

PPGTER/DES.46.2026.JOG

Processo de Desenvolvimento do DEBUGGERS - Jogo Analógico para apoiar o Ensino e Aprendizagem de Fundamentos de Programação

Autores

Lucas Ziquinatti Piccini Nunes
ziquinattipiccini@gmail.com

Rafael Rezende Lehnhart
lehnhart@gmail.com

Giliane Bernardi
giliane.bernardi@ufsm.br

Andre Zanki Cordenonsi
andre.cordenonsi@ufsm.br

Giani Petri
giani.petri@ufsm.br

Solange de Lurdes Pertile
solange.pertile@ufsm.br



Versão 1.0
Status: Final
Distribuição: Externa
JUNHO 2026



2026 PPGTER – Programa de Pós-Graduação em Tecnologias Educacionais em Rede

Atribuição-Não Comercial 4.0 Internacional (CC BY-NC 4.0)

Você tem o direito de compartilhar, copiar e redistribuir o material em qualquer suporte ou formato; adaptar, remixar, transformar, e criar a partir do material, de acordo com o seguinte: você deve dar o crédito apropriado, prover um link para a licença e indicar se mudanças forem feitas. Você deve fazê-lo em qualquer circunstância razoável, mas de nenhuma maneira que sugira que o licenciante apoia você ou seu uso. Você não pode usar o material para fins comerciais.



Editoria Técnica do PPGTER

Universidade Federal de Santa Maria

Av. Roraima n. 1000

Centro de Educação, Prédio 16, sala 3146

Santa Maria – RS – CEP 97105-900

Fone / FAX: 55 3220 9414

ppgter@ufsm.br

edtec.ppgter@gmail.com

Relatórios Técnicos do Programa de Pós-Graduação em Tecnologias Educacionais em Rede / Programa de Pós-Graduação em Tecnologias Educacionais em Rede, Universidade Federal de Santa Maria. – Vol. 8. n. 1 (2026) Jan/Jul. – Santa Maria: PPGTER/UFSM, 2019.

Periodicidade semestral.

1. Tecnologia Educacional.
 2. Desenvolvimento de Tecnologias Educacionais.
 3. Gestão de Tecnologias Educacionais.
- I. Universidade Federal de Santa Maria. Programa de Pós-Graduação em Tecnologias Educacionais em Rede.

Como citar este relatório:

NUNES, L.Z.P., LEHNHART, R.R., BERNARDI, G, CORDENONSI, A.Z., PETRI, G., PERTILE, S.L.
Processo de Desenvolvimento do DEBUGGERS - Jogo Analógico para apoiar o Ensino e Aprendizagem de Fundamentos de Programação. Santa Maria: 2026. Relatórios Técnicos do Programa de Pós-Graduação em Tecnologias Educacionais em Rede, v. 8., n.1. Disponível em:
<https://www.ufsm.br/cursos/pos-graduacao/santa-maria/ppgter/relatorios-tecnicos-ppgter>

Resumo

Este Relatório Técnico apresenta o projeto do *DEBUGGERS: A Batalha Contra os Bugs*, jogo educacional híbrido, desenvolvido para facilitar a aprendizagem dos fundamentos de lógica de programação de forma interativa e cooperativa, tendo como referência lúdica a linguagem de programação Python. Os jogadores assumem o papel de programadores que precisam construir algoritmos para completar desafios antes que bugs corrompam o sistema ou que o projeto atinja o modo de crise, impossibilitando sua continuidade. Como principal justificativa para o projeto, ressalta-se o desafio educacional que o ensino de lógica de programação representa, onde conceitos previamente desconhecidos ou pouco desenvolvidos nas escolas, frequentemente resultam em dificuldade de compreensão e baixa motivação por parte dos estudantes. Nesse contexto, o jogo surge como uma estratégia inovadora que busca superar essa barreira ao transformar o aprendizado em uma experiência interativa, mais palpável e engajadora, oferecendo aos alunos um ambiente seguro e controlado para reforçar o desenvolvimento de habilidades e obter mais oportunidades práticas de aprendizagem. Para o seu design, foi utilizado o ENDO-GDC, um *Game Design Canvas* para jogos educacionais endógenos, que busca melhorar a comunicação entre os diferentes membros de uma equipe interdisciplinar, facilitar a concepção do jogo e auxiliar no equilíbrio entre objetivos pedagógicos com a parte de entretenimento do game (TAUCEI, 2019). O resultado é um jogo de cartas e tabuleiro com suporte digital de aplicativo web onde a mecânica central de “codificar” com cartas está intrinsecamente ligada ao objetivo de aprendizado de estruturar o pensamento algorítmico. Diante disso, se espera que o jogo auxilie os estudantes do ensino técnico em informática e dos primeiros semestres de cursos de computação a compreenderem componentes básicos da programação como: declaração de tipos de dados e variáveis, comandos de entrada e saída, condicionais e laços de repetição.

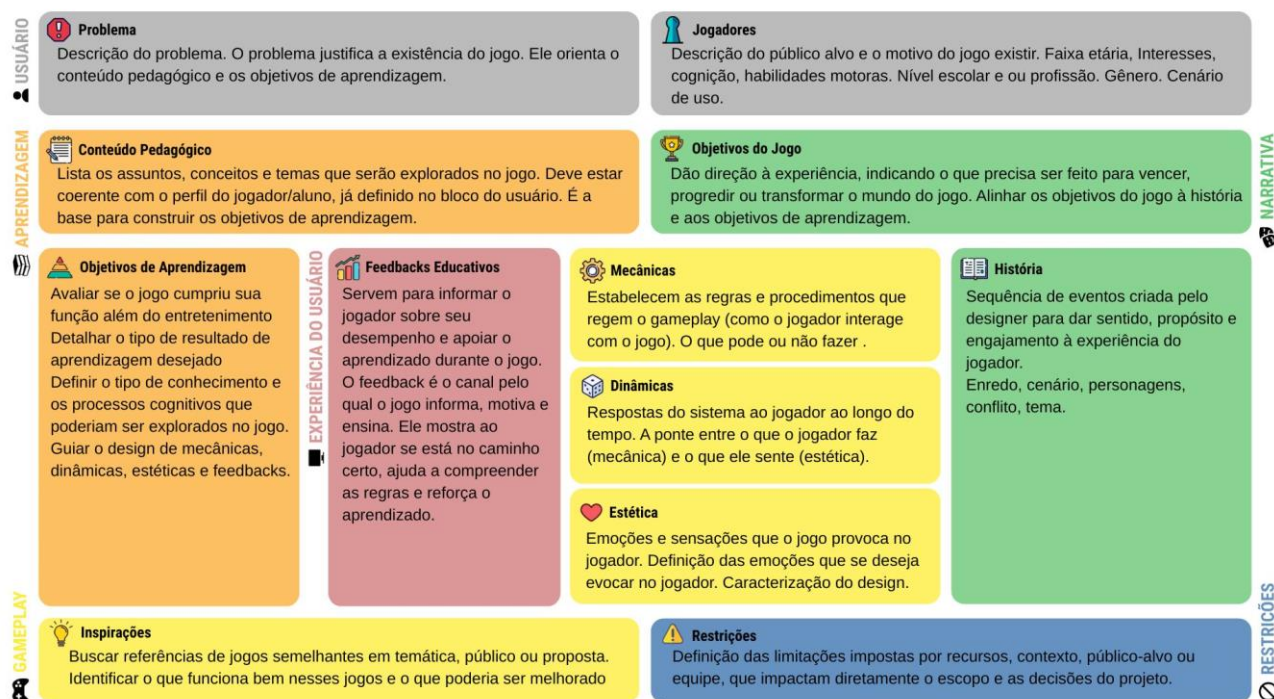
1. Introdução

Este relatório técnico apresenta as etapas de desenvolvimento do jogo educacional DEBUGGERS: A Batalha Contra os Bugs. Este jogo tem como objetivo ensinar de forma lúdica os fundamentos da lógica de programação e estruturação de código, e foi desenvolvido durante a realização da disciplina Jogos na Educação, do Programa de Pós-Graduação em Tecnologias Educacionais em Rede, no primeiro semestre de 2025.

A temática do jogo aborda o universo do desenvolvimento de software, onde jogadores colaboram como uma equipe de programadores para entregar um projeto funcional. A principal problematização que o jogo busca atacar é a natureza, muitas vezes abstrata e desmotivadora do aprendizado inicial de programação. Ao transformar conceitos iniciais de programação, como algoritmos, tipos de dados, variáveis e condicionais em componentes de um jogo cooperativo, busca-se tornar o aprendizado mais interativo, engajador e acessível, mesmo para aqueles que possuem pouco ou nenhum conhecimento introdutório sobre lógica de programação.

Para a sua construção, foi utilizado o ENDO-GDC, um Game Design Canvas para jogos educacionais endógenos, que busca melhorar a comunicação entre os diferentes membros de uma equipe interdisciplinar, facilitar a concepção do jogo e auxiliar no equilíbrio entre objetivos pedagógicos com a parte de entretenimento do game (TAUCEI, 2019). A Figura 1 apresenta o canvas, com a disposição visual de seus blocos e seções.

Figura 1 – Estrutura do Game Design Canvas ENDO-GDC



Fonte: adaptado de Taucei(2019)

A estrutura do ENDO-GDC é composta por seis blocos principais: **usuário**, que aborda o problema que justifica a criação do jogo e define o perfil do jogador/aluno; **aprendizagem**, que aborda o conteúdo pedagógico e os objetivos de aprendizagem; **experiência do usuário**, que inclui aspectos de feedback pedagógico; **narrativa**, que contempla a história e os objetivos do jogo; **gameplay**, que descreve as mecânicas, dinâmicas e estética, incluindo também uma seção de inspirações, que registra referências e influências criativas e, embora faça parte do gameplay, esta seção é preenchida antes da definição do Mecânicas, Dinâmicas e Estética (MDA); e, por fim, o bloco de **restrições**, que explicita limitações tecnológicas, contextuais ou relacionadas ao público-alvo que impactam o design.

2. Desenvolvimento do jogo Debuggers

Esta seção tem como objetivo apresentar o desenvolvimento do jogo DEBUGGERS: A Batalha Contra os Bugs, por meio dos blocos da abordagem ENDO-GDC.

2.1. Bloco do Usuário

O Bloco do Usuário define quem é o jogador ou aluno e qual é o problema que justifica a criação do jogo. Ele serve como base para todas as decisões pedagógicas e de design, orientando a escolha de conteúdos, mecânicas e estética.

2.1.1. Problema

O problema central é a dificuldade de aprendizagem e a baixa motivação frequentemente associadas aos conteúdos introdutórios de lógica de programação. Muitos estudantes consideram o tema abstrato e desinteressante, o que leva à evasão e à dificuldade de aplicar os conceitos na prática. O jogo visa transformar esse cenário, oferecendo uma abordagem prática, interativa e colaborativa.

2.1.2. Jogador/Aluno

O público-alvo principal são:

- Alunos dos primeiros semestres de cursos de graduação da área de Computação;
- Alunos de cursos técnicos em Informática que estão tendo o primeiro contato com programação.

O jogo foi projetado para não exigir conhecimento prévio de programação e foi idealizado para ser jogado por 4 participantes, podendo ser adaptado para 2 a 5 jogadores. O cenário de uso pode ser tanto uma sala de aula ou laboratório, bem como em ambientes informais de estudo.

2.2. Bloco de Aprendizagem

Este bloco estrutura os aspectos pedagógicos centrais do jogo, garantindo que o conteúdo de aprendizagem esteja integrado de forma endógena ao gameplay.

2.2.1. Conteúdo Pedagógico

Os principais conceitos e temas trabalhados são:

- **Pensamento algorítmico e sequência lógica:** é a base para a resolução dos desafios, instigando os alunos a desenvolverem esse tipo de pensamento.
- **Componentes básicos da programação na linguagem python:** incluindo a declaração de variáveis (Tipos: *Int*, *Float*, *Bool*, *Str*), estruturas de controle (*if*, *elif*, *else*), estruturas de repetição (*while*), operações aritméticas e lógicas, e ações de entrada e saída (*input* e *print*).
- **Boas práticas de programação:** mesmo não sendo obrigatório na linguagem python, é incentivado o uso de *Type Hint* para introduzir a importância de declarar tipos de dados de forma didática.

2.2.2. Objetivos de Aprendizagem

Com base na Taxonomia Revisada de Bloom, espera-se que ao final do jogo, o aluno seja capaz de:

- **Lembrar** os diferentes tipos de cartas e as funções que elas representam dentro da programação (tipos de dados, variáveis, condicionais, controle e ações).
- **Compreender** a função de um laço de repetição ou de uma estrutura condicional no contexto de um algoritmo.
- **Aplicar** (construir) uma sequência lógica de cartas, analogamente chamado de código, para resolver um problema proposto em uma carta de desafio.
- **Analisar**, por meio da decomposição de um problema, analogamente chamado de desafio, em etapas menores para serem construídas com as cartas de código.

2.3. Bloco de Experiência do Usuário

Este bloco trata dos mecanismos de feedback pedagógico que serão oferecidos ao jogador. Feedbacks são mecânicas essenciais em jogos educativos, atuando para informar o jogador sobre seu desempenho e apoiar o aprendizado durante o jogo.

2.3.1. Feedback Pedagógico

Com o intuito de motivar e ensinar o conteúdo nos cursos de programação, o jogo Debuggers propõe os seguintes feedbacks pedagógicos:

- **Compilação:** a ação de “Compilar” oferece o feedback mais direto. Usando um aplicativo web, os jogadores recebem uma resposta imediata e binária: sucesso ou falha. Uma falha não apenas informa que o algoritmo está incorreto, mas gera uma consequência tangível, o surgimento de um bug crítico e de bugs simples, de acordo com a dificuldade do desafio. O aplicativo consiste em uma aplicação web que valida a sequência de cartas informada pelos jogadores, sequência esta, obtida a partir dos códigos presentes nas cartas do jogo.

Figura 2 – Tela do aplicativo web



Fonte: dos autores

- **Dicas:** a ação “Revelar Dica” fornece um feedback de apoio. Os jogadores podem gastar uma ação para obter ajuda, mas isso tem um custo chamado Bug Simples, ensinando sobre o conceito de *trade-offs*, ou seja, escolher entre duas ou mais opções, sabendo que cada uma tem suas vantagens e desvantagens. A Figura 2 apresenta alguns exemplos de dicas.

Figura 3 – Exemplo de dicas



Fonte: dos autores

- **Estabilidade do sistema:** o marcador de “Estabilidade” é um feedback constante sobre o desempenho geral da equipe. Se ele estiver caindo, serve como alerta para que a equipe mude sua estratégia e foque na ação de “Debugar”.

2.4. Bloco de Narrativa

Este bloco dá sentido e engajamento ao jogo, articulando os objetivos pedagógicos a uma história que deve ser envolvente e que busca motivar o jogador a participar e persistir nos desafios.

2.4.1. História

- **Enredo:** os jogadores formam uma equipe de programadores contratada para desenvolver um sistema. Eles precisam construir diferentes módulos (os desafios), mas são constantemente *confrontados* por bugs que surgem no projeto. Ao acumular bugs, a equipe corre o risco de entrar em modo de crise, fazendo com que perca o jogo.
- **Cenário:** uma equipe de desenvolvimento de software trabalhando em conjunto para manter a estabilidade do projeto.
- **Personagens:** os jogadores são os “programadores”.
- **Antagonistas:** os “Bugs” (simples e críticos) são os inimigos que os jogadores precisam combater. A cada ciclo de desenvolvimento, novos bugs podem surgir.
- **Conflito:** a luta para construir um código funcional e estável antes que os bugs acumulados levem a equipe ao modo de crise causando assim o cancelamento do projeto, ou seja, a derrota completa.

2.4.2. Objetivos do Jogo

O objetivo central dos jogadores é atingir a condição de vitória ao trabalhar em equipe para construir e validar com sucesso os algoritmos de todos os módulos (desafios) propostos antes que os *bugs* corrompam o sistema. Este objetivo é mantido claro e visível durante toda a partida, pois as cartas de desafio a serem desenvolvidas permanecem na área de jogo, dando direção e propósito à experiência. O alinhamento com a aprendizagem é totalmente endógeno, uma vez que a ação de vencer, ou compilar o código corretamente, está diretamente ligado aos conceitos de programação. Por outro lado, a condição de derrota é o *crash* do sistema, que ocorre se a equipe não conseguir reverter o modo de crise, um estado crítico causado pelo acúmulo de bugs, o que adiciona tensão e reforça a importância da gestão de problemas. A Figura 3 apresenta dois exemplos de desafios e sua carta correspondente de solução.

Figura 4 – Exemplos de Desafios e Soluções



Fonte: dos autores

2.5. Bloco de Gameplay

Este bloco articula os aspectos relacionados à experiência de jogo, desde as inspirações que fundamentam as escolhas até a definição formal do modelo Mecânicas, Dinâmicas e Estética (MDA).

2.5.1. Inspirações

Para a concepção do jogo Debuggers, duas influências principais se destacam pela sua contribuição à mecânica e à estrutura do jogo, ajudando a construir um gameplay coerente e alinhado aos objetivos educacionais: Scratch e o jogo de tabuleiro cooperativo Eldritch Horror. Para a estética, serviram como inspiração para o desenho das cartas e para o tabuleiro, placas de circuito e um dashboard de um sistema para o tabuleiro.

O Scratch é uma linguagem de programação visual baseada em blocos, desenvolvida pelo MIT Media Lab, que permite a iniciantes, especialmente crianças e jovens, aprender os fundamentos da programação de forma intuitiva, sem a necessidade de se preocupar com

a sintaxe complexa do código. Os usuários criam programas arrastando e encaixando blocos coloridos que representam comandos, variáveis e estruturas de controle.

Eldritch Horror, por sua vez, é um jogo de tabuleiro cooperativo de aventura e terror, onde os jogadores assumem o papel de investigadores que viajam pelo mundo para resolver mistérios e impedir que uma entidade cósmica ancestral destrua o mundo.

2.5.2. Mecânicas

As mecânicas definem as regras e os procedimentos que regem o jogo, estabelecendo o que os jogadores podem ou não fazer. As mecânicas foram desenhadas para estarem diretamente relacionadas aos objetivos de aprendizagem.

- **Ações do jogador:** cada jogador pode realizar duas das seguintes ações por rodada:
 - **Codar:** jogar as cartas de código, obedecendo a ordem lógica do algoritmo;
 - **Pesquisar:** descartar cartas da mão para comprar novas;
 - **Revelar Dica:** revelar cartas de dicas sobre um desafio específico;
 - **Refatorar:** reorganizar cartas já jogadas;
 - **Compilar:** inserir a sequência das cartas no aplicativo web e validar o algoritmo;
 - **Debugar:** descartar cartas da mão para remover bugs simples da mesa.
- **Construção de algoritmos:** a mecânica principal é a construção de algoritmos, ou seja, posicionar as cartas de código em sequência lógica. As cartas de código são identificadas por cores conforme sua função.
- **Gerenciamento de mão:** comprar e descartar cartas de código para otimizar as jogadas e resolver o desafio, evitando o aparecimento de bugs.
- **Ciclo de desenvolvimento:** ao fim de cada rodada, o marcador de etapa avança, reduzindo a estabilidade e gerando bugs simples, criando um ambiente de ritmo e pressão constantes.
- **Sistema de bugs:** são os antagonistas do jogo, podendo ser de dois tipos: bugs simples e bugs críticos com efeitos distintos (flash e contínuo) que afetam o jogo de maneira significativa, a ponto de fazer com que a estabilidade caia e o projeto entre em modo crise, o qual não deve persistir por mais de duas rodadas, caso contrário o projeto é cancelado e o jogo é perdido.

2.5.3. Dinâmicas

São os comportamentos que emergem da interação entre as mecânicas e os jogadores, como cooperação, competição, exploração ou desafio. No caso do jogo Debuggers, destacam-se as seguintes dinâmicas:

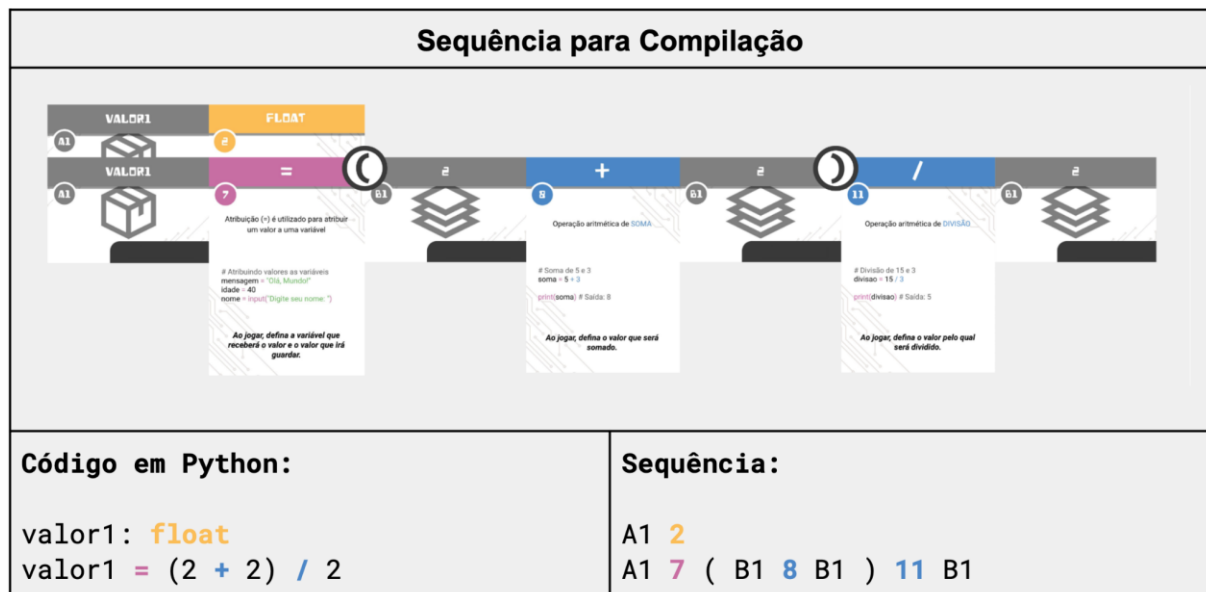
- **Cooperação, comunicação e negociação:** os jogadores devem decidir juntos quais desafios realizar, como utilizar as cartas de maneira otimizada e quando gastar ações para debugar ou refatorar.
- **Resolução de problemas em equipe:** a equipe precisa analisar as cartas disponíveis e o desafio para construir o algoritmo de forma colaborativa e evitar entrar no modo de crise, o que pode causar o crash do jogo.
- **Gerenciamento de risco e pressão:** a equipe precisa constantemente decidir entre avançar nos desafios ou controlar a proliferação de bugs, especialmente quando a estabilidade está baixa.

2.5.4. Estética

Refere-se às emoções e experiências sensoriais que o jogo busca provocar, como curiosidade, tensão, empatia ou diversão. Inclui também elementos visuais, sonoros e táteis.

- **Desafio:** satisfação de resolver um quebra-cabeça lógico ao construir um algoritmo corretamente. A Figura 4 apresenta um exemplo de resolução (modo de compilação) de desafio.

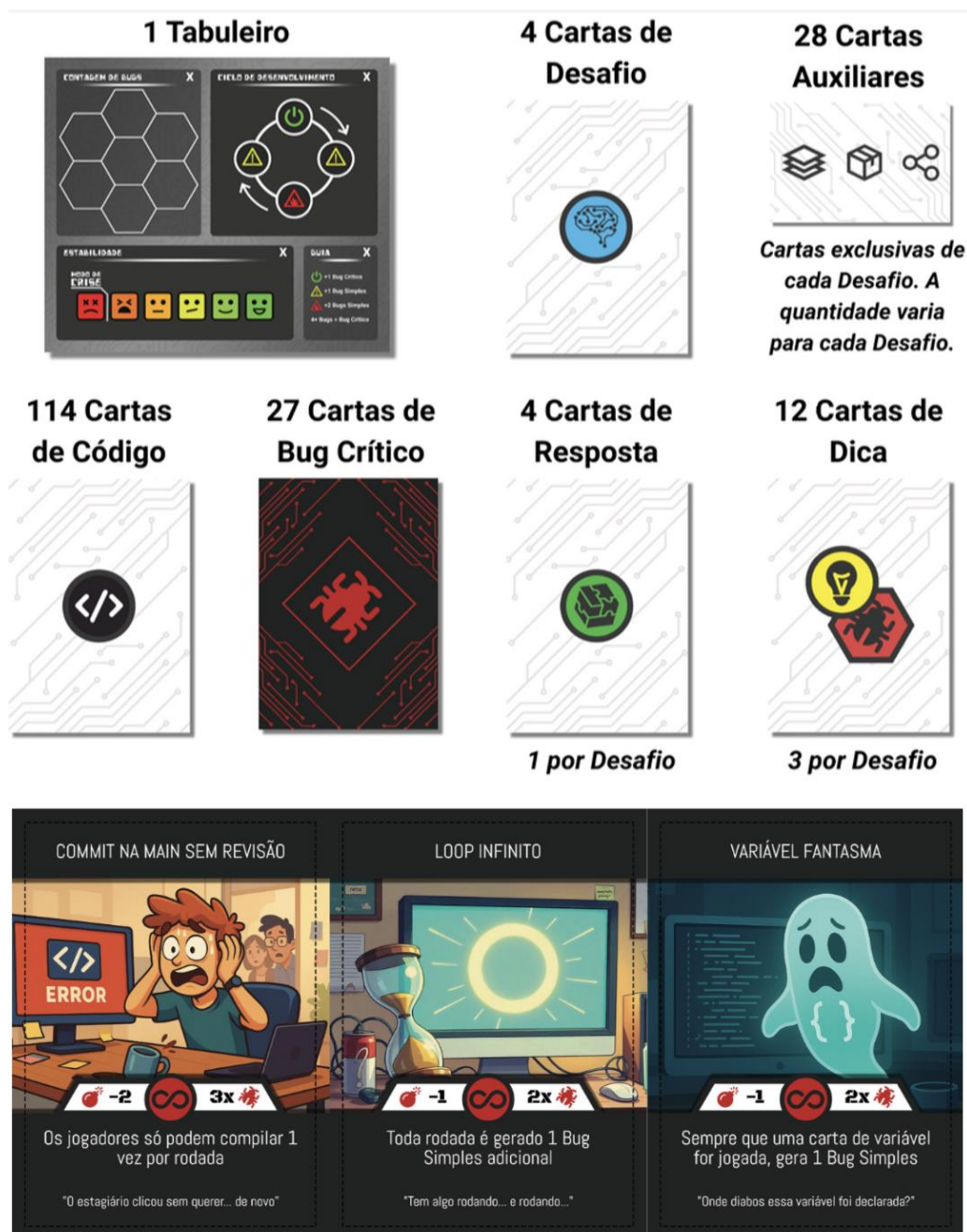
Figura 5 – Exemplo de resolução de desafio



Fonte: dos autores

- **Tensão e alívio:** ansiedade gerada pelo acúmulo de bugs e a queda da estabilidade, seguida pelo alívio ao conseguir debugar um bug crítico ou compilar um desafio com sucesso antes que o modo de crise seja ativado.
- **Camaradagem:** sensação de trabalho em equipe ao vencerem (ou perderem) juntos.
- **Fantasia:** imersão na temática de ser um programador lutando contra as falhas de um sistema.
- **Elementos visuais:** cartas coloridas de acordo com suas funções específicas e utilização de ícones que facilitam o entendimento sobre a ação de cada carta. Tabuleiro que simula uma placa circuito com intuito de estabelecer uma conexão entre os elementos do jogo. A Figura 5 apresenta exemplos de componentes e cartas do jogo.

Figura 6 – Exemplos de elementos do jogo



Fonte: dos autores

2.6. Bloco de Restrições

O bloco de Restrições tem como objetivo a definição das limitações impostas por recursos, contexto, público-alvo ou equipe, que impactam diretamente o escopo e as decisões do projeto. Os recursos são os instrumentos que serão utilizados para desenvolver o jogo. As restrições são as limitações impostas por esses instrumentos e pelas outras camadas ao jogo. Serão abordadas as seguintes restrições.

2.6.1. Tecnológicas

O jogo exige o uso de um dispositivo digital (computador, smartphone ou tablet) com o aplicativo web oficial para a ação de compilar ou validar. Isso impede que o jogo seja jogado em ambientes sem acesso a essa tecnologia.

2.6.2. De Contexto

O jogo é otimizado para 4 jogadores, e a experiência pode ser alterada em grupos maiores ou menores. O tempo de partida, embora não especificado, precisa ser considerado para aplicação em aulas com duração fixa (exemplo: 50 minutos).

2.6.3. Do Público-Alvo

A linguagem e os ícones precisam ser claros para jogadores sem experiência prévia. A complexidade dos desafios deve ser balanceada para ser acessível a alunos do ensino médio, mas ainda interessante para alunos do ensino técnico em informática ou universitários dos cursos de computação.

3. Aplicação do Jogo

Debuggers pode ser aplicado em laboratórios de informática ou salas de aula como uma atividade prática após a introdução teórica de um conceito de programação. O professor pode atuar como um facilitador, explicando as regras e mediando a discussão pós-jogo para reforçar a conexão entre as mecânicas (exemplo: usar uma carta FLOAT) e o conceito de programação relacionados aos tipos de dados.

4. Avaliação do Jogo

Para garantir uma avaliação robusta e sistemática do jogo Debuggers, sugere-se a utilização de um instrumento de avaliação de sua qualidade. No contexto deste projeto, indica-se o uso do modelo de avaliação de jogos educacionais MEEGA+. Esta avaliação é fundamental para validar a eficácia do jogo e teve como base os pressupostos de Petri; Von Wangenheim; Borgatto (2019).

O modelo MEEGA+ (*Model for the Evaluation of Educational GAMES*) oferece um instrumento de medição confiável e válido, permitindo a coleta de evidências sobre a qualidade do jogo e suas contribuições para a aprendizagem a partir de dois fatores principais:

- **Experiência do Jogador e Usabilidade.** O fator “Experiência do Jogador” é multifacetado, abrangendo dimensões que medem o engajamento e a motivação, como atenção focada, desafio, confiança e relevância, além de aspectos afetivos como a diversão e a interação social.
- **Aprendizagem percebida,** avaliando a percepção dos alunos sobre o quanto o jogo contribuiu para seu conhecimento.

A aplicação do questionário do modelo MEEGA+ utiliza uma escala Likert de 5 pontos, variando de “discordo totalmente” a “concordo totalmente” para mensurar a percepção dos jogadores em relação a 35 itens padronizados. Esses itens cobrem todas as dimensões dos fatores de experiência do jogador e usabilidade, garantindo uma cobertura completa da avaliação.

A etapa de análise dos resultados é focada em estatística descritiva para descrever e apresentar os dados coletados. A documentação do MEEGA+ oferece suporte para esta fase, disponibilizando uma planilha de análise de dados padronizada que automatiza parte do processo.

Em síntese, a aplicação do modelo MEEGA+ pode permitir uma avaliação sistemática, confiável e válida da qualidade do jogo Debuggers. Os resultados obtidos fornecerão evidências concretas sobre sua eficácia como ferramenta educacional, medindo aspectos cruciais como usabilidade, engajamento, motivação e, principalmente, a percepção de aprendizagem, fundamentando tanto futuras melhorias quanto sua adoção na prática pedagógica.

5. Considerações Finais

O desenvolvimento do jogo Debuggers, orientado pelo ENDO-GDC resultou em um artefato educacional onde o aprendizado da lógica de programação é a própria essência da jogabilidade. A ferramenta se mostrou eficaz para estruturar a concepção do jogo, garantindo que a narrativa, as mecânicas e os objetivos estivessem alinhados de forma coesa. O produto final é um jogo cooperativo com potencial para engajar estudantes e facilitar a compreensão de temas fundamentais da computação, servindo como uma valiosa ponte entre a teoria e a prática.

Todos os arquivos do jogo, incluindo manual de regras e aplicativo web, encontram-se disponíveis em <https://ziquinatti.github.io/Debuggers-Game/>, tendo sido organizados de forma a facilitar a impressão por parte de professores interessados, aspecto considerado essencial considerando a importância da publicização e a acessibilidade de recursos educacionais.

Referências

PETRI, G., von WANGENHEIM, C. G. & BORGATTO, A. F. (2019). **MEEGA+: A Model for the Evaluation of Educational Games for Computing Education (MEEGA+: Um Modelo para a Avaliação de Jogos Educacionais para o ensino de Computação)**. Brazilian Journal of Computers in Education (Revista Brasileira de Informática na Educação - RBIE), 27(3), 52-81. DOI: 10.5753/RBIE.2019.27.03.52

TAUCEI, Bernardo Blasquez. **Endo-GDC: desenvolvimento de um Game Design Canvas para concepção de jogos educativos endógenos**. Dissertação (Mestrado em Engenharia de Sistemas e Computação) – Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia (COPPE), Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2019.