

Coding Fight: Jogo para auxiliar o aprendizado dos fundamentos da programação na disciplina de algoritmos

**Daniel Muller Rezende¹, Matheus dos Reis Casarim¹,
Mathews Edwirds Gomes Almeida¹, Pedro Luiz Bonorino Braga¹**

¹Departamento de ciência da computação
Universidade Federal de Juiz de Fora (UFJF) – Juiz de fora, Brasil.

{daniel.rezende, matheus.casarim, pedro.bonorino}@estudante.ufjf.br

{mathews.gomes}@engenharia.ufjf.br

Abstract.

Resumo.

1. Introdução

2. Uso de Jogos no Ensino e Aprendizagem de Algoritmos

Desde o início da vida, as pessoas têm contato com jogos de alguma forma. O estímulo natural gerado pelo desafio dos jogos é um impulso que vem sendo explorado há tempo por professores e educadores, principalmente na fase infantil, onde são desenvolvidas habilidades de coordenação, raciocínio e tomada de decisão. Nessa fase, as crianças são estimuladas a aprender com o uso de jogos, sejam eles físicos ou digitais, como jogos em que a criança deve colocar uma figura geométrica no buraco com a forma correspondente. Dessa forma, nos últimos anos tem-se observado a criação de muitas alternativas de ensino no formato remoto, bem como o auxílio de aplicativos de forma integrada ao ensino. Simultaneamente, também se observa o crescimento da quantidade de jogos digitais, que se tornaram populares em meio a indivíduos de todas as idades, mas especialmente entre crianças e jovens [Fernandes 2010].

Jogos tornam-se ferramentas eficientes quando utilizadas de forma correta no ensino, motivando e facilitando o aprendizado. Ao fazer uso de um jogo educativo, o jogador aumenta a capacidade de reter o conteúdo aprendido. A atração do jogo deve-se ao fato dos jogos possuírem regras, metas, interação e oferecerem ao usuário uma experiência divertida. Desta forma, os jogos forçam o aluno a decidir e priorizar, benefícios que auxiliam no aprendizado não apenas da computação, mas também para a vida pessoal e acadêmica [Monclar et al. 2018].

O mau aprendizado da lógica de programação influencia o futuro do aluno como desenvolvedor, já que sem uma boa base da lógica, se torna mais difícil o aprendizado de conceitos mais avançados. Dessa forma, a qualidade do software desenvolvido e o tempo de desenvolvimento são prejudicados. Nesse sentido, muitos alunos da disciplina de Algoritmos não necessariamente trabalharão como programadores. Grande parte desses estudantes estão relacionados a outros cursos sem foco em programação, como Matemática, Química e Engenharias. Porém, grande parte destes utilizam ou utilizarão a programação de alguma forma, seja em pesquisas ou estudos que necessitam do auxílio de softwares

a serem desenvolvidos pelos próprios estudantes. Geralmente, os alunos de outras áreas são os que mais sentem dificuldades no aprendizado de algoritmos, já que não tiveram contato com práticas e conceitos da área anteriormente [Marcussi et al. 2016].

Focando na parte de raciocínio, a lógica ensinada em disciplinas de exatas é grande parte da dificuldade no aprendizado na disciplina de Algoritmos. Algoritmos são sequências de instruções, escritas de forma lógica, para resolver um problema específico. Para isso, é necessário o conhecimento da lógica e de uma linguagem de programação. Em meio ao modelo tradicional de ensino, jogos, principalmente digitais, surgem para dar uma nova experiência de aprendizado e ensino de algum tópico, permitindo o conhecimento e diversão de forma conjunta, melhorando a experiência de ambas as partes, aluno e professor [Souza et al. 2013]. O uso de jogos nesse aspecto, auxilia professores usando o estímulo gerado pelo jogo, junto ao conteúdo ensinado por ele, para combinar foco e competitividade com a fixação de conceitos e conteúdos [Pontes 2013]. Dessa forma, cria-se um laço afetivo do aluno com o aprender, facilitando e estimulando a aprendizagem.

Tendo isso em vista, um dos maiores desafios dessa abordagem de ensino é o desenvolvimento de um jogo que consiga equilibrar os elementos de narrativa e estética que prendam a atenção dos usuários com estratégias e mecânicas de jogabilidade que garantam o aprendizado dos alunos. Como mostram os trabalhos apresentados na seção 4, priorizar a estética deixando de lado as mecânicas podem prejudicar a experiência de jogabilidade, como é o caso dos resultados observados pelos desenvolvedores do trabalho [Ferreira et al. 2021] em que os alunos que participaram da pesquisa definiram o jogo como pouco desafiador.

Por outro lado, ignorar os conceitos de estética e história e focar apenas na jogabilidade pode não ser suficiente para prender a atenção dos jogadores. A elaboração de narrativas se configura como um aspecto essencial para garantir a eficácia da aprendizagem, uma vez que adotadas estratégias que mesclam o conteúdo da disciplina ensinada com aspectos e elementos do cotidiano dos alunos, o engajamento e dedicação dos estudantes tende a aumentar. Um exemplo disso são trabalhos que desenvolvem jogos de aventura para conduzir os desafios e missões de seu jogo, tornando a experiência mais atrativa [Venson et al. 2022].

3. *Coding Fight*: Concepção e Desenvolvimento

Dada a dificuldade dos estudantes no aprendizado de Algoritmos, abordada na seção anterior, o presente trabalho buscou elaborar um método alternativo para o ensino da disciplina. Isso foi feito com o intuito de amenizar a evasão e estimular o aprendizado dos estudantes ingressados nos cursos de exatas da Universidade Federal de Juiz de Fora (UFJF). No projeto desenvolvido, o jogo *Coding Fight* pretende auxiliar professores e estudantes da disciplina de Algoritmos no ensino e aprendizado de fundamentos de programação, com foco inicial na linguagem C. No entanto, acredita-se que existe a possibilidade de incluir módulos que também transmitam esses fundamentos para outras disciplinas e linguagens de programação.

De forma geral, o *Coding Fight* integra elementos de jogos de luta *single-player* com sistemas de perguntas e respostas (*quiz*) para compor as mecânicas do jogo, vide figura 1. Ao acertar a resposta de um desafio, o jogador inflige dano ao inimigo, podendo

receber dano caso responda de forma incorreta. O conteúdo da disciplina é dividido em seções da matéria, dando a oportunidade do aluno treinar um conteúdo específico da disciplina e poder se orientar nele. Além disso, a aplicação permite que o professor analise quais as maiores dificuldades dos seus alunos, podendo variar o tempo gasto, quantidades e tipos de desafio por módulo, bem como as metodologias de ensino dos conteúdos.



Figura 1. Tela de desafios do jogo.

A página de desafios do jogo conta com dois avatares de personagens, um representando o aluno e outro representando o seu oponente. Cada avatar possui inicialmente 100 pontos de vida, valor que pode ser reduzido à medida que o aluno progride nos desafios. No entanto, deve-se salientar que ao deixar de responder as questões no tempo determinado, o jogador automaticamente leva dano do oponente. Quando a vida de um dos dois jogadores é zerada, o jogo acaba e uma mensagem de comemoração ou de motivação é exibida. Efeitos visuais e sonoros serão adicionados no intuito de reproduzir o dano sendo causado pelo oponente ou pelo jogador, como resposta às ações do aluno no jogo.

Outro ponto que deve-se destacar é a mecânica de ranqueamento. De forma resumida, o jogador recebe mais pontos conforme a dificuldade dos desafios aumenta. Isso também é refletido na quantidade de dano que ele causa no inimigo ao acertar uma questão. Assim, o aluno sente a sua progressão à medida que responde os desafios e não é muito penalizado quando erra questões difíceis, o que o estimula a continuar jogando e aprendendo. A tabela 1 mostra de forma mais detalhada como essa mecânica funciona. Adicionalmente, o aluno pode habilitar ou desabilitar a exibição de sua pontuação no *ranking* geral, acessando as configurações da sua conta. Contudo, a sua pontuação privada continuará sendo salva no banco de dados para que o professor possa avaliar o desempenho individual e geral de seus alunos.

Por fim, uma mecânica de *Combos* foi implementada no jogo. Essa ferramenta é frequentemente utilizada em jogos de luta e é ativada quando o jogador executa uma sequência de acertos, geralmente recebendo mais pontos ou causando mais dano que o normal aos adversários. De forma similar a outros jogos de luta, os combos imple-

Tabela 1. Mecânica de ranqueamento detalhada.

Dificuldade do desafio	Dano recebido (ao errar)	Dano causado (ao acertar)	Pontos adquiridos (ao acertar)
Fácil	7 a 10	1 a 3	3
Média	4 a 6	4 a 6	7
Difícil	1 a 3	7 a 10	11

mentados no *Coding Fight* são ativados quando o usuário acerta 3 ou mais questões em sequência, sendo recompensado com um bônus (*buff*) aleatório, como regeneração de vida, mais tempo para resolver as questões, mais dano ou recebendo um multiplicador de pontuação. Isso visa estimular o interesse dos jogadores a continuar evoluindo através dos módulos propostos pelo professor.

No sistema do jogo existem dois tipos de usuário: aluno e professor. Ambos precisam realizar o cadastro e efetuar o *login* antes de acessar qualquer página. Após conectar ao sistema, os alunos têm acesso a apenas algumas funcionalidades como: cadastro, seleção de módulos, desafios, reclamações/sugestões e *ranking* geral. Ao passo que os professores podem usufruir das mesmas funcionalidades dos alunos, com a adição da criação de turmas e desafios, atribuição de alunos às turmas e *ranking* privado dos estudantes. Sendo importante destacar que apenas os professores podem adicionar novos desafios e que estes devem ser compostos de uma pergunta com seus respectivos campos de conteúdo, dica e respostas devidamente preenchidos.

Atualmente, o jogo se encontra na etapa de criação das animações, efeitos sonoros e finalização da interface gráfica.

3.1. Ferramentas de desenvolvimento

As páginas do jogo foram desenvolvidas a partir do *framework* *Vue.js*, baseado na linguagem *JavaScript*. Um dos maiores diferenciais desse *framework* é a possibilidade de reutilização de componentes de forma simples e rápida, ao mesmo tempo em que permite que as interfaces sejam intuitivas e fáceis de desenvolver, valorizando assim, a experiência do usuário e do programador.

Para o armazenamento de dados e autenticação, foram utilizadas as ferramentas *Realtime Database* e *Authentication* da plataforma *Firebase*, na versão gratuita. Entre as vantagens estão a possibilidade de analisar, em tempo real, o comportamento dos usuários, além de permitir o sincronismo dos dados entre os desafios inseridos pelo professor e a pontuação dos alunos. Para disponibilização aos usuários, o código fonte do jogo foi disponibilizado em <https://github.com/PedroBonorin0/Daniel_Matheus_Mathews_Pedro>.

4. Trabalhos Relacionados

Pesquisas relacionadas ao uso de ferramentas tecnológicas de ensino e aprendizagem têm sido amplamente conduzidas para favorecer o aprendizado dos estudantes de qualquer nível de ensino, a partir do uso de jogos e/ou sistemas gamificados. Para esse propósito, é importante que haja a integração de diversas metodologias, recursos pedagógicos e tecnologias a fim de se permitir que os professores tenham mais liberdade para planejar e

aplicar diferentes estratégias de ensino às necessidades de sua turma [Schmitt 2018]. Em vista disso, os trabalhos tratados nessa seção abordam o uso da informática na educação, relacionando, mais especificamente, a eficácia do uso de jogos no aprendizado tanto de disciplinas comuns da educação básica, quanto de disciplinas mais específicas de determinados cursos do ensino superior.

Começando pela educação básica, o jogo “Apolo” [Melo et al. 2019] mostra a criação de um jogo baseado no *framework Progressive Web App (PWA)* com o objetivo de apoiar o ensino em sala de aula de forma lúdica e estimulante para os alunos. O *software* foi inspirado no jogo de tabuleiro conhecido como “Perfil”, sendo importante evidenciar que varias regras do jogo original foram adaptadas de acordo com as demandas dos professores. Um dos pontos semelhantes ao *Coding Fight* é a possibilidade de personalização, registro e acompanhamento individual ou coletivo dos desafios concluídos pelos alunos/equipes. Outra vantagem é versatilidade do *framework* utilizado, já que ele torna a experiência de uso dinâmica e acessível por dispositivos móveis, tablets e computadores. O estudo de caso para a disciplina de Geografia de uma turma do Ensino médio revelou a avaliação positiva tanto dos estudantes quanto dos docentes e também mostrou que existe potencial para reforçar os recursos pedagógicos para o ensino brasileiro.

Para o ensino de idiomas, um exemplo de jogo é o *Complex World* [Montanher et al. 2021]. Desenvolvido com o *RPG Maker* e *Dialogflow*, o jogo foi idealizado como ferramenta de auxílio ao aluno no aprendizado de inglês como segunda língua, baseando-se, principalmente nos conceitos presentes na metodologia do Pensamento Complexo [Morin and Lisboa 2007]. No jogo, dois ambientes foram implementados e o objetivo do jogador é usar o seu personagem para cumprir missões interagindo com o cenário e evoluindo na narrativa por meio de *chatbots*. A avaliação do trabalho se deu por meio de questionário. Os resultados mostraram que o jogo conseguiu manter a atenção e interesse dos jogadores ao mesmo tempo em que proporcionou a sensação de progresso e aprendizado. Além disso, a fase de experimentação mostrou que existe a possibilidade de se inserir novos conteúdos e modelos de avaliação, bem como aplicar o jogo em paralelo com avaliações em sala de aula e com o ensino tradicional.

No contexto do ensino superior, é possível destacar o uso de jogos sérios como forma de aumentar o engajamento dos estudantes e consequentemente proporcionar a formação de profissionais com melhor qualificação para o mercado de trabalho. A partir disso, o trabalho [Venson et al. 2022] descreve o processo de desenvolvimento de um jogo digital para o aprendizado dos conteúdos relacionados a sistemas de energia fotovoltaica nos cursos de engenharia. Nele, os jogadores assumem o papel de engenheiros responsáveis por garantir o funcionamento dos equipamentos fotovoltaicos do planeta. O *software* foi desenvolvido na plataforma *Unity*¹ e elaborado com base no *Heuristic Framework* [Dickey 2006], o qual utiliza a estrutura de jogos de aventura para a elaboração de narrativas voltadas para a educação com foco na resolução de problemas. Esse modelo de desenvolvimento se torna um diferencial uma vez os jogadores tendem a aumentar seu engajamento à medida que as narrativas se desenvolvem. De acordo com os desenvolvedores, os resultados obtidos pelas suas pesquisas, mostram uma boa aceitação dos estudantes em relação ao uso do jogo para o aprendizado na área.

¹<https://unity.com/>

Ainda nesse contexto, o trabalho [Ferreira et al. 2021] elabora um jogo em 2D para navegadores que visa o uso de técnicas de *storytelling* para a apresentação dos conceitos fundamentais sobre Ecossistemas de *Software*. Nele, o jogador é inserido em um mundo de fantasia e interage com os elementos e conteúdos da área de Ecossistemas de *Software* através de histórias e metáforas. O jogo foi desenvolvido na plataforma Construct 2² através do ENgAGED (*EducatioNAl GamEs Development*), um processo de desenvolvimento de jogos que explora etapas de análise, desenvolvimento e avaliação do jogo. Para a avaliação, os desenvolvedores utilizaram o MEEGA+ [Petri et al. 2018], um modelo para a medição de qualidade de jogos educacionais, e os resultados mostram que o jogo possui uma boa usabilidade, porém alguns dos grupos de estudantes analisados consideraram o jogo monótono e pouco desafiador. De acordo com os autores, isso pode estar relacionado com a simplicidade das mecânicas implementadas, o que acarreta uma maior facilidade de assimilação do funcionamento do jogo por parte dos jogadores e consequentemente faz com que eles passem mais tempo assistindo *cutscenes* do que jogando.

Alguns trabalhos têm focado na adaptação/construção de jogos que visem facilitar o ensino de lógica de programação, disciplina que é de suma importância para os cursos de computação em geral. Nesse contexto o artigo [Silva et al. 2021], que foi conduzido sobre alunos ingressantes do ensino superior. Nele, o jogo *ProgramSE* foi desenvolvido com inspiração em jogos do gênero “*Escape the Room*”³ para auxiliar o professor no ensino de novos conceitos de lógica de programação e ajudar os alunos a fixar conhecimentos prévios da disciplina. Os resultados do trabalho indicaram que é possível utilizar jogos para o ensino, no entanto, eles devem ser pensados para além do entretenimento do jogador, permitindo que sua progressão ocorra naturalmente na disciplina, sem a necessidade de memorização. Por fim, deve-se ressaltar que também foram sugeridas melhorias como: inclusão de novas mecânicas e fases e eliminação de pontos de repetição e monotonia.

Por fim, também pode-se citar o artigo [Panegalli et al. 2019], que desenvolveu um jogo inspirado no design, história e jogabilidade do jogo *Super Mario World*. Porém, no jogo proposto pelo artigo, ao invés do jogador controlar o personagem de maneira convencional, ele monta o movimento do personagem utilizando blocos de comando. Essa estratégia corresponde aos conceitos introduzidos pela disciplina de lógica de programação. Outro ponto a se observar é que a apresentação do conteúdo se dá de maneira gradativa, ou seja, começando de estruturas mais simples para as mais complexas. A avaliação do jogo foi realizada por meio um experimento com alunos utilizando o método *GameFlow*, o qual rendeu notas médias - boas. Entretanto apesar de apresentar de maneira palatável os fundamentos de lógica de programação, o jogo não dispõe os pseudo-códigos gerados pelos alunos durante as fases, o que dificulta um maior aprofundamento nas disciplinas de computação.

5. Análises Comparativas

Tendo em vista o que foi discutido anteriormente, esta seção busca realizar comparações entre o *Coding Fight* e os trabalhos apresentados na seção 4 de forma a ponderar as

²<https://www.construct.net/>

³Uma modalidade de jogo de desafio, misturando aventura e raciocínio, cujo objetivo é encontrar uma maneira de escapar de um lugar misterioso, utilizando todo tipo de objetos que encontrar.

melhores estratégias para a construção de um jogo educacional. Para isso, contruiu-se a tabela 2 elencando as métricas de comparação entre cada jogo analisado. Dentre elas, a linguagem de programação em que o jogo foi desenvolvido, o tema, o conteúdo a ser ensinado, a mecânica de jogo, a metodologia de avaliação e os resultados obtidos com a pesquisa.

Tabela 2. Comparação dos Projetos.

Nome	Desenvolvimento	Mecânica	Tema	Conteúdo	Avaliação
<i>Coding Fight</i>	<i>Web App</i>	Quizz	Luta	Algoritmos	-
<i>Apolo</i>	<i>Web App</i>	Tabuleiro	Perfil	Educação Básica	Estudo de Caso
<i>Complex World</i>	RPG Maker	Pensamento Complexo	RPG	Inglês	Questionário
<i>Phototype</i>	<i>Unity</i>	<i>Heuristic Framework</i>	Aventura	Engenharia	Questionário
Árvore de Ecos	<i>Web App</i>	<i>storytelling</i>	Fantasia	Ecossistemas	ENgAGED
<i>ProgramSE</i>	<i>Construct 3</i>	<i>point and click</i>	<i>Escape the Room</i>	Programação	Experimental
Super Mario Logic	<i>Interad</i>	Plataforma	Estratégia	Programação	<i>GameFlow</i>

Primeiramente, observou-se que em grande parte do artigos analisados, existe uma propensão ao uso de algum tipo de questionário ou *quiz* personalizado. Um exemplo disso é o jogo “Super Mario Logic” [Panegalli et al. 2019]. Nele, o *quiz* se torna parte do jogo, uma vez que as respostas aos desafios são blocos de comando que decidem o comportamento do personagem, onde o objetivo final é acertar todos os comandos e concluir a fase. Esse modelo de jogabilidade pode ser uma boa alternativa para futuros aprimoramentos no *Coding Fight*, tendo em vista que o projeto consiste apenas na obtenção de pontos a partir de acertos. A vantagem da utilização de *quizes* como mecânica de jogo é a possibilidade de identificar mais facilmente as dificuldades dos alunos, uma vez que as respostas corretas já estão definidas, sem a necessidade de interpretações externas para sua validação. Mas, em contrapartida, isso pode gerar a desestimulação dos alunos, uma vez que as fases do jogo podem se assemelhar muito a uma prova ou lista de atividades. Nesse caso, por exemplo, uma escolha interessante seria a dinamização dos desafios de forma que o jogador precisasse escolher a melhor forma de resolvê-los para causar um dano maior ao adversário. Isso tornaria o jogo mais desafiador e consequentemente aumentaria o engajamento dos alunos.

A disciplina de Algoritmos normalmente é dividida em módulos que vão desde introdução à programação, passando por funções, condicionais, *strings*, entre outros, até estruturas de dados. Essa divisão do conteúdo se faz presente na ementa da disciplina [UFJF 2022] e na abordagem tradicional aplicada em sala de aula, tanto na UFJF quanto em outras instituições ou cursos online. Esse tipo de divisão também ocorre em outros jogos e trabalhos relacionados, como é o caso do “Árvore de ECOS” e “*ProgramSE*” que foram abordados anteriormente [Silva et al. 2021, Ferreira et al. 2021]. Por esse motivo, acredita-se que jogos ou sistemas *Gamificados* podem ser úteis para incentivar o aprendizado na disciplina, uma vez que o aluno será estimulado a alcançar melhores pontuações e a realizar mais *Combos* durante os desafios. Isso faz com que o aluno se sinta menos forçado a estudar conteúdos que nunca fizeram parte da sua formação acadêmica ou que apresentam alta curva de aprendizado.

Sob o ponto de vista de implementação, a escolha da linguagem e da plataforma de desenvolvimento podem ser diferenciais na aceitação e eficácia do jogo. Alguns projetos foram desenvolvidos em ambientes mais elaborados e direcionados para a criação de jogos, como o *Phototype* que utilizou a plataforma *Unity*. A utilização desses ambientes

pode demandar um nível de conhecimento maior da ferramenta mas possibilitam a criação de jogos mais elaborados tanto do ponto de vista visual e auditivo quanto em questão de mecânicas e jogabilidade. O *Coding Fight* por exemplo, é baseado em um sistema *web*, o que trás algumas limitações para o projeto. O *design* e elementos visuais precisam ser mais simples para não causar impacto no desempenho. Mas, por outro lado, por se tratar de um sistema *web*, ele se torna acessível a uma gama maior de usuários.

Outro aspecto a ser analisado, é a temática do *game*. Esse, por sua vez, não possibilita apontar de forma precisa os benefícios e prejuízos na adoção de cada um. Isso se deve ao fato de que cada pessoa possui um gosto diferente. Nesse caso, é necessário analisar os temas mais populares ou que mais estão presentes no cotidiano do público alvo, além de considerar as formas como eles podem se encaixar ao conteúdo a ser ensinado. Para o *Coding Fight* por exemplo, optou-se por utilizar o tema de jogos de luta em que a cada acerto o jogador aplica dano ao oponente. Mas os projetos analisados também exploram outras possibilidades de temas interessantes para o contexto de jogos educacionais, como é o caso de *Árvore de Ecos* que usa metáforas do mundo da fantasia para ensinar os conceitos de ecossistemas.

Por fim, é necessário considerar o método de avaliação utilizado por cada projeto apresentado. Como discutido anteriormente, não há uma forma correta ou incorreta de realizar a etapa de avaliação, é necessário escolher o modelo que melhor se encaixa no contexto do projeto e na forma como o experimento foi organizado. Tendo isso em vista, nota-se que alguns trabalhos como *Apolo* e *Phototype* escolheram o questionário como forma de avaliar os resultados dos seus jogos. Outros optaram por modelos baseados em heurísticas de pontuação como é o caso de *Árvore de Ecos* que utilizou o método ENgAGED. Já no caso do *Coding Fight*, como o jogo ainda se encontra na etapa de desenvolvimento, não há uma método de avaliação definido.

6. Considerações Finais

Referências

- Dickey, M. D. (2006). Game design narrative for learning: Appropriating adventure game design narrative devices and techniques for the design of interactive learning environments. *Educational technology research and development*, 54(3):245–263.
- Fernandes, N. A. (2010). Uso de jogos educacionais no processo de ensino e de aprendizagem.
- Ferreira, T. d. S. D., Viana, D., and dos Santos, R. P. (2021). *Árvore de ecos: Um jogo para ensino de conceitos de ecossistemas de software*. *Revista Brasileira de Informática na Educação*, 29:273–300.
- Marcussi, L. D., Guedes, K., Dal Molin Filho, R. G., Santiago Filho, R. M., and Junior, C. R. B. (2016). Pesquisa no ensino de algoritmos e programação nas engenharias: Estudos e resultados preliminares. In *Simpósio de Engenharia de Produção*.
- Melo, M. V. C. d., Segato, T., and Mombach, J. (2019). *Apolo: Versão digital de um jogo de tabuleiro para apoio ao ensino na educação básica*. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 30, page 873.

- Monclar, R. S., Silva, M. A., and Xexéo, G. (2018). Jogos com propósito para o ensino de programação. *Anais do XVII Simpósio Brasileiro de Jogos e Entretenimento Digital-SBGames*, pages 1132–1140.
- Montanher, R. C., Zadi, I. C., and Monteiro, A. M. (2021). Complex world: um jogo para auxiliar o aprendizado de inglês como segunda língua utilizando a abordagem pedagógica do pensamento complexo. In *Anais do XXXII Simpósio Brasileiro de Informática na Educação*, pages 417–428. SBC.
- Morin, E. and Lisboa, E. (2007). *Introdução ao pensamento complexo*, volume 3. Sulina Porto Alegre.
- Panegalli, F. S., Bernardi, G., and Cordenonsi, A. Z. (2019). Super mario logic: um jogo sério para auxiliar no processo de ensino e aprendizagem de lógica de programação. *RENOTE*, 17(1):244–253.
- Petri, G., von Wangenheim, C. G., and Borgatto, A. F. (2018). Meega+: A method for the evaluation of educational games for computing education. *INCoD–Brazilian Institute for Digital Convergence*, pages 1–47.
- Pontes, H. P. (2013). Desenvolvimento de jogos no processo de aprendizado em algoritmos e programação de computadores. *Proceedings of the XII Simpósio Brasileiro de Games e Entretenimento Digital (SBGames)*. São Paulo.
- Schmitt, C. (2018). A integração das tdc à educação matemática um estudo sobre o uso de ferramentas digitais e metodologias ativas no ensino e aprendizagem de matemática.
- 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.
- Souza, M. d., Jaeger, E. V., and da Silva Cardoso, B. M. (2013). Ensino de algoritmos apoiado pelo uso de jogos digitais educativos. *RENOTE*, 11(3).
- UFJF (2022). Algoritmos - ufjf. Acessado: 10 de Julho de 2022.
- Venson, R., Callaghan, M., and Marcelino, R. (2022). Phototype: Um jogo sério para fixação de conhecimento em sistemas fotovoltaicos. *ETD-Educação Temática Digital*, 24(2):275–295.