

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GUILHERME DE SOUZA CERDEIRA

PEDRO RAMOS PEREIRA

**DESENVOLVIMENTO DE UM JOGO EDUCACIONAL PARA APOIO AO
AUTOESTUDO DE CANDIDATOS À OBTENÇÃO DA CARTEIRA NACIONAL
DE HABILITAÇÃO**

PONTA GROSSA

2026

**GUILHERME DE SOUZA CERDEIRA
PEDRO RAMOS PEREIRA**

**DESENVOLVIMENTO DE UM JOGO EDUCACIONAL PARA APOIO AO
AUTOESTUDO DE CANDIDATOS À OBTENÇÃO DA CARTEIRA NACIONAL
DE HABILITAÇÃO**

**Development of a Educational Game to Support Self-Study for Driver's
License Applicants**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Ciência da Computação do
Curso de Bacharelado em Ciência da Computação
da Universidade Tecnológica Federal do Paraná.
Orientador: Prof. Dr. Richard Duarte Ribeiro
Coorientador: Prof. Dr. Vinícius Camargo Andrade

**PONTA GROSSA
2026**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

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

**GUILHERME DE SOUZA CERDEIRA
PEDRO RAMOS PEREIRA**

**DESENVOLVIMENTO DE UM JOGO EDUCACIONAL PARA APOIO AO
AUTOESTUDO DE CANDIDATOS À OBTENÇÃO DA CARTEIRA NACIONAL
DE HABILITAÇÃO**

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

Data de aprovação: 26 / junho / 2026

Richard Duarte Ribeiro
Doutorado
Universidade Tecnológica Federal do Paraná

Luiz Rafael Schmitke
Doutorado
Universidade Tecnológica Federal do Paraná

Helyane Bronoski Borges
Doutorado
Universidade Tecnológica Federal do Paraná

**PONTA GROSSA
2026**

AGRADECIMENTOS

Agradecemos o nosso professor orientador Dr. Richard Duarte Ribeiro e o nosso coorientador Prof. Dr. Vinícius Camargo Andrade, que nos apoiaram e nos guiaram nesse processo de construção do nosso trabalho. Agradecemos também todos os familiares e amigos que estiveram do nosso lado nesse ciclo muito importante de nossas vidas.

Gostaríamos também de agradecer ao restante do corpo docente que compõe a universidade e pelos ensinamentos que nos foram passados ao longo do curso.

Ficam os nossos sinceros agradecimentos a todos previamente citados e também a todos os outros que contribuíram positivamente na nossa jornada acadêmica.

RESUMO

A educação para o trânsito no Brasil enfrenta o desafio constante de modernizar suas metodologias, especialmente diante da flexibilização do ensino teórico promovida pela Resolução Conselho Nacional de Trânsito (CONTRAN) nº 1.020/2025, a qual flexibilizou a formação teórica ao permitir modalidades presenciais e a distância. e amplia o incentivo da criação de ferramentas criativas de autoestudo. Diante do alto índice de acidentes viários e da limitação de métodos tradicionais estáticos para reter a atenção dos jovens, este trabalho propõe o desenvolvimento de um jogo educacional *desktop/web*, construído no motor gráfico *Godot*, voltado ao apoio e preparação de candidatos à Carteira Nacional de Habilitação (CNH). Inspirado no sucesso de engajamento no jogo da campanha australiana *Dumb Ways to Die*, o software desenvolvido mostra uma abordagem punitiva ao utilizar o humor e uma estética cartunesca para ilustrar as consequências de infrações e da falta de direção defensiva. Por meio da integração das perguntas oficiais do Banco Nacional de Questões às mecânicas interativas de tentativa e erro, o jogo busca transformar a fixação dos conteúdos em uma experiência lúdica e visual. Ao decorrer do trabalho, é feita a análise, o design, a implementação e a testagem do software proposto. Como conclusão, o software desenvolvido e suas funcionalidades são apresentados, juntamente com tópicos sobre possíveis trabalhos futuros.

Palavras-chave: jogos; gamificação; trânsito; habilitação.

ABSTRACT

Traffic education in Brazil faces the ongoing challenge of modernizing its methodologies, especially in light of the flexibilization of theoretical instruction promoted by Resolution CONTRAN No. 1,020/2025, which allows courses to be conducted in the Distance Learning modality and further encourages the creation of creative self-study tools. Given the high rate of traffic accidents and the limitations of traditional static methods in retaining the attention of younger audiences, this work proposes the development of a *desktop/web* educational game, built using the *Godot* game engine, aimed at supporting and preparing candidates for obtaining a drivers license (CNH). Inspired by the engagement success of the Australian campaign game *Dumb Ways to Die*, the developed software adopts a punitive approach by using humor and a cartoon-style aesthetic to illustrate the consequences of traffic violations and the lack of defensive driving. By integrating official questions from the National Question Bank with interactive trial-and-error mechanics, the game seeks to transform the reinforcement of theoretical content into a fun and visual learning experience. Throughout the development of this work, the analysis, design, implementation, and testing of the proposed software are presented. As a conclusion, the developed software and its functionalities are discussed, along with topics addressing possible future work.

Keywords: games; gamification; traffic; driver's license.

LISTA DE FIGURAS

Figura 1 – Metodologia cascata para desenvolvimento de software	18
Figura 2 – Diagrama de Caso de Uso <i>Unified Modeling Language</i> (UML).....	25
Figura 3 – Diagrama de Classes UML	26
Figura 4 – Diagrama de Sequência UML do modo desafio	28
Figura 5 – Diagrama de Sequência UML do modo treino.....	29
Figura 6 – Exemplos do <i>feedback</i> visual	31
Figura 7 – Ícone do jogo.	34
Figura 8 – Menu principal do jogo.	34
Figura 9 – Interface do modo treino.	35
Figura 10 – Interface do placar de líderes.....	35
Figura 11 – Exemplo de um desafio.....	36
Figura 12 – Tela de desafio completo.....	36
Figura 13 – Tela de desafio fracassado	37
Figura 14 – Tela de final da partida.....	37
Figura 15 – Proporção de participantes que possuem CNH	38
Figura 16 – Resultados da avaliação do jogo baseada no modelo <i>Model for the Evaluation of Educational Games</i> (MEEGA+) simplificado	39
Figura 17 – Média de avaliação por pergunta do modelo MEEGA+	39

LISTA DE QUADROS

Quadro 1 – História de usuário 1: Iniciar o jogo	22
Quadro 2 – História de usuário 2: Inserir nome de jogador	22
Quadro 3 – História de usuário 3: Jogar modo desafio	23
Quadro 4 – História de usuário 4: Jogar modo treino	23
Quadro 5 – História de usuário 5: Realizar um desafio	23
Quadro 6 – História de usuário 6: Visualizar questão oficial	24
Quadro 7 – História de usuário 7: Visualizar placar de líderes	24

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CNH	Carteira Nacional de Habilitação
CONTRAN	Conselho Nacional de Trânsito
CTB	Código de Trânsito Brasileiro
EaD	Ensino a Distância
GUT	<i>Godot Unit Testing</i>
JERED	Jeitos Engraçados de Reprovar no Exame do Detran
JSON	<i>JavaScript Object Notation</i>
MEEGA+	<i>Model for the Evaluation of Educational Games</i>
SENATRAN	Secretaria Nacional de Trânsito
SESC	Serviço Social do Comércio
UI	<i>User Interface</i>
UML	<i>Unified Modeling Language</i>
UX	<i>User Experience</i>

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Objetivos	11
1.1.1	Objetivo geral	11
1.1.2	Objetivos específicos	12
1.2	Organização do trabalho	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	Jogos educacionais	13
2.2	Normas de trânsito no Brasil	15
2.3	Conceitos do trânsito brasileiro	15
2.4	Softwares Similares.....	16
3	METODOLOGIA	18
3.1	Análise	19
3.2	Projeto	19
3.3	Implementação	19
3.4	Testagem.....	20
4	DESENVOLVIMENTO.....	22
4.1	Elicitação de requisitos	22
4.2	Projeto do sistema	26
4.2.1	Estrutura do sistema	26
4.2.2	Fluxos do sistema	27
4.2.3	Testes unitários	29
4.3	Implementação do sistema	30
4.3.1	Comunicação entre componentes por meio de sinais	30
4.3.2	Sistema de <i>feedback</i> visual ao jogador	30
4.3.3	Testes unitários	31
4.3.4	Extensão e adição de novos desafios	32
5	RESULTADOS E DISCUSSÕES	34
5.1	JERED: Jeitos Engraçados de Reprovar no Exame do Detran	34
5.2	Avaliação do jogo	37
5.2.1	Perfil dos Participantes	37
5.2.2	Respostas	38
6	CONCLUSÃO	41
6.1	Trabalhos futuros	41
	REFERÊNCIAS.....	43
	APÊNDICE A QUESTIONÁRIO DE PESQUISA	47
	APÊNDICE B MATRIZ DE RESPOSTAS DA AVALIAÇÃO	49

1 INTRODUÇÃO

O crescimento acelerado da frota de veículos no Brasil, que hoje ultrapassa a marca de 124 milhões de unidades, transformou a dinâmica das cidades e tornou o trânsito um ambiente cada vez mais hostil (Infleet, 2025). O aumento constante no número de acidentes e a dificuldade em formar condutores devidamente preparados são desafios que as autoescolas enfrentam diariamente. Nesse cenário, métodos de ensino tradicionais muitas vezes não conseguem prender a atenção do aluno, abrindo espaço para a gamificação como uma forma de tornar o aprendizado mais prático, seguro e envolvente (Aguiar; Ferreira, 2025).

A formação de condutores no Brasil passa por uma transição normativa crítica com a Resolução CONTRAN nº 1.020/2025. Esta nova resolução estabelece diretrizes para um modelo de ensino mais flexível, permitindo que a carga horária teórica seja cumprida de forma assíncrona via Ensino a Distância (EaD) (Brasil, 2025). Esta mudança normativa transfere parte da responsabilidade do treinamento e aprendizado para o estudo autônomo do candidato, que antes era compromisso da autoescola.

Essa flexibilização das exigências relacionadas à formação de condutores tem sido alvo de debates na sociedade. Entre os argumentos favoráveis à mudança, o professor da UnB e doutor em Estudos de Transporte pela Universidade de Londres, Paulo Cesar Marques, defendeu a flexibilização, apontando a redução de custos para a obtenção da carteira de habilitação como o principal motivo (G1, 2025). Também comparou esse novo formato com o processo de obtenção da carteira de motorista dos países escandinavos e do Reino Unido, países estes que, de acordo com Paulo Cesar Marques, são referência na área de segurança no trânsito.

Em contrapartida, os argumentos de David Duarte Lima, doutor em segurança de trânsito pela Universidade Livre de Bruxelas, demonstram a preocupação do profissional perante as mudanças estabelecidas. O doutor reforça a “necessidade de um sistema de ensino robusto, e não o enfraquecimento dele” (G1, 2025). Comparou a estratégia defendida com a realizada pela Espanha, que segundo ele, possuía um trânsito violento e, após massivos investimentos na formação obrigatória dos candidatos à carteira de habilitação, resultou em uma queda de 80% na mortalidade no trânsito em menos de 10 anos.

Adicionalmente, o coronel Ricardo Silva, vice-presidente do Departamento de Trânsito de Santa Catarina, afirma que o foco dessa flexibilização foi a redução de custos para o candidato à habilitação, e que os critérios estruturais de segurança no trânsito não foram priorizados (CNN Brasil, 2025). Em uma de suas falas, ele comenta que pode haver um prejuízo na formação, e também um desinteresse por parte dos novos condutores em conhecer a legislação de trânsito.

A motivação do projeto proposto fundamenta-se em pilares legais e pedagógicos. Sob a ótica legal, o Marco Legal da Indústria de Jogos Eletrônicos (Lei nº 14.852/2024) mostra o desenvolvimento de jogos como vetor de progresso cultural e social, incentivando seu uso para fins didáticos (Brasil, 2024). Paralelamente, a Resolução CONTRAN nº 1.020/2025 fomenta

o uso de tecnologias inovadoras, criando uma demanda direta por softwares de suporte ao aprendizado individual (Brasil, 2025).

Alinhado à essa demanda, sob o ponto de vista pedagógico, o presente projeto baseia-se na campanha *Dumb Ways to Die*¹. Criada originalmente para promover a segurança ferroviária, a abordagem evidenciou que empregar uma estética carismática, combinada com a apresentação de consequências letais em um cenário recreativo, pode provocar um engajamento maior do que as campanhas de conscientização convencionais (Ward, 2015).

Com base no jogo desenvolvido pela campanha e nos argumentos apresentados pelos profissionais, este trabalho propõe o desenvolvimento de um jogo educacional, visando mitigar esses futuros prováveis riscos dessa flexibilização. A gamificação neste projeto proposto, similar ao do jogo da campanha *Dumb Ways to Die* consiste principalmente na utilização dos seguintes elementos de jogos, segundo Sauter (2024): Pontos, Competição e Desafio.

Inspirado na mecânica da campanha citada anteriormente, o *software* proposto adotará dinâmicas de decisão rápida para simular situações baseadas no banco oficial de questões da Secretaria Nacional de Trânsito (SENATRAN). A jogabilidade divide-se em dois modos principais: o "Modo Desafio", no qual o usuário deve solucionar cenários sob um cronômetro regressivo para acumular pontos; e o "Modo Treino", um ambiente de prática que permite jogar fases individualmente, de maneira que o usuário poderá praticar e revisar as regras e questões abordadas.

Fazendo uma ligação ao elemento de competição segundo Sauter (2024), o jogo contará com um *leaderboard* no qual irá conter o nome e suas respectivas pontuações de outros jogadores. Essa funcionalidade será integrada por meio da *SimpleBoards Application Programming Interface* (API), operando de forma remota e atualizando os dados automaticamente.

1.1 Objetivos

Nesta seção, são apresentados o objetivo geral e os objetivos específicos do trabalho desenvolvido.

1.1.1 Objetivo geral

O objetivo geral deste trabalho consiste em desenvolver um jogo multiplataforma, utilizando o motor gráfico *Godot*², para auxiliar candidatos à obtenção da CNH (Carteira Nacional de Habilitação) no processo de autoestudo.

¹ <https://www.dumbwaystodie.com/>

² <https://godotengine.org/pt-br/>

1.1.2 Objetivos específicos

Estão definidos, para esse trabalho, os seguintes objetivos específicos:

1. Levantar requisitos pedagógicos baseados na legislação vigente da nova resolução CONTRAN nº 1.020/2025 e no Banco Nacional de Questões da SENATRAN (Brasil, 2025).
2. Modelar a arquitetura do software, integrando mecânicas de gamificação, como sistemas de pontuação e feedback imediato (Aguiar; Ferreira, 2025).
3. Implementar o jogo no motor gráfico *Godot*, simulando cenários reais de sinalização e normas de circulação.
4. Testar o software desenvolvido por meio de testes unitários com a ferramenta *Godot Unit Testing* (GUT)³ e avaliá-lo, junto aos usuários participantes, por meio de um formulário.

1.2 Organização do trabalho

Este trabalho foi dividido em 6 capítulos principais, abordando desde a ideia e a conceitualização do projeto, até seus resultados e discussões. O primeiro capítulo serve como uma introdução ao leitor sobre o tema abordado no artigo. O segundo é focado na parte acadêmica, mostrando definições, conceitos, exemplos e projetos similares, retirados diretamente da literatura. O terceiro capítulo comenta detalhadamente a metodologia utilizada neste trabalho. O quarto, fala sobre o processo de desenvolvimento do software apresentado anteriormente, seguindo os padrões da metodologia definida. Já o quinto capítulo mostra quais foram os resultados obtidos, apresentando o software em seu estado final, junto com algumas discussões que foram levantadas no decorrer do processo de escrita deste trabalho. Por fim, o sexto capítulo conclui este trabalho, trazendo algumas considerações sobre o projeto e relatando ideias para trabalhos futuros.

³ <https://godotengine.org/asset-library/asset/1709>

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são mostrados conceitos, definições, tecnologias e trabalhos relacionados, retirados da literatura, em que este trabalho se baseia.

Na Seção 2.1 é comentado sobre técnicas de jogos educacionais e suas aplicações neste projeto. Na Seção 2.2, são abordados aspectos das normas de trânsito no Brasil, e como elas estão relacionadas com o jogo produzido neste trabalho. E por fim, na seção 2.4, são mostrados alguns dos softwares similares ao desenvolvimento deste trabalho que avaliamos, apresentando suas características principais.

2.1 Jogos educacionais

O uso de jogos no contexto pedagógico vai além do entretenimento, consolidando-se como uma estratégia para a construção e retenção de conhecimentos complexos. Para compreender esse potencial, é necessário distinguir entre jogos digitais comuns e jogos educacionais com design intencional de aprendizagem. Gee (2003) argumenta que bons jogos eletrônicos incorporam princípios sofisticados de aprendizagem em sua própria estrutura, não como camada adicional, mas como parte fundamental de seu funcionamento. O autor identifica que jogos bem projetados situam o significado em experiências incorporadas, ou seja, o jogador não apenas recebe informações abstratas, mas as experimenta dentro de contextos de ação que conferem sentido prático ao conteúdo aprendido.

Esse processo pode ser potencializado utilizando o conceito de *microlearning*, que consiste na entrega de conteúdos em pequenas unidades focadas e de curta duração (Fadel *et al.*, 2023). No desenvolvimento de jogos para candidatos à habilitação, essa técnica permite que o usuário exercite o domínio da legislação de trânsito por meio de desafios rápidos e sequenciais, facilitando a retenção do conteúdo através da repetição orientada e da interação baseada no Banco Nacional de Questões da SENATRAN. A combinação entre *microlearning* e jogos cria ciclos de aprendizagem curtos porém intensos, nos quais o candidato recebe um estímulo, responde, recebe retorno imediato e avança para o próximo nível.

Nesse sentido, Prensky (2001) destaca que o engajamento em jogos eletrônicos ocorre porque eles oferecem objetivos claros e *feedback* imediato, elementos essenciais para manter a motivação do aprendiz. No contexto do autoestudo para a CNH, a utilização de mecânicas de tentativa e erro permite que o candidato aprenda com os próprios erros em um ambiente seguro, sem as consequências reais que uma falha no trânsito acarretaria. Esse aspecto é diretamente relacionado ao que Gee (2003) denomina de moratória psicossocial, isto é, um espaço de aprendizagem onde os riscos do mundo real são reduzidos, permitindo que o aprendiz explore, erre e revise suas estratégias sem punições definitivas.

Complementarmente, a gamificação, entendida como a aplicação de elementos e mecânicas típicas de jogos em contextos que não são jogos, oferece uma estrutura metodológica

para tornar o processo de estudo mais envolvente (Sauter, 2024). Entre os principais elementos gamificados aplicáveis ao ensino da legislação de trânsito destacam-se: sistemas de pontuação que recompensam o acerto e incentivam a revisão; *rankings* e placares que estimulam a competição saudável entre usuários; conquistas e medalhas que reconhecem marcos de aprendizagem; e barras de progresso que conferem visibilidade ao avanço do candidato.

No software desenvolvido neste trabalho, intitulado Jeitos Engraçados de Reprovar no Exame do Detran (JERED), esses conceitos da gamificação foram implementados por meio de dinâmicas estruturadas em dois modos de jogo distintos: o Modo Desafio e o Modo Treino. No Modo Desafio, o elemento de desafio é imposto por um sistema de progressão de dificuldade temporal combinada com um limitador de três vidas, em que o candidato deve interagir rapidamente com cenários, baseados no Banco Nacional de Questões da SENATRAN, antes que uma barra de progresso chegue no final. O sistema de pontos quantifica o sucesso do usuário a cada rodada e, ao final da partida, essa pontuação é submetida a um placar de líderes global, ativando o elemento de competição ao permitir a comparação de desempenho entre diferentes candidatos.

Em contrapartida, o Modo Treino remove a pressão punitiva das vidas e do tempo para focar na fixação puramente cognitiva, oferecendo um catálogo de livre escolha onde o usuário estuda cenários específicos e recebe *feedback* imediato de acerto ou erro, consolidando a estratégia de *microlearning* e os conceitos de um espaço de aprendizagem seguro descrito por Gee (2003) anteriormente.

Gee (2003) também reforça esse argumento ao afirmar que bons jogos educativos funcionam como simuladores de experiência, nos quais o jogador não apenas memoriza informações, mas as vivencia em situações próximas da realidade. No contexto da CNH, isso significa transformar regras abstratas do Código de Trânsito Brasileiro (CTB) em simulações de decisão aplicadas, onde o candidato precisa avaliar cenários, identificar infrações e escolher condutas corretas, desenvolvendo um repertório que tende a transcender a memorização se aproximando da compreensão funcional das normas. Essa distinção entre memorização superficial e compreensão situada é, para Gee (2003), o cerne da diferença entre aprendizagem passiva e aprendizagem ativa.

No entanto, a mera inserção de elementos de gamificação e conteúdos educacionais não assegura, por si só, a qualidade técnica de um *serious game*. A literatura aponta que a validação de jogos educacionais exige metodologias empíricas rigorosas que analisem o artefato sob múltiplos fatores, equilibrando os objetivos de aprendizagem com a experiência de entretenimento (Martinez; Menéndez-menéndez; Bustillo, 2022). Diante disso, avaliar um jogo pedagógico requer instrumentos que mensurem não apenas aspectos isolados de *software*, como a sua interface, mas também o engajamento provocado, a motivação gerada e, essencialmente, a percepção do usuário quanto ao conhecimento retido durante a atividade de autoestudo.

Para responder a essa necessidade de validação no cenário acadêmico, destaca-se o MEEGA+, um framework utilizado para a avaliação da qualidade de jogos educacionais sob

a perspectiva dos estudantes. Esse modelo estrutura-se através de um questionário padronizado com respostas utilizando a escala Likert (5 opções de respostas, variando de “discordo totalmente” a “concordo totalmente”) (Devellis; Thorpe, 2021).

O MEEGA+ divide a percepção do jogador sobre o jogo em duas dimensões principais: a Usabilidade (experiência do usuário, englobando estética, clareza e facilidade de aprendizado) e a Experiência do Jogador (focada em aspectos motivacionais, como desafio, imersão, diversão e o aprendizado percebido) (Petri; Wangenheim; Borgatto, 2019). A aplicação desse modelo permite tentar traduzir a experiência subjetiva dos candidatos à CNH em dados estatísticos quantificáveis, visando fornecer os dados para avaliar se os elementos gamificados, implementados no jogo educacional, cumprem o seu papel no suporte ao aprendizado do conteúdo em questão.

2.2 Normas de trânsito no Brasil

No Brasil, a legislação que rege a circulação de veículos e pedestres é o CTB (Brasil, 1997). Segundo o Código de Trânsito Brasileiro, o trânsito em condições seguras é um direito de todos, sendo dever dos órgãos do Sistema Nacional de Trânsito priorizar a defesa da vida, nela incluída a preservação da saúde e do meio-ambiente (Brasil, 1997), classificando comportamentos de risco em diferentes naturezas de infração, cada qual vinculada a uma penalidade específica.

Ao simular situações de risco baseadas no Banco Nacional de Questões da SENATRAN, o sistema permite que o usuário internalize não apenas a regra teórica, mas a lógica de segurança necessária para evitar as sanções administrativas e, primordialmente, os acidentes reais.

2.3 Conceitos do trânsito brasileiro

Dentre os temas fundamentais exigidos nos exames teóricos da SENATRAN, destacam-se a Direção Defensiva, as Regras de Circulação e Conduta e o entendimento da Sinalização Viária.

A Direção Defensiva compreende o conjunto de técnicas e atitudes que permitem ao condutor antecipar e reconhecer situações de perigo, agindo de maneira a prevenir acidentes, independentemente das condições adversas do ambiente ou das ações incorretas de outros usuários da via (Brasil, 2005).

As Regras de Circulação e Conduta, estabelecem a hierarquia de responsabilidades no trânsito. O CTB determina claramente que os veículos de maior porte são responsáveis pela segurança dos menores, os motorizados pelos não motorizados e, em conjunto, todos têm o dever de zelar pela incolumidade dos pedestres (Brasil, 1997).

O *software* proposto atua como um instrumento interativo para a consolidação dos conceitos supracitados. Por meio de minijogos e cenários práticos, o usuário será desafiado a aplicar, por exemplo, as regras de circulação e conduta, devendo identificar quais veículos encontram-se em situações irregulares na via.

2.4 Softwares Similares

A análise de softwares similares é fundamental para identificar práticas bem-sucedidas e lacunas no mercado de educação viária. Nesta seção, destacam-se ferramentas que unem a temática do trânsito à gamificação e ao entretenimento.

Um caso de sucesso nessa vertente é a campanha e jogo australiano *Dumb Ways to Die*, lançado pela operadora *Metro Trains* de Melbourne. Criado para reduzir acidentes nas vias ferroviárias, o *software* optou por não utilizar imagens de acidentes reais, adotando personagens animados amigáveis, humor e uma estética minimalista para mostrar acidentes absurdos causados por desatenção (Torres, 2014). A ideia do jogo baseia-se no fato de focar na conexão emocional e na jogabilidade divertida antes de entregar a mensagem de segurança. A eficácia desse modelo foi comprovada estatisticamente: a campanha ajudou a reduzir em mais de 30% os índices de acidentes na região de Melbourne, em seu primeiro ano (Kwon, 2016).

No contexto acadêmico e de autoescolas, pesquisas recentes mapearam outros jogos sérios voltados à educação no trânsito que servem como referência direta. Entre eles, destaca-se o jogo *Transit Kämpfer*, desenvolvido por alunos da escola Serviço Social do Comércio (SESC) do Rio de Janeiro (Ferreira *et al.*, 2019), voltado para o ensino de normas de trânsito a partir de mecânicas de *quiz* com *feedbacks* imediatos, que demonstrou aumento significativo no engajamento e no aprendizado dos usuários (Hobold; Silva; Werlich, 2025). Outro exemplo pertinente é o aplicativo *Traffic*, uma ferramenta de gamificação focada no ensino das placas do CTB, cujo diferencial é a tradução de conteúdos técnicos e legislativos para uma linguagem acessível e atrativa, especialmente para os jovens (Hobold; Silva; Werlich, 2025).

Entre outros exemplos de jogos educacionais identificados na literatura nacional e internacional destacam-se: *CivLusorie* (Andrade; Almeida; Bittencourt, 2015), por exemplo, atua como um simulador de gestão municipal focado na mobilidade urbana de Feira de Santana (BA), promovendo reflexões sobre a responsabilidade política e social no trânsito. Já o *Beyond Fun* (Rodrigues *et al.*, 2015) aposta na imersão, consistindo em um simulador 3D no qual o uso de controles alternativos, como volantes e *tablets*, demonstrou aumentar o engajamento no aprendizado prático de regras viárias e sinalizações. Por fim, o título *Rules on Wheels* (Ismail; Abdennadher; Abouelsaadat, 2016), foca no ensino sobre as leis de trânsito no Egito. Utilizando grupos e controles de testes o estudo conclui que participantes que jogaram o jogo obtiveram resultados superiores aos que utilizaram o site oficial do governo egípcio. Uma síntese contendo as principais características dos *softwares* e campanhas mencionados pode ser observada na Tabela 1.

Apesar dessas iniciativas, a literatura ressalta que muitos jogos educacionais ainda focam estritamente na memorização estática e na perspectiva exclusiva do motorista (Hobold; Silva; Werlich, 2025). O jogo proposto neste trabalho busca preencher essa lacuna ao unir o rigor do Banco Nacional de Questões da SENATRAN exigido pelas resoluções do CONTRAN (Brasil, 2025) ao estilo de jogabilidade engajador e cômico inspirado em *Dumb Ways to Die*, criando um *software* moderno e focado no autoestudo do jogador.

Tabela 1 – Comparativo de *softwares* educacionais e jogos de trânsito

Software	Origem	Características principais
<i>Dumb Ways to Die</i>	Metro Trains (Austrália)	Personagens animados, humor e estética minimalista para prevenção de acidentes de forma lúdica.
<i>Transit Kämpfer</i>	SESC (Rio de Janeiro)	Mecânica de <i>quiz</i> com <i>feedback</i> imediato sobre normas de trânsito e alto engajamento.
<i>Traffic</i>	Acadêmico (Brasil)	Gamificação voltada ao ensino das placas do CTB com linguagem acessível aos jovens.
<i>CivLusorie</i>	Acadêmico (Brasil)	Simulador de gestão municipal com foco em decisões de mobilidade urbana e cidadania.
<i>Beyond Fun</i>	Acadêmico (Brasil)	Simulador 3D que utiliza periféricos não tradicionais (volantes e <i>tablets</i>) para aumentar a imersão.
<i>Rules on Wheels</i>	Acadêmico (Egito)	Jogo voltado ao ensino de placas de trânsito validada por grupos de controle.

Fonte: Autoria própria (2026).

3 METODOLOGIA

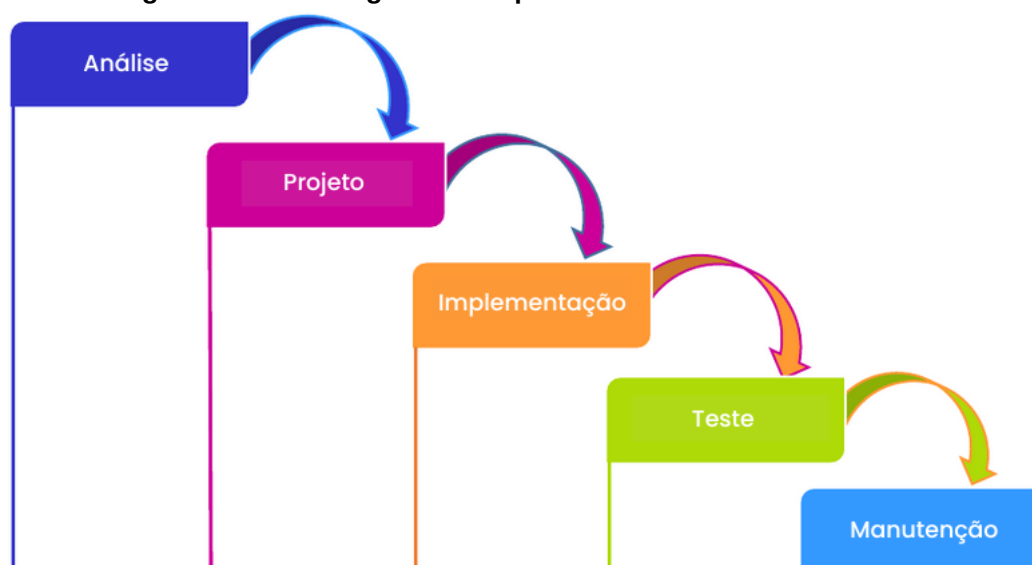
Neste capítulo é abordada a metodologia que foi utilizada neste projeto, começando pela apresentação da metodologia escolhida, os motivos da escolha, e as suas etapas (Seções 3.1, 3.2, 3.3 e 3.4).

Para o desenvolvimento deste software, optou-se pela utilização do modelo em Cascata (*Waterfall*) (Pressman; Maxim, 2014), uma abordagem de desenvolvimento de software sequencial e linear. Este modelo é caracterizado por um fluxo unidirecional, assemelhando-se a uma queda d'água, onde o progresso flui de forma constante para baixo através das fases de concepção, iniciação, análise, projeto, construção e teste. De acordo com Senarath (2021), uma das principais vantagens desta abordagem é a disciplina que ela impõe: cada fase possui entregáveis específicos e um processo de revisão próprio, o que facilita o controle do cronograma em projetos com prazos rígidos, como é o caso deste trabalho acadêmico.

Diferente de metodologias ágeis, que permitem mudanças constantes de escopo, o modelo *Waterfall* exige que os requisitos sejam detalhados e compreendidos logo no início (Pressman; Maxim, 2014). Para o desenvolvimento de um jogo educativo voltado à legislação de trânsito, essa característica é benéfica, visto que o conteúdo pedagógico é baseado em normas jurídicas estáveis (as Resoluções do CONTRAN), permitindo que o *design* e a implementação ocorram sobre uma base sólida e pouco sujeita a alterações durante o ciclo de desenvolvimento.

A escolha por esta metodologia justifica-se pelo objetivo deste projeto acadêmico, que possui requisitos bem definidos e um cronograma fixo. Abaixo, detalham-se as etapas adaptadas para este estudo, conforme a Figura 1.

Figura 1 – Metodologia cascata para desenvolvimento de software



Fonte: Senarath (2021)

Conforme proposto por Senarath (2021), a metodologia Cascata original prevê 5 etapas, incluindo a etapa de Manutenção, voltada à manutenção do software. Entretanto, para este trabalho, essa etapa foi descartada por se tratar de um projeto de entrega única para fins acadêmicos.

3.1 Análise

Nesta fase inicial, foram levantadas as necessidades pedagógicas e técnicas do jogo. O foco concentrou-se no estudo da legislação de trânsito vigente, especificamente na Resolução CONTRAN nº 1.020/2025, para identificar os conteúdos essenciais ao autoestudo dos candidatos. Definiram-se, nesta fase, as funcionalidades do sistema, como o controle de progresso do usuário e a lógica das questões teóricas.

3.2 Projeto

Com os requisitos estabelecidos, a etapa de *Design* focou na arquitetura técnica e na experiência do usuário, conhecida como *User Experience* (UX). Foram definidos:

- Mecânicas de Jogo: Como o usuário interage com o conteúdo educativo;
- Interface Visual: fluxo de navegação em telas;
- Estrutura de Dados: Organização das cenas, *nodes* (nós) e recursos dentro do motor gráfico *Godot*.

3.3 Implementação

A fase de implementação consistiu na codificação propriamente dita, utilizando a linguagem *GDScript* na versão estável 4.6.1¹ do motor gráfico *Godot*. Nesta etapa, os requisitos e o *design* foram convertidos em código funcional, integrando elementos visuais e sonoros à lógica de programação para criar a ferramenta de apoio ao aprendizado.

As plataformas escolhidas para o desenvolvimento do sistema foram *desktop* e *web*. Essa escolha foi realizada visando proporcionar um ambiente de estudo mais focado e controlado, reduzindo possíveis distrações, comuns em dispositivos móveis, como notificações constantes de aplicativos e redes sociais. Dessa forma, o usuário pode dedicar um momento específico ao estudo do conteúdo relacionado às normas de trânsito, utilizando o jogo como uma ferramenta complementar de aprendizagem. Ainda assim, considerando a crescente utilização de dispositivos móveis para atividades educacionais, não se descarta a possibilidade de um porte futuro para plataformas mobile, ampliando a acessibilidade do software (ver Capítulo 6.1).

¹ <https://godotengine.org/article/maintenance-release-godot-4-6-1/>

Para o desenvolvimento do jogo foi escolhido o motor gráfico *Godot*, uma ferramenta de desenvolvimento de jogos *open source*, utilizada tanto em projetos acadêmicos quanto independentes. A escolha dessa ferramenta se deu principalmente por sua longevidade e sustentabilidade tecnológica (Bradfield, 2018), uma vez que, por ser um software de código aberto, não depende de decisões comerciais de empresas proprietárias para sua continuidade. Além disso, a *Godot* apresenta características importantes para o escopo do projeto, como baixo consumo de recursos, facilidade de prototipagem e suporte multiplataforma (Bradfield, 2018), fatores que contribuem para o desenvolvimento e manutenção do software ao longo do tempo.

Para a implementação do sistema de ranking (*leaderboard*), utilizou-se a API *SimpleBoards*², selecionada por sua arquitetura leve e facilidade de integração via requisições REST. A escolha justifica-se pela necessidade de um *backend* persistente que não exigisse a manutenção de um servidor próprio, permitindo que o foco do desenvolvimento permanecesse nas mecânicas pedagógicas do jogo.

A comunicação entre o jogo e a API foi realizada através do nó *HTTPRequest* da *Godot*. O processo seguiu o fluxo padrão de comunicação *web* (Fielding, 2000):

- Envio de Pontuação: Ao final de cada partida, o sistema envia uma requisição *POST* contendo o identificador do jogador, seu nome e a pontuação obtida, utilizando uma chave de API (*API Key*) para autenticação básica.
- Recuperação de Dados: Para a exibição do ranking global, o jogo realiza uma requisição *GET*, recebendo como resposta um objeto em formato *JavaScript Object Notation* (JSON).
- Tratamento de Dados: Utilizou-se a classe JSON nativa do *GDScript* para converter a resposta da API em um dicionário de dados, permitindo a ordenação e exibição dinâmica dos nomes e pontuações na interface do usuário (*User Interface* (UI)).

Essa integração via API externa assegura que os dados de desempenho dos candidatos sejam armazenados de forma centralizada, promovendo um elemento de gamificação que incentiva a competitividade saudável e o engajamento contínuo com o conteúdo das normas de trânsito.

3.4 Testagem

O processo de avaliação e testagem do sistema dividiu-se em duas etapas complementares: a testagem técnica do código-fonte e a avaliação da qualidade pedagógica e de usabilidade do artefato sob a perspectiva dos usuários. Para a primeira etapa, o projeto integrou o uso do *framework* GUT³ para a execução de testes unitários, com foco na validação individual de funções e módulos cruciais do *software* (como o sistema de pontuação, a verificação de acertos

² <https://simpleboards.dev/>

³ <https://godotengine.org/asset-library/asset/1709>

nas simulações e o fluxo de navegação pelas interfaces), assegurando a integridade técnica do programa antes de sua distribuição.

Para a segunda etapa, visando mensurar a qualidade do jogo e a aceitação do *software*, estruturou-se uma metodologia de avaliação com usuários reais embasada no MEEGA+ proposto por Petri, Wangenheim e Borgatto (2019). Adotou-se o *one-shot post-test only design*, um delineamento experimental de grupo único no qual o candidato à obtenção CNH interage com o jogo e, imediatamente após a sessão de jogo, responde a um questionário estruturado em uma escala Likert de cinco pontos, variando de “discordo totalmente” a “concordo totalmente” (Petri; Wangenheim; Borgatto, 2019).

Com a intenção de otimizar a coleta de dados no ambiente de autoestudo e evitar a fadiga do respondente, a qual reduz a taxa e a confiabilidade de respostas quando os formulários são excessivamente longos, optou-se pela utilização de uma versão compactada do instrumento original do MEEGA+. O questionário foi reduzido para oito itens classificados como essenciais, selecionados visando a adequação da pergunta em relação ao jogo. Separadas nas 2 dimensões (Usabilidade e Experiência do Jogador), as subdimensões que foram escolhidas são:

- Usabilidade: Mensurada por meio de três itens cobrindo as subdimensões de Estética (avaliação do *design* visual e interface), Aprendizabilidade (facilidade em compreender a mecânica básica do jogo) e Operabilidade (clareza das regras apresentadas).
- Experiência do Jogador: Composta por cinco itens focados nas percepções psicossociais e pedagógicas dos candidatos, cobrindo as dimensões de Desafio (adequação da dificuldade temporal), Diversão, Atenção Focada (nível de envolvimento e imersão nas simulações), Relevância (associação com a ementa da formação de condutores) e Aprendizagem Percebida (retenção do conteúdo sobre a legislação viária).

Ao preservar apenas as questões estatisticamente mais significativas do modelo original, de acordo com as cargas fatoriais descritas por Petri, Wangenheim e Borgatto (2019), a metodologia adquire maior viabilidade operacional, evitando a fadiga sem perder o rigor científico de validação. Os dados numéricos coletados por este questionário simplificado fornecem as informações necessárias para compor as análises descritivas apresentadas no Capítulo 5.

4 DESENVOLVIMENTO

Este capítulo apresenta o desenvolvimento do projeto, utilizando como base os conceitos e exemplos abordados no capítulo anterior. Nesta etapa, são elicitados os requisitos, como mostra a seção 4.1, projetado a estrutura e o fluxo do sistema, na seção 4.2, e por fim, é mostrado a implementação do software produto desse trabalho (Seção 4.3).

4.1 Elicitação de requisitos

Para a elicitação de requisitos, foi optado em utilizar um formato de *user stories* ou histórias de usuário (Raharjana; Siahaan; Fatichah, 2021; Wautelet *et al.*, 2017). Os quadros 1, 2, 3, 4, 5, 6 e 7 representam as histórias de usuário e os critérios de aceitação escolhidos para descrever o sistema produzido por este trabalho.

Quadro 1 – História de usuário 1: Iniciar o jogo

Como um...	Candidato à CNH
Eu quero...	Iniciar o jogo pelo menu principal
Para que...	Eu possa acessar os modos disponíveis de aprendizado

Fonte: Autoria própria (2026)

- A tela inicial deve carregar ao abrir o aplicativo.
- Devem estar visíveis os botões para “Modo Desafio”, “Modo Treino”, “Placar de Líderes” e o espaço para inserir o nome do usuário.
- Os botões devem responder ao clique e redirecionar o jogador para a tela correspondente.
- Deve existir um botão para sair do jogo.

Quadro 2 – História de usuário 2: Inserir nome de jogador

Como um...	Candidato à CNH
Eu quero...	Inserir o meu nome na caixa de texto do menu
Para que...	Eu possa jogar o modo desafio do jogo

Fonte: Autoria própria (2026)

- Deve haver um campo de texto visível e interativo para a entrada do nome do usuário.
- O sistema não deve permitir o início do Modo Desafio se o campo estiver vazio, exibindo um aviso na tela.
- O campo deve ter um limite máximo de 17 caracteres para não quebrar a formatação do Placar de Líderes.

Quadro 3 – História de usuário 3: Jogar modo desafio

Como um...	Candidato à CNH
Eu quero...	Jogar o modo desafio do jogo
Para que...	Eu possa testar meus conhecimentos em desafios progressivamente mais difíceis

Fonte: Autoria própria (2026)

- O jogo deve começar com a dificuldade base, definida no código.
- A dificuldade deve aumentar de acordo com o número de fases concluídas corretamente.
- O jogador deve começar com 3 “vidas” restantes.
- Ao acabar as “vidas”, o jogo deve ser interrompido e redirecionar para a tela de fim de jogo.

Quadro 4 – História de usuário 4: Jogar modo treino

Como um...	Candidato à CNH
Eu quero...	Jogar fases individualmente no modo treino
Para que...	Eu possa memorizar e treinar situações específicas de trânsito

Fonte: Autoria própria (2026)

- A tela deve apresentar uma lista com as fases disponíveis para seleção livre.
- Ao selecionar um desafio, deve aparecer na tela a questão referente a ele.
- Deve estar visível o botão de “Jogar” para jogar o desafio selecionado.
- O jogador não deve possuir “vidas” neste modo.
- Ao concluir uma fase, erro ou acerto, o jogador deve ter a opção de ver a questão ou voltar ao menu de seleção.
- A pontuação obtida neste modo não deve ser enviada para o placar de líderes.
- Deve existir um botão para voltar ao menu principal.

Quadro 5 – História de usuário 5: Realizar um desafio

Como um...	Candidato à CNH
Eu quero...	Interagir com os elementos da fase para resolver o desafio proposto
Para que...	Eu consiga completar a simulação corretamente

Fonte: Autoria própria (2026)

- Os elementos visuais interativos devem emitir um *feedback* claro ao serem clicados.
- O sistema deve validar instantaneamente se a interação foi a correta.
- Em caso de acerto, o jogo deve emitir um *feedback* positivo e seguir para a próxima tela.
- Em caso de erro, o jogo deve emitir um *feedback* negativo. A fase deve continuar até que o jogador clique na interação correta ou que o tempo acabe.

Quadro 6 – História de usuário 6: Visualizar questão oficial

Como um...	Candidato à CNH
Eu quero...	Visualizar a questão original do Banco Nacional de Questões após a fase
Para que...	Eu possa compreender a resposta correta e reforçar meu aprendizado

Fonte: Autoria própria (2026)

- Uma imagem da questão original do Banco Nacional de Questões deve aparecer na tela de fim de jogo se o jogador clicar no botão “ver questão”.
- A alternativa correta deve estar claramente destacada em verde para o jogador.
- Deve haver um botão claro e funcional para fechar a visualização da questão original.

Quadro 7 – História de usuário 7: Visualizar placar de líderes

Como um...	Candidato à CNH
Eu quero...	Visualizar o placar de líderes
Para que...	Eu possa comparar minha pontuação com a de outros jogadores

Fonte: Autoria própria (2026)

- O sistema deve realizar uma requisição à API para mostrar o ranking mais recente.
- O placar deve exibir os nomes e pontuações em ordem decrescente.
- Deve estar visível o botão “Atualizar”, que atualiza as entradas no placar de líderes ao ser clicado.
- Deve existir um botão para voltar ao menu principal.

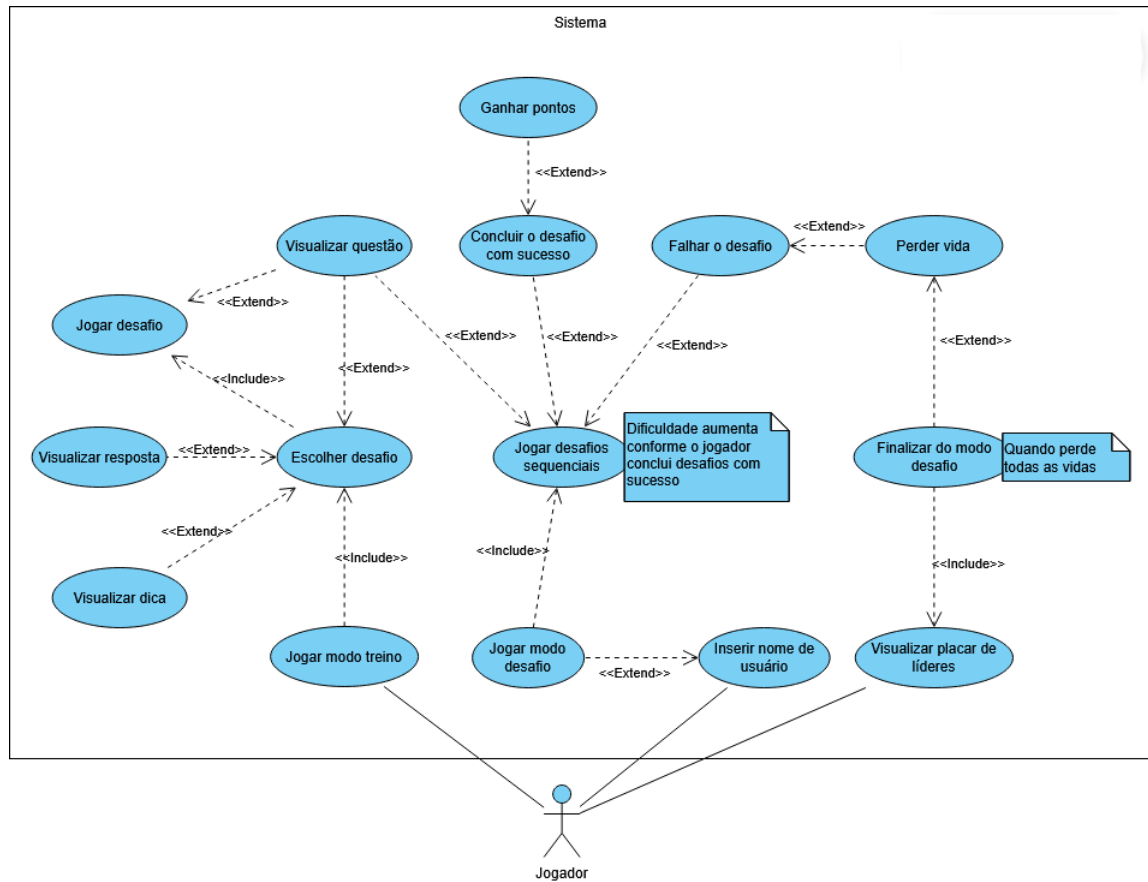
Além das histórias de usuário, uma outra forma de definir os requisitos é pelo diagrama de casos de uso. A Figura 2 apresenta o Diagrama de Caso de Uso UML do sistema proposto, ilustrando as principais interações entre o jogador e as funcionalidades do jogo educacional desenvolvido.

A gamificação neste projeto proposto, similar ao do jogo da campanha *Dumb Ways to Die*, consiste principalmente na utilização dos seguintes elementos de jogos, segundo Sauter (2024): Pontos, Competição e Desafio.

De maneira similar ao jogo supracitado, o jogador realiza desafios propostos pela fase para alcançar um objetivo específico, conforme o tempo é decrementado de maneira proporcional à progressão total do jogador. Inicialmente o jogador possui três “vidas”, que são perdidas após falhar na simulação, ou não conseguir realizar o desafio até o final do tempo pré-determinado. Ao perder todas as “vidas”, o jogo encerra, contabilizando a pontuação final do jogador e voltando para o menu inicial.

Os desafios descritos anteriormente, são simulações de situações reais caracterizadas por perguntas do Banco Nacional de Questões da SENATRAN. Nesses desafios o usuário precisa interagir com a simulação por meio de cliques do *mouse* realizando a ação que parece correta naquele momento. Ao término da fase, a questão demonstrada na simulação pode ser

Figura 2 – Diagrama de Caso de Uso UML



Fonte: Autoria própria (2026)

visualizada pelo usuário. Esse recurso serve tanto para fixar o conteúdo visto na simulação, quanto para indicar o que o jogador deve fazer, caso ele não tenha entendido.

O jogo proposto possui 2 modos distintos. Um principal, contabilizando os pontos da forma descrita anteriormente, e outro de treino, sendo possível jogar os desafios individualmente. O jogo também conta com um placar de líderes mostrando as melhores pontuações dos outros jogadores, caracterizando o elemento de Competição de Sauter (2024).

Um exemplo de uma situação proposta para o usuário seria: um ambiente no qual o veículo está aumentando as RPM (Rotações por Minuto) e o usuário deve pressionar a embreagem (clcando em sua respectiva imagem) e trocar de marcha (novamente clicando em sua respectiva imagem) para finalizar a fase com sucesso. Caso as RPM atinjam um valor muito alto (contador de RPM acende vermelho), o jogador perde a fase, consequentemente perdendo uma vida. A simulação, vista na fase, é baseada em uma questão retirada diretamente do Banco Nacional de Questões, e pode ser visualizada pelo jogador após o término da fase.

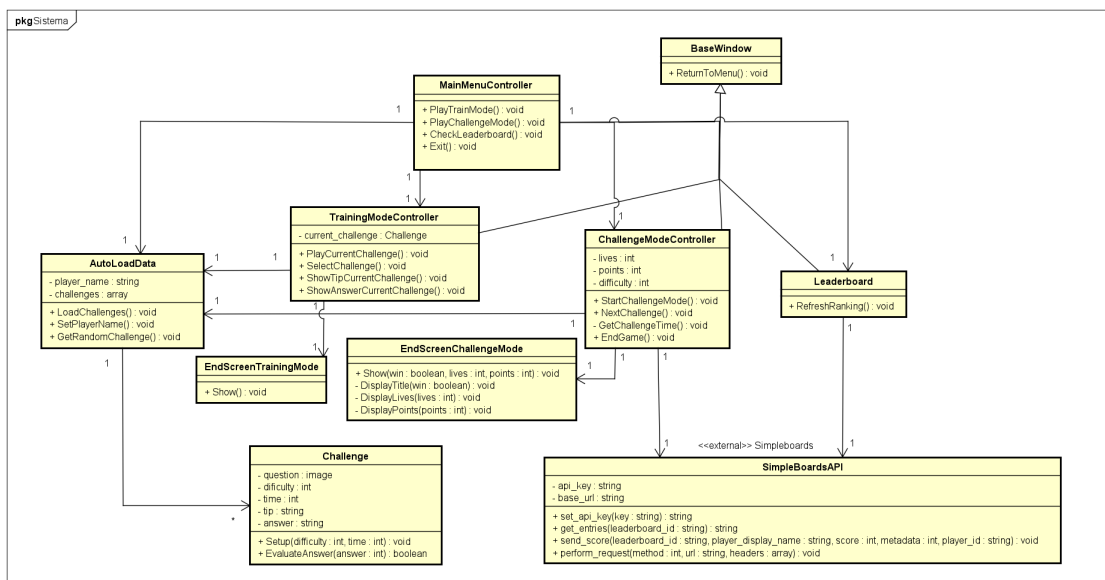
4.2 Projeto do sistema

4.2.1 Estrutura do sistema

A estrutura do software foi projetada de forma modular, buscando separar as responsabilidades entre os diferentes componentes do sistema. Para representar essa organização, foi elaborado um Diagrama de Classes UML que ilustra os principais elementos estruturais do jogo e as relações entre eles.

A arquitetura do sistema é baseada em controladores responsáveis pela lógica de cada funcionalidade principal do jogo, além de classes responsáveis pela representação de dados e pela comunicação com serviços externos. A Figura 3 apresenta o Diagrama de Classes UML desenvolvido para representar essa estrutura.

Figura 3 – Diagrama de Classes UML



Fonte: Autoria própria (2026)

O fluxo inicial do sistema é controlado pela classe *MainMenuController*, responsável por gerenciar as interações do usuário no menu principal do jogo. A partir dessa interface, o jogador pode inserir o seu nome, selecionar entre os modos de jogo disponíveis, visualizar o placar de líderes ou encerrar a aplicação. Dessa forma, essa classe atua como ponto central de navegação entre as diferentes funcionalidades do sistema.

Os modos de jogo são gerenciados por controladores específicos. O modo desafio é administrado pela classe *ChallengeModeController*, que mantém informações relacionadas ao estado da partida, como número de vidas do jogador, pontuação atual e nível de dificuldade. Essa classe também é responsável por selecionar os desafios de forma aleatória, controlar o tempo disponível e a dificuldade para resolução dos desafios e mostrar a tela de desafio

finalizado após o término dele. Essa tela de finalização é responsável por mostrar os pontos atuais do jogador e a quantidade de vidas restantes.

Já o modo de treinamento é gerenciado pela classe *TrainingModeController*, cujo objetivo é permitir que o jogador pratique desafios individualmente, sem as penalidades presentes no modo desafio. Esse modo oferece uma experiência mais livre de aprendizado, permitindo que o jogador explore o conteúdo educacional de forma mais flexível.

Os desafios do jogo são representados pela classe *Challenge*, que encapsula as informações necessárias para a execução de cada atividade, incluindo a imagem da questão que descreve a situação apresentada ao jogador, o nível de dificuldade associado e o tempo disponível para resolução.

O carregamento dos desafios e do nome do usuário é realizado pela classe *AutoLoadData*. Essa classe centraliza o acesso aos dados do jogo, armazenando informações como o nome do jogador e a lista de desafios disponíveis. Dessa forma, evita-se a duplicação de dados entre diferentes partes do sistema.

A interface gráfica compartilhada entre diferentes telas do jogo é representada pela classe *BaseWindow*, que fornece funcionalidades comuns de navegação, como o retorno ao menu principal. Essa abordagem reduz redundâncias na implementação das diferentes interfaces do sistema.

Por fim, o sistema também integra um mecanismo de placar de líderes por meio da classe *Leaderboard*, responsável por atualizar e exibir as pontuações registradas pelos jogadores. A comunicação com o serviço externo de armazenamento de pontuações é realizada por meio da classe *SimpleBoardsAPI*, que encapsula as requisições necessárias para envio e recuperação de dados do placar.

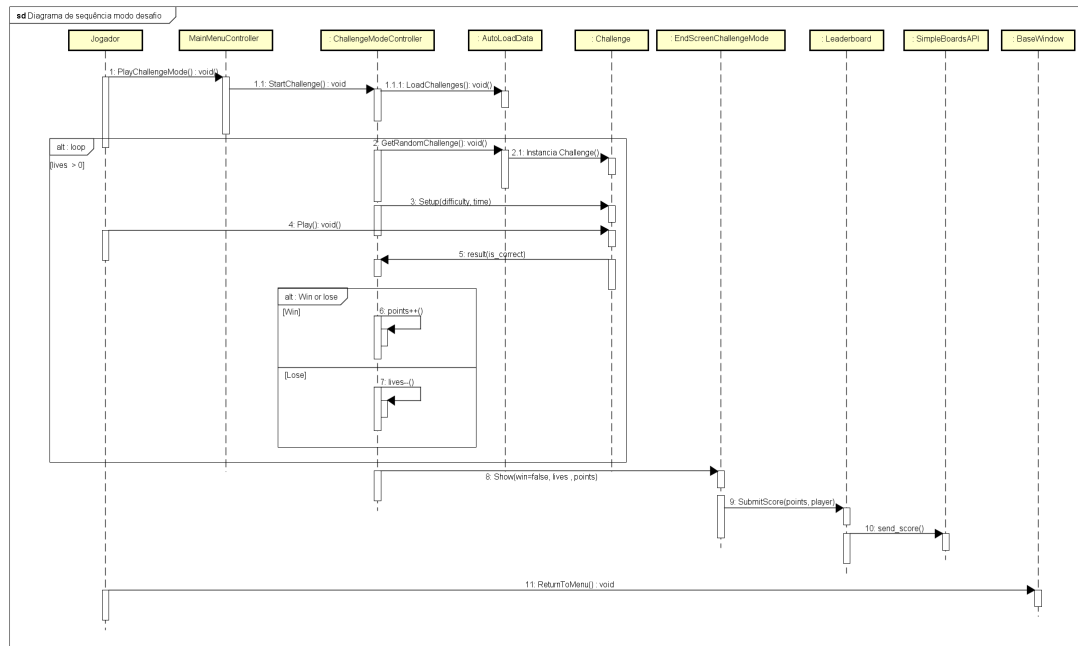
Essa organização estrutural permite separar claramente as responsabilidades de cada componente do sistema, facilitando a manutenção do código, a expansão futura das funcionalidades e a compreensão geral da arquitetura do software desenvolvido.

4.2.2 Fluxos do sistema

Além da representação estrutural apresentada pelo diagrama de classes, também foram elaborados dois Diagramas de Sequência UML para ilustrar os 2 fluxos principais do jogo, um de uma partida no modo desafio (Figura 4) e outro da seleção de desafios no modo de treino (Figura 5). Esse tipo de diagrama permite visualizar a ordem das interações entre os diferentes componentes do sistema durante a execução do jogo, evidenciando como os controladores, os dados e os serviços externos se comunicam ao longo da experiência do usuário.

O Diagrama de Sequência UML do modo desafio ilustra a dinâmica competitiva do sistema, na qual o jogador é submetido a uma progressão de dificuldades com foco na retenção de conteúdo por meio de recompensas e penalidades. O fluxo inicia-se no *MainMenuController* por meio da chamada *PlayChallengeMode()*, que transfere o controle para o *ChallengeModeCon-*

Figura 4 – Diagrama de Sequência UML do modo desafio



Fonte: Autoria própria (2026)

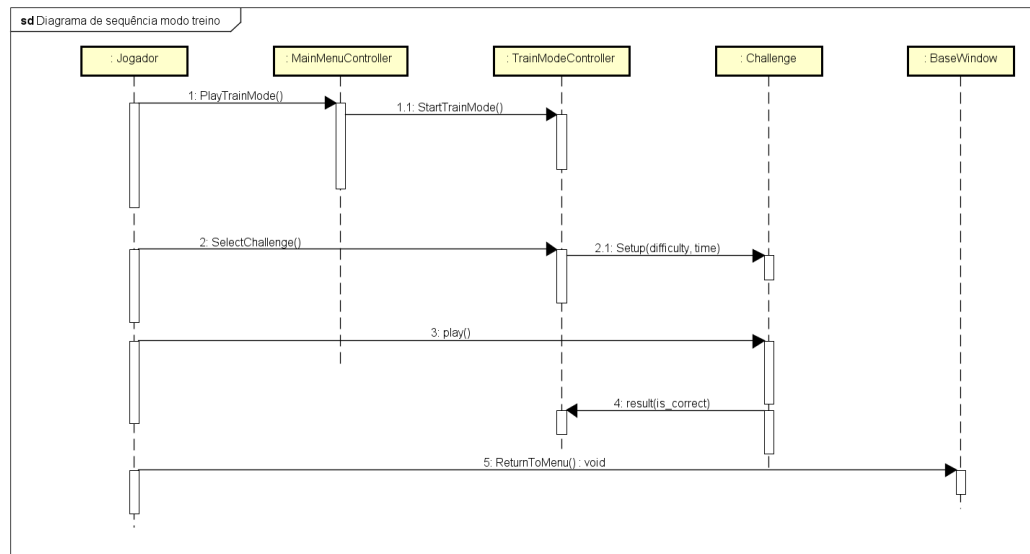
troller. Após o carregamento dos dados via *AutoLoadData*, o sistema entra em uma estrutura de repetição que permanece ativa enquanto o jogador possuir vidas disponíveis.

Dentro deste ciclo, os desafios são instanciados aleatoriamente, e a cada interação do jogador, o controlador avalia o resultado enviado pela classe *Challenge*. Caso a resposta seja correta, a pontuação é incrementada; em caso de erro, uma vida é decrementada. Ao final de cada desafio, o sistema exibe a tela de encerramento (*EndScreenChallengeMode*), porém o usuário perde todas as vidas, ao invés do botão de próximo desafio, é exibido o botão de finalizar partida, este botão realiza a submissão da pontuação final para o *Leaderboard* e volta ao menu principal. Esta tela de encerramento, por sua vez, comunica-se com a *SimpleBoardsAPI* para registrar o desempenho no ranking global, consolidando o elemento de competição do jogo.

Por outro lado, o diagrama de sequência do modo de treino apresenta uma estrutura voltada à prática livre e ao *microlearning*, permitindo que o candidato à habilitação foque em situações específicas de trânsito. O fluxo começa com a ativação do *TrainModeController* pelo menu principal. Após a listagem dos desafios pelo *AutoLoadData*, o jogador assume o controle da experiência ao utilizar o método *SelectChallenge*, o que possibilita o estudo direcionado de temas nos quais possua maior dificuldade.

Diferente do modo anterior, nota-se a ausência de mecanismos de pontuação global ou perda de vidas. Durante a execução do método *play*, o foco reside no *feedback* visual imediato gerado pelo sinal *result*. Esse retorno permite que o usuário aprenda com o erro em um ambiente seguro e sem pressões externas, reforçando o caráter educativo da ferramenta. O

Figura 5 – Diagrama de Sequência UML do modo treino



Fonte: Autoria própria (2026)

encerramento do modo ocorre por meio da classe *BaseWindow*, que gerencia o retorno seguro do jogador ao menu principal.

4.2.3 Testes unitários

Durante o desenvolvimento do software, também foi considerada a implementação de testes automatizados com o objetivo de verificar o funcionamento correto de partes específicas do sistema. Para isso, foi utilizada a ferramenta GUT¹, um *framework* de testes unitários desenvolvido especificamente para o *Godot*.

O GUT permite a criação de testes automatizados para *scripts* desenvolvidos em *GDS-crypt*, possibilitando a validação de métodos, verificação de resultados esperados e identificação de comportamentos incorretos durante a execução do código. Essa abordagem contribui para aumentar a confiabilidade do software e facilitar a identificação de erros ao longo do processo de desenvolvimento.

No contexto deste projeto, os testes unitários são aplicados principalmente em componentes responsáveis pela lógica do jogo, como controladores de modo de jogo e manipulação de desafios. Dessa forma, torna-se possível verificar o correto funcionamento de operações como seleção de desafios, controle de pontuação e gerenciamento de estados da partida.

A utilização de testes automatizados também contribui para a manutenção futura do software, permitindo que alterações no código sejam validadas rapidamente por meio da execução dos testes previamente definidos.

¹ <https://godotengine.org/asset-library/asset/1709>

4.3 Implementação do sistema

Após a definição da arquitetura do sistema, dos requisitos e da modelagem estrutural apresentada nas seções anteriores, foi realizada a implementação do software utilizando o motor gráfico *Godot*. Durante essa etapa, todos os componentes previamente planejados foram desenvolvidos, incluindo os controladores de modo de jogo, o sistema de carregamento de desafios, a integração com o placar de líderes e as interfaces principais do jogo.

Além dos elementos descritos durante a fase de planejamento, algumas funcionalidades adicionais foram implementadas com o objetivo de melhorar a organização do código e a comunicação entre os diferentes componentes do sistema. Essas decisões de implementação aproveitaram recursos próprios do *Godot*, contribuindo para uma arquitetura mais modular e desacoplada.

4.3.1 Comunicação entre componentes por meio de sinais

Durante a implementação, foi utilizado o sistema de sinais (*signals*) do *Godot* para realizar a comunicação entre diferentes partes do sistema. Os sinais são um mecanismo de comunicação baseado em eventos que permite desacoplar os componentes da aplicação, evitando dependências diretas entre classes².

No contexto do jogo desenvolvido, desafios individuais emitem sinais personalizados ao final de sua execução, informando ao controlador do modo de jogo o resultado da interação do jogador. Esses sinais indicam, por exemplo, se o jogador concluiu corretamente ou se falhou em finalizar o desafio.

O controlador do modo desafio escuta esses sinais e, a partir das informações recebidas, atualiza os parâmetros da partida, como pontuação, número de vidas restantes e progressão da dificuldade. Essa abordagem permite que os desafios permaneçam independentes da lógica geral do jogo, facilitando a manutenção e a expansão futura do sistema.

4.3.2 Sistema de *feedback* visual ao jogador

Para melhorar a experiência do usuário e fornecer um retorno visual imediato durante a resolução dos desafios, foi implementado um sistema de *feedback* visual baseado em efeitos personalizados. Esse sistema foi implementado para a classe base *Challenge*, ou seja, esse recurso pode ser utilizado facilmente em todos os desafios implementados.

Quando o jogador realiza uma ação dentro de um desafio, o sistema verifica se a escolha realizada está alinhada com a solução correta. A partir dessa verificação, é emitido um efeito visual indicando se o jogador está no caminho correto ou não.

² https://docs.godotengine.org/en/stable/classes/class_signal.html

Quando a escolha está correta, um efeito contendo um símbolo de verificação (*check*) verde é exibida na interface, aonde o jogador clicou. Em contrapartida, quando a escolha é incorreta, o sistema emite um efeito contendo um símbolo de erro (X) vermelho. Esse mecanismo permite que o jogador compreenda rapidamente as consequências de suas decisões dentro do desafio, reforçando o caráter educativo do jogo.

A Figura 6 apresenta exemplos do sistema de *feedback* visual implementado no jogo.

Figura 6 – Exemplos do *feedback* visual



Fonte: Autoria própria (2026)

4.3.3 Testes unitários

Com o objetivo de aumentar a confiabilidade do software desenvolvido, foram implementados testes unitários utilizando a ferramenta GUT. O GUT é uma ferramenta de testes automatizados desenvolvido especificamente para o *Godot*, permitindo a validação de *scripts* escritos em *GDScript*.

Durante o processo de desenvolvimento, foram criados testes unitários para as principais classes do sistema, especialmente aquelas responsáveis pela lógica do jogo. Para cada classe implementada, foi desenvolvido ao menos um teste com o objetivo de verificar o funcionamento correto de seus métodos principais.

Essa abordagem permite validar o comportamento esperado do sistema, identificando possíveis inconsistências durante o desenvolvimento e contribuindo para uma maior robustez do software. Além disso, a presença de testes automatizados facilita futuras modificações no código, permitindo que alterações sejam verificadas rapidamente por meio da execução dos testes existentes.

O Código-fonte 4.1 mostra um exemplo da implementação de um teste unitário da classe *MainMenuController* utilizando o GUT. Este código implementa os testes *test_save_name_valid*, *test_save_name_empty*, *test_ready_loads_player_name*, que testam se é um nome válido, se o tratamento de nome vazio está correto e se está carregando corretamente o nome do jogador no campo principal. Os métodos *before_each* e *after_each* são executados automaticamente antes

Código-Fonte 4.1 – Exemplo de teste unitário da classe MainMenuController utilizando GUT

```

1 extends GutTest
2
3 var controller
4
5 func before_each():
6     controller = MainMenuController.new()
7
8     controller.name_input = LineEdit.new()
9     controller.warning_name = Label.new()
10    controller.warning_name.visible = false
11
12    add_child(controller)
13    controller.add_child(controller.name_input)
14    controller.add_child(controller.warning_name)
15
16 func after_each():
17     controller.free()
18
19 func test_save_name_valid():
20     controller.name_input.text = "Pedro"
21
22     var result = controller.save_name_autoload_data()
23
24     assert_true(result)
25     assert_eq(AutoloadData.player_name, "Pedro")
26     assert_false(controller.warning_name.visible)
27
28 func test_save_name_empty():
29     controller.name_input.text = ""
30
31     var result = controller.save_name_autoload_data()
32
33     assert_false(result)
34     assert_true(controller.warning_name.visible)
35
36 func test_ready_loads_player_name():
37     AutoloadData.player_name = "Guilherme"
38
39     controller._ready()
40
41     assert_eq(controller.name_input.text, "Guilherme")

```

Fonte: Autoria própria (2026)

e após cada teste, respectivamente. Estes são utilizados para instanciar e deletar os elementos utilizados nos testes.

4.3.4 Extensão e adição de novos desafios

Durante o desenvolvimento do sistema, não foi implementado um mecanismo automatizado para criação dinâmica de novos desafios por meio da interface do jogo. Essa decisão foi tomada considerando que os desafios presentes no sistema possuem características específicas e personalizadas, envolvendo lógica própria, elementos visuais e interação particular com o jogador.

Apesar disso, a arquitetura do software foi projetada de forma modular, permitindo que novos desafios sejam adicionados diretamente no código do projeto. Cada desafio é implemen-

tado como uma unidade independente que se comunica com os controladores do jogo por meio do sistema de sinais descrito anteriormente.

Na prática, durante o desenvolvimento deste trabalho, a criação da estrutura básica de um novo desafio demandava aproximadamente entre cinco e dez minutos de implementação. Esse tempo correspondia principalmente à definição da lógica estrutural do desafio e sua integração com os controladores do sistema.

Entretanto, etapas como seleção de elementos visuais, criação ou adaptação de *assets*, ajustes de interface e polimento da experiência do jogador demandaram um tempo maior de desenvolvimento. Ainda assim, a organização estrutural adotada no projeto demonstrou facilitar significativamente o processo de expansão do conteúdo do jogo.

A implementação de ferramentas que permitam a criação de novos desafios de forma mais automatizada ou acessível para usuários finais constitui uma possibilidade interessante de evolução do sistema, sendo discutida com maior detalhe na seção de trabalhos futuros (Seção 6.1).

5 RESULTADOS E DISCUSSÕES

Neste capítulo, apresenta-se o artefato desenvolvido como resultado final deste trabalho. O software foi nomeado como “JERED: Jeitos Engraçados de Reprovar no Exame do Detran”. O nome, além de ser um acrônimo, é uma homenagem e inspiração direta ao jogo da campanha *Dumb Ways to Die*, adaptando o conceito de situações absurdas para o contexto da formação de condutores no Brasil. O ícone do jogo pode ser visto na Figura 7

Figura 7 – Ícone do jogo.



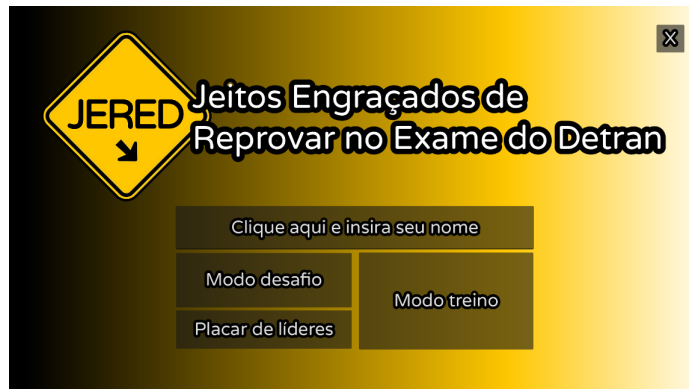
Fonte: Autoria própria (2026)

5.1 JERED: Jeitos Engraçados de Reprovar no Exame do Detran

O jogo foi desenvolvido para execução em *desktop* e em navegadores *web*, visando a portabilidade e facilidade de acesso pelos candidatos. Seguindo as diretrizes do Marco Legal dos Games (Brasil, 2024), o projeto está disponível de forma aberta. O código-fonte pode ser consultado no *GitHub*¹ e a versão para jogar via navegador está hospedada na plataforma *Itch.io*².

A interface do menu principal pode ser vista na Figura 8, permitindo que o usuário insira seu nome, jogue o modo desafio, o modo de treino e visualize o placar de líderes.

Figura 8 – Menu principal do jogo.



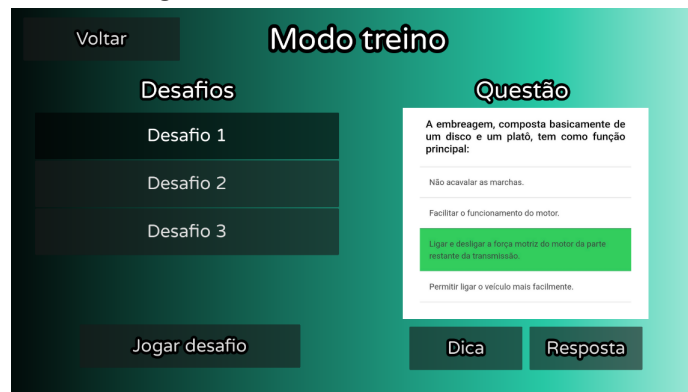
Fonte: Autoria própria (2026)

¹ <https://github.com/GuilhermeCerdeira/JERED>

² <https://prpedro.itch.io/jered>

O modo treino (Figura 9), foi implementado para oferecer um ambiente livre de pressão, focado na fixação do conteúdo. O jogador pode selecionar o desafio na parte esquerda da tela, visualizar a questão na parte direita, e escolher jogar o desafio selecionado pressionando o botão “Jogar desafio”, no canto inferior esquerdo da tela. O jogador também tem a opção de ver uma dica ou ver a resposta do desafio selecionado por meio dos botões abaixo da questão. Ambos os botões mostram um texto explicativo indicando o que o usuário deve realizar.

Figura 9 – Interface do modo treino.



Fonte: Autoria própria (2026)

Para incentivar o jogador, o jogo conta com um placar de líderes, referente a pontuação obtida no modo desafio, representado pela Figura 10, que permite comparar o desempenho entre usuários de diferentes localidades, adicionando uma camada competitiva ao estudo e expondo um dos aspectos da gamificação do software desenvolvido.

Figura 10 – Interface do placar de líderes.



Fonte: Autoria própria (2026)

Durante os desafios, como o exemplificado na Figura 11, o jogador deve interagir com situações que simulam questões da prova teórica. O objetivo é realizar a ação que indica a resposta correta da questão relacionada a situação, memorizando a alternativa correta através do reforço visual e a repetição.

Neste exemplo citado anteriormente, a questão do Banco Nacional de Questões está relacionada com a embreagem do carro, explicando a sua funcionalidade e utilização, logo na

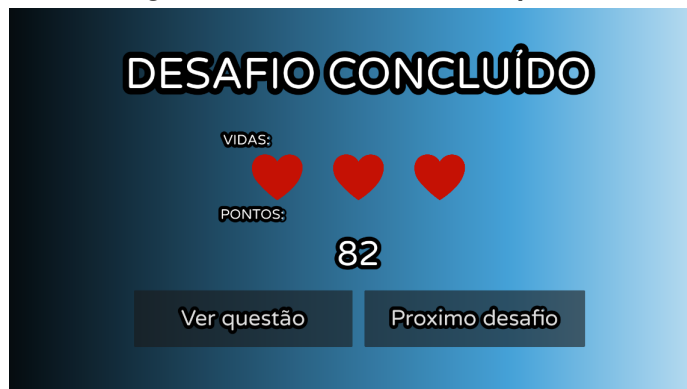
Figura 11 – Exemplo de um desafio.



Fonte: Autoria própria (2026)

simulação o jogador precisa, pressionar a embreagem, clicando no pedal correto, e trocar a marcha, clicando na alavanca do câmbio, para cumprir o desafio com sucesso. No modo desafio, a tela de sucesso é representada na Figura 12.

Figura 12 – Tela de desafio completo.



Fonte: Autoria própria (2026)

Caso o jogador apenas tente trocar a marcha, ou o tempo restante, mostrado pela barra verde de progresso, acabe, o jogador falha o desafio, mostrando a tela de derrota (Figura 13) e perdendo uma vida.

Em ambas as telas de finalização, vitória e derrota, o jogador pode visualizar a questão representada na simulação ou ir para o próximo desafio. Nessa tela o jogador pode ver a quantidade de pontos da partida atual e a quantidade de vida restante.

Quando não restam mais vidas, a partida do modo desafio é finalizada, mostrando a tela da Figura 14. Nessa tela final, o jogador pode, ou ver a questão relacionada com o desafio, ou finalizar a partida, registrando a pontuação dele no placar de líderes e voltando ao menu principal.

Figura 13 – Tela de desafio fracassado



Fonte: Autoria própria (2026)

Figura 14 – Tela de final da partida.



Fonte: Autoria própria (2026)

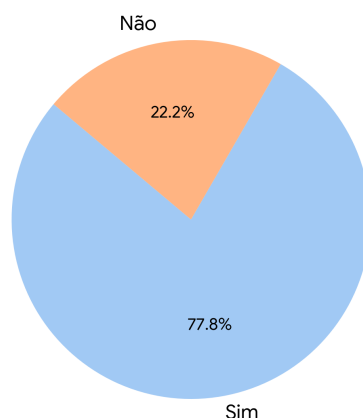
5.2 Avaliação do jogo

Esta seção destina-se à apresentação e discussão dos resultados obtidos a partir da avaliação do jogo educacional JERED. A coleta de dados foi realizada por meio de um questionário de satisfação (Apêndice A) baseado em uma versão simplificada do modelo MEEGA+. O formulário foi enviado para 15 participantes, porém 9 participantes o responderam após interagirem com o jogo. A matriz completa contendo as respostas brutas coletadas pode ser consultada no Apêndice B.

5.2.1 Perfil dos Participantes

Uma primeira etapa do questionário buscou traçar o perfil demográfico dos usuários. A amostra contou com a participação de 9 indivíduos, caracterizando-se por um público jovem, em que a maioria, aproximadamente 89%, concentra-se na faixa etária entre 18 e 26 anos.

Em relação à posse da CNH, a maioria dos participantes (77,8%) afirmou já possuir o documento, enquanto apenas 22,2% informaram não possuir, conforme ilustra a Figura 15.

Figura 15 – Proporção de participantes que possuem CNH

Fonte: Autoria própria (2026)

É importante ressaltar que o intuito principal do JERED é apoiar o autoestudo de candidatos que não possuem a CNH ou que estão em processo de obtenção. Portanto, a amostra majoritariamente composta por indivíduos já habilitados não representa o cenário ideal de testagem do público-alvo primário. Contudo, essa avaliação mostrou a perspectiva de motoristas já habilitados que, em teoria, possuem o conhecimento prático e teórico necessário para validar de forma crítica o conteúdo de legislação de trânsito abordado no jogo.

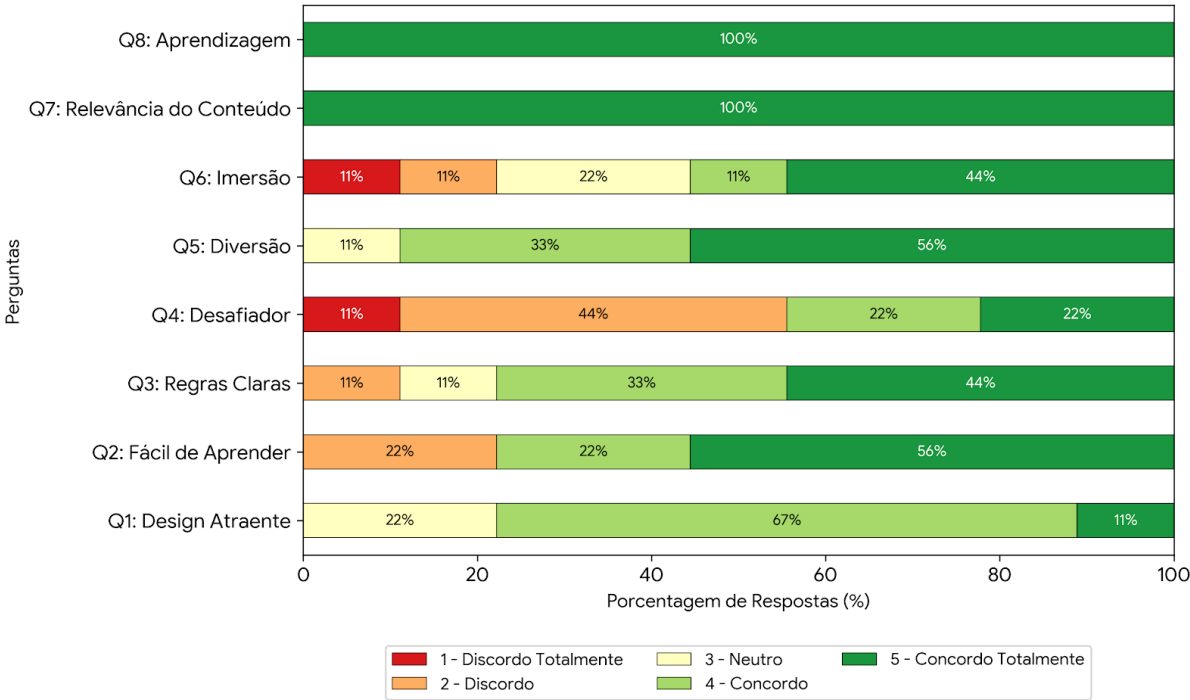
5.2.2 Respostas

Para avaliar a usabilidade e a experiência proporcionada pelo jogo, os participantes responderam a oito afirmações utilizando uma escala Likert de 5 pontos (Devellis; Thorpe, 2021), variando de “1 - Discordo Totalmente” a “5 - Concordo Totalmente”. A Figura 16 apresenta o percentual de respostas para cada uma das dimensões avaliadas.

Para compreender de forma mais clara em quais aspectos o jogo conseguiu cumprir o seu propósito e em quais ele demonstrou necessidade de aprimoramento, foi calculada a média das respostas para cada questão (onde 1 é a pior nota e 5 é a nota máxima). Os resultados estão ordenados na Figura 17.

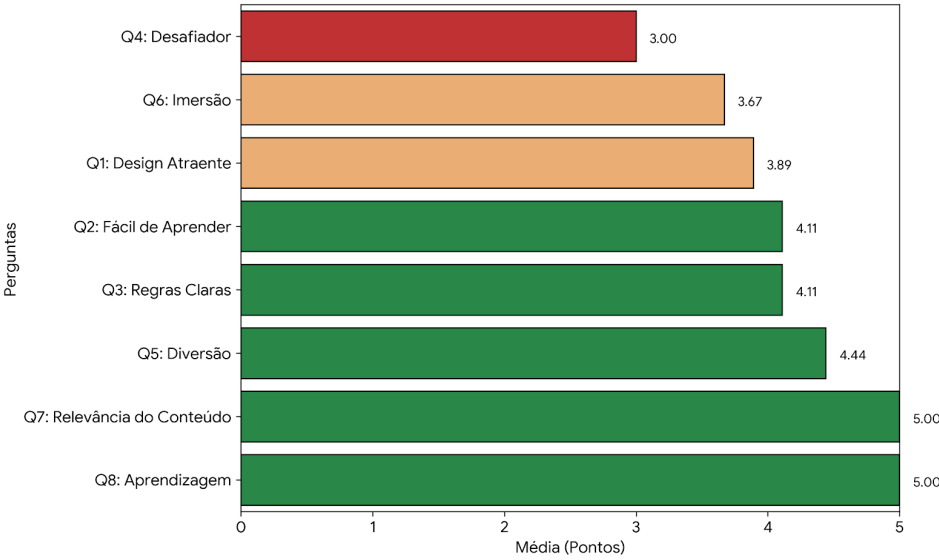
A análise das médias revela os pontos fortes do JERED. O jogo obteve a nota máxima em dois quesitos fundamentais: “Relevância do Conteúdo” (Q7) e “Aprendizagem” (Q8). Isso significa que, do ponto de vista educacional, o software cumpriu integralmente o seu propósito, deixando nítida a relação do jogo com a educação no trânsito e sendo percebido como uma ferramenta agregadora para o conhecimento dos usuários. A “Diversão” (Q5), com média de 4,44, e as dimensões de usabilidade, como “Regras Claras” (Q3) e “Fácil de Aprender” (Q2), com médias de 4,11, também foram pontos fortes da aplicação.

Figura 16 – Resultados da avaliação do jogo baseada no modelo MEEGA+ simplificado



Fonte: Autoria própria (2026)

Figura 17 – Média de avaliação por pergunta do modelo MEEGA+



Fonte: Autoria própria (2026)

Por outro lado, o gráfico também expõe onde o jogo não obteve resultados positivos e onde há margem para melhorias futuras. A afirmação “Este jogo é desafiador para mim” (Q4) obteve a menor média da pesquisa (3,00). Esse dado é um possível reflexo direto do perfil dos participantes: por serem majoritariamente habilitados, o conteúdo não apresentou um nível

de dificuldade estimulante. Isso mostra uma possível necessidade de implementar níveis de dificuldade variáveis para atender diferentes níveis de conhecimento prévio.

Além disso, a “Imersão” (Q6), com média de 3,67, e o “Design Atraente” (Q1), com média de 3,89, indicam que a parte audiovisual e narrativa não foi suficiente para fazer com que o jogador se desconectasse do ambiente ao seu redor. Embora o jogo produzido neste trabalho não esteja completo, elas apontam que futuros investimentos em polimento gráfico, UI e *design* sonoro são necessários para elevar a qualidade do engajamento estético e imersivo do jogador.

6 CONCLUSÃO

Neste capítulo são apresentadas algumas considerações sobre o sistema desenvolvido, bem como possíveis direções para a continuidade e evolução do projeto em trabalhos futuros (Seção 6.1).

A implementação realizada neste trabalho teve como foco principal a construção de uma base estrutural sólida para o sistema. Durante o desenvolvimento, buscou-se organizar a arquitetura do software de forma modular, priorizando a separação de responsabilidades entre os diferentes componentes do sistema e a comunicação entre eles por meio de mecanismos desacoplados, como sinais e controladores específicos para cada funcionalidade. Essa abordagem tem como objetivo facilitar a manutenção do código e permitir a ampliação do sistema de forma progressiva.

Dessa forma, as decisões de projeto foram direcionadas não apenas para a implementação das funcionalidades necessárias ao funcionamento do jogo educacional proposto, mas também para garantir que novas funcionalidades possam ser incorporadas futuramente com menor esforço de desenvolvimento. A estrutura atual permite, por exemplo, a inclusão de novos tipos de desafios, a modificação de regras de interação e a expansão da lógica do sistema sem a necessidade de alterações profundas na arquitetura existente.

Já os resultados obtidos na pesquisa de satisfação, embora feita com uma quantidade amostral pequena de usuários, tende a mostrar o potencial da solução desenvolvida. A totalidade dos participantes reconheceu a clareza da relação do jogo com a educação no trânsito e confirmou a sua contribuição para a aprendizagem (100% de aprovação nesses quesitos).

Além disso, os índices de usabilidade e diversão mostraram que o objetivo de engajar o usuário foi alcançado. Considerando as respostas positivas (acima da nota neutra), 77,8% dos avaliadores consideraram o jogo fácil de aprender e com regras claras, enquanto 88,9% afirmaram ter se divertido com a experiência, validando a abordagem tecnológica proposta.

Embora a amostra reduzida não permita uma afirmação definitiva de que a ferramenta soluciona o problema proposto, os indícios apontam que o JERED apresenta características promissoras. A recepção inicial sugere que *serious games* são alternativas viáveis no contexto da educação para o trânsito, podendo atuar de forma complementar para democratizar o estudo e auxiliar na formação de futuros condutores.

6.1 Trabalhos futuros

A arquitetura desenvolvida neste trabalho abre espaço para diferentes possibilidades de evolução do sistema. Embora o protótipo implementado tenha como foco a criação de um jogo educacional voltado ao apoio ao estudo das regras de trânsito, a organização modular adotada durante o desenvolvimento sugere que o software possui potencial para evoluir para uma plataforma mais ampla de criação e execução de “mini-desafios” educacionais.

Nesse contexto, uma possível direção de continuidade seria transformar o sistema em um ambiente que permita não apenas jogar os desafios existentes, mas também criar novos, dentro da própria aplicação. Tal abordagem permitiria que desenvolvedores ou educadores produzissem novos conteúdos educacionais a partir da estrutura já existente, ampliando significativamente o alcance e a reutilização do sistema.

Uma estratégia interessante para viabilizar esse tipo de expansão seria a incorporação de mecanismos baseados em programação visual, como sistemas de programação em nós (*node-based programming*) (Gregory, 2018). Nesse tipo de abordagem, a lógica das aplicações é construída por meio da conexão entre blocos ou nós que representam operações, eventos ou fluxos de execução. Ambientes desse tipo têm sido amplamente utilizados em ferramentas de desenvolvimento modernas, especialmente em motores de jogos e sistemas interativos, por permitirem a criação de comportamentos complexos de forma visual e acessível (Kuhail *et al.*, 2021).

A adoção de um sistema baseado nesse paradigma poderia permitir que novos desafios ou jogos fossem definidos por meio da composição de nós que representassem eventos do jogo, ações do jogador, verificações de regras ou mudanças de estado. Dessa forma, usuários poderiam desenvolver novas atividades educacionais sem a necessidade de modificar diretamente o código-fonte do sistema.

Um outro possível direcionamento para trabalhos futuros consiste na portabilização do software para dispositivos móveis, ampliando o alcance da ferramenta educacional. O jogo foi desenvolvido utilizando o motor gráfico *Godot*, que possui suporte nativo à exportação para múltiplas plataformas, incluindo *Android* e *iOS*, o que torna esse processo relativamente simples do ponto de vista técnico. Considerando que a interação principal do jogo se baseia em cliques do *mouse* na tela e em botões, a adaptação para interfaces *touch* tende a exigir apenas ajustes mínimos de interface e escala de elementos gráficos. Dessa forma, ações realizadas atualmente por meio do mouse poderiam ser facilmente substituídas por toques na tela, mantendo a mesma lógica de interação. A disponibilização do jogo em dispositivos móveis poderia aumentar significativamente sua acessibilidade.

REFERÊNCIAS

- AGUIAR, R. N.; FERREIRA, A. A. Gamificação como ferramenta educacional. **Revista Educação, Cultura e Sociedade**, v. 16, n. 2, p. 28–39, 2025. ISSN 2237-1648.
- ANDRADE, P. H. M. O. de; ALMEIDA, F. B. de; BITTENCOURT, R. A. Um jogo educacional 2d sobre trânsito e cidadania. *In*: ANAIS DO XIV SIMPÓSIO BRASILEIRO DE JOGOS E ENTRETENIMENTO DIGITAL (SBGAMES). 2015, Teresina. **Anais [...]** Teresina: SBC, 2015.
- BRADFIELD, C. **Godot Engine Game Development Projects**. [S.l.]: Packt Publishing, 2018.
- BRASIL. **Direção Defensiva: Trânsito seguro é um direito de todos**. Brasília, DF: , 2005. Disponível em: <https://www.gov.br/transportes/pt-br/assuntos/transito/arquivos-senatran/docs/direcaodefensiva.pdf>.
- Brasil. **Lei nº 14.852, de 3 de maio de 2024**: Cria o marco legal para a indústria de jogos eletrônicos. Brasília: Brasil, 2024. Diário Oficial da União.
- Brasil. **Lei nº 9.503, de 23 de setembro de 1997**: Institui o código de trânsito brasileiro. Brasília: Brasil, 1997. Disponível em: http://www.planalto.gov.br/ccivil_03/leis/19503.htm.
- Brasil. **Resolução CONTRAN nº 1.020, de 1º de dezembro de 2025**: Normatiza os procedimentos sobre a aprendizagem, a habilitação e a expedição de documentos de condutores. Brasília: CONTRAN, 2025. Diário Oficial da União.
- CNN Brasil. **CNH sem autoescola? Especialistas alertam sobre impactos na segurança**. [S.l.]: , 2025. Disponível em: <https://www.cnnbrasil.com.br/auto/cnh-sem-autoescola-especialistas-alertam-sobre-impactos-na-seguranca/>. Acesso em: 2026-03-27.
- DEVELLIS, R. F.; THORPE, C. T. **Scale development: Theory and applications**. [S.l.]: Sage publications, 2021.
- FADEL, L. M. *et al.* Gamificação e microlearning: Estratégias para retenção de conhecimento na era digital. **Revista Brasileira de Informática na Educação (RBIE)**, SBC v. 31, n. 1, p. 112–130, 2023.
- FERREIRA, C. *et al.* Transit kämpfer: uma proposta de jogo digital para educação no trânsito. *In*: ANAIS DO XXV WORKSHOP DE INFORMÁTICA NA ESCOLA. 2019, Porto Alegre, RS, Brasil. **Anais [...]** Porto Alegre, RS, Brasil: SBC, 2019. p. 1269–1273. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/wie/article/view/13299>.
- FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**. 2000. Tese (Doutorado) — University of California, Irvine 2000.
- G1. **Fim da obrigatoriedade de autoescolas: veja os argumentos a favor e os contra a mudança**. [S.l.]: , 2025. Disponível em: <https://g1.globo.com/podcast/o-assunto/noticia/2025/10/05/fim-da-obrigatoriedade-de-autoescolas-veja-os-argumentos-a-favor-e-os-contra-a-mudanca.ghtml>. Acesso em: 2026-03-27.
- GEE, J. P. **What Video Games Have to Teach Us About Learning and Literacy**. New York: Palgrave Macmillan, 2003.
- GREGORY, J. **Game Engine Architecture**. 3. ed. [S.l.]: CRC Press, 2018.

- HOBOLD, W.; SILVA, G.; WERLICH, C. Uma revisão da literatura sobre jogos educativos de trânsito. *In: ANAIS DO XXIV SIMPÓSIO BRASILEIRO DE JOGOS E ENTRETENIMENTO DIGITAL*. 2025, Porto Alegre, RS, Brasil. **Anais [...]** Porto Alegre, RS, Brasil: SBC, 2025. p. 1679–1687. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/sbgames/article/view/37463>.
- INFLEET. **Frota brasileira já é a maior da história**. [S.l.]: , 2025. Disponível em: <https://infleet.com.br/materiais/frota-brasileira-ja-e-a-maior-da-historia/>. Acesso em: 15 mar. 2026.
- ISMAIL, S.; ABDENNADHER, S.; ABOUELSAADAT, W. Rules on wheels: A serious game for teaching traffic signs. *In: ANAIS DA 8TH INTERNATIONAL CONFERENCE ON GAMES AND VIRTUAL WORLDS FOR SERIOUS APPLICATIONS (VS-GAMES)*. 2016, Barcelona. **Anais [...]** Barcelona: IEEE, 2016. p. 1–7.
- KUHAIL, M. A. *et al.* Characterizing visual programming approaches for end-user developers: A systematic review. **IEEE Access**, v. 9, n. 1, p. 14181–14202, 2021.
- KWON, S. Analysis on signification of actant for representation in Dumb Ways to Die - as the centre semiotic analysis of A. J. Greimas. **Journal of Korea Multimedia Society**, v. 19, n. 6, p. 1095–1105, 2016.
- MARTINEZ, K.; MENÉNDEZ-MENÉNDEZ, M. I.; BUSTILLO, A. A new measure for serious games evaluation: Gaming educational balanced (geb) model. **Applied Sciences**, v. 12, n. 22, 2022. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/12/22/11757>.
- PETRI, G.; WANGENHEIM, C. G. von; BORGATTO, A. F. Meega+: Um modelo para a avaliação de jogos educacionais para o ensino de computação. **Revista Brasileira de Informática na Educação**, v. 27, n. 3, p. 52–81, set. 2019. Disponível em: <https://journals-sol.sbc.org.br/index.php/rbie/article/view/4576>.
- PRENSKY, M. **Digital Game-Based Learning**. New York: McGraw-Hill, 2001.
- PRESSMAN, R.; MAXIM, D. B. R. **Software Engineering: A Practitioner's Approach**. McGraw-Hill Education, 2014. ISBN 9780078022128. Disponível em: <https://books.google.com.br/books?id=i8NmnAEACAAJ>.
- RAHARJANA, I. K.; SIAHAAN, D.; FATICHAH, C. User stories and natural language processing: A systematic literature review. **IEEE Access**, v. 9, n. 1, p. 53811–53826, 2021.
- RODRIGUES, M. A. F. *et al.* Beyond fun: an interactive and educational 3d traffic rules game controlled by non-traditional devices. *In: PROCEEDINGS OF THE 30TH ANNUAL ACM SYMPOSIUM ON APPLIED COMPUTING (SAC)*. 2015, Nova Iorque. **Anais [...]** Nova Iorque: ACM, 2015. p. 239–246.
- SAUTER, G. A. **Aplicativo de caminhadas fundamentado em técnicas de gamificação**. 2024. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) - Universidade Tecnológica Federal do Paraná, 2024.
- SENARATH, U. S. **Waterfall Methodology, Prototyping and Agile Development**. [S.l.], 2021. Disponível em: https://www.researchgate.net/publication/353324450_Waterfall_Methodology_Prototyping_and_Agile_Development. Acesso em: 22 abr 2026.
- TORRES, P. O. A. **Análisis comunicacional de la campaña “Dumb Ways To Die” (Maneras Tontas de Morir) de Metro Trains en 2012**. 2014. Dissertação (Mestrado) — Universidad de San Carlos de Guatemala Guatemala 2014.

WARD, M. **Has Dumb Ways To Die Been Effective?** [S.l.]: , 2015. Mumbrella. Disponível em: <https://mumbrella.com.au/dumb-ways-die-stopped-dumb-behavior-around-trains-270751>. Acesso em: 15 mar 2026.

WAUTELET, Y. *et al.* User-story driven development of multi-agent systems: A process fragment for agile methods. **Computer Languages, Systems & Structures**, Elsevier v. 50, n. 1, p. 159–176, 2017.

APÊNDICE A – Questionário de pesquisa

Este apêndice apresenta o instrumento de coleta de dados utilizado para avaliar o jogo JERED, estruturado com base em uma versão simplificada do modelo MEEGA+. O questionário foi disponibilizado aos participantes via formulário eletrônico e é dividido em duas seções.

Seção 1

- **Qual a sua idade?**

- ☐ Menos de 18 anos
- ☐ Entre 18 e 26 anos
- ☐ Acima de 26 anos

- **Você possui CNH?**

- ☐ Sim
- ☐ Não

Seção 2

As perguntas desta seção procuram mensurar a qualidade do *design*, usabilidade e potencial educativo do jogo. Todas as afirmações foram avaliadas utilizando uma escala Likert (Devellis; Thorpe, 2021) de 5 pontos, sendo: (1) Discordo Totalmente; (2) Discordo; (3) Neutro; (4) Concordo; e (5) Concordo Totalmente.

1. O *design* do jogo é atraente (interface, gráficos, etc.).
2. Foi fácil para mim aprender a jogar o JERED.
3. As regras do jogo são claras e compreensíveis.
4. Este jogo é desafiador para mim.
5. Eu me diverti com o jogo.
6. Eu esqueci sobre o ambiente ao meu redor enquanto jogava este jogo.
7. Ficou claro para mim como o conteúdo do jogo está relacionado com a Educação no Trânsito.
8. O jogo contribuiu para a minha aprendizagem na Educação no Trânsito.

APÊNDICE B – Matriz de respostas da avaliação

Este apêndice apresenta os dados brutos coletados por meio da pesquisa de satisfação apresentada no Apêndice A. A amostra é composta por 9 participantes identificados de P1 a P9.

Na Tabela 2, as colunas de Q1 a Q8 representam as oito questões relacionadas à avaliação do jogo. Os valores numéricos de 1 a 5 correspondem às respostas fornecidas pelos usuários segundo a escala Likert (Devellis; Thorpe, 2021) apresentada no questionário.

Tabela 2 – Matriz de dados brutos coletados na avaliação do JERED

Participante	Idade	CNH	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
P1	Acima de 26	Não	5	5	5	5	5	5	5	5
P2	Entre 18 e 26	Sim	4	2	2	1	4	4	5	5
P3	Entre 18 e 26	Sim	4	5	4	2	5	5	5	5
P4	Entre 18 e 26	Sim	4	5	5	5	5	5	5	5
P5	Entre 18 e 26	Sim	3	4	4	2	3	2	5	5
P6	Entre 18 e 26	Sim	4	5	5	2	4	3	5	5
P7	Entre 18 e 26	Não	3	5	5	4	4	3	5	5
P8	Entre 18 e 26	Sim	4	2	3	4	5	1	5	5
P9	Entre 18 e 26	Sim	4	4	4	2	5	5	5	5

Fonte: Autoria própria (2026)