

Programando com o Swift: da concepção à avaliação de um jogo desplugado

Walquiria P. Santos¹, Rozelma S. de França²

¹Departamento de Computação, ²Departamento de Educação

Universidade Federal Rural de Pernambuco (UFRPE)

CEP: 52171-900 – Recife – PE – Brasil

walquiria.psantos, rozelma.franca@ufrpe.br

Abstract. *Teaching Computer Science in Basic Education has the potential to develop fundamental skills such as critical and creative thinking and the ability to solve problems; in order for these objectives to be effectively achieved, the provision of diverse teaching resources is essential. This article describes the adaptation of the digital game “Learn to Program 1” available in the Swift Playgrounds app. Based on the Unplugged Computer Science approach that works on alternative ways of learning programming concepts without relying on digital devices, the goal of this game is to demonstrate that it is possible to program unplugged with a programming language. Using a Swift language, the player works on syntax and learns basic programming concepts such as functions, loops, conditionals and logical operators, having the necessary commands printed to solve the challenges.*

Resumo. *O ensino de Computação na Educação Básica tem o potencial de desenvolver habilidades fundamentais como o pensamento crítico, criativo e a capacidade de resolver problemas; para que esses objetivos sejam efetivamente alcançados a disponibilização de recursos didáticos diversos é essencial. Este artigo propõe a adaptação e avaliação do jogo digital “Aprenda a programar 1” disponível no aplicativo Swift Playgrounds. Baseado na abordagem Ciência da Computação Desplugada que trabalha formas alternativas de aprender conceitos de programação sem depender de dispositivos digitais, o objetivo deste é demonstrar que é possível programar de forma desplugada com uma linguagem de programação. Utilizando a linguagem swift, o jogador trabalha a sintaxe e aprende conceitos básicos da programação como funções, loops, condicionais e operadores lógicos tendo os comandos necessários impressos para resolver os desafios.*

1. Introdução

Vivemos num mundo em que a tecnologia digital se faz mais presente a cada dia. Dispositivos inteligentes, aplicativos e sistemas automatizados têm se tornado cada vez mais comuns e acessíveis às pessoas, e compreender como eles funcionam é de vital importância, não apenas para aqueles que os desenvolvem, mas também para os que utilizam. Aprender a programar não é apenas escrever códigos, mas é compreender como os computadores funcionam de forma lógica e estruturada, desenvolvendo desta forma um pensamento crítico, lógico e estruturado, conhecido como pensamento computacional (PC).

O PC é uma habilidade que envolve resolver problemas a partir de técnicas da Computação, criar estratégias e tomar decisões com base em dados e informações (Wing, 2006). Isso evidencia a importância do ensino de computação na educação básica, pois amplia as competências cognitivas e sociais dos estudantes, promovendo

uma formação integral. Além disso, introduzir conceitos de programação e lógica aos estudantes pode os tornar não apenas consumidores, mas também criadores conscientes de tecnologias digitais, o que é essencial para sua inserção crítica e ativa na sociedade.

A Base Nacional Comum Curricular (BNCC), em seu complemento de Computação (Brasil, 2022), já prevê o desenvolvimento de competências e habilidades dessa natureza, organizadas em três eixos: Pensamento Computacional, Mundo Digital e Cultura Digital. Nesse contexto, o ensino de computação na educação básica deve contemplar a formação de cidadãos digitais que são capazes de usar e compreender as tecnologias de forma crítica, ética e responsável, como também desenvolver habilidades técnicas, como programação, para resolução de problemas por meio da tecnologia, assim como desenvolver o PC para resolver problemas de forma lógica e eficiente.

Em um contexto em que há escassez de infraestrutura tecnológica, a programação desplugada surge como uma alternativa viável e eficaz, pois a mesma oportuniza o aprendizado de conceitos básicos de programação sem o uso de computadores e suas tecnologias associadas. Materiais como jogos, cartas, blocos lógicos, tabuleiros e outros objetos manipuláveis são essenciais para criar experiências práticas que ajudam os estudantes a compreender noções abstratas como algoritmos, lógica e estruturas de dados. Esses recursos podem favorecer a aprendizagem colaborativa e o desenvolvimento de habilidades cognitivas e socioemocionais, ao mesmo tempo em que tornam o processo de ensino mais acessível, principalmente em contextos onde o acesso a computadores e internet é limitado.

Diante disso, a metodologia de Computação Desplugada (CD) surge como uma possibilidade de trabalhar o PC; esta proposta foi sistematizada por Bell, Written e Fellows (2009) no projeto *Computer Science Unplugged*, o qual utilizava jogos e atividades físicas para ensinar conceito de computação. Alguns trabalhos também reforçam a efetividade do uso da CD nas escolas com recursos limitados. Em Monteiro Rodrigues *et al.* (2021) a CD foi trabalhada juntamente com o Scratch: primeiro foi trabalhado a CD, o que trouxe maior engajamento dos estudantes e, por conseguinte, a introdução gradual dos conceitos de programação para iniciação no Scratch.

A utilização de jogos no processo de ensino-aprendizagem tem sido também amplamente estudada e utilizada como uma estratégia eficiente para promover engajamento, motivação e desenvolvimento cognitivo. Segundo Gee (2003), jogos oferecem contextos ricos para o aprendizado, pois jogadores, ao interagirem com os desafios, constroem significados e estratégias de forma autônoma e contextualizada. Para Martins, Reis e Marques (2016) o uso de jogos educativos para ensinar programação não apenas torna a aprendizagem envolvente, mas também ajuda os estudantes a aplicar conceitos teóricos em situações práticas.

Nesta pesquisa, a proposta de adaptação do jogo digital Swift Playgrounds em um jogo físico surge como uma alternativa pedagógica relevante especialmente em contextos educacionais onde há limitações quanto ao acesso a dispositivos tecnológicos e à internet. Muitas escolas da educação básica, em especial em regiões periféricas ou com menos recursos, não dispõem de laboratórios de informática adequados, número suficiente de computadores ou conexão estável à internet para realizar atividades que dependem exclusivamente de recursos digitais. Além disso, a proposta amplia o acesso

às experiências educativas de Computação, o que pode garantir a inclusão de todos os estudantes, independentemente de sua realidade socioeconômica; neste ínterim.

O restante do artigo está organizado como segue: a Seção 2 faz um estudo comparativo de trabalhos relacionados a jogos desplugados no ensino da computação com o intuito de identificar semelhanças e contribuições para a aprendizagem de computação na educação básica; as Seções 3 e 4 apresentam o jogo desplugado desenvolvido, e resultados de sua aplicação em sala. Por fim, a seção 5 tece as considerações finais acerca do trabalho e possibilidades de trabalhos futuros.

2. Trabalhos Relacionados

Desenvolver as habilidades do PC na educação básica tem sido um tema amplamente trabalhado por diversos educadores. Desta forma, esta seção traz um quadro comparativo de contribuições de diferentes artigos ao ensino de computação na educação básica com o uso de artefatos desplugados. Para isto, foi realizada uma criteriosa busca em fontes acadêmicas reconhecidas, como a biblioteca digital da Sociedade Brasileira de Computação (SBC OpenLib) e o Portal de Periódicos da CAPES. As palavras chaves utilizadas nos campos de pesquisa foram: “Jogo desplugado”, “jogo de tabuleiro”, “unplugged”, “jogo” e “desplugado”; a seleção dos artigos teve como foco identificar experiências e propostas pedagógicas que abordassem o ensino de computação, especialmente com ênfase no PC e no uso de atividades desplugadas através de jogos. A síntese desses resultados é apresentada no Quadro 1.

Quadro 1. Trabalhos correlatos com o uso de jogos desplugados para ensino de Computação

Título do Artigo	Ano	Autores	Público-Alvo	Artefato Produzido	Contribuições para o Ensino de Computação
Warfall: ensino de representação de imagens por meio de pixel art em jogo de tabuleiro	2024	Juliete L. S. Cavalcante, Augusto C. P. S. Montalvão	Estudantes do Ensino Médio	Jogo de tabuleiro tipo RPG (Warfall)	Ensina representação de imagens com pixel art e promove o PC com uso de monogramas em formato desplugado.
Desenvolvimento e Avaliação de Material Didático Desplugado para o Ensino de Computação na Educação Básica	2021	Marcia E. J. K. da Cruz, Samanta G., Wilk, Marques Oliveira.	Professores e estudantes do 4º e 5º ano do EF	Material Didático com 31 desafios organizados em 7 fascículos	Apoia formação docente e ensino de algoritmos e padrões, com impacto positivo na aprendizagem e alta aceitação pelos professores.
Desenvolvimento e Avaliação de um Jogo de Tabuleiro para Ensinar o Conceito de Algoritmos na Educação Básica	2019	Christiane G. von Wangenheim et al.	estudantes do Ensino Fundamenta I (K-12)	Jogo de tabuleiro "SplashCode"	Enfoca o ensino de algoritmos e programação com design lúdico e uso de personagens Kawaii, promovendo motivação e interação social.
CodeBô Unplugged: Um jogo desplugado	2023	Tamilis O. Cerqueira, Andreia P. S. Silva, Luis G. J. Araujo	estudantes do Ensino Fundamenta	Jogo desplugado "CodeBô"	Ensina estrutura de dados "pilha" com movimentação em

para o ensino de Pilha			I II	Unplugged	mapa matricial, usando cartões de ação e lógica para compreensão prática.
Um jogo de tabuleiro para integrar Matemática e Pensamento Computacional no Ensino Fundamental	2024	Mauro R. Silva, Ana P. Nahirne, Hênio D. F. Oliveira, Clodis Boscaroli	estudantes do Ensino Fundamental I (Escola do campo)	Jogo de tabuleiro "Trator Wars"	Integra conceitos de plano cartesiano com pilares do pensamento computacional, de forma lúdica e desplugada.
Trabalhando habilidades da BNCC Computação: Jogo não digital com foco no Desenvolvimento do Pensamento Computacional Desplugado	2023	Graziela F. Guarda et al	estudantes dos Anos Iniciais do Ensino Fundamental I	Customização do jogo "Cara a Cara – Personalidades Históricas"	Alinha habilidades da BNCC com os pilares do pensamento computacional, propondo atividade acessível e adaptável para professores.

Foram priorizados estudos que apresentassem dados sobre o público-alvo, artefatos didáticos desenvolvidos e suas aplicações práticas em contextos escolares. A análise permitiu reunir produções que, embora diversas em abordagem e objetivos, compartilham a proposta de tornar o ensino de computação mais acessível, desplugado, lúdico e alinhado à BNCC. O jogo desplugado “programando com Swift” aqui proposto foi idealizado com a mesma proposta de tornar o aprendizado de conceitos computacionais mais acessível, especialmente em contextos educacionais com infraestrutura tecnológica limitada e, no caso deste, a interação com a sintaxe de uma linguagem de programação. Também, ele difere dos trabalhos correlatos por proporcionar uma experiência de aprendizagem tátil e visual, que favorece o entendimento de conceitos abstratos, como algoritmos e estruturas sequenciais. Também, o artefato construído pode aproximar os estudantes de uma linguagem de programação usada profissionalmente, além de promover o trabalho em equipe, a argumentação e o raciocínio lógico coletivo.

3. O jogo “Programando com Swift”

O desenvolvimento do “Programando com Swift” parte de uma adaptação do jogo digital “Aprenda a programar 1”, disponível no aplicativo educacional Swift Playground da Apple voltado a ensinar lógica de programação e a linguagem Swift de forma interativa e divertida especialmente para iniciantes. No Swift Playground, por meio da resolução de desafios dentro de pequenos “mundos” 3D, o jogador deve ajudar o personagem (geralmente o Byte, uma criatura simpática) a coletar gemas, acionar interruptores e chegar a determinados pontos usando código Swift para controlá-lo e cada nível apresenta um desafio lógico, que só pode ser resolvido aplicando conceitos de programação. Tal jogo digital também é dividido em lições e capítulos começando do mais simples (comandos básicos) até desafios mais complexos (funções, loops aninhados, condicional aninhada) e o progresso é gradual: cada desafio ensina um conceito novo e reutiliza os anteriores para reforço.

Deste contexto, o “Programando com Swift” aqui proposto assume a forma de um jogo físico de tabuleiro, idealizado durante as aulas de um clube de programação de uma escola particular para os Anos Finais do Ensino Fundamental, como um curso extracurricular no contraturno escolar. O clube funcionava em um laboratório de fabricação digital da escola, e os professores propuseram a ideia de transpor para o meio físico o jogo digital já familiar aos estudantes. O intuito foi trabalhar a computação desplugada favorecendo outras formas de interações e aprendizagem de programação. Nesse processo, a primeira autora deste trabalho atuou como auxiliar dos professores, sendo responsável pela adaptação e implementação do protótipo físico, garantindo a fidelidade à proposta inicial e a viabilidade técnica da execução no laboratório de fabricação digital da escola.

O jogo “Programando com Swift” é composto de 7 desafios de código referentes às sete fases do jogo digital. De cada fase, foi selecionado um desafio que pudesse ser reproduzido no tabuleiro, juntamente com as peças de código para a resolução. O mesmo foi concebido como uma ferramenta pedagógica alternativa, visando apoiar o processo de introdução à linguagem de programação entre estudantes dos Anos Finais do Ensino Fundamental em fase inicial de aprendizagem, promovendo assim o desenvolvimento da habilidade EF06CO02 da BNCC, que prevê a criação de algoritmos que envolvam instruções sequenciais, de repetição e de seleção usando uma linguagem de programação.

3.1. Visão geral do processo de desenvolvimento

A versão física foi idealizada e produzida no laboratório de fabricação digital da escola em conjunto com colaboradores do espaço: estagiários com conhecimento em modelagem 3D e corte em impressora a laser, e seguiu algumas etapas, conforme ilustra a Figura 1.



Figura 1. Processo de desenvolvimento do jogo “Programando com Swift”

A seguir, essas etapas são detalhadas e ilustradas.

3.1.1 Análise do jogo digital Swift Playgrounds

O Swift Playgrounds¹ (Figura 2) é um aplicativo desenvolvido pela Apple que oferece um ambiente interativo para aprender e praticar programação utilizando a linguagem swift². O aplicativo está disponível para iPad e MacOS e é amplamente utilizado em contextos educacionais para promover o desenvolvimento de PC e a criatividade. Para a criação da versão física, foi utilizado o jogo Aprenda a Programar 1, dentre os vários associados ao Swift Playgrounds.

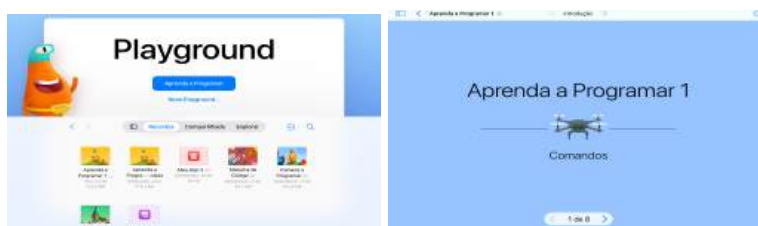


Figura 2. Tela Inicial do aplicativo Swift Playgrounds

3.1.2. Escolha de materiais

Para a confecção da base do tabuleiro (Figura 3), das peças e dos códigos foi utilizado MDF devido à sua resistência, rigidez e facilidade de corte e personalização. O MDF tem uma estrutura sólida, o que evita deformações ao longo do tempo e permite diversos acabamentos, como a gravação a laser, o que confere um aspecto profissional e atrativo ao jogo. O TNT destaca-se devido à sua leveza, versatilidade e baixo custo. Ele foi utilizado como revestimento de algumas peças do tabuleiro que representavam a água (caminho que o jogador não poderia passar).

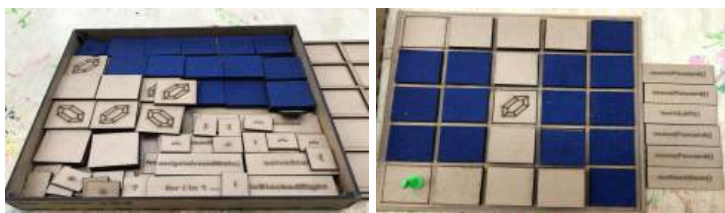


Figura 3. Jogo na caixa (esquerda) e tabuleiro montado de um desafio com a resposta ao lado (direita)

3.1.3. Modelagem no software e corte na cortadora a laser

A modelagem do tabuleiro e dos blocos de código foi realizada no software Inkscape³, com o objetivo de viabilizar a fabricação por meio de corte a laser em MDF. O tabuleiro foi projetado com dimensões de 24 x 24 cm, subdividido em uma malha regular de quadrados de 4 x 4 cm, o que facilita o encaixe e a organização dos blocos. Os blocos de código, por sua vez, foram modelados com alturas fixas de 2,5 cm e larguras variáveis entre 2 e 8 cm, permitindo representar diferentes comandos da sintaxe da linguagem de programação Swift. Cada bloco foi desenhado com precisão vetorial no Inkscape, considerando margens de segurança e espaçamento para corte a laser, e gravado com o

¹Disponível em <https://www.apple.com/br/swift/playgrounds/>

²Disponível em <https://www.swift.org/>

³ Disponível em <https://inkscape.org/p-t-br/>

conteúdo textual correspondente à sua função sintática, como comandos, estruturas de controle e operadores. A escolha do MDF como material base se deu por sua resistência, baixo custo e boa resposta ao corte e gravação a laser, permitindo a produção de peças educativas reutilizáveis para ambientes de ensino de programação física e tangível.

3.1.4. Seleção e adaptação das regras

A adaptação das regras foi uma das etapas mais relevantes, pois foi preciso selecionar alguns desafios de código que eram possíveis serem reproduzidos no tabuleiro levando em consideração o tamanho do mesmo e as limitações, sendo necessário adequar a complexidade algorítmica do ambiente virtual para procedimentos que poderiam ser executados na versão física pelos jogadores. Os jogos selecionados foram: *Acionando um comando* da fase **Comandos** (Figura 4); *Aninhando Padrões* da fase **Funções** (Figura 5); *Loops por todos os lados* da fase **Loops “For”** (Figura 6); *Subida Condicional* da fase **Código Condicional** (Figura 7); *Espiral de Não* da fase **Operadores Lógicos** (Figura 8); *Retângulos Aleatórios* da fase **Loops “While”** (Figura 9) e *Conquistando um Labirinto* da fase **Algoritmos** (Figura 10).



Figura 4. Acionando um Comando - Comandos. Jogo virtual no aplicativo Swift Playgrounds (esquerda) Jogo físico adaptado (direita)

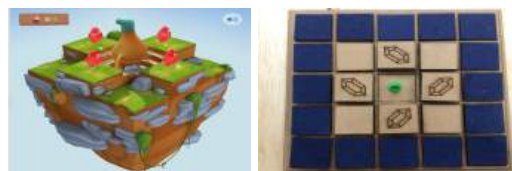


Figura 5. Aninhando Padrões - Funções. Jogo virtual no aplicativo Swift Playgrounds (esquerda) Jogo físico adaptado (direita)

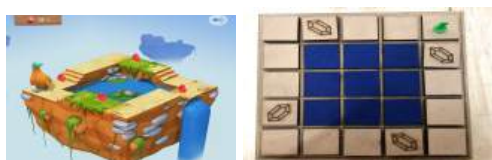


Figura 6. Loops por todos os lados - Loops “For”. Jogo virtual no aplicativo Swift Playgrounds (esquerda) Jogo físico adaptado (direita)



Figura 7. Subida Condicional -Código Condicional. Jogo virtual no aplicativo Swift Playgrounds (esquerda) Jogo físico adaptado (direita)

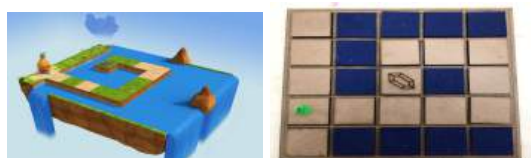


Figura 8. Espiral de Não - Operadores Lógicos. Jogo virtual no aplicativo Swift Playgrounds (esquerda) Jogo físico adaptado (direita)



Figura 9. Retângulos Aleatórios - Loops “While”. Jogo virtual no aplicativo Swift Playgrounds (esquerda) Jogo físico adaptado (direita)



Figura 10. Conquistando um Labirinto - Algoritmos. Jogo virtual no aplicativo Swift Playgrounds (esquerda) Jogo físico adaptado (direita)

A simplificação do jogo foi necessária tendo em vista a jogabilidade e o equilíbrio, essenciais para o engajamento e motivação dos estudantes. Ademais, o processo de adaptação não comprometeu a essência do jogo. Em vez disso, ressignificou através de uma experiência lúdica, na qual o jogador pode explorar novas formas de interação e interpretação das mecânicas originais. No quadro 2 são apresentadas as diferenças e semelhanças entre as versões.

Quadro 2. Comparativo visual entre o jogo digital e o físico

Aspecto	Jogo Digital Swift Playgrounds	Jogo Físico Programando com Swift
Formato	Aplicativo interativo em 3D no iPad	Tabuleiro físico com peças representando comandos e funções
Execução	O código digitado é executado automaticamente pelo sistema	Um jogador ou mediador interpreta e executa os comandos fisicamente
Objetivo	Coletar todas as joias no cenário virtual	Coletar todas as joias no tabuleiro
Interação	Toque e digitação para inserir comandos	Montagem manual de peças em sequência
Feedback	Visual imediato com animação e som	Visual e simbólico, dependendo da imaginação ou da execução por uma pessoa
Público-alvo	Crianças e iniciantes em programação com acesso a dispositivos eletrônicos	Crianças e iniciantes em programação em ambientes sem necessidade de tecnologia digital
Vantagem	Visualização interativa e automática do código	Interatividade física e possibilidade de trabalho em grupo
Limitação	Depende de um dispositivo compatível para instalação	Necessita de interpretação manual e imaginação para simular os movimentos espaciais

O quadro comparativo entre as versões digital e física do jogo permite visualizar de forma clara e objetiva as características, semelhanças e diferenças entre as duas abordagens. Esse recurso auxilia na análise pedagógica evidenciando como cada versão

contribui para o desenvolvimento do raciocínio lógico, do pensamento computacional e das habilidades de resolução de problemas. Além disso, o comparativo facilita a escolha da metodologia mais adequada para diferentes contextos de ensino, considerando recursos disponíveis, perfil dos estudantes e objetivos de aprendizagem. No quadro 3 estão listados os comandos impressos do jogo físico e seus significados para auxiliar a compreensão de como cada comando atua no jogo.


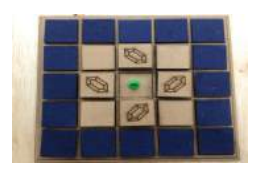
Quadro 3. Comandos e seus significados

Comandos impressos	Significado
moveForward()	Move o personagem uma casa ou passo para frente na direção em que ele está "virado".
collectGem()	Ação de coletar uma gema que está no local onde o personagem se encontra.
turnLeft()	Faz o personagem girar 90 graus para a esquerda (no sentido anti-horário).
turnRight()	Faz o personagem girar 90 graus para a direita (no sentido horário).
func turnAround()	Define uma função chamada turnAround() que faz o personagem girar 180 graus (para voltar na direção oposta)
func solveStair()	Define uma função chamada solveStair() que resolve o problema de subir e descer escadas, movendo-se adequadamente.
func navigateAroundWalls()	Define uma função chamada navigateAroundWalls() que ajuda o personagem a contornar ou evitar obstáculos (paredes) no caminho.
navigateAroundWalls()	Chama a função navigateAroundWalls() para executar a navegação ao redor de obstáculos
turnAround()	Chama a função definida turnAround() , fazendo o personagem girar 180 graus
solveStair()	Chama a função definida solveStair() que resolve o problema de subir e descer escadas, movendo-se adequadamente.
isOnGem	Verifica se o personagem está em cima de uma gema que pode ser coletada.
isBlocked	Verifica se há um obstáculo ou lago bloqueando o caminho do personagem.
isBlockedRight	Verifica se há um obstáculo (parede) ou lago bloqueando o caminho à direita do personagem.
for i in 1 ...	Inicia um loop que itera de $i = 1$ até um valor especificado (geralmente um número inteiro) realizando uma ação repetidamente.
if	Condicional que verifica se uma expressão é verdadeira. Se for, executa o código dentro do bloco.

else	Bloco condicional que é executado quando a condição do if não é verdadeira.
while	Cria um loop que continuará a executar enquanto uma condição for verdadeira.
!	Operador lógico de negação. Inverte o valor da expressão. Exemplo: !isBlocked significa "não está bloqueado".
&&	Operador lógico "E". Retorna verdadeiro apenas se ambas as condições forem verdadeiras. Exemplo: if isBlocked && isOnGem { ... } .
{ }	Chaves - delimitam um bloco de código como estrutura condicional, loop ou função
0...9	Representa os números inteiros de 0 a 9. Usado em expressões numéricas, contagem ou como valores em loops

A compreensão de como funciona cada comando é importante para que os usuários estejam aptos a resolver os desafios de forma mais eficiente e ágil. No quadro 4 temos uma síntese dos desafios propostos no jogo físico, assim como potenciais resoluções para coleta de jóias e os conceitos que podem ser ensinados a partir deles.

Quadro 4. Desafios propostos - resolução e conceitos ensinados

Tabuleiro	Desafio	Comandos disponíveis impressos nas peças (resolução dos desafios)	Conceitos ensinados
	Acionando um Comando	<pre> moveForward() moveForward() turnLeft() moveForward() moveForward() collectGem() </pre>	Lógica sequencial e a importância da ordem dos comandos
	Aninhando Padrões	<pre> func turnAround() { turnLeft() turnLeft() } func solveStair() { moveForward() collectGem() turnAround() moveForward() turnLeft() } solveStair() solveStair() solveStair() solveStair() </pre>	Ensina a criar e chamar funções para agrupar comandos, incentivando a reutilização de código e o conceito de modularidade na programação.

		<pre>while !isOnGem { navigateAroundWalls() } collectGem()</pre>	
--	--	--	--

As regras comuns a todas as fases tem como objetivo principal coletar todas as jóias posicionadas no tabuleiro, seguindo as instruções corretas e na ordem adequada. O peão só pode andar pelos quadrados marrons (caminho livre), pois os quadrados azuis representam o lago e são bloqueios, não sendo permitido passar por eles. Para resolver o desafio de cada fase, o jogador utiliza apenas os comandos impressos disponíveis para a fase (funções, loops, condicionais, operadores). Os comandos devem ser escritos antes da execução e exatamente na sintaxe do Swift, incluindo parênteses e chaves quando necessário. O “código” é então interpretado fisicamente no tabuleiro (movendo o peão passo a passo conforme as instruções). Cada fase introduz um novo conceito de programação, mas mantém a lógica de deslocamento e coleta como desafio central.

3.1.5 Documentação

A documentação foi uma etapa importante do processo pois ela é uma sistematização das informações referentes às regras, aos objetivos e às dinâmicas do jogo, bem como sua reprodutibilidade, compreensão e o uso adequado por diferentes públicos em variados contextos. Também, desempenha um papel importante na avaliação e validação do jogo, uma vez que as regras podem ser testadas, avaliadas e reformuladas. Foi elaborado um tutorial⁴ para os usuários, no qual são explicadas as regras, como montar o tabuleiro e dicas para resolver os desafios.

3.1.6. Testes e avaliação

O jogo foi implementado e analisado pela primeira autora deste artigo com a participação de estudantes do clube de programação ao longo de três encontros com duração média de 1h20min, cada. Cinco estudantes interagiram com o jogo criado. O clube era um curso extracurricular gratuito oferecido no contraturno da escola para os estudantes que estavam nos Anos Finais do Ensino Fundamental, de uma escola particular de padrão socioeconômico médio-alto.

Os estudantes apresentavam um perfil marcado pelo acesso prévio a recursos tecnológicos e oportunidades educacionais diferenciadas, possuíam conhecimentos prévios básicos em programação adquiridos nos Anos Iniciais do Ensino Fundamental por meio de plataformas de programação em blocos e proficiência em inglês suficiente para compreender tutoriais, interfaces de softwares na língua inglesa. Vale salientar que os estudantes do clube estavam acostumados a trabalhar com programação em bloco e que por isso tinham certa resistência para utilizar o Swift playground, tendo em vista que o mesmo é em linha de código.

Para a validação do jogo criado, inicialmente os participantes interagiram com a versão digital e, posteriormente, com a versão física (Figura 11), sendo submetidos aos mesmos desafios em ambas as modalidades com o objetivo de identificar semelhanças e diferenças entre os dois formatos.

⁴ Disponível em <https://drive.google.com/file/d/1eVX4EJtPeFWDUXXoysoWpdqDVeOCdUAs/view>

A experimentação envolveu um total de sete desafios e foi realizada ao longo de três encontros da oficina. Os estudantes que já conheciam a versão digital testaram a versão física e aqueles que não, começaram pela versão digital e depois experimentaram a versão física. Entre estes apenas dois estudantes possuíam experiência prévia com todas as fases do jogo digital. No primeiro encontro, apenas um estudante conseguiu concluir todos os sete desafios, fato atribuído ao tempo necessário tanto para a montagem do tabuleiro quanto para a resolução das tarefas utilizando os comandos disponíveis. No segundo encontro um estudante concluiu todos os desafios e dois concluíram cinco desafios, no último encontro um estudante concluiu seis desafios. Os estudantes tiveram autonomia para decidir sua forma de organização, optando de maneira espontânea por jogar individualmente ou em duplas. De modo geral, o desafio que apresentou maior grau de dificuldade foi o último intitulado “Conquistando um Labirinto”, devido à sua complexidade. Após a atividade, durante a qual os estudantes foram avaliados de acordo com o grau de motivação e engajamento, os mesmos responderam a um formulário estruturado, com o intuito de avaliar a experiência proporcionada pelo modelo físico do jogo. Além disso, foi conduzida observação participante em cada encontro.



Figura 11. estudantes testando o jogo físico

O formulário⁵ aplicado na avaliação foi elaborado com base no modelo de avaliação de jogos educacionais proposto por Savi (2011), o qual adota a percepção dos estudantes como principal fonte de dados para mensurar a qualidade dos jogos. Esse modelo busca equilibrar de forma sistemática os aspectos relacionados à motivação, à experiência do usuário (usabilidade) e aos resultados de aprendizagem. Os resultados dessa avaliação são detalhados na seção seguinte.

4. Resultados e Discussão

Tendo em vista que a proposta do jogo de tabuleiro “Programando com Swift” surgiu em um clube de programação com perfis diversos de estudantes, mas na faixa etária dos Anos Finais do Ensino Fundamental, a avaliação inicial também foi conduzida nesse contexto. A partir das observações conduzidas, pode-se identificar que durante as interações com o jogo físico os estudantes ficaram engajados e empolgados ao longo da resolução dos desafios. Dois participantes se destacaram pelo excelente desempenho: um deles, com experiência prévia em programação explorou abordagens alternativas para resolver os desafios, demonstrando criatividade e flexibilidade cognitiva; o outro seguiu rigorosamente o passo a passo oferecido pelo jogo concluindo as tarefas em menor tempo aliado a um elevado grau de motivação e atenção. Os três participantes restantes apresentaram desempenho mediano: um deles, mais disperso, buscava criar

5 Disponível em https://drive.google.com/file/d/1w0FcVaUFSMoa1NZ_I7SaMqwR_FBOunLd/view

novos comandos sem concluir integralmente as tarefas, enquanto dois optaram por trabalhar em dupla devido a pouca experiência em programação, mas enfrentaram dificuldades para superar alguns obstáculos propostos principalmente os desafios mais complexos que envolviam funções e operadores lógicos. Esses resultados indicam que o desempenho excelente esteve associado sobretudo ao conhecimento prévio e à dedicação dos jogadores. Vale salientar que durante toda a atividade não havia limite de tempo e os participantes eram incentivados a dialogar e trocar ideias sobre possíveis soluções sendo as sessões realizadas no contexto das aulas do clube de programação.

Em relação aos dados quantitativos, os resultados obtidos por meio do questionário aplicado ao fim de cada encontro apresentaram limitações quanto à sua representatividade, em virtude do reduzido número de respondentes, uma vez que a implementação foi feita no âmbito do clube de programação e o mesmo só contava com cinco estudantes regulares no período. Embora 100% dos estudantes tenham participado da validação, o quantitativo de participantes compromete a generalização dos dados, embora este não seja o foco deste trabalho. Devido a esse número de participantes, uma só resposta pode impactar nos índices registrados nas dimensões de motivação e experiência do usuário.

As Figuras 12 e 13 apresentam, respectivamente, os resultados das dimensões motivação e experiência do usuário. As cores das barras horizontais representam os níveis de concordância dos participantes em relação às afirmações dos questionários aplicados: azul escuro(-2), vermelho(-1), verde(0), roxo(1) e azul claro(2). Já os números que aparecem dentro das barras (porcentagens como 40,0%) indicam a proporção de respostas que aquele item recebeu em determinada categoria da escala. Como pode-se observar, em termos de motivação, de forma geral, foi identificado que ela varia de moderada a baixa: apesar de alguns casos apresentarem percentuais relevantes de respostas positivas (valores 1 e 2), há predominância de respostas neutras (0) e negativas (-1 e -2), considerando respostas dos cinco estudantes.

Um dos motivos do índice moderado de motivação é que os estudantes já tinham uma certa resistência ao uso do *Swift Playgrounds*, principalmente por se tratar de uma ferramenta baseada em código escrito, enquanto estavam habituados a trabalhar com programação em blocos. Além disso, sua experiência prévia estava fortemente vinculada à criação de jogos ao passo que a proposta do *Swift Playgrounds* na oficina consistia em resolver desafios interativos voltados ao ensino de conceitos básicos de programação. Essa diferença de foco contribuiu para a baixa adesão à plataforma, uma vez que o interesse principal do grupo era o desenvolvimento de jogos próprios. Diante desse cenário, o jogo físico foi concebido como uma alternativa para deslocar o foco do ambiente digital e proporcionar uma experiência desplugada, que permitisse aos estudantes testar seus conhecimentos de forma diferente; contudo, a resistência inicial ao jogo digital possivelmente impactou de maneira negativa a receptividade ao jogo físico; ainda assim considera-se que, caso a versão física seja desenvolvida com a coparticipação dos estudantes há potencial para resultados mais positivos visto que, durante a implementação muitos apresentaram ideias distintas para a resolução dos desafios.

Quanto à experiência do usuário, o padrão predominante é experiência do usuário mais positiva que negativa, com várias ocorrências de satisfação máxima. Ainda assim, a presença de valores negativos (-1 e -2) indica que existem pontos específicos que afetam negativamente parte dos usuários e que poderiam ser trabalhados para elevar a experiência com o uso do jogo físico.

Em suma, embora o jogo digital ofereça um ambiente digital imersivo e visualmente atraente para o ensino de programação, o jogo físico apresenta vantagens, pois não depende de dispositivos eletrônicos, conexão à internet ou conhecimentos prévios de digitação e sintaxe, tornando-se mais inclusivo para públicos com menor familiaridade com tecnologia; a dinâmica presencial do tabuleiro estimula interações sociais, permitindo que estudantes trabalhem em duplas ou grupos, negociem estratégias e aprendam de forma colaborativa; pode ser facilmente adaptado em termos de dificuldade, regras ou narrativa, incorporando sugestões dos próprios estudantes, o que fortalece o senso de pertencimento e participação ativa no processo de aprendizagem, além disso ao eliminar a necessidade de lidar com sintaxe de programação escrita, o tabuleiro permite que os estudantes concentrem-se nos conceitos lógicos fundamentais, facilitando a transição posterior para linguagens baseadas em texto.

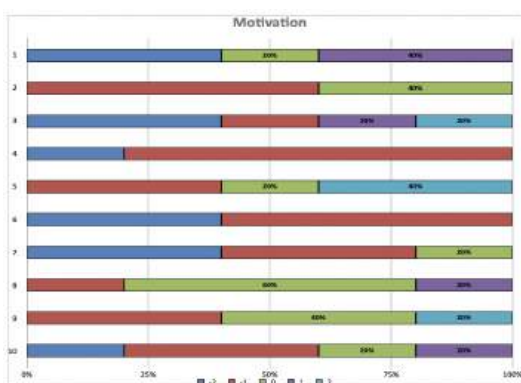


Figura 12. Gráfico de frequências de resultados da dimensão motivação

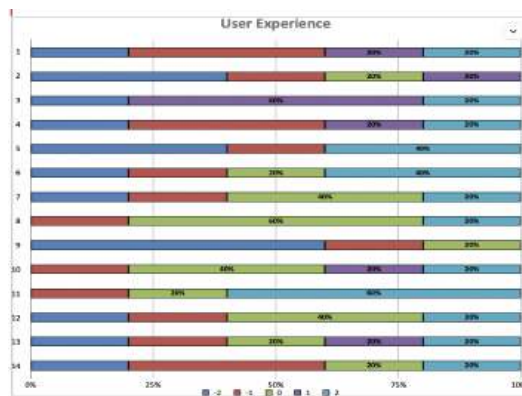


Figura 13. Gráfico de frequências de resultados da dimensão experiência do usuário

Ressalta-se, novamente, que a generalização dos resultados não é foco desta pesquisa, tendo sua base quantitativa limitada em virtude do baixo número de participantes. Dessa forma, os dados coletados devem ser interpretados como indicativos preliminares úteis para compreender tendências, levantar hipóteses e orientar melhorias futuras na proposta didática e não como evidência conclusiva sobre a eficácia do jogo físico em larga escala. A análise qualitativa descrita se soma aos resultados quantitativos detalhados, possibilitando identificar pontos fortes e de melhoria no jogo criado.

Diante dos resultados, sistematizou-se algumas implicações importantes para esta pesquisa. Em relação aos potenciais de melhorias, pode ser relevante revisar o jogo físico criado para tornar a experiência mais envolvente para os usuários. Isto pode se dar pela criação de novos desafios de códigos para serem resolvidos em duplas ou em grupos, por exemplo. Além disso, imprimir mais blocos de programação para aumentar

a margem de soluções dos desafios, ampliar as regras do jogo e usar exemplos práticos nas instruções pode potencializar a experiência e motivação dos jogadores nas interações.

A distribuição dos dados quantitativos mostra que os aspectos de motivação, experiência do usuário e aprendizagem foram avaliados de maneira semelhante. Isso sugere que, no geral, o jogo impactou os participantes de forma equilibrada sem grandes discrepâncias entre eles. Como não há valores extremos nos gráficos, pode-se sugerir que atividade do jogo desplugado precise de um tempo maior para gerar efeitos significativos, ou seja, pequenas adaptações contínuas podem aprimorar os resultados ao longo do tempo.

6. Considerações Finais

Esta pesquisa teve como objetivo, fundamentado na literatura sobre o ensino de computação e PC, desenvolver um recurso didático acessível e eficaz para o ensino de programação na educação básica. Para isso, foi idealizado, prototipado, criado e testado um jogo físico desplugado no formato de tabuleiro, que utiliza blocos com a sintaxe inspirada em uma linguagem de programação real (Swift) para promover a compreensão de conceitos fundamentais de algoritmos e lógica. O jogo foi implementado em um clube de programação com estudantes dos Anos Finais do Ensino Fundamental, servindo como ambiente experimental para coleta de dados sobre sua aplicabilidade.

A aplicação do jogo físico voltado ao ensino de programação revelou-se promissora quanto ao seu potencial educativo, ainda que tenha contado com um número limitado de participantes em sua avaliação. Apesar da amostra reduzida comprometer a possibilidade de generalizações mais amplas, algo que não é foco desta etapa da pesquisa, as interações observadas e os relatos coletados indicam que a proposta desperta interesse e contribui para o desenvolvimento de habilidades relacionadas ao PC, como a lógica, a decomposição de problemas e a sequência de comandos.

Sugere-se, para investigações futuras, a ampliação da amostra para incluir uma diversidade maior de perfis de usuários, o que permitirá uma análise mais robusta e generalizável dos aspectos de motivação, aprendizagem e experiência dos usuários. De forma complementar, uma coleta qualitativa por meio da observação da interação dos usuários com o jogo ao longo do tempo também poderá permitir uma compreensão mais profunda sobre a motivação e satisfação dos usuários na interação com o jogo, e fatores que influenciam seus resultados.

Outra linha de investigação relevante é comparar a experiência dos usuários com o jogo criado em comparação com sua versão digital. Essa comparação pode permitir identificar preferências, percepções de engajamento, usabilidade e aprendizado, revelando quais elementos de cada formato mais contribuem para a motivação e a participação dos estudantes. Também, outra possibilidade de trabalho futuro diz respeito à co-criação de novos desafios do jogo físico junto aos próprios usuários. Isto pode permitir o alinhamento das funcionalidades do jogo com as expectativas e necessidades do público-alvo, além de estimular um engajamento e aprendizagem de computação mais efetivos, e contribuir para o aprimoramento contínuo do recurso didático.

Agradecimentos

Gostaria de expressar meus sinceros agradecimentos à minha orientadora, pelo apoio, dedicação e valiosas orientações durante todo o desenvolvimento desta pesquisa. Agradeço também à banca examinadora pelas contribuições e sugestões que enriqueceram este trabalho. Meu reconhecimento à equipe do FabLab, pela colaboração e infraestrutura disponibilizadas, e à escola parceira, pela abertura e suporte essenciais para a realização do estudo. Por fim, agradeço imensamente aos estudantes participantes, cuja participação e empenho foram fundamentais para o sucesso desta pesquisa.

Referências

- Bell, T., Witten, I. H., Fellows, M., Adams, R., & McKenzie, J. (2011). Ensinando Ciência da Computação sem o uso do computador. *Computer Science Unplugged ORG*.
- Cavalcante, S., & Montalvão, S. (2024). Warfall: ensino de representação de imagens por meio de pixel art em jogo de tabuleiro. *Revista Principia - Divulgação Científica E Tecnológica Do IFPB*, 62.
<https://doi.org/10.18265/2447-9187a2025id8672>
- de Oliveira Cerqueira, T., Silva, A. P. S., & de Jesus Araujo, L. G. (2023, April). Codebo unplugged: Um jogo desplugado para o ensino de pilha. In *Simpósio Brasileiro de Educação em Computação (EDUCOMP)* (pp. 04-05). SBC.
- Co, J. W. (2025). Base Nacional Comum Curricular - Educação é a Base. Mec.gov.br.
<https://basenacionalcomum.mec.gov.br/abase/>
- da Cruz, M. E. K., Marques, S. G., & Oliveira, W. (2021). Desenvolvimento e avaliação de material didático desplugado para o ensino de computação na educação básica. *Revista Brasileira de Informática na Educação*, 29, 160-187.
- Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in entertainment (CIE)*, 1(1), 20-20.
- von Wangenheim, C. G., Araújo, G., de Medeiros, S., Missfeldt Filho, R., Petri, G., da Cruz Pinheiro, F., ... & Hauck, J. C. (2019). Desenvolvimento e Avaliação de um Jogo de Tabuleiro para Ensinar o Conceito de Algoritmos na Educação Básica. *Revista Brasileira de Informática na Educação*, 27(03), 310-335.
- Guarda, G. F., de Moraes Stella, A. C., de Holanda Oliveira, G., Trigo, G. M., Cruz, L. G., de Brito, M. E. M., ... & Silveira, I. F. (2023, November). Trabalhando habilidades da BNCC Computação: Jogo não digital com foco no Desenvolvimento do Pensamento Computacional Desplugado. In *Workshop de Informática na Escola (WIE)* (pp. 810-820). SBC.
- Martins, R., Reis, R., & Marques, AB (2016, novembro). Inserção da programação no ensino fundamental Uma análise do jogo Labirinto Clássico da Code. org através de um modelo de avaliação de jogos educacionais. In *Anais do Workshop de Informática na Escola* (Vol. 22, No. 1, pp. 121-130).

Monteiro Rodrigues, A. K., Mundim Silva, A. P., & Guimarães Carneiro, M. (2021). Ensino de Pensamento Computacional para alunos do ensino básico usando Computação Desplugada e Scratch. Em *Extensao*, 20(2).

Savi, R. (2011). Avaliação de jogos voltados para a disseminação do conhecimento.

Silva, M. R., Nahirne, A. P., de Oliveira, H. D. F., & Boscarioli, C. (2024). Um jogo de tabuleiro para integrar Matemática e Pensamento Computacional no Ensino Fundamental. *TANGRAM-Revista de Educação Matemática*, 7(3), 131-150.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

([S.d.]). Gov.br. Recuperado 12 de agosto de 2025, de <https://www.gov.br/mec/pt-br/escolas-conectadas/BNCCCComputaoCompletoDiagramado.pdf>