Coisas, Inteligência Artificial (IA) e Automação. Para a criação de um carro autônomo, é imprescindível a realização de treinamentos, testes e validações. Levando em consideração os riscos que um carro pode causar no trânsito, é fundamental a utilização de softwares que possam simular um ambiente controlável com alto grau de realidade.

O software de simulação deve ser capaz de entregar todas as informações necessárias para que a Inteligência Artificial do veículo possa analisá-las e tratá-las como em um ambiente real. Sendo assim, o simulador de direção autônoma Car Learning to Act (CARLA) (DOSO-VITSKIY et al., 2017) se mostra promissor em quesito compatibilidade, facilitando a integra-

ção com outros softwares e permitindo um estudo mais aprofundado sobre as várias ocasiões no trânsito.

Existem várias situações adversas no trânsito, como acidentes, chuvas fortes, direção noturna, neve, entre outras. Tendo isso em vista, é fundamental que a simulação aborde uma variedade de cenários, como por exemplo o trabalho realizado pelos autores Ramakrishna et al. (2022b), que abordaram uma extensão para o simulador CARLA capaz de auxiliar na construção de cenários como esses. Também é ressaltado a importância dos testes e validação desses casos, destacando a crucialidade dos mesmos antes da aplicação no mundo real e a extrema complexidade para a resolução desses problemas através da computação visual.

Diante desse contexto, com o intuito de avaliar o desempenho de um carro autônomo em situações adversas, este trabalho propõe uma simulação controlada e auxiliada por uma Inteligência Artificial no simulador CARLA. Desta forma, o trabalho abordou o treinamento de uma IA que reaja às situações adversas em um veículo autônomo, como: evitar colisões com buracos e manter-se no limite da faixa durante chuva pesada, através da leitura dos sensores disponíveis no carro, e implementações de software utilizando *reinforcement learning*.

O restante do artigo está organizado da seguinte forma: na Seção 2, foram apresentados conceitos cruciais utilizados como base, tal como definição de veículos autônomos e o CARLA; na Seção 3, são apresentados os trabalhos correlatos; na Seção 4, é descrito a metodologia utilizada no processo, tal como os materiais usados, e o método de trabalho; na Seção 5, são descritos e demonstrados os resultados iniciais, vide a reprodução de cenários adversos e os resultados obtidos com a utilização de um modelo *Learning by Cheating* (LBC) pra uma simulação; na Seção 6, é apresentado o cronograma de tarefas que foram planejadas para o segundo semestre de 2024.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Veículos autônomos

De acordo com Michaelis (2024), autônomo diz-se do que possui capacidade de governar a si próprio, direito de se administrar livremente; no ramo tecnológico, é descrito por um sistema ou equipamento que pode manter seu funcionamento sem necessitar da ação de um agente externo. Logo, é possível assumir que um carro autônomo é um automóvel com habilidade de tomar suas próprias decisões sem a necessidade de intervenção direta do ser humano.

SAE International (2018) complementa o significado de um veículo autônomo (VA) sendo um veículo que, através de uma série de sensores, tem a capacidade de tomar decisões lógicas programáveis. Contudo, um VA pode variar em diferentes níveis de autonomia, desde tomar decisões sutis até adotar abordagens agressivas suficientes para minimizar a intervenção humana.

Levando em consideração essas diversas atuações de autônomos dentro do trânsito, fo-

ram padronizados 5 níveis de autonomia para um carro descrito na Tabela 1. Essa classificação é baseada na capacidade do automóvel de realizar tarefas operacionais, como controle lateral e longitudinal, monitoramento do ambiente, planejamento e execução de respostas a mudanças do ambiente, realização de manobras, denominadas Tarefas de Direção Dinâmica (DDT, do inglês *Dynamic Driving Task*).

**Tabela 1 – Níveis de Autonomia**

Nível	Nome	Descrição
0	Nenhuma automação	A direção é mantida pelo motorista sempre que esses recursos de suporte estão ativados, mesmo que os pés estejam fora dos pedais e o motorista não esteja controlando manualmente a aceleração ou a frenagem.
1	Assistência ao motorista	Execução de ação específica pelo sistema de automação que exerce ou controle lateral ou controle longitudinal do veículo, enquanto o motorista assume o controle da outra direção (mas não ambos simultaneamente), com a expectativa de que o motorista realize o restante do DDT.
2	Automação parcial da direção	Execução, pelo sistema de automação, do controle lateral e longitudinal do veículo com a expectativa de que o motorista faça apenas a etapa final da ação ou fique como supervisor das ações do veículo
3	Automação condicional da direção	Execução automatizada da completude de uma ação com controle lateral e longitudinal do veículo, ativada pelo motorista com a expectativa de que o motorista seja apenas supervisor e esteja pronto para assumir o comando em caso de erro, risco à segurança ou quando solicitado
4	Automação da direção	Execução completamente automatizada do veículo, ativada pelo motorista, sem a expectativa de que este assuma o controle em algum momento mas permitindo a intervenção
5	Automação completa da direção	Execução completamente automatizada do veículo a todos os momentos, sem a expectativa de que o motorista assuma o controle em algum momento

**Fonte:** SAE International (2018)

Cada decisão que o autônomo realiza é baseado em um conjunto de informações de entrada capturadas por instrumentos de medição denominados sensores. Um VA pode conter vários sensores com diferentes finalidades, como, por exemplo: câmeras, sensores de distância, GPS, LIDAR, entre outros. Varghese et al. (2015) classifica os sensores em dois tipos, sensores internos e externos. Os sensores internos têm como finalidade capturar as informações do veí-

culo como rotação, velocidade, inclinação do veículo e outros. Já os externos são usados para receber informações externas e de diferentes aspectos como imagens, distância entre objetos, umidade, nível de brilho e localização.

## 2.2 CARLA

O Car Learning to Act (CARLA) é um simulador *open-source* que foi criado para auxiliar o desenvolvimento e treinamento de veículos autônomos. Segundo Dosovitskiy et al. (2017), o simulador fornece diversos recursos como:

- a) Servidor multi cliente que possibilita a vários clientes acessar o mesmo nó e controlar atores diferentes.
- b) API que permite aos desenvolvedores controlar todos os aspectos relacionados à simulação, incluindo geração de tráfego, comportamento de pedestres, clima, sensores e muito mais.
- c) Conjunto de sensores de direção autônoma permitindo aos usuários configurar diversos sensores, incluindo LIDARs (do inglês *light detection and ranging*), múltiplas câmeras, sensores de profundidade e GPS, entre outros.
- d) Entidades que podem interagir com os veículos da simulação como pedestres, veículos, semáforos entre outros.
- e) Gerenciador de tráfego para controlar os personagens não-jogáveis (NPC, do inglês *non-playable character*) dentro do ambiente de reprodução.

Tendo o cliente em Python, uma das linguagens mais populares em todo o mundo, o CARLA conta com uma comunidade ativa, que desenvolve novos objetos, veículos, sensores e novas bibliotecas, o que torna uma plataforma atualizada pelas adições da comunidade.

O CARLA também provê objetos como construções, estradas, veículos e condições climáticas para construção de cenários, a fim de reproduzir variados escopos do uso de um veículo autônomo. Todos estes recursos tornam o programa uma ótima e completa opção se tratando de simuladores de veículos autônomos.

## 2.3 Learning by Cheating

A direção de veículos autônomos baseada em visão ainda é difícil, pois o sistema precisa aprender a perceber o mundo e agir nele no mesmo momento. Por isso, o *Learning by Cheating* é um método de aprendizagem de máquina que simplifica o processo decompondo em duas etapas, usando a lógica de aluno e professor. Primeiro, existe um agente que tem acesso a

informações do layout real do ambiente e as posições de todos os participantes do tráfego. Na segunda etapa, o agente privilegiado atua como um professor que treina um agente puramente *sensorimotor* baseado em visão, que recebe informações sensoriais e toma decisões tendo-as como base, seu objetivo é criar uma política que imite as ações do professor.

De acordo com Chen et al. (2019a), o *Learning by Cheating* é uma promissora técnica que tem como vantagens: o agente privilegiado generaliza melhor, ou seja, pode aplicar o aprendizado a mais situações, a supervisão do agente privilegiado para o agente *sensorimotor* é mais forte, e o agente privilegiado fornece respostas em tempo real durante o treinamento.

Em relação a estas vantagens, ele generaliza melhor porque atua sobre uma representação muito compacta, ou seja, ele trabalha com informações simplificadas e essenciais. Isso permite que ele foque em decisões de alto nível sem ser sobrecarregado por dados desnecessários ou ruído visual. Dessa forma, o aprendizado é mais eficiente e menos suscetível a variações, o que o ajuda a adaptar-se melhor a diferentes cenários durante o treinamento.

A supervisão é mais eficaz porque ocorre em várias etapas do processo de tomada de decisão, não apenas no resultado final. Em vez de corrigir apenas o comportamento final, o agente privilegiado supervisiona as representações intermediárias — as etapas de raciocínio que o agente *sensorimotor* usa para chegar à decisão. Isso resulta em um aprendizado mais detalhado e orientado, ajudando o aluno a compreender melhor como processar informações visuais e tomar decisões precisas, o que melhora seu desempenho geral.

Sobre as respostas em tempo real, como são usados apenas pares de mapas e imagens, o agente *sensorimotor* pode ser treinado de forma mais eficaz. Essa abordagem permite questionarativamente o agente privilegiado durante o treinamento para ajustar as políticas de ação. Isso resulta em um processo de aprendizado mais dinâmico que melhora com base em consultas diretas ao agente privilegiado.

Um detalhe de implementação sobre o processo é que o agente privilegiado recebe as informações do ambiente e produz um mapa de calor que passa por uma camada contida na Rede Neural Convolucional (CNN, do inglês *Convolutional Neural Network*) com a função *softmax* (SA), definindo pontos de referências para todos os comandos, também melhorando o desempenho do processo.

### 3 TRABALHOS CORRELATOS

Existe uma necessidade de segurança, confiabilidade e eficiência para que os veículos autônomos sejam difundidos ao redor do mundo. Em consequência, surgem estudos e melhorias relacionados ao que hoje pode ser um dos maiores empecilhos à adesão desta tecnologia, as situações adversas de trânsito.

Elallid et al. (2023) realizaram estudos e ações que apresentam uma abordagem para controlar um veículo autônomo (VA) em um cenário complexo com tráfego denso, envolvendo pedestres, ciclistas e outros veículos. Utilizando uma estrutura modificada do *Deep Q-Network*,

um método difundido de Aprendizagem por Reforço, validaram com sucesso um cenário em que o veículo precisa se locomover através de um cruzamento para chegar ao seu destino final, considerando gastar o tempo mínimo de viagem, e sem colisões. O método proposto foi validado utilizando o simulador CARLA, e os resultados demonstraram a eficiência do modelo em termos de aprendizagem, estabilidade e rapidez.

Os sensores instalados em veículos autônomos podem ser facilmente afetados por bloqueios (por exemplo, chuva, neve, poeira, neblina e outros) cobrindo a superfície deles. No artigo Jeon et al. (2022), foi desenvolvido o quadro de precisão baseado no simulador CARLA e na avaliação do desempenho de detecção de pista sob bloqueio de sensores por chuva forte. Como resultado da análise, descobriu-se que o desempenho da detecção de faixas seria degradado mais severamente nas faixas curvas do que nas faixas retas. Em conclusão ao trabalho, foi proposta uma equação de taxa crítica de precipitação que causaria falhas de segurança, estabelecida com base em conjuntos de dados experimentais de chuva.

Como citado no artigo Sural et al. (2023), os carros autônomos devem ser testados nos mais diversos cenários, além disso, testá-los no mundo real é custoso em termos de tempo, esforço e recurso. Em especial, operar em condições adversas, como más condições climáticas e áreas em obras inacessíveis em uma aplicação real. O co-simulation é um ambiente chamado CoSim que cria uma interface para veículos autônomos utilizando *CARLA Simulator* que possibilita a simulação destes cenários.

Através da estratégia feita por Muzahid et al. (2022), o processo de decisão de Markov integrado com *Deep Reinforcement Learning-Based* tem como objetivo evitar cadeia de colisões de veículos, diminuir a severidade e aumentar a segurança para veículos autônomos. A solução desde desafio é impraticável apenas pelo uso de sensores. Logo, é necessário utilizar uma estratégia de tomada de decisão baseada em Aprendizagem por Reforço e analisar a eficiência de segurança dos métodos existentes na condução da segurança. Os resultados demonstraram eficiência na prevenção de colisão de múltiplos veículos.

O artigo Manikandan et al. (2023) propõe um algoritmo utilizando *Deep Reinforcement Learning* para definir tomadas de decisão de um carro autônomo ao desviar de obstáculos. Os autores compararam resultados de três tipos diferentes de algoritmos de *Deep Learning*: *Deep Q-Network*, *Double Deep Q-Network* e *Dueling Double Deep-Q Network*, aplicados dentro de um cenário simulado no CARLA, utilizando dos sensores disponibilizados pelo software. No final, optaram por *Dueling Double Deep-Q Network*, na qual apresentou o menor número de resultados negativos dentre os algoritmos testados.

Similarmente, no artigo Kuan et al. (2020), os autores também testaram a aplicação de um algoritmo *Deep Q-Learning* em simulações ad-hoc no CARLA para detectar e desviar de buracos em estradas, respeitando regras de eficiência energética e velocidade de execução para o funcionamento saudável dos sensores em uma implementação real. Na conclusão, decidiram manter apenas o sistema de detecção de buracos, pois os resultados do sistema de desvio não foi satisfatório, apresentando resultados percentuais abaixo da taxa de sucesso em cenários na quais pilotos humanos conseguiriam desviar.

## 4 METODOLOGIA

Nesta seção, são explorados os materiais e métodos fundamentais utilizados para a implementação de um veículo autônomo enquadrado no nível 3 descrito na Tabela 1 e análise dos resultados.

### 4.1 Materiais

Foi utilizado a linguagem de programação *Python* para a implementação de uma Inteligência Artificial (IA) com algoritmo *Deep Reinforcement Learning* que controla o veículo com base nas APIs disponibilizadas pelo simulador CARLA.

No CARLA foram utilizados os mapas das cidades disponíveis no simulador como ambiente de teste, no qual contém edifícios, semáforos, tráfego de pedestres e veículos, entre outros atores que ajudam na simulação e desenvolvimento de um veículo autônomo, como na Figura 1.

**Figura 1 – Mapa de cidade disponível no CARLA**



**Fonte:** Elaborado pelos autores

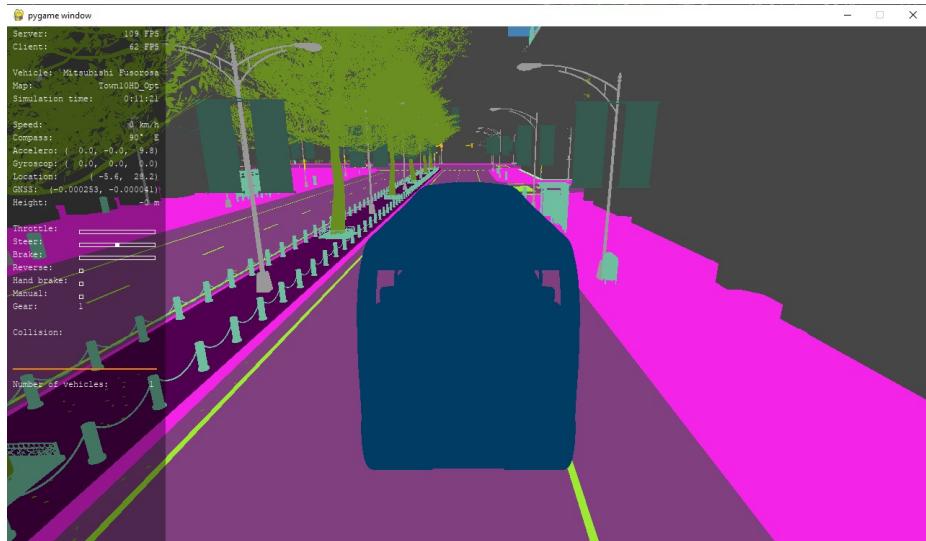
As imagens utilizadas no treinamento da IA foram obtidas pelo *sensor camera rgb* fornecido pelo CARLA, que funciona como uma câmera normal capturando as fotos da cena onde o veículo está localizado. A velocidade em que o sensor captura os dados pode ser definida pelo *sensor tick*, sendo que foi adotado um padrão com o atraso mínimo permitido para obter os dados o mais rápido possível.

Utilizando o *sensor lidar ray cast* que simula um sensor LIDAR rotativo de projeção de raios de luz, é possível gerar uma nuvem de pontos da superfície externa de um objeto que, ao final, pode ser utilizado para detectar a distância que o objeto se encontra.

O sensor *camera semantic segmentation* obtém imagens classificando cada objeto com

uma cor de acordo com a etiqueta que o elemento recebeu no início da simulação e assim ajudando a definir melhor o que é cada objeto do cenário onde o veículo está localizado, como pode ser observado na Figura 2.

**Figura 2 – Visão do sensor de segmentação semântica no CARLA**



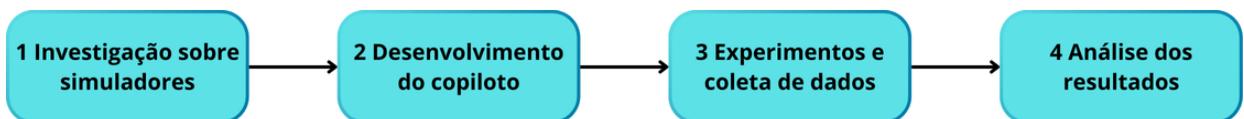
Fonte: Elaborado pelos autores

Ainda sobre sensores, foi utilizado o *sensor other collision* que registra um evento, sempre que há uma colisão com um ator dentro do ambiente do CARLA. Para processar e tratar as imagens obtidas, foi utilizado a biblioteca *OpenCV*, responsável pelos algoritmos de visão computacional fornecendo os dados para a Inteligência Artificial que foi construída com ajuda da ferramenta *Pytorch*, que trata os dados para a tomada de decisões, refletindo na direção dentro do CARLA.

O simulador CARLA, embora seja estável e bem elaborado, apresenta limitações na criação de cenários complexos, como vias com muitos carros ou a presença de pedestres em determinados momentos. Para superar essas limitações e construir cenários adversos para treinar veículos autônomos, foi recorrido à iniciativa de código aberto denominada ANTI-CARLA (RAMAKRISHNA et al., 2022a). Esta ferramenta, é mencionada pelos autores em seu artigo como uma estrutura de testes adversa em CARLA para sistemas de direção autônoma, permitindo a simulação de condições climáticas extremas, variações na luminosidade e outros desafios que podem ser encontrados nas ruas.

## 4.2 Método

Nesta subseção, é apresentado o método para o estudo, desenvolvimento e análise do copiloto de situações adversas e seus resultados. A Figura 3 ilustra a divisão do trabalho, que consistiu em quatro etapas: investigação sobre o tema, desenvolvimento do copiloto, experimento e coleta de dados e análise dos resultados.

**Figura 3 – Método de Trabalho**

**Fonte:** Elaborado pelos autores

Inicialmente, foi conduzido um estudo bibliográfico com o propósito de examinar a utilização de direção de veículos autônomos em simuladores e compreender a efetividade do seu uso para treinamento e direção de veículos em um cenário controlado. Durante essa fase, foi possível observar que existem simuladores que estão munidos com as principais ferramentas e sensores, e, além disso, possuem constantes atualizações, o que contribui com o avanço da tecnologia de direção autônoma, resultando em notáveis transformações.

Em decorrência da realização dos treinamentos, para avaliar o desenvolvimento da solução, foi necessária a busca de simuladores que pudessem auxiliar, tanto na execução, quanto demonstração das evoluções do projeto. Conforme apontado por Rosique et al. (2019), existem diversas soluções genéricas para validação e simulação de um carro autônomo, no entanto, baseado na necessidade de obter semelhança à realidade, sensores funcionais às análises dos cenários desejados, é prudente a busca de simuladores consolidados que possuam as características essenciais ao problema tratado.

Dentre os principais simuladores encontrados, houve uma análise em relação ao AirSim (SHAH et al., 2017), na qual se trata de um simulador multiplataforma *open-source*, utilizado para treinamento de drones, carros e outros, tendo suporte à simulações de software e hardware em loop. Foi criado por engenheiros da Microsoft como uma plataforma para pesquisadores de IA experimentarem algoritmos de aprendizado profundo, visão computacional e aprendizado por reforço para carros autônomos.

Outro simulador com um propósito de ensino e pesquisa é o Udacity Self Driving Simulator, desenvolvido pela Udacity, uma organização educacional com fins lucrativos, a fim de complementar seu curso abordando carros autônomos (Udacity, 2024). O simulador, utilizando de um ambiente virtual criado com o Unity 3D para coletar os dados capturados por 3 câmeras localizadas no carro, tem como objetivo principal educar os alunos sobre como treinar carros para navegar em estradas.

Uma outra opção é o Deepdrive (DEEPDRIVE, 2024), simulador com suporte a Windows e Linux, que possui interface de desenvolvimento utilizando da API do Gym, uma biblioteca Python de *open-source* para desenvolver e comparar algoritmos de Aprendizagem por Reforço. Esse simulador oferece sensores GPS, LIDAR e Radar, que podem ser utilizados em um desenvolvimento voltado à veículos autônomos, além de um conjunto robusto de ferramentas de desenvolvimento proporcionadas pelo Unreal Engine, obtendo uma qualidade gráfica extremamente realista e detalhada.

Diante dessas possibilidades, em relação a definição do simulador que seria utilizado, o CARLA é um dos mais competentes para com as necessidades deste trabalho, tendo foco princi-

pal na simulação, implementação e execução de algoritmos em veículos autônomos, possuindo suporte abrangente em sensores automotivos e uma comunidade de desenvolvimento ativa.

O CARLA é um simulador de direção autônoma de *open-source* Dosovitskiy et al. (2017), criado previamente para ser uma API modular abordando uma série de tarefas envolvendo direção. Inclui ativos digitais como layouts urbanos, edifícios e veículos para criação de mapas, o que amplia mais ainda os escopos para teste. O programa também pode ser usado para comparar o desempenho de três métodos de pipelines modulares de direção autônoma: um modelo treinado usando aprendizado por imitação, um modelo treinado através de aprendizado por reforço e um modelo ensinado através de aprendizado por reforço. Algumas das principais características do CARLA incluem escalabilidade através de uma arquitetura de servidor multi-cliente, um sensor de direção autônoma, uma API flexível, simulação rápida para planejamento e controle, criação de mapas, simulação de cenários de tráfego e bases de direção autônoma.

O uso de simuladores para treinamento de veículos autônomos é crucial para implementar algoritmos de Inteligência Artificial e expô-lo aos mais diversos cenários, resultando em uma melhoria substancial no treinamento em momentos normais e adversos. Dito isso, é importante treinar o modelo em diversas situações cotidianas possíveis vivenciadas por um motorista, bem como a cenários adversos que não podem ser reproduzidos com constância no mundo real. Dessa forma, o uso de uma tecnologia que reproduz o mundo real contribui para um aprimoramento na confiabilidade no algoritmo implementado.

Este artigo busca validar o uso de aprendizado por reforço em veículos autônomos para reagir em situações adversas no simulador CARLA, possibilitando um treinamento extensivo da IA em diversos cenários e tornando o modelo mais preparado para situações adversas. Essa abordagem não só aumenta o preparo dos copilotos existentes, como também melhora a robustez e a confiabilidade dos modelos.

A utilização do simulador no teste e treinamento do modelo de aprendizado por reforço é crucial, mas não isenta a necessidade de validação no mundo real. Pelo simulador, é apenas possível construir uma aplicação robusta e confiável e reduzir o tempo de treinamento, além de possibilitar o treinamento em cenários adversos. Outro benefício é a possibilidade de reproduzir problemas encontrados e a facilidade na otimização do treinamento, garantindo um produto de alta responsividade e eficiência para validar a viabilidade de veículos autônomos.

Na segunda parte do estudo, foi utilizado os cenários de trânsito disponibilizados pelo ANTI-CARLA para automatizar a criação dos ambientes e automação dos testes. Para a criação de um cenário é necessário definir os parâmetros referente a densidade de trânsito, rota e condições climáticas. Posto isso, foi construído um ambiente controlável e replicável para validação do algorítimo. O próximo passo foi a escolha com base nos estudo apresentado por Manikandan et al. (2023) do agente copiloto *Learning by Cheating* (LBC) para validação das suas capacidades de conclusão das rotas propostas.

Os dados coletados durante as execuções foram armazenados para estudo e análise. Desta forma, por meio de boas métricas é possível avaliar o nível confiabilidade do veículo autônomo. As informações coletadas consistem em: colisões do veículo com outros objetos,

tempo de reação do veículo, estabilidade de direção, entre outras informações obtidas pelos sensores internos (velocidade e rotação), como já mencionado anteriormente.

Na fase de avaliação dos resultados foram analisados os dados coletados e mapeado o desempenho do veículo ao se manter estável na pista, acurácia do sistema autônomo e a validação de possíveis novos casos que podem ocorrer numa situação real de trânsito.

## 5 EXPERIMENTOS E RESULTADOS

Como resultados preliminares, para avaliar o modelo proposto, foi utilizado o ANTI-CARLA para submeter o copiloto nos cenários de testes adversos no trânsito Figura 4a. Durante o processo de teste foi utilizado um mapeamento de visão com diferentes posições de câmera como entrada para o agente de direção, como exemplificado na Figura 4b.

**Figura 4 – Imagens ilustrativas do ANTI-CARLA**



(a) ANTI-CARLA em situação de baixa visibilidade

Fonte: Elaborado pelos autores



(b) ANTI-CARLA criando situações de risco

Fonte: Elaborado pelos autores

Para o copiloto foi utilizado o modelo *Learning by Cheating* (LBC) introduzido por Chen et al. (2019b), que usa o processo de aprendizagem por imitação para o treinamento do agente de direção. Além disso, em cada simulação o agente percorre uma rota do ponto A até o B, onde foram avaliados os seguintes critérios:

- Porcentagem da rota concluída: É definida pela razão da distância percorrida e a distância total do percurso;
- Porcentagem de extravio da faixa: Essa métrica é calculada pela razão entre a distância total percorrida fora das faixas permitidas e a distância total do percurso concluído. O sistema monitora continuamente a posição do veículo em relação a pontos de referência, chamados de *waypoints*, ao longo da rota. Se a distância entre a posição do veículo e a faixa permitida ultrapassa um limite pré-definido, o veículo é considerado fora da faixa.
- Testes de colisão: A métrica de colisão é calculada com base na contagem de eventos disparados pelo sensor de colisão do CARLA. Cada vez que o veículo colide com um objeto, o sensor gera um evento que é registrado, aumentando a contagem da métrica.

O sistema contabiliza colisões com outros veículos, objetos estáticos e pequenas quedas, aplicando filtro para ignorar colisões pequenas ou repetidas com o mesmo objeto;

- d) Tempo Esgotado: A métrica de tempo esgotado controla um limite de tempo na simulação usando o tempo interno do CARLA. Quando o tempo especificado é alcançado, o status da classe muda para *FALHA*, indicando que o tempo limite foi atingido;

Os critérios foram definidos com a finalidade de avaliar as funções mais básicas do copiloto, principalmente de se manter em uma pista sem colidir até o fim do percurso. Com estas métricas pode-se entender o desempenho do LBC.

Como já definido, o percurso consiste em percorrer de um ponto A do mapa até um ponto B, e é desejado que o veículo complete-o totalmente. Nos casos de teste demonstrados, foram utilizados apenas percursos curtos, como estradas retas sem cruzamentos e com curvas suaves, com a finalidade de focar apenas nas situações adversas.

Nos casos de teste, foram consideradas tanto a invasão de faixas erradas quanto a saída para calçadas. Quando detectado um extravio, o sistema acumula a distância percorrida fora da faixa e calcula o percentual correspondente da rota em que o veículo não manteve a direção correta.

Para criar os cenários de testes, é possível manipular os eventos e características no mapa da simulação através dos arquivos de configurações do próprio ANTI-CARLA. Os parâmetros são agrupados em 3 itens: 1 - Clima, 2 - Densidade de tráfego e 3 - Densidade de pedestres.

Para manipulação de clima foi ajustado alguns parâmetros como nebulosidade, que varia de 0 a 100, sendo 0 com nenhuma nebulosidade e 100 com a maior quantidade permitida pelo CARLA. Além disso foi manipulado a precipitação que também vai de 0 a 100 com a mesma lógica da nebulosidade. Outras características manipuladas foram a densidade do nevoeiro, também variando de 0 a 100, e a distância de visibilidade do nevoeiro de 0 a 100 onde 0 e 0 metros de distância e 100 são 1000 metros de distância. Para manipular o dia e noite é utilizado o Ângulo de altitude do sol que vai de 0 a 90, onde 0 representa 0° de referência ao horizonte e 90 é 90°.

A simulação ocorre em 5 cenários diferentes criados pelo próprio ANTI-CARLA, onde foi executado 1 teste para cada cenário em dois tipos de condições:

- a) Condições com boa visibilidade (Figuras 5a e 6a);
- a) Condições com baixa visibilidade (Figuras 5b e 6b);

**Figura 5 – Imagens do cenário 1 gerada pelo ANTI-CARLA**



**(a) Cenário 1 em boas condições de visibilidade**

Fonte: Elaborado pelos autores



**(b) Cenário 1 em baixas condições de visibilidade**

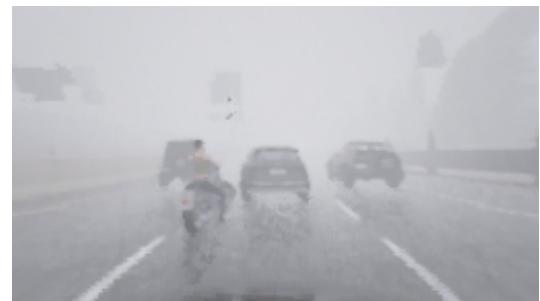
Fonte: Elaborado pelos autores

**Figura 6 – Imagens do cenário 2 gerada pelo ANTI-CARLA**



**(a) Cenário 2 em boas condições de visibilidade**

Fonte: Elaborado pelos autores



**(b) Cenário 2 em baixas condições de visibilidade**

Fonte: Elaborado pelos autores

Um dos 5 cenários de trânsito gerado pelo ANTI-CARLA é a inserção de trânsito intenso em uma avenida que pode ser observado nas Figuras 6 e 7 :

**Figura 7 – Simulação de trânsito intenso**



Fonte: Elaborado pelos autores

O *Learning by Cheating* (LBC) se mostrou eficaz ao lidar com as situações de boa visibilidade, no entanto, em situações adversas foram observadas falhas em vários cenários.

Como é possível ver nas Tabelas 2 e 3, o modelo obteve um desempenho satisfatório em cenários de clima favorável.

**Tabela 2 – Tempo de Execução - Condições normais**

<b>Hora de Início</b>	2024-06-07 21:02:43
<b>Hora de Término</b>	2024-06-07 21:03:28
<b>Duração (Tempo do Sistema)</b>	44.51s
<b>Duração (Tempo de Simulação)</b>	32.7s
<b>Razão (Tempo do Sistema / Tempo de Simulação)</b>	0.735

**Tabela 3 – Resultados dos Testes de Critério - Condições normais**

<b>Critério</b>	<b>Resultado</b>	<b>Valor</b>
Teste de Conclusão de Rota	SUCESSO	100 %
Teste de Rota Fora da Faixa	SUCESSO	0 %
Teste de Colisão	SUCESSO	0 vezes
Tempo Esgotado	SUCESSO	

Ao submeter o algoritmo às situações adversas foram obtidos resultados com uma eficiência variável e pouco satisfatória. As Tabelas 4 e 5 contém a exposição e avaliação dos mesmos critérios, desta vez em condições de baixa visibilidade.

**Tabela 4 – Tempo de Execução - Condições adversas**

<b>Hora de Início</b>	2024-06-12 23:00:03
<b>Hora de Término</b>	2024-06-12 23:02:38
<b>Duração (Tempo do Sistema)</b>	154.82s
<b>Duração (Tempo de Simulação)</b>	115.05s
<b>Razão (Tempo do Sistema / Tempo de Simulação)</b>	0.743

**Tabela 5 – Resultados dos Testes de Critério - Condições adversas**

<b>Critério</b>	<b>Resultado</b>	<b>Valor</b>
Teste de Conclusão de Rota	FALHA	50.35 %
Teste de Rota Fora da Faixa	FALHA	10.15 %
Teste de Colisão	FALHA	2 times
Tempo Esgotado	SUCESSO	

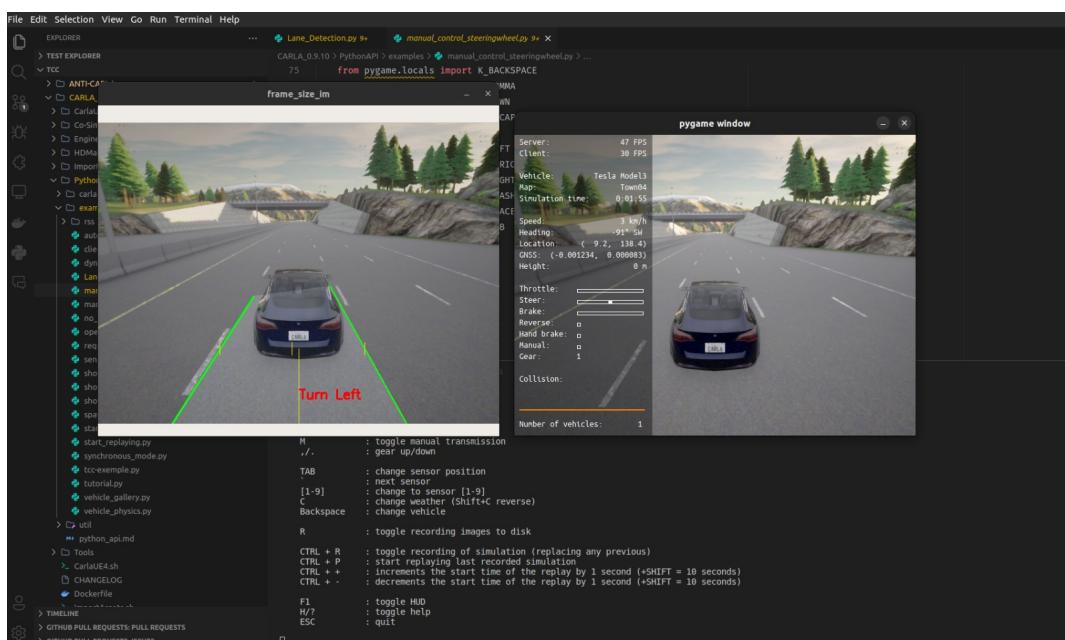
É possível concluir que a variação dos resultados de autonomia do veículo estão diretamente ligadas às condições aplicadas ao cenário. Existe uma incapacidade de tomada de

decisão assertiva quando se trata de baixa visibilidade, possivelmente devido ao desempenho insuficiente dos sensores, que podem ser atrapalhados por reflexos, ou pouca amplitude de visualização. A baixa visibilidade compromete a capacidade do sistema de detectar corretamente obstáculos e faixas de rodagem, levando a um aumento no número de falhas, como colisões e desvio de rota. Além disso, a ineficiência no tempo de resposta do sistema em relação às condições ideais reforça a necessidade de aprimoramentos no algoritmo.

Apesar dos cenários disponibilizados pelo ANTI-CARLA serem relativamente simples, como o trânsito em vias movimentadas, foram identificadas algumas dificuldades que o *Learning by Cheating* (LBC) enfrenta quando exposto a diferentes situações. Através dos resultados dos testes, é visível que o modelo tem uma boa eficiência nas simulações predefinidas pelo ANTI-CARLA. No entanto, podem ser desenvolvidas melhorias para abranger também as situações adversas supracitadas.

Como iniciativa para os próximos passos do trabalho, foi realizado um estudo de possíveis algoritmos que podem ser utilizados para auxílio no desenvolvimento do trabalho. Um deles é o de detecção das linhas apresentado por (John Kim, 2020) para manter o veículo na faixa corretamente. Utilizando algoritmos de reconhecimento de padrão com o auxílio do OpenCV foi possível obter os resultados da Figura 8.

**Figura 8 – Detecção das faixas**



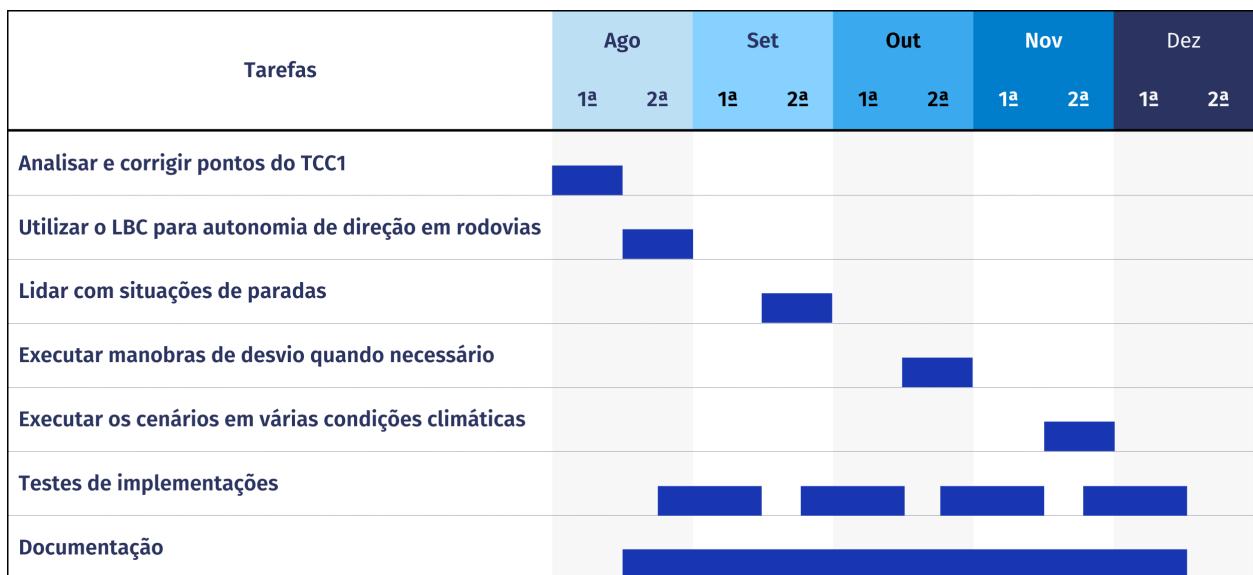
**Fonte:** Elaborado pelos autores

Os resultados obtidos neste trabalho servirão como base para o desenvolvimento do TCC2, onde será aprimorado o modelo de copiloto para situações adversas, com o objetivo de tornar o veículo autônomo mais seguro e confiável.

## 6 CRONOGRAMA

O planejamento para a continuação do desenvolvimento do trabalho durante o segundo semestre de 2024 baseia-se na revisão e continuação dos pontos discutidos no TCC1, como o aprimoramento do modelo de aprendizagem de imitação introduzido pelo *Learning By Cheating* (LBC) para aumentar a precisão na execução de manobras de desvio nos cenários existentes. Além disso, outro requisito importante é a introdução de novos cenários adversos para tornar o modelo mais especializado na tomada de decisão nos mesmos. Em paralelo, novos testes serão implementados e documentados durante o processo como exposto na Figura 9.

**Figura 9 – Cronograma TCC 2**



**Fonte:** Elaborado pelos autores

## Referências

- AL-OGAILI, Ali Saadon et al. Review on scheduling, clustering, and forecasting strategies for controlling electric vehicle charging: Challenges and recommendations. **IEEE Access**, v. 7, p. 128353–128371, 2019.
- BONON, Carlos Eduardo de Almeida. Estudo comparativo dos softwares de simulação para carros autônomos CARLA e Udacity. Universidade Federal Fluminense, 2019.
- CHEN, Dian et al. Learning by cheating. **CoRR**, abs/1912.12294, 2019. Disponível em: <<http://arxiv.org/abs/1912.12294>>.
- CHEN, Dian et al. **Learning by Cheating**. 2019.
- DEEPDRIVE. **DeepDrive: Open Source Autonomous Driving Simulator**. 2024. <<https://deepdrive.io/>>. [Acessado em: 18-maio-2024].
- DETTRAN-BA. **Evento promovido pelo Detran-BA trouxe expoentes do tema no Brasil**. 2023. Acessado: 5 de junho de 2024. Disponível em: <<https://www.detran.ba.gov.br/noticia.php?n=90-dos-sinistros-de-transito-sao-causados-por-falhas-humanas-revelam-especialistas-no-i-seminario-estadual-de-transito-da-bahia>>.
- DOSOVITSKIY, Alexey 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.
- ELALLID, Badr Ben et al. A reinforcement learning based approach for controlling autonomous vehicles in complex scenarios. In: **2023 International Wireless Communications and Mobile Computing (IWCMC)**. [S.l.: s.n.], 2023. p. 1358–1364.
- JEON, Hyeonjae et al. Carla simulator-based evaluation framework development of lane detection accuracy performance under sensor blockage caused by heavy rain for autonomous vehicle. **IEEE Robotics and Automation Letters**, v. 7, n. 4, p. 9977–9984, 2022.
- John Kim. **CARLA Lane Detection**. 2020. Acessado: 5 de junho de 2024. Disponível em: <<https://github.com/angelkim88/CARLA-Lane\Detection/blob/master/README.md>>.
- KUAN, Chi-Wei; CHEN, Wen-Hui; LIN, Yu-Chen. Pothole detection and avoidance via deep learning on edge devices. In: **2020 International Automatic Control Conference (CACS)**. [S.l.: s.n.], 2020. p. 1–6.
- MANIKANDAN, N. S.; KALIYAPERUMAL, Ganesan; WANG, Yong. Ad hoc-obstacle avoidance-based navigation system using deep reinforcement learning for self-driving vehicles. **IEEE Access**, v. 11, p. 92285–92297, 2023.
- MEHRA, Aryan et al. Reviewnet: A fast and resource optimized network for enabling safe autonomous driving in hazy weather conditions. **IEEE Transactions on Intelligent Transportation Systems**, v. 22, n. 7, p. 4256–4266, 2021.
- MICHAELIS. **Autonomia**. 2024. Accessed: 2024-05-15. Disponível em: <<https://michaelis.uol.com.br/busca?id=EMrj>>.
- MUZAHID, Abu Jafar Md et al. Deep reinforcement learning-based driving strategy for avoidance of chain collisions and its safety efficiency analysis in autonomous vehicles. **IEEE Access**, v. 10, p. 43303–43319, 2022.

RAMAKRISHNA, Shreyas et al. **ANTI-CARLA: An Adversarial Testing Framework for Autonomous Vehicles in CARLA**. 2022.

RAMAKRISHNA, Shreyas et al. Anti-carla: An adversarial testing framework for autonomous vehicles in carla. In: **2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2022. p. 2620–2627.

REDMON, Joseph; FARHADI, Ali. **YOLOv3: An Incremental Improvement**. 2018.

ROSIQUE, Francisca et al. A systematic review of perception system and simulators for autonomous vehicles research. **Sensors**, v. 19, n. 3, 2019. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/19/3/648>>.

SAE International. **Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles**. 2018. Disponível em: <[https://www.sae.org/standards/content/j3016\\\_\\\_201806/](https://www.sae.org/standards/content/j3016\_\_201806/)>.

SHAH, Shital et al. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: **Field and Service Robotics**. [s.n.], 2017. Disponível em: <<https://arxiv.org/abs/1705.05065>>.

SURAL, Shounak et al. Cosim: A co-simulation framework for testing autonomous vehicles in adverse operating conditions. In: **2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2023. p. 2098–2105.

Udacity. **Self-Driving Car Simulator**. 2024. <<https://github.com/udacity/self-driving-car-sim>>. Accessed: Date Accessed.

VARGHESE, Jaycil Z; BOONE, Randy G et al. Overview of autonomous vehicle sensors and systems. In: SN. **International Conference on Operations Excellence and Service Engineering**. [S.l.], 2015. v. 2015.