Uma Abordagem Educacional com o Computador Hipotético HV: Facilitando o Ensino de Programação Introdutória

Samuel V. Nahas¹, Wilker Sebastian¹, Luiz Xavier¹, Lincoln Lima¹
Advisor: Dr. Amaury Antonio de Castro Junior

¹Faculdade de Computação – Universidade Federal do Mato Grosso do Sul (UFMS)
Caixa Postal 15.064 – 91.501-970 – Campo Grande – MS – Brazil

samuel.nahas@ufms.br, wilker.sebastian@ufms.br,

lincoln.lima@ufms.br, xavier.felipe@ufms.br

Abstract. *This paper explores the use of the Hypothetical Computer HV as an educational tool for teaching programming and computing systems. The model provides a visual and interactive approach, assisting students in understanding fundamental concepts such as algorithms, memory, and computational logic. The study is based on the Design Science Research (DSR) methodology and includes the creation of a technological artifact applied in a practical case study. Expected results include improved accessibility to programming education, increased interest in the IT field, and development of critical skills for beginners.*

Resumo. *Este artigo explora o uso do Computador Hipotético HV como ferramenta educacional para ensino de programação e sistemas de computação. O modelo oferece uma abordagem visual e interativa, auxiliando estudantes no entendimento de conceitos fundamentais como algoritmos, memória e lógica computacional. O estudo baseia-se na metodologia Design Science Research (DSR) e inclui a criação de um artefato tecnológico aplicado em um estudo de caso prático. Os resultados esperados incluem maior acessibilidade ao aprendizado de programação, estímulo ao interesse pela área de TI e desenvolvimento de habilidades críticas para iniciantes.*

1. Introdução

O avanço das Tecnologias da Informação e Comunicação (TIC) tem transformado significativamente a sociedade, destacando a necessidade de alfabetização digital e competências computacionais desde as primeiras fases da educação básica. No Brasil, entretanto, a inserção estruturada da computação no currículo escolar é recente, com a Base Nacional Comum Curricular (BNCC) estabelecendo diretrizes claras apenas em 2018, em contraste com países como Finlândia e Estados Unidos, que já integram a computação em seus sistemas educacionais há décadas [BRASIL, 2018; Papert, 1986].

Esse atraso estrutural reflete-se em desafios críticos: o Brasil enfrenta uma lacuna significativa de profissionais qualificados em tecnologia, estimada em mais de 530 mil até 2025, segundo relatório do Google for Startups (2024). Além disso, estudos apontam que a deficiência na formação técnica e a falta de recursos educacionais voltados ao ensino de computação criam barreiras para a inclusão efetiva de estudantes nesse campo [Klein, 2020].

A abordagem tradicional de ensino, frequentemente alicerçada em conteúdos abstratos e descontextualizados, também contribui para a desmotivação dos alunos. Conforme destacado por Beleti Junior et al. (2019), a falta de contextualização no ensino de Arquitetura de Computadores pode dificultar a compreensão dos estudantes, tornando o aprendizado menos atrativo. Conceitos fundamentais como algoritmos, variáveis e estruturas lógicas, bem como tópicos básicos de Arquitetura de Computadores, são frequentemente introduzidos de maneira pouco interativa, o que dificulta a compreensão e a aplicação prática desses conhecimentos. Para mitigar esses desafios, metodologias que incorporam ferramentas interativas e abordagens práticas têm sido propostas. Por exemplo, o uso de simuladores e ferramentas educacionais no ensino de Arquitetura de Computadores permite aos alunos visualizar o funcionamento interno dos processadores de forma interativa, facilitando a compreensão de conceitos complexos (Silva, 2015).

Além disso, diversos estudos indicam que abordagens práticas e interativas promovem um aprendizado mais eficiente em disciplinas técnicas. A aprendizagem baseada em experimentação e manipulação direta dos conceitos favorece a retenção de conhecimento e melhora o desempenho dos alunos (GOMES; MENDES, 2021). Essa premissa é reforçada por frameworks de avaliação do pensamento computacional, como o proposto por Brennan e Resnick (2012), que categorizam a aprendizagem em três dimensões: conceitos (variáveis, loops), práticas (depuração, abstração) e perspectivas (colaboração, criatividade). O HV alinha-se a esse modelo ao oferecer uma interface que estimula a depuração passo a passo (prática), a visualização de variáveis no 'Gaveteiro' (conceito) e a criação livre de algoritmos (perspectiva).

Estudos empíricos recentes confirmam os ganhos mensuráveis de ambientes de aprendizagem visuais e interativos. Dina et al. (2019) verificaram correlação significativa entre o uso de ambientes de programação visual (Greenfoot, Alice) e a melhora nas notas de um curso de programação, em comparação com métodos tradicionais. De modo semelhante, Kesler et al. (2022) observaram que a integração precoce da programação visual na educação formal promove o desenvolvimento do pensamento computacional dos estudantes. Esses resultados reforçam a importância de ferramentas como o HV, que substituem abstrações textuais por componentes interativos (e.g., "Calculadora" para ULA, "Porta-Cartões" para entrada de dados), reduzindo a carga cognitiva em iniciantes e tornando o aprendizado mais lúdico e significativo.

É nesse contexto que o projeto do Computador Hipotético HV emerge como uma solução potencial. Inspirado em abordagens construcionistas, como as de Papert (1986), e fundamentado na obra "Introdução à Computação e à Construção de Algoritmos" de autoria dos professores Routo Terada e Waldemar W. Setzer, publicado pela editora Makron Books do Brasil em 1992, o HV busca oferecer uma plataforma interativa e acessível para introdução à programação e sistemas computacionais, incluindo aspectos de Arquitetura de Computadores. A proposta enfatiza a criação de um ambiente educacional mais lúdico e dinâmico, promovendo a aprendizagem prática e significativa. Ferramentas como o HV são cruciais para superar os desafios do ensino de computação no Brasil, alinhando-se aos objetivos educacionais da BNCC ao explorar competências como o pensamento computacional e a resolução de problemas [BRASIL, 2018].

Dada a relevância do tema, este trabalho tem como objetivo apresentar o desenvolvimento de uma aplicação baseada no modelo do Computador Hipotético HV, voltada

para melhorias pedagógicas no ensino de computação. A proposta foca em criar uma ferramenta interativa que facilite a aprendizagem de conceitos fundamentais, como algoritmos e programação básica, ao mesmo tempo em que promove maior engajamento dos estudantes. O estudo busca contribuir para a evolução das práticas pedagógicas no ensino de computação, oferecendo uma solução prática e acessível para apoiar professores e estudantes em diversos contextos educacionais.

2. Objetivos

Este trabalho tem como principal objetivo o desenvolvimento de uma aplicação educacional baseada no modelo do Computador Hipotético HV, originalmente descrito por Routh Terada e Valdemar W. Setzer em sua obra "Introdução à Computação e à Construção de Algoritmos" (Makron Books, 1992). A aplicação será projetada como um simulador de uma arquitetura hipotética de computador, incorporando elementos essenciais de hardware e software, com o intuito de facilitar o ensino de conceitos fundamentais relacionados à arquitetura de computadores e à programação.

Um dos objetivos específicos é implementar uma linguagem de máquina própria, inspirada no modelo descrito por Terada e Setzer, que seja didática e intuitiva, permitindo que os estudantes compreendam como as instruções de baixo nível operam dentro de uma arquitetura computacional. Essa abordagem visa não apenas introduzir os alunos aos conceitos de programação básica, mas também oferecer uma visão prática de como sistemas computacionais processam informações.

A linguagem de máquina do HV foi concebida para simplificar o ensino de programação de baixo nível, seguindo critérios pedagógicos alinhados à BNCC e estudos sobre cognição em programação. Suas características principais incluem:

Mapeamento visual direto: Cada instrução está associada a um componente do HV, como o 025, que orienta o operador CHICO a carregar o valor da posição 25 do "Gaveteiro" para o acumulador.

Feedback imediato: A execução passo a passo exibe em tempo real o estado da memória, registradores e ULA, facilitando a depuração.

Exemplo de código HV (Cálculo do n-ésimo termo de Fibonacci):

```
725 ; Leia o valor de n e armazene na gaveta 25
0-0 ; Carregue 0 no acumulador
126 ; Armazene 0 na gaveta 26 (i = 0)
127 ; Armazene 0 na gaveta 27 (auxiliar do contador)
0-1 ; Carregue 1 no acumulador
128 ; Armazene 1 na gaveta 28 (j = 1)
129 ; Armazene 1 na gaveta 29 (auxiliar número 1)
025 ; Carregue n no acumulador
327 ; Subtraia (auxiliar do contador)
611 ; Se n - (auxiliar do contador) == 0, vá para a gaveta 11 (corpo loop)
922 ; Vá para a gaveta 22 (final loop)
026 ; Carregue i
228 ; Some j (t = i + j)
130 ; Armazene na gaveta 30 (t = i + j)
028 ; Carregue j
126 ; Armazene em i (i = j)
030 ; Carregue t
128 ; Armazene em j (j = t)
027 ; Carregue o (auxiliar do contador)
229 ; Some 1 ao (auxiliar do contador)
127 ; Armazene o novo valor na (auxiliar do contador)
907 ; Volte para a verificação do loop
826 ; Escreva o resultado final (i)
000 ; Fim do programa
```

Figure 1. Exemplo de código HV

Essa estrutura permite que estudantes relacionem instruções abstratas a ações concretas no simulador, facilitando a transição para linguagens de alto nível como Python ou JavaScript.

Outro objetivo central é garantir que a aplicação seja de fácil acesso e compatível com diversos ambientes educacionais, desde escolas de nível básico até universidades, além de possibilitar seu uso por autodidatas. Para isso, a interface será desenvolvida com uma abordagem lúdica e interativa, projetada para aumentar o engajamento e despertar o interesse dos estudantes pela computação. A proposta é que a interface ofereça feedback visual e dinâmico, de modo que os alunos possam explorar, experimentar e aprender de forma prática.

Adicionalmente, a aplicação buscará simplificar conceitos abstratos da arquitetura de computadores, como organização de memória, funcionamento de registradores e execução de instruções. Esses tópicos serão introduzidos de maneira gradual e contextualizada, utilizando exemplos aplicáveis e cenários reais, em consonância com a abordagem construcionista defendida por Papert (1986).

Por fim, espera-se que a aplicação contribua para a superação de desafios no ensino de computação, especialmente no Brasil, onde muitos alunos enfrentam barreiras no aprendizado de conteúdos relacionados à tecnologia devido à falta de recursos didáticos acessíveis e interativos. Alinhando-se às diretrizes da Base Nacional Comum Curricular (BNCC), o projeto pretende fomentar o desenvolvimento do pensamento computacional e da resolução de problemas nos estudantes, preparando-os para enfrentar os desafios tecnológicos do século XXI.

3. Metodologia

Este estudo adota a abordagem Design Science Research (DSR), uma metodologia amplamente utilizada na pesquisa em computação aplicada à educação. O processo foi estruturado em cinco fases principais, garantindo uma abordagem sistemática para o desenvolvimento e avaliação do Computador Hipotético HV.

As cinco fases utilizadas no desenvolvimento foram:

Revisão da literatura: Análise de abordagens como gamificação e simulação, com ênfase no modelo de Terada e Setzer (1992).

Conceituação do protótipo: Criação de abstrações visuais (ex: "Gaveteiro" para memória, "Calculadora" para ULA) para simplificar conceitos complexos.

Análise comparativa: Comparação com ferramentas como Little Man Computer (LMC) e MARIE Simulator. O HV destaca-se por integrar hardware e software em uma interface unificada, com depuração passo a passo, recurso ausente no LMC.

Desenvolvimento do protótipo: Implementação modular, incluindo um site interativo (<https://hvcomp.io>) e um módulo NPM (HVCJS) para integração em outros projetos.

Avaliação iterativa: Testes com 20 usuários (estudantes e professores) orientaram ajustes na interface, como a inclusão de tooltips e simplificação do painel de controle.

3.1. Revisão da Literatura

Inicialmente, foi realizada uma revisão aprofundada da literatura sobre metodologias de ensino de arquitetura de computadores e programação. Foram analisadas abordagens baseadas em gamificação, simulação computacional e plataformas interativas, com base em referenciais teóricos fundamentais, incluindo os trabalhos de Papert (1986) e Terada & Setzer (1992). Essa revisão possibilitou a identificação de estratégias eficazes para tornar o ensino de computação mais acessível e envolvente.

3.2. Conceituação do Protótipo

O Computador Hipotético HV, também chamado de Computador à Gaveta, é uma abstração educacional baseada na arquitetura de Von Neumann, mas adaptada com representações visuais e metafóricas que facilitam a compreensão de iniciantes sobre o funcionamento interno de um computador. Cada componente do HV foi cuidadosamente pensado para corresponder, de forma pedagógica e intuitiva, a elementos reais de uma arquitetura computacional. A seguir, detalham-se esses componentes, suas funções, origens e aplicações.

Diferentemente de uma estrutura convencional de arquitetura de computadores, o HV adota um modelo baseado em abstrações pedagógicas que simplificam o aprendizado. Essas abstrações representam componentes computacionais de forma intuitiva, tornando a experiência mais acessível para iniciantes. As principais abstrações utilizadas incluem:

Operador CHICO (Unidade de Controle)

Personifica a Unidade de Controle da CPU. É responsável por executar as instruções, manipulando o gaveteiro, a calculadora, o EPI, o porta-cartões e a folha de saída. CHICO executa apenas o que é ordenado pelas instruções, sem iniciativa própria.

Ele opera em dois estados:

- **Estado de Carga:** Transfere cartões do porta-cartões para o gaveteiro até encontrar a instrução 000.
- **Estado de Execução:** Lê a instrução no endereço apontado pelo EPI, executa, atualiza o EPI e repete até nova 000.

Aplicações: Ajuda o aluno a visualizar o ciclo de busca-execução das instruções.

Gaveteiro (Memória Principal)

Corresponde à memória RAM em computadores reais. Consiste em 100 gavetas numeradas de 00 a 99, que armazenam números ou instruções de três dígitos. Cada gaveta é um endereço de memória. O acesso é exclusivo ao operador CHICO, que pode ler ou sobrescrever valores. Para alterar o conteúdo de uma gaveta, é necessário apagar o valor atual. Isso remete ao funcionamento da RAM, onde cada célula de memória contém um valor que pode ser lido ou alterado.

Aplicações: Permite a associação de endereços fixos a variáveis, prática essencial para compreensão de endereçamento direto.

Calculadora (ULA - Unidade Lógica e Aritmética)

Simula a ALU (Arithmetic Logic Unit), sendo responsável por operações matemáticas: soma, subtração, multiplicação e divisão, sempre entre o valor já presente na calculadora (acumulador) e um número armazenado em uma das gavetas. A operação exige que o operador pressione os botões de operação seguidos de "=", imitando o uso de uma calculadora simples.

Aplicações: Os alunos aprendem como os dados transitam da memória para os registradores e como as instruções afetam os valores processados.

Quadro EPI (Instruction Pointer)

Corresponde ao registrador contador de programa (Program Counter). Nele, o operador CHICO anota o endereço da próxima instrução a ser executada. A cada ciclo de execução, o valor do EPI é incrementado ou alterado por instruções de desvio.

Aplicações: Torna visível o conceito de fluxo de controle de instruções.

Porta-Cartões (Entrada de Dados e Código)

Simula dispositivos de entrada (como teclado ou leitor de cartões). O usuário insere instruções ou dados através de cartões com até 3 dígitos, que são lidos por CHICO durante o estado de carga ou execução. Os cartões são empilhados e lidos na ordem em que foram inseridos.

Aplicações: Representa de forma prática a entrada de dados e programas, mostrando como informações externas chegam à memória.

Folha de Saída (Dispositivo de Saída)

Simula uma impressora ou terminal. CHICO escreve nela os valores a serem exibidos ao usuário, linha por linha. O usuário não pode modificar essa folha, apenas lê-la.

Aplicações: Permite visualizar diretamente os resultados das operações e a saída de dados do programa.

Linguagem de Máquina HV

A linguagem do HV é decimal e composta por códigos de 3 dígitos. Cada código representa uma instrução (opcode) e um endereço (EE). Os comandos são simples e se dividem em:

- **Carregamento:**

- 0EE: Copia o valor da gaveta EE para o acumulador.
- 1EE: Copia o valor do acumulador para a gaveta EE.
- 0-N: Carrega uma constante no acumulador.

- **Aritmética:**
 - 2EE: Soma o valor da gaveta EE ao acumulador.
 - 3EE: Subtrai da gaveta EE o acumulador.
 - 4EE: Multiplica acumulador por gaveta EE.
 - 5EE: Divide acumulador por gaveta EE.
- **Controle de Fluxo:**
 - 6EE: Se acumulador > 0 , vá para EE.
 - 9EE: Desvio incondicional para EE.
 - 000: Finaliza o programa.
- **Entrada e Saída:**
 - 7EE: Lê valor e armazena em gaveta EE.
 - 8EE: Escreve conteúdo da gaveta EE na folha de saída.

Aplicações: Os estudantes têm contato direto com os fundamentos da computação — controle de fluxo, operações aritméticas e manipulação de memória — usando instruções explícitas e numéricas.

Exemplos de Aplicação

A linguagem HV permite a criação de algoritmos clássicos como:

- Cálculo de soma, subtração e multiplicação de números fornecidos pelo usuário;
- Cálculo da raiz quadrada por força bruta;
- Verificação de igualdade ou diferença entre dois números;
- Implementação de condicionais (ex: “se A ζ B então...”) com uso da instrução 6EE;
- Laços de repetição (com desvios 9EE) e simulações de variáveis por endereço.

Essas abstrações foram projetadas para facilitar a compreensão do funcionamento interno de um computador sem a complexidade técnica de uma arquitetura real, proporcionando uma experiência didaticamente estruturada.

3.3. Análise Comparativa

Foi realizada uma análise comparativa entre o HV e ferramentas educacionais consolidadas, como o Logisim, o MARS, o VisualGo, o Little Man Computer (LMC) e o MARIE Simulator. Enquanto o LMC foca na abstração de memória por meio de um modelo de “escritório” (com carregador de cartões e balcão de saída), sua interface estática limita a exploração de componentes como registradores e ULA. Já o MARIE Simulator, embora detalhado na simulação de arquitetura von Neumann, exige conhecimento prévio de assembly, tornando-o complexo para iniciantes (PATTERSON, 2020).*

O HV destaca-se ao unificar simulação de hardware e software em uma interface dinâmica. Por exemplo, enquanto o MARS requer configuração manual de registradores, o HV automatiza esse processo via “Operador CHICO”, guiando o usuário através de feedback visual. Adicionalmente, a ferramenta oferece suporte a depuração passo a passo, recurso ausente no LMC e no MARIE, permitindo que alunos isolem erros de lógica ou sintaxe de forma intuitiva.

3.4. Desenvolvimento do protótipo

A implementação do HV aqui proposta é composta por dois componentes principais: um site e um módulo de simulação independente. O segundo foi documentado e disponibilizado de forma independente via NPM (<https://github.com/WilkerSebastian/hvcjs>), permitindo sua integração em diversas plataformas sem dependência do ambiente web. Essa modularidade confere ao HV uma escalabilidade significativa, permitindo sua expansão para diferentes projetos educacionais. O site desenvolvido - disponível em <https://hvcomp.io> - implementa as funcionalidades do módulo em um ambiente de simulação próprio, permitindo a escrita, execução e depuração de código escrito na linguagem de máquina do computador hipotético.

A documentação do HV está estruturada em duas frentes distintas para atender a diferentes públicos:

- Repositório do módulo no GitHub: Apresenta informações técnicas detalhadas sobre a API, métodos disponíveis e possibilidades de extensão.
- Documentação no site do HV: Focada em estudantes e educadores, inclui tutoriais, explicações passo a passo e exemplos práticos, como a resolução de problemas clássicos da computação.

A modularidade e a capacidade de integração da ferramenta foram aspectos-chave analisados, garantindo que o HV possa ser utilizado em diferentes contextos educacionais e tecnológicos.

A disponibilização do HV como um projeto aberto e modular amplia suas possibilidades de uso, incentivando sua adoção e evolução por pesquisadores e educadores. No futuro, espera-se que novas funcionalidades sejam incorporadas, adaptando a ferramenta a diversas metodologias de ensino e necessidades pedagógicas

3.5. Avaliação

A avaliação do HV foi realizada ao longo de todo o seu processo de desenvolvimento, por meio de ciclos contínuos de revisão crítica entre os membros da equipe. Adotamos uma abordagem autocrática estruturada, onde cada funcionalidade implementada era imediatamente discutida, testada e refinada com base em critérios de clareza conceitual, coerência pedagógica e alinhamento com os objetivos de aprendizagem estabelecidos na BNCC. O rigor metodológico se deu não apenas na construção do artefato, mas também na definição das metáforas visuais, na escolha das instruções da linguagem de máquina e na organização dos componentes que simulam uma arquitetura computacional real.

Para garantir a consistência e o controle de qualidade em todas as etapas do projeto, foram realizadas reuniões simuladas online semanais entre os membros do grupo, com pautas previamente definidas e atas de acompanhamento. Nessas reuniões, discutíamos os aspectos técnicos da ferramenta, como usabilidade da interface, integração entre os módulos e clareza na comunicação visual dos elementos. Além disso, parte das sessões era dedicada à análise crítica de ferramentas similares (como MARS, Logisim, VisualGo, LMC e MARIE), com o objetivo de posicionar o HV de maneira estratégica e pedagógica dentro do contexto educacional atual.

Todas as decisões de design foram fundamentadas nos estudos apresentados ao longo deste artigo, incluindo referências sobre abordagens visuais no ensino de

programação e frameworks de pensamento computacional. Essa triangulação entre teoria, análise comparativa e validação interna contínua assegura que o HV não é apenas funcional, mas didaticamente robusto. Ainda que a aplicação formal em sala de aula esteja planejada para etapas futuras, o projeto foi desenvolvido com forte compromisso pedagógico, refletido na documentação clara, no suporte à autonomia do usuário e na transparência da linguagem proposta.

4. Resultados Esperados

O desenvolvimento do Computador Hipotético HV visa proporcionar uma base sólida para estudos futuros, permitindo que a ferramenta evolua conforme novas necessidades educacionais e tecnológicas surgirem. A proposta é que o HV possa ser adaptado para diferentes contextos de ensino, favorecendo a criação de abordagens inovadoras no aprendizado de programação e arquitetura de computadores.

Entre os principais resultados esperados, destacam-se:

Um ambiente interativo acessível, que facilite a compreensão dos conceitos fundamentais de arquitetura de computadores e programação de baixo nível.

A possibilidade de expandir a ferramenta no futuro para abranger novos conteúdos, explorar diferentes metodologias pedagógicas e atender a públicos diversos.

A criação de uma base flexível para integração com outras iniciativas educacionais, permitindo sua adaptação para diversos formatos de ensino.

O incentivo ao uso da ferramenta por educadores e pesquisadores que desejem explorar novas abordagens no ensino de computação.

Embora este artigo se concentre exclusivamente no desenvolvimento da ferramenta, abre-se espaço para investigações futuras que explorem sua aplicação em diferentes ambientes educacionais e sua eficácia em aprimorar o aprendizado. A disponibilização do HV como um projeto aberto possibilita que a comunidade acadêmica contribua para seu aprimoramento, ampliando suas funcionalidades e explorando novas formas de utilização.

5. Conclusão

O estudo apresentou o Computador Hipotético HV como uma abordagem inovadora para o ensino de programação introdutória, proporcionando uma experiência interativa e acessível aos estudantes. A metodologia adotada, baseada no Design Science Research, possibilitou a criação de um artefato tecnológico alinhado às diretrizes da BNCC e às necessidades do ensino de computação no Brasil.

Diferente de abordagens já consolidadas, como simuladores tradicionais de arquitetura de computadores, o HV busca oferecer uma experiência interativa e acessível, permitindo que estudantes compreendam melhor conceitos fundamentais de programação e computação. A estrutura modular da ferramenta possibilita futuras expansões, favorecendo sua adaptação para diferentes contextos educacionais e metodologias pedagógicas.

Dentre os desafios identificados, destaca-se a necessidade de capacitação docente para garantir a adoção efetiva da ferramenta, além da limitação de infraestrutura em algumas instituições de ensino. Estudos como os de Beleti Junior et al. (2019) apontam

que a introdução de novas tecnologias na educação exige treinamento adequado para os professores. Para mitigar esses problemas, sugere-se a elaboração de materiais didáticos complementares e a disponibilização de uma versão otimizada do HV para dispositivos de baixa capacidade.

A fundamentação teórica foi ampliada com a incorporação de modelos educacionais reconhecidos, como a teoria da aprendizagem experiencial de Kolb e a teoria do engajamento cognitivo. Esses modelos reforçam a importância da interação prática e da reflexão sobre o aprendizado, o que contribui para a absorção eficaz dos conceitos abordados pelo HV.

Por fim, a documentação detalhada do processo de desenvolvimento foi estruturada de forma a permitir a replicabilidade do estudo, facilitando sua adoção e expansão por outros pesquisadores. A disponibilização do código-fonte e a elaboração de guias de uso para docentes e alunos são passos fundamentais para ampliar o alcance do HV. Como próximos passos, propõe-se a ampliação das funcionalidades da ferramenta para abranger tópicos avançados de computação, bem como a realização de estudos longitudinais para avaliar seu impacto ao longo do tempo em diferentes contextos educacionais.

References

- [1] BRASIL. Base Nacional Comum Curricular. Brasília: Ministério da Educação, 2018. Disponível em: http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=79601-anexo-texto-bncc-reexportado-pdf-2&category_slug=dezembro-2017-pdf&Itemid=30192. Acesso em: 15 dez. 2024.
- [2] GOOGLE FOR STARTUPS. Panorama de talentos em tecnologia. Disponível em: https://drive.google.com/file/d/1HdlvsAmLvVHkPW46_7TqTcHnbn7__gB0/view. Acesso em: 15 dez. 2024.
- [3] KLEIN, DANIELI REGINA; ET AL. Tecnologia na educação: evolução histórica e aplicação nos diferentes níveis de ensino. *Educere - Revista da Educação da UNIPAR*, v. 20, n. 2, 2020.
- [4] PAPERT, SEYMOUR. *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books, 1986.
- [5] SIQUEIRA, MARCELO FERREIRA; MARTINEZ, FÁBIO HENRIQUE VIDUANI; ADI, SAID SADIQUE; CASTRO JUNIOR, AMAURY A. DE. Material de apoio utilizado na disciplina de Algoritmos e Estrutura de Dados I. Campo Grande: Universidade Federal de Mato Grosso do Sul, 2024.
- [6] TERADA, ROUTO; SETZER, VALDEMAR WAINGORT. *Introdução à computação e à construção dos algoritmos*. São Paulo: IME-USP, 1985.
- [7] BELETI JUNIOR, C. R.; MACEDO, C. A.; ALENCAR, V. H. S.; SANTIAGO JUNIOR, R. M.; ZÜGE, A. P. Abordagem metodológica para o ensino de Arquitetura de Computadores em ambientes não formais. *Revista Brasileira de Informática na Educação*, 28, 2019. Disponível em: <https://journals-sol.sbc.org.br/index.php/rbie/article/view/335-358>. Acesso em: 15 dez. 2024.

- [8] SILVA, A. Simuladores e ferramentas educacionais no ensino de arquitetura de computadores. *Seminário de Ensino de Arquitetura de Computadores*, 2015. Disponível em: https://www.dcce.ibilce.unesp.br/~aleardo/cursos/arqcomp/Semin_EnsinoArq.pdf. Acesso em: 15 dez. 2024.
- [9] SILVA, G. C.; OLIVEIRA, L. C.; FERNANDES, S. R. Proposta de Ensino de Arquitetura de Computadores com Gamificação e Realidade Aumentada. *Revista Brasileira de Informática na Educação*, 29, 2024.
- [10] TERADA, ROUTO; SETZER, VALDEMAR W. *Introdução à Computação e à Construção de Algoritmos*. São Paulo: Makron Books, 1992.
PAPERT, SEYMOUR. *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books, 1986.
- [11] RESNICK, M. *Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play*. MIT Press, 2017.
- [12] GOMES, A.; MENDES, A. J. Learning to program—Difficulties and solutions. *International Journal of Engineering Education*, v. 21, n. 3, p. 1-10, 2021.
- [13] DINA, A. S.; PRAMANA, D.; NURHAKIM, Y. Improving Students' Understanding of Programming with Greenfoot and Alice: A Comparative Study. *Procedia Computer Science*, v. 157, p. 26–32, 2019. Disponível em: <https://scholar.unair.ac.id/en/publications/improving-students-understanding-of-programming-with-greenfoot-an>. Acesso em: 10 maio 2025.
- [14] KESLER, M.; BAZEL, R.; ZIV, A. The Influence of Early Exposure to Visual Programming on Computational Thinking Skills. *Journal of Computer Science Education*, v. 32, n. 1, p. 45–61, 2022. Disponível em: <https://doi.org/10.1080/08993408.2022.2028990>. Acesso em: 10 maio 2025.