

# Utilização de Jogos Interativos como Método de Ensino de Programação para Iniciantes

Igor G. A. Rocha<sup>1</sup>, Charles T. B. Garrocho<sup>1</sup>

<sup>1</sup> Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG)  
Campus Ouro Branco – Ouro Branco, MG – Brasil

igorguilhermedev@gmail.com

**Abstract.** *With technological advancements, learning to program has become essential but also challenging. Many students drop out due to initial difficulties. This study investigates the use of games as a didactic tool to facilitate this learning process. The game Script Hero was developed with a focus on programming logic and mechanics that stimulate student engagement. Its effectiveness was assessed through a questionnaire based on the MEEGA+ framework (Model-Driven Educational Game Development Approach). The results are expected to help shift educators' perspectives on the use of games as educational tools, thereby reducing dropout rates in related courses.*

**Resumo.** *Com o avanço tecnológico, aprender programação se tornou essencial, mas também desafiador. Muitos alunos desistem por dificuldades iniciais. Este trabalho investiga o uso de jogos como ferramenta didática para facilitar esse aprendizado. Foi desenvolvido o jogo Script Hero, um jogo com foco na lógica de programação e mecânicas que estimulam o engajamento. A eficácia do jogo foi avaliada por meio de um questionário feito com base no framework MEEGA+ (Model-Driven Educational Game Development Approach). Espera-se que os resultados ajudem a mudar a forma com que educadores enxergam o uso de jogos como ferramenta educacional e consequentemente reduzir a taxa de abandono nos cursos da área.*

## 1. Introdução

A programação é a prática de criar instruções detalhadas e precisas que permitem que um computador execute tarefas específicas, basicamente é o ato de transformar uma linguagem compreendida pelos humanos em uma linguagem que pode ser entendida por um computador [McCracken 1957 apud Blackwell 2002]. Essa habilidade é fundamental atualmente para o desenvolvimento de aplicativos, jogos, sites, sistemas embarcados e muitas outras aplicações tecnológicas que fazem parte do cotidiano de todas as pessoas. A programação requer o conhecimento de conceitos lógicos, estruturas de dados e algoritmos, além da capacidade de transformar ideias abstratas em algo funcional.

No entanto, um aspecto que está despertando a atenção dos pesquisadores é a elevada taxa de abandono dos alunos nos cursos de tecnologia e disciplinas relacionadas à programação. Segundo Hoed (2016), as áreas relacionadas à matemática, física e computação possuem uma taxa de evasão maior do que a taxa média dos cursos superiores no geral, com a computação possuindo uma porcentagem superior em cinco anos consecutivos em relação à média nacional. Silva et al. (2024), ao analisar a taxa de evasão

nos cursos de Computação da Universidade Federal do Amapá, os autores identificaram que disciplinas como Programação I e II estão entre aquelas com maior taxa de insucesso, contribuindo significativamente para a evasão dos estudantes. Um questionário de natureza quantitativa aplicado por Fukao et al. (2023), revelou uma variedade de fatores que contribuem para as dificuldades enfrentadas pelos estudantes. Entre os principais motivos apontados destacam-se: a didática adotada pelos professores, a falta de motivação pessoal dos alunos e a elevada exigência e complexidade das disciplinas do curso. Esses resultados também vão de encontro com a pesquisa feita por Silva et al. (2024), que lista os principais motivos de evasão ordenados por relevância, sendo o 4º motivo mais relevante a deficiência didática dos professores.

Diante dessa situação, é crucial considerar soluções que possam aprimorar o processo de aprendizagem em programação, como a adoção de abordagens inovadoras de ensino. Existem hoje, várias técnicas educacionais que estão sendo aplicadas em escolas e universidades, incluindo a gamificação e o aprendizado baseado em jogos [Pho and Dinscore 2015].

A abordagem da gamificação busca utilizar elementos inspirados em jogos, como recompensas, desafios e competição, como forma de motivar e engajar os alunos durante o processo de aprendizagem. Através da implementação de sistemas de pontuação, níveis e conquistas, os estudantes são estimulados a progredir, superar obstáculos e alcançar metas estabelecidas, tornando o aprendizado mais cativante e empolgante. Essa abordagem proporciona um ambiente dinâmico e interativo, onde os alunos podem experimentar um senso de conquista e satisfação ao atingirem marcos significativos, tornando o processo de aprendizagem uma experiência envolvente e recompensadora [Busarello 2016].

O aprendizado baseado em jogos difere da gamificação, pois em vez de utilizar apenas elementos de jogos no processo de ensino, ele incorpora um jogo real como parte integrante da experiência de aprendizagem. Os autores de [Netto et al. 2017] criaram um jogo chamado *Game Logic*, um jogo focado no auxílio da aprendizagem de lógica de programação, um questionário foi aplicado em 10 alunos que jogaram o jogo, 70% dos participantes deram uma nota 8 de 10 para o jogo, os outros 30% deram uma nota de 4 a 7 no geral tivemos resultados positivos em relação a aplicabilidade de um jogo sério nesse contexto.

Contudo, o objetivo deste artigo é analisar a eficácia de jogos no auxílio à aprendizagem de programação. Para isso, foi desenvolvido o jogo *Script Hero*, construído com base em princípios do *framework Octalysis*. O jogo dá ênfase ao aprendizado de conceitos básicos de programação, como variáveis, estruturas condicionais, estruturas de repetição e funções. Atualmente, o jogo está disponível para computadores e foi avaliado com base no questionário *MEEGA+*.

Este artigo está dividido em quatro seções. A seção 2 apresenta a fundamentação teórica do trabalho e está subdividida em duas partes: base conceitual e trabalhos relacionados. A seção 3 trata da proposta do trabalho e da metodologia empregada no desenvolvimento do jogo. A seção 4 corresponde à análise da eficácia, abordando a coleta e a análise dos resultados obtidos com a aplicação do jogo. Por fim, a seção 5 apresenta as conclusões do estudo.

## 2. Fundamentação Teórica

Nesta seção, o objetivo primordial é construir uma base sólida de conhecimento sobre o tema central desta pesquisa. Com isso em mente, busca-se fornecer informações detalhadas que sirvam de *background* e fundamentem as seções subsequentes do trabalho.

### 2.1. Base Conceitual

Nesta seção, são apresentados os principais conceitos utilizados ao longo deste artigo, explicando seus significados e destacando como são aplicados tanto em contextos acadêmicos quanto profissionais em alguns casos. O objetivo é oferecer uma base teórica clara para a compreensão do trabalho, facilitando a conexão entre a fundamentação e a proposta desenvolvida.

#### 2.1.1. Gamificação e Jogo Sério

A gamificação é uma técnica que está nos holofotes de um tempo para cá. Segundo KAPP (2012), a gamificação é o ato de colocar elementos característicos de jogos, como recompensas, níveis e medalhas, em contextos diferentes de jogos. Esses elementos ajudam no engajamento e satisfação do usuário, muitas vezes fazendo com que o mesmo se sinta mais disposto a continuar utilizando a ferramenta. Atualmente existem diversas ferramentas que utilizam gamificação em diversas áreas, como saúde, educação e esportes. Alguns exemplos são *Duolingo*, *Waze* e *Nike Run Club*.

Jogo sério é um termo utilizado para jogos tipicamente *no entertainment*, ou seja, são jogos onde o foco principal não está no entretenimento [Silva et al. 2012]. No contexto educacional esses jogos tem como objetivo o aprendizado acima do entretenimento. Através dos jogos sérios, os educadores têm uma poderosa ferramenta capaz de engajar alunos de maneira efetiva, aumentando seu foco e consequentemente a retenção de conhecimento.

*CodeCombat* e *Human Resource Machine* são alguns exemplos de jogos sérios na área da programação.

#### 2.1.2. Octalysis

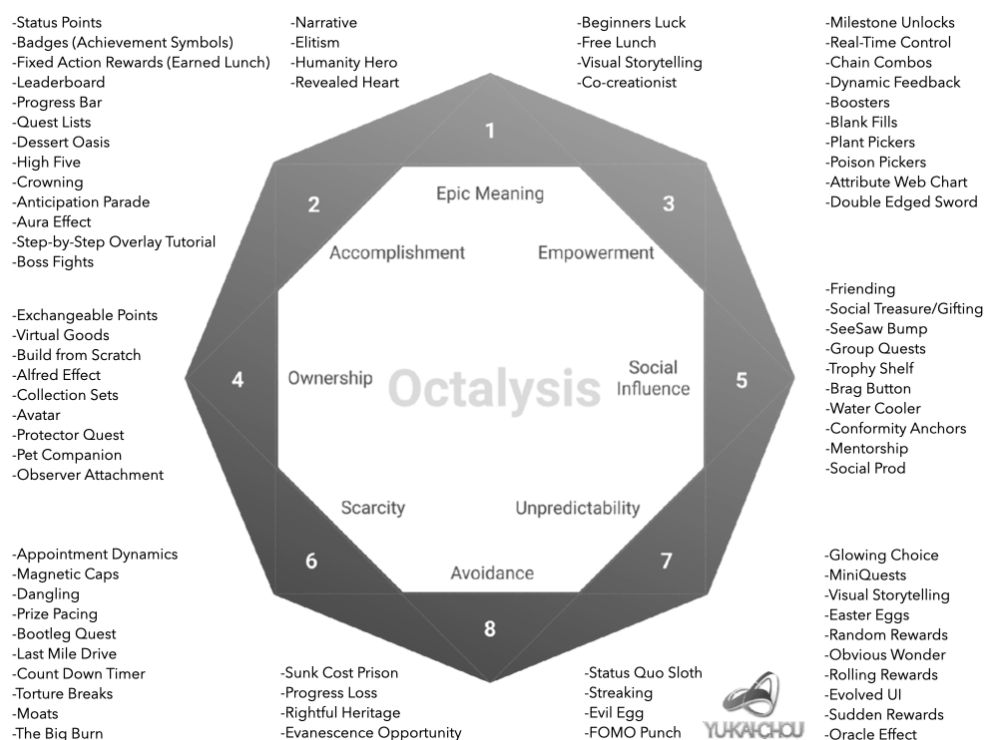
O *Octalysis* é um modelo criado por Yu-kai Chou para entender o que motiva as pessoas em experiências gamificadas. Ele se baseia em oito motivações principais:

1. Significado Épico e Chamado;
2. Desenvolvimento e Realização;
3. Empoderamento da Criatividade e Feedback;
4. Propriedade e Posse;
5. Influência Social e Relacionamento;
6. Escassez e Impaciência;
7. Imprevisibilidade e Curiosidade;
8. Evitação de Perda.

A ideia do framework é mostrar que, para engajar de verdade, um sistema não deve focar só em pontos e prêmios, mas sim em motivações humanas mais profundas [kai Chou 2014]. A figura 1 mostra mais a fundo os principais tópicos desse modelo.

O Octalysis já está sendo utilizado em diversos projetos, inclusive em projetos com cunho educacional, alguns exemplos são:

1. Aplicativos para ensino de programação, como o desenvolvido por Christopher e Waworuntu (2021), que utilizou o Octalysis para criar um aplicativo de aprendizado da linguagem Java, 74,27% dos alunos que avaliaram o aplicativo, concordaram que o aplicativo foi útil no processo de aprendizagem;
2. No ensino de línguas, Sulispera e Recard (2020), aplicaram o framework Octalysis em aulas de inglês com alunos do ensino fundamental. Usando elementos como desafios, narrativa e colaboração em grupo, os autores observaram um aumento no engajamento e na participação dos estudantes, mostrando que a gamificação pode tornar o aprendizado mais envolvente e eficaz.



**Figura 1. Os 8 Fatores-Motivadores da Gamificação.**  
Disponível em: [kai Chou 2014]

### 2.1.3. MEEGA+

O MEEGA+ (*Model for the Evaluation of Educational Games*) é um modelo de avaliação criado especificamente para jogos educacionais voltados ao ensino de computação. Ele surgiu como uma evolução do modelo MEEGA original, com o objetivo de oferecer uma forma mais precisa e confiável de medir a qualidade desses jogos. O modelo avalia fatores como a experiência do jogador, usabilidade e percepção de aprendizagem, usando um

questionário estruturado com base em validações estatísticas. Isso permite que desenvolvedores, pesquisadores e professores tenham um instrumento sólido para entender como os jogos impactam o aprendizado e o engajamento dos estudantes, orientando melhorias e decisões sobre sua adoção em sala de aula [Petri et al. 2019].

## 2.2. Trabalhos Relacionados

Para identificar os trabalhos relacionados ao presente estudo, foi realizada uma busca exploratória com base em artigos acadêmicos publicados nos últimos anos, com foco em jogos sérios e ambientes gamificados aplicados ao ensino de programação. A pesquisa foi conduzida manualmente por meio da leitura de títulos, resumos e palavras-chave, priorizando artigos que apresentassem ferramentas, aplicativos ou jogos voltados ao ensino de lógica de programação.

O projeto *Game Logic* apresenta um jogo educacional para dispositivos Android, com foco no ensino de lógica de programação por meio de blocos e fluxogramas. A proposta visa facilitar a compreensão de algoritmos de forma lúdica e acessível. Os resultados indicaram boa aceitação pelos estudantes, 70% deles deram uma nota geral entre 8 e 10 e 30% uma nota entre 4 e 7, com destaque para o potencial do jogo como ferramenta auxiliar no processo de aprendizagem [Netto et al. 2017].

Christopher e Waworuntu (2021), desenvolveram um aplicativo gamificado, baseado no framework Octalysis, para o ensino da linguagem Java. A aplicação incorpora mecânicas como pontos, rankings e missões, visando promover engajamento e motivação. Avaliado com base no modelo *UTAUT*, o aplicativo demonstrou bons resultados, obtendo 74,27% de aprovação entre os alunos, sendo considerado eficaz para apoiar o ensino de programação.

*Code Dungeon* é um jogo sério voltado ao ensino de programação introdutória e pensamento computacional. Com foco em estruturas condicionais, o jogo foi desenvolvido com base no framework *SGDA* e avaliado pelo modelo *MEEGA+*. Os resultados evidenciaram contribuições positivas para o processo de ensino-aprendizagem, especialmente quanto ao engajamento e à assimilação dos conteúdos.

Conforme Oliveira e Boff (2023), “O jogo demonstrou-se viável como ferramenta de ensino. Na aplicação dos questionários, com duas turmas de Programação de Computadores I, 28 alunos responderam, em grande parte, com um feedback bastante positivo sobre o jogo desenvolvido”.

*ProgramSE* é um jogo educacional voltado ao ensino de lógica de programação para iniciantes no ensino superior. A proposta utiliza programação em blocos e desafios baseados em ações do cotidiano, com foco na fixação de conceitos básicos. O desenvolvimento foi guiado por princípios do Construtivismo e da Taxonomia de Bloom. A avaliação foi feita com 32 estudantes, utilizando o modelo *MEEGA+*, indicou boa aceitação, sugerindo que o jogo pode ser uma ferramenta útil para apoiar o aprendizado inicial em programação. Segundo as palavras dos autores Silva et al. (2021), “A percepção dos alunos em relação à forma com que os conceitos de programação foram ensinados foi positiva.”

*Variant Scape* é um jogo digital educacional que se encontra em fase de prototipagem, desenvolvido pelos autores de Araújo et al. (2023), por meio da ferramenta

*Figma.* O jogo tem como principal objetivo auxiliar o ensino de conceitos introdutórios de lógica de programação e pensamento computacional. Para avaliar sua aceitação e potencial pedagógico, foi realizado um estudo exploratório com cinco discentes do curso de Ciência da Computação da Universidade do Estado do Amazonas (UEA), sendo um ingressante, um concluinte e três estudantes entre o 5º e o 7º período. A análise dos dados foi conduzida por meio dos instrumentos Emoti-SAM e MEEGA+, que avaliaram aspectos emocionais, de usabilidade e de experiência do jogador. Os resultados indicaram que quatro dos cinco participantes demonstraram reações positivas durante a interação com o jogo, resultando em um índice de aproveitamento de 80%.

A Tabela 1 apresenta um conjunto de informações relevantes para a realização de um comparativo entre o projeto proposto neste artigo e os trabalhos relacionados. No geral, observou-se que apenas um dos trabalhos analisados disponibilizou um link de acesso direto ao artefato descrito, permitindo sua execução e validação prática. Nos demais casos, não foi possível localizar ou acessar o jogo/aplicativo mencionado, o que limitou a verificação de alguns aspectos.

Dessa forma, os dados apresentados na tabela foram preenchidos com base nas descrições fornecidas pelos próprios artigos, bem como nas imagens, gráficos e demais evidências documentais disponíveis.

**Tabela 1. Comparação entre os trabalhos relacionados e o jogo proposto**

<b>Critério</b>	<b>Java App</b>	<b>Game Logic</b>	<b>Code Dungeon</b>	<b>ProgramSE</b>	<b>Variant Scape</b>	<b>Script Hero</b>
<b>Principais tecnologias utilizadas no desenvolvimento</b>	React Native com Firebase	Não especificado	Unity3D	Construct 3	Figma	A definir
<b>Framework de gamificação ou guia de desenvolvimento</b>	Octalysis	Não especificado	SGDA (Serious Game Design Assessment)	Construtivismo + Taxonomia de Bloom	Não especificado	A definir
<b>Framework de avaliação</b>	UTAUT	Questionário informal	MEEGA+	MEEGA+	MEEGA+ e Emoti-SAM	MEEGA+
<b>Número de avaliadores</b>	40 alunos	10 alunos	28 alunos	32 alunos	5 alunos	A definir
<b>Tópicos abordados</b>	Java em geral	Lógica de programação	Estruturas condicionais e pensamento computacional	Sequência, Seleção, Repetição, Estruturas Condicionais	Repetição, Estruturas Condicionais, Variáveis, Operações Matemáticas	A definir
<b>Integração com IA</b>	Não mencionada	Não mencionada	Não mencionada	Não mencionada	Não mencionada	A definir
<b>Acesso ao aplicativo</b>	Indisponível	Indisponível	Indisponível	Disponível	Indisponível	A definir
<b>Feedback Geral</b>	Positivo	Positivo	Positivo	Positivo	Positivo	A definir

A seguir, são descritos os critérios utilizados na Tabela 1, acompanhados de sua respectiva relevância para a análise comparativa:

1. **Principais tecnologias utilizadas no desenvolvimento:** Identifica as ferramentas, linguagens e plataformas empregadas na construção do jogo. Esse critério é importante para avaliar a viabilidade técnica, a compatibilidade com diferentes dispositivos e a complexidade do desenvolvimento;

2. **Framework de gamificação ou guia de desenvolvimento:** Indica se o projeto seguiu alguma abordagem metodológica para promover engajamento, motivação e progressão do usuário. A presença de frameworks bem definidos sugere maior atenção à experiência do jogador e ao potencial pedagógico do jogo;
3. **Framework de avaliação:** Refere-se ao modelo utilizado para validar a eficácia do jogo como ferramenta de ensino. Adoção de métodos como MEEGA+ ou UTAUT demonstra preocupação com uma avaliação sistemática e estruturada;
4. **Número de avaliadores:** Aponta o tamanho da amostra envolvida na etapa de validação. Um número maior de avaliadores tende a fornecer resultados mais confiáveis e representativos sobre a aceitação e usabilidade do jogo;
5. **Tópicos abordados:** Descreve os conteúdos pedagógicos contemplados, como estruturas de controle, repetição, funções ou lógica de programação. Esse critério permite avaliar o alinhamento do jogo com os objetivos de aprendizagem;
6. **Integração com IA:** Verifica se o jogo incorpora inteligência artificial para personalização, adaptação de conteúdo ou feedback dinâmico. Embora não essencial, essa integração representa um avanço tecnológico relevante;
7. **Acesso ao aplicativo:** Indica se o artefato está disponível publicamente para download ou uso. A acessibilidade é fundamental para reprodutibilidade, disseminação do trabalho e impacto educacional;
8. **Feedback Geral:** Resume a percepção dos usuários ou participantes sobre o jogo, com base nas avaliações reportadas. Esse critério serve como indicativo da aceitação e da efetividade da ferramenta no contexto educacional.

### 3. Proposta

O jogo desenvolvido, intitulado *Script Hero*, é um jogo educacional projetado para auxiliar no ensino de lógica de programação por meio da resolução de desafios baseados em movimentação de um personagem em um tabuleiro. Ele foi inteiramente implementado na linguagem Java, utilizando o framework JavaFX, com o objetivo de garantir compatibilidade com diferentes sistemas operacionais e facilitar sua execução em uma ampla variedade de dispositivos. O Script Hero está disponível para download, acessível em: <https://site-tcc-mauve.vercel.app>. Caso o site esteja indisponível, é possível acessar as *releases* do projeto pelo *GitHub*: <https://github.com/IgorGuilhermeDev/ReleasesScriptHero>.

#### 3.1. Metodologia

Esta seção apresenta o passo a passo seguido no desenvolvimento do Script Hero, desde a fase inicial de prototipagem até a definição da versão final do sistema.

##### 3.1.1. GDD

A primeira etapa do projeto consistiu na elaboração do *Game Design Document (GDD)*, documento fundamental para guiar a concepção e o escopo inicial do jogo. Nele, foram esboçadas as tecnologias previstas para o desenvolvimento, sendo escolhidas a linguagem Java e a biblioteca gráfica JavaFX.

O GDD definiu uma estrutura composta por três capítulos, cada um contendo três fases, organizadas de forma a abordar gradativamente conteúdos introdutórios de programação:

- 1º Capítulo: Métodos e variáveis;
- 2º Capítulo: Estruturas condicionais;
- 3º Capítulo: Estruturas de repetição.

Essa estrutura pedagógica foi pensada para promover o aprendizado progressivo, alinhado ao nível de dificuldade de cada etapa.

### 3.1.2. Arquitetura

Com base nas definições estabelecidas no GDD, foi elaborado um diagrama de componentes, com o objetivo de abstrair e organizar a arquitetura proposta para o jogo.

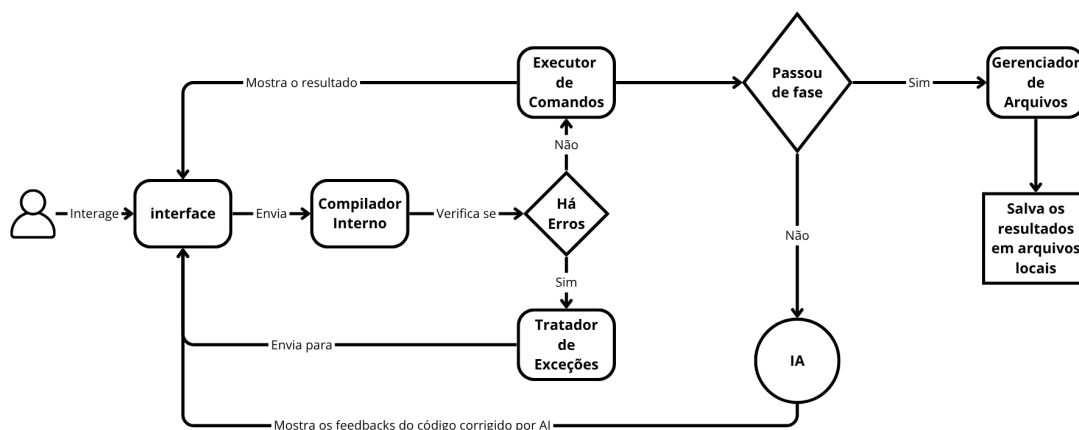


Figura 2. Arquitetura proposta para o sistema. Fonte: elaborado pelos autores.

### 3.1.3. Aplicação do Framework Octalysis

Com a arquitetura e o escopo inicial definidos no GDD, o planejamento do jogo já se encontrava praticamente concluído. No entanto, ainda faltavam elementos de gamificação que incentivassem o jogador a continuar jogando, promovendo maior envolvimento e uma experiência mais significativa. Para suprir essa lacuna, adotou-se o framework Octalysis, conforme apresentado anteriormente na Figura 1, a qual ilustra os oito *Core Drives* centrais.

Dentre esses oito princípios, foram selecionados três Core Drives para serem implementados no jogo

- Core Drive 1 – Significado Épico e Vocação: Implementado por meio de visual *storytelling* com *assets* temáticos relacionados ao universo da tecnologia, como computadores, pendrives e inimigos representados por vírus, reforçando a imersão e o senso de propósito;



- Core Drive 2 – Desenvolvimento e Realização: Representado por barras de progresso, lista sequencial de fases e um sistema de estrelas obtidas de acordo com o desempenho do jogador em cada etapa, visando promover sensação de progresso e conquista;
- Core Drive 3 – Empoderamento da Criatividade e Feedback: Aplicado por meio de feedbacks dinâmicos ao longo das fases, permitindo ao jogador experimentar soluções e receber retornos imediatos, promovendo aprendizado ativo.



Figura 3. Alguns dos assets do Script Hero. Fonte: feito pelos autores.

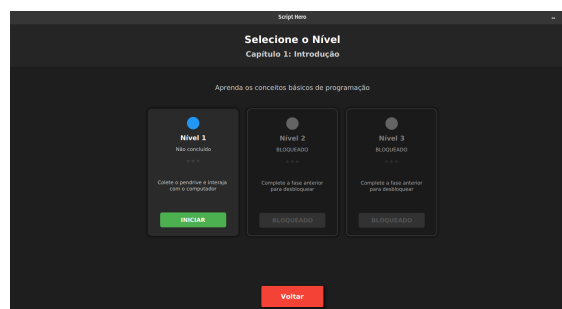


Figura 4. Exemplificação da aplicação do core 2 Fonte: elaborado pelos autores.



Figura 5. Exemplificação da aplicação do core 3 . Fonte: elaborado pelos autores.

#### 4. Coleta e Análise dos Resultados

#### 5. Conclusão

## Referências

- Araújo, L., Macena, J., Melo, R., Pessoa, M., and Pires, F. (2023). Variant scape: um jogo para exercitar conceitos de introdução à lógica de programação. In *Anais do XXXIV Simpósio Brasileiro de Informática na Educação*, pages 1789–1800, Porto Alegre, RS, Brasil. SBC.
- Blackwell, A. F. (2002). What is programming? In *PPIG*, page 20. Citeseer.
- Busarello, R. I. (2016). *Gamification: princípios e estratégias*. Pimenta Cultural.
- Christopher, L. and Waworuntu, A. (2021). Java programming language learning application based on octalysis gamification framework. *IJNMT (International Journal of New Media Technology)*, 8(1):65–69.
- Fukao, A., Colanzi, T., Martimiano, L., and Feltrim, V. (2023). Estudo sobre evasão nos cursos de computação da universidade estadual de maringá. In *Anais do III Simpósio Brasileiro de Educação em Computação*, pages 86–96, Porto Alegre, RS, Brasil. SBC.
- Hoed, R. M. (2016). Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de computação. Dissertação (mestrado profissional em computação aplicada), Universidade de Brasília, Brasília. xvi, 164, [8] f., il.
- kai Chou, Y. (2014). Octalysis: Complete gamification framework. Acesso em: 01 jul. 2025.
- Netto, D., Medeiros, L. M., de Pontes, D., and de Moraes, E. (2017). Game logic: Um jogo para auxiliar na aprendizagem de lógica de programação. In *Anais do XXV Workshop sobre Educação em Computação*, pages 2297–2306, Porto Alegre, RS, Brasil. SBC.
- Oliveira, G. C. and Boff, E. (2023). Code\_dungeon: um serious game para auxiliar no aprendizado de programação. In *Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, pages 581–590. SBC.
- Petri, G., Von Wangenheim, C. G., and Borgatto, A. F. (2019). Meega+: Um modelo para a avaliação de jogos educacionais para o ensino de computação. *Revista Brasileira de Informática na Educação*, 27(03):52–81.
- Pho, A. and Dinscore, A. (2015). Game-based learning. *Tips and trends*, pages 1–5.
- Silva, M. B. d., Furtado, S. D. d. F., and Furtado, J. C. C. (2024). A evasão dos estudantes em cursos de computação: um estudo sobre a universidade federal do amapá. *Caderno Pedagógico*, 21(9):e7820.
- Silva, R. R., Rivero, L., and Dos Santos, R. P. (2021). Programse: Um jogo para aprendizagem de conceitos de lógica de programação. *Revista Brasileira de Informática na Educação*, 29:301–330.
- Silva, T. G. d. et al. (2012). Jogos sérios em mundos virtuais: uma abordagem para o ensino-aprendizagem de teste de software.