



**UNIVERSIDADE FEDERAL DO OESTE DO PARÁ
INSTITUTO DE CIÊNCIAS DA EDUCAÇÃO
LICENCIATURA EM INFORMÁTICA EDUCACIONAL**

THAYANE PATRICIA ROCHA MORAES

**COMPUTAÇÃO DESPLUGADA COM JOGOS: DESENVOLVENDO O
PENSAMENTO COMPUTACIONAL NO COLÉGIO CÍVICO-MILITAR JOSÉ DE
ALENCAR**

**SANTARÉM-PA
2024**

THAYANE PATRICIA ROCHA MORAES

**COMPUTAÇÃO DESPLUGADA COM JOGOS: DESENVOLVENDO O
PENSAMENTO COMPUTACIONAL NO COLÉGIO CÍVICO-MILITAR JOSÉ DE
ALENCAR**

**Trabalho de Conclusão de Curso apresentado ao
Programa de Ciências Exatas para obtenção de
grau de Licenciatura em Informática
Educativa: Universidade Federal do Oeste do
Pará, Instituto de Ciências da Educação.
Orientador: Enoque Calvino Melo Alves**

**SANTARÉM-PA
2024**

Dados Internacionais de Catalogação-na-Publicação (CIP)
Sistema Integrado de Bibliotecas – SIBI/Ufopa

M828c Moraes, Thayane Patricia Rocha
 Computação desplugada com jogos: desenvolvendo o pensamento computacional no Colégio Cívico Militar José de Alencar./ Thayane Patricia Rocha Moraes. – Santarém, 2024.
 25 p.: il.
 Inclui bibliografias.

 Orientador: Enoque Calvino Melo Alves.
 Trabalho de Conclusão de Curso (Graduação) – Universidade Federal do Oeste do Pará, Instituto de Ciências da Educação, Programa de Ciências Exatas, Licenciatura em Informática Educacional.

 1. Pensamento computacional. 2. Desplugados. 3. Jogos educativos. I. Alves, Enoque Calvino Melo, *orient.* II. Título.

CDD: 23 ed. 370.115


THAYANE PATRICIA ROCHA MORAES

COMPUTAÇÃO DESPLUGADA COM JOGOS: Desenvolvendo o pensamento computacional no colégio cívico-militar José de Alencar


Trabalho de Conclusão de Curso de Licenciatura em Informática Educacional para obter a aprovação na disciplina Seminário de Apresentação de TCC pela Universidade Federal do Oeste do Pará - UFOPA, Instituto de Ciências da Educação – ICED.

Conceito: 10


Data de Aprovação 25/10/2024

Documento assinado digitalmente
 ENOQUE CALVINO MELO ALVES
Data: 29/10/2024 20:43:53-0300
Verifique em <https://validar.iti.gov.br>

Prof. MSc. Enoque Calvino Melo Alves- Orientador Universidade Federal do Oeste do Pará - Brasil

Documento assinado digitalmente
 CLAUDIR OLIVEIRA
Data: 31/10/2024 10:04:53-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Claudir Oliveira
Universidade Federal do Oeste do Pará - Brasil

Documento assinado digitalmente
 SOCORRO VANIA LOURENCO ALVES
Data: 30/10/2024 22:35:52-0300
Verifique em <https://validar.iti.gov.br>

Prof^a. MSc. Socorro Vânia Lourenço Alves
Universidade Federal do Oeste do Pará - Brasil

RESUMO

Este trabalho explora o impacto da introdução do Pensamento Computacional (PC) no ensino fundamental e médio, utilizando métodos lúdicos e desplugados para estimular habilidades de resolução de problemas e cognitivas em estudantes. Inspirado nos conceitos de Jeannette Wing e outros pesquisadores, o estudo estrutura o Pensamento Computacional em quatro pilares principais: decomposição, abstração, reconhecimento de padrões e criação de algoritmos. A pesquisa incluiu uma revisão bibliográfica e a aplicação prática de jogos educativos, como a Cifra de César, Pixel Arte e Labirinto, para trabalhar esses pilares. As atividades foram implementadas em duas turmas de um colégio cívico-militar, com avaliações iniciais e finais para medir o progresso dos alunos. Os resultados indicam uma melhoria significativa no entendimento e interesse dos alunos em programação em blocos, com destaque para a efetividade dos jogos na introdução de conceitos de programação e na promoção de habilidades de pensamento crítico.

Palavras-Chave: Pensamento Computacional. Desplugados. Jogos Educativos.

Abstract

This study investigates the impact of introducing Computational Thinking (CT) in primary and secondary education through unplugged, playful methods to stimulate problem-solving and cognitive skills in students. Based on the concepts of Jeannette Wing and other researchers, Computational Thinking is structured into four main pillars: decomposition, abstraction, pattern recognition, and algorithm creation. The research involved a literature review and the practical application of educational games, such as Caesar Cipher, Pixel Art, and Maze, to work on these pillars. The activities were implemented in two classes at a civic-military school, with initial and final assessments to measure student progress. Results indicate a significant improvement in students' understanding and interest in block programming, highlighting the effectiveness of games in introducing programming concepts and promoting critical thinking skills.

Keywords: Computational Thinking. Unplugged. Educational Games.

ÍNDICE DE FIGURAS

Figura 1 - Similaridade entre raças de cachorros (reconhecimento de Padrões).....	9
Figura 2 - Exemplo da Cifra de César.....	14
Figura 3 - Exemplo da mensagem criptografada passada para os alunos	15
Figura 4 - Exemplo da matriz utilizada pelos alunos	16
Figura 5 - Modelo do labirinto apresentado aos alunos.....	17
Figura 6 - Gráfico representando as porcentagem das respostas dos alunos	18
Figura 7 - Gráfico representando as porcentagem das respostas dos alunos	20

SUMÁRIO

1	INTRODUÇÃO.....	5
2	REVISÃO BIBLIOGRÁFICA.....	6
2.1	Decomposição.....	7
2.2	Abstração.....	8
2.3	Reconhecimento de padrões.....	9
2.4	Criação de algoritmos.....	10
2.5	Computação desplugada.....	10
3	METODOLOGIA.....	11
4	JOGOS APLICADOS.....	13
4.1	Cifra de César.....	13
4.1.1	Materiais.....	13
4.1.2	Objetivo.....	13
4.1.3	Instruções.....	13
4.2	Pixel arte.....	15
4.2.1	Materiais.....	15
4.2.2	Objetivo.....	15
4.3	Labirinto.....	16
4.3.1	Materiais.....	16
4.3.2	Objetivo.....	16
4.3.3	Instruções.....	16
5	AVALIAÇÃO INICIAL.....	17
6	AVALIAÇÃO FINAL.....	18
7	CONCLUSÃO.....	20
	REFERÊNCIAS.....	21

1 INTRODUÇÃO

O Pensamento Computacional (PC) tem se consolidado como uma competência essencial no âmbito da educação contemporânea, estendendo-se para além dos domínios da ciência da computação. Popularizado por Jeannette Wing em 2006, o conceito de PC engloba habilidades que são fundamentais para a solução de problemas complexos, através de processos estruturados e analíticos. Embora Wing (2006) tenha desempenhado um papel crucial na disseminação desse termo, suas raízes remontam a disciplinas como matemática e engenharia, com antecedentes em trabalhos de pesquisadores como Seymour Papert(1980), que já exploraram o uso de abordagens computacionais para o aprendizado na década de 1970.

A relevância do Pensamento Computacional na educação moderna é sublinhada pela sua aplicabilidade em diversos contextos, desde o ensino básico até ao superior, promovendo o desenvolvimento de competências como abstração, distribuição, reconhecimento de padrões e criação de algoritmos. Essa abordagem tem sido apontada como um diferencial na formação de indivíduos capazes de enfrentar os desafios do mundo digital e de contribuir para a inovação tecnológica. Nesse sentido, a integração do Pensamento Computacional no currículo escolar surge como uma necessidade crescente, conforme apontado por Kurtz et al. (2008), que demonstra como essa inclusão pode melhorar significativamente a capacidade de resolução de problemas e preparar os estudantes para futuras carreiras

Este trabalho busca investigar a implementação do Pensamento Computacional em um contexto educacional prático, por meio de jogos lúdicos e desplugados, com o objetivo de verificar seu impacto no aprendizado e no desenvolvimento de habilidades cognitivas e técnicas. Para tanto, foi realizada uma revisão bibliográfica abrangente sobre o tema, além da aplicação de jogos educativos em turmas do ensino fundamental e médio de um colégio cívico-militar. O presente estudo busca

contribuir para a compreensão da importância do Pensamento Computacional no processo de ensino-aprendizagem e propor metodologias práticas para sua integração.

2 REVISÃO BIBLIOGRÁFICA

O termo Pensamento Computacional foi popularizado em 2006 por Wing em seus trabalhos, mas não foi ela quem criou o termo, embora tenha ajudado a popularizá-lo, é importante notar que o conceito já existia anteriormente. No entanto, identificar o primeiro ano exato em que o termo foi utilizado pode ser difícil, já que suas raízes estão em diversas disciplinas, como ciência da computação, matemática e educação, e podem remontar a décadas anteriores.

Jeannette Wing (2006) retrata o pensamento computacional como uma habilidade fundamental para todos, não somente para cientistas da computação. Uma habilidade fundamental, não mecânica, é algo que toda pessoa deve saber para viver na nossa sociedade, a habilidade mecânica é algo de rotina (Wing 2006).

Bundy (2007) e Nunes (2011) definem o Pensamento Computacional como habilidades comumente utilizadas na criação de programas computacionais como uma metodologia para resolver problemas específicos nas mais diversas áreas.

A definição e o entendimento do pensamento computacional evoluíram ao longo do tempo com a pesquisa e prática na área. Diversos estudos e ramificações bibliográficas têm contribuído para a compreensão dessa abordagem e sua aplicação em diferentes contextos educacionais e profissionais (Seymour Papert e Cynthia Solomon, 1972), (Papert 1980), (Brackmann 2017), (Liukas 2015), (Bezerra & Silveira 2011) e (Darling-Hammond e Bransford 2005).

À medida que o pensamento computacional se tornou mais amplamente compreendido, cresceu também o interesse em aplicá-lo de maneira prática nas salas de aula. Essa evolução teórica e empírica levou à consideração de como essa abordagem pode ser integrada de forma eficiente no currículo escolar, buscando

não apenas o desenvolvimento técnico, mas também o aprimoramento das habilidades cognitivas e de resolução de problemas dos alunos.

A integração do pensamento computacional no currículo escolar tem sido um tema recorrente. Estudos como o de Kurtz, et al. (2008) em "*Teaching Computational Thinking to Kids*" demonstram que a incorporação do pensamento computacional em atividades educacionais pode melhorar significativamente as habilidades de resolução de problemas dos alunos e prepará-los para carreiras em áreas técnicas e científicas, Kurtz (2008).

Com o crescente interesse na adoção do pensamento computacional na educação, torna-se essencial explorar os seus elementos centrais, que formam a base para seu ensino e aplicação. A partir de uma compreensão mais aprofundada, é possível perceber como esses pilares se conectam diretamente com as habilidades que os alunos precisam desenvolver para enfrentar desafios complexos.

Os pilares do pensamento computacional representam os fundamentos que sustentam essa abordagem, permitindo a resolução de problemas de maneira lógica e estruturada. Eles incluem a decomposição, o reconhecimento de padrões, a abstração e os algoritmos, cada um desempenhando um papel essencial na análise e formulação de soluções eficientes. Compreender esses pilares é fundamental para aplicar o pensamento computacional em diferentes contextos, facilitando tanto o aprendizado quanto a inovação em diversas áreas do conhecimento.

2.1 Decomposição

Jeannette Wing, em seu trabalho seminal de 2006 intitulado "Computational Thinking", descreve a decomposição como uma das quatro habilidades chaves do Pensamento Computacional, juntamente com padrões, abstração e algoritmos. A publicação é uma habilidade cognitiva central, que possibilita a abordagem de problemas complexos por meio de sua divisão em componentes menores e mais gerenciáveis. Esse processo é essencial para facilitar a compreensão e a resolução de questões no contexto da ciência da computação .

A decomposição não apenas permite que os alunos compreendam melhor o problema como um todo, mas também viabiliza a criação de algoritmos específicos para cada parte. Segundo Grover e Pea (2013) , essa prática é especialmente relevante no ensino de ciências da computação para estudantes, pois auxilia no desenvolvimento de habilidades analíticas e criativas permitidas para a resolução eficiente de problemas.

Dividir um problema em partes menores permite que os alunos se concentrem em cada componente, tornando o processo de resolução mais estruturado e objetivo. Dessa forma, a discussão se destaca como uma habilidade essencial tanto para estudantes quanto para profissionais da área, contribuindo diretamente para o desenvolvimento de soluções inovadoras e eficazes.

2.2 Abstração

David Parnas (1972) introduz o conceito de abstração de módulos e argumenta que a decomposição de um sistema em módulos é fundamental para gerenciar a complexidade e promover a clareza no design de software. Parnas defende que cada módulo deve ser projetado para encapsular uma funcionalidade específica, ocultando os detalhes de implementação e fornecendo uma interface clara e consistente para interagir com outros módulos, Parnas (1972).

A abstração também desempenha um papel crucial em diferentes paradigmas de programação, como a programação orientada a objetos (POO). Na POO, a abstração é implementada através de classes e objetos, que permitem que os desenvolvedores criem modelos do mundo real com propriedades e comportamentos encapsulados. O conceito de encapsulamento em POO, introduzido por autores como Barbara Liskov (1994) e Jeannette Wing (2006), é uma forma de abstração que facilita a criação de sistemas modulares e expansíveis. Em seu trabalho "A Behavioral Notion of Subtyping" (1994), Liskov discute como a abstração e o encapsulamento contribuem para a criação de hierarquias de classes robustas e reutilizáveis, Liskov (1994).

A abstração não se limita apenas à programação, mas também é um conceito importante em áreas como o design de sistemas e a engenharia de software. Em

sistemas complexos, a abstração permite a construção de arquiteturas que encapsulam detalhes de implementação, facilitando a comunicação entre diferentes partes do sistema e permitindo uma evolução mais flexível e controlada.

2.3 Reconhecimento de padrões

O Reconhecimento de Padrões é definido como a prática de identificar semelhanças e padrões com o objetivo de resolver problemas complexos de maneira mais eficaz. Esse processo envolve a busca por elementos que sejam iguais ou bastante semelhantes em diferentes problemas, Liukas (2015).

No texto de Brackman (2017) existe um exemplo prático, conforme pode ser visto na **Figura 1**.

Figura 1 - Similaridade entre raças de cachorros (reconhecimento de Padrões)



Fonte: Brackmann 2017

Note que, embora todos os cachorros tenham olhos, rabo e pelos, suas características podem variar, como a cor dos olhos, o comprimento do rabo e a cor dos pelos. No Pensamento Computacional, essas variações são chamadas de Padrões. Ao reconhecer um padrão para cachorros, é possível descrever outros cachorros com base nesse padrão, fazendo ajustes nas características.

2.4 Criação de algoritmos

Um dos autores mais influentes no campo da criação de algoritmos é Donald E. Knuth, cujas contribuições para a teoria e prática dos algoritmos são amplamente reconhecidas. Em seu trabalho *The Art of Computer Programming* (1968) Knuth explora a matemática e a lógica subjacentes à criação de algoritmos e apresenta uma abordagem sistemática para o design e análise de algoritmos. Knuth descreve algoritmos com rigor matemático, oferecendo uma base sólida para a compreensão e a aplicação de técnicas algorítmicas, Knuth (1968).

Knuth enfatiza a importância da análise de algoritmos, que envolve a avaliação da complexidade do tempo e do espaço dos algoritmos. A análise de complexidade ajuda a determinar a eficiência de um algoritmo, o que é crucial para o desenvolvimento de sistemas que precisam lidar com grandes volumes de dados ou realizar cálculos intensivos.

Outro autor relevante na área é Robert Sedgewick, que em seus trabalhos como *Algorithms* (1983) e *Introduction to Algorithms* (coautoria com Kevin Wayne, 2009), aborda uma ampla gama de técnicas algorítmicas e estruturas de dados. Sedgewick fornece uma visão detalhada sobre como diferentes algoritmos são aplicados a problemas específicos e discute os impasses (*trade-offs*) envolvidos na escolha de algoritmos apropriados para diferentes contextos (Sedgewick & Wayne, 2009).

A criação de algoritmos não é apenas um aspecto técnico, mas também uma habilidade criativa que envolve a inovação e a experimentação. O desenvolvimento contínuo de novos algoritmos e técnicas é fundamental para enfrentar os desafios emergentes da tecnologia, como o processamento de grandes volumes de dados, a otimização de sistemas e a implementação de soluções inteligentes.

2.5 Computação desplugada

É uma abordagem que visa ensinar conceitos fundamentais da Ciência da Computação sem a necessidade de dispositivos eletrônicos. Segundo Vieira, Passos e Barreto (2013), essa metodologia permite que os alunos aprendam por

meio de brincadeiras e atividades práticas que estão diretamente relacionadas a conteúdo curricular, promovendo um ambiente de aprendizado mais lúdico e acessível.

Tim Bell (2015), professor da Universidade de Canterbury, em conjunto com Ian Witten e Mike Fellows, realizou o projeto "Computer Science Unplugged" como uma resposta à necessidade de ensinar ciência da computação de maneira acessível e divertida, mesmo em contextos onde a necessidade de ensinar ciência da computação de maneira acessível e divertida, mesmo em contextos onde a tecnologia não está disponível. Este trabalho é uma das primeiras tentativas de formalizar o ensino de conceitos complexos de computação, como algoritmos, criptografia, estruturas de dados, e representação da informação, por meio de atividades práticas que não excluem o uso de computadores.

O impacto do trabalho de Tim Bell (2015), e seus colegas é significativo, especialmente em contextos educacionais onde o acesso a informática e tecnologias é limitado. Ao criar atividades que não dependam de informática, eles abriram novas possibilidades para o ensino de ciência da computação, particularmente em países e escolas com poucos recursos tecnológicos.

A computação desplugada busca transformar conceitos abstratos em experiências concretas, usando atividades interativas, jogos e desafios para ilustrar os princípios básicos da computação. As atividades propostas são projetadas para serem simples, envolventes e adequadas para estudantes desde o ensino fundamental, até mesmo para adultos que queiram aprender conceitos de computação.

3 METODOLOGIA

O presente trabalho baseou-se na revisão bibliográfica e na aplicação de jogos práticos em um ambiente educacional. A revisão bibliográfica incluiu uma análise abrangente dos principais trabalhos relacionados ao Pensamento Computacional (PC), desde sua popularização em 2006 por Jeannette Wing até os desenvolvimentos mais recentes, destacando-se os pilares centrais do PC, como decomposição, reconhecimento de padrões, abstração e algoritmos. Autores como Bundy (2007), Nunes (2011), Knuth (1968) e Brackman (2017) foram revisados

para enriquecer a base teórica.

Para a etapa prática, foi implementado um conjunto de jogos baseados no ensino do Pensamento Computacional em duas turmas do Colégio Cívico-Militar José de Alencar: uma turma do 9º ano e outra do 2º ano. Essas turmas foram selecionadas por possuírem alunos com idades adequadas e maior maturidade para participar dos jogos. As disciplinas de matemática foram escolhidas devido à sua carga horária, permitindo o desenvolvimento das atividades de maneira mais aprofundada. A primeira turma era composta por 27 alunos, com idades entre 14 a 16 anos, enquanto a segunda contava com 25 alunos, com idades entre 16 a 18 anos.

Foram aplicados três jogos didáticos: Cifra de César, Pixel Arte e Labirinto, que buscavam trabalhar diferentes aspectos do Pensamento Computacional, como reconhecimento de padrões, abstração e algoritmos. Os jogos foram aplicados em grupos e contaram com materiais práticos como convites criptografados, matrizes de coordenadas e blocos de programação tridimensionais. A tabela abaixo apresenta a relação entre os jogos propostos nesta metodologia e os pilares do pensamento computacional. Conforme pode ser observado na tabela, o 'X' indica a presença do pilar no respectivo jogo. Os jogos permitiram o trabalho com todos os pilares do pensamento computacional nas turmas propostas.

Tabela 1 - Jogos e seus pilares realizados na sala de aula

Jogos	Decomposição	Reconhecimento de Padrões	Abstração	Algoritmo
Cifra de César	X	X	X	
Pixel Arte	X			X
Labirinto	X	X	X	X

Fonte: Autor (2024)

A metodologia seguiu uma abordagem lúdica e educativa, integrando conceitos teóricos com práticas interativas, que facilitam o entendimento de conceitos abstratos do Pensamento Computacional. A aplicação foi monitorada para avaliar o desempenho e o engajamento dos alunos, observando como eles desenvolviam habilidades de resolução de problemas por meio da decomposição de tarefas complexas.

4 JOGOS APLICADOS

Nesta seção apresentaremos os jogos aplicados nas duas turmas do Colégio Cívico-Militar José de Alencar, para cada jogo destacamos os materiais necessários, o objetivo do jogo e as instruções de como aplicar cada um deles em sala de aula.

4.1 Cifra de César

4.1.1 Materiais

- Convites com texto criptografado
- Lápis e borracha
- Tabela de decodificação

4.1.2 Objetivo

Trabalhar o reconhecimento de padrões e análise de dados. A criptografia funciona com a troca do alfabeto de acordo com as escolhas da casa, podendo ser de 3 a 7 casas. A descryptografia funciona da direita para esquerda e a criptografia da direita para esquerda.

4.1.3 Instruções

Divida os alunos em grupos e entregue um convite para cada grupo para que possam decifrar o texto do convite usando o conceito de Cifra de César.

A **Cifra de César** é implementada como um tipo de cifra de substituição. Neste jogo, os caracteres de um texto fornecido são substituídos por outros, selecionados com base em um deslocamento predefinido ao longo do alfabeto, podendo ser para a direita ou para a esquerda. Este deslocamento é uma componente crítica do processo de cifragem. Para criptografar uma palavra inteira você pode deslocar duas casas a direita. Por exemplo, 'amigo' criptografado é 'cokiq'.

- $a + 2 = c$
- $m + 2 = o$
- $i + 2 = k$
- $g + 2 = i$
- $o + 2 = q$

Retornar o texto ao normal é chamado de descryptografia (**Figura 2**). Para

descriptografar uma palavra, basta subtrair a chave em vez de adicioná-la:

- o $c - 2 = a$
- o $o - 2 = m$
- o $k - 2 = i$
- o $i - 2 = g$
- o $q - 2 = o$

Figura 2 - Exemplo da Cifra de César



Fonte: Autor (2023)

A seguir a **Figura 3** mostra o texto repassado para que os alunos pudessem descriptografar, foi baseado no assunto em que a professora estava direcionado aos alunos.

Figura 3 - Exemplo da mensagem criptografada passada para os alunos

O bloco apresenta duas seções de texto em um fundo de grade azul. A primeira seção, intitulada "Cifra de Cesár criptografado", contém o texto: "E jyrçãs hs 2º kvey sy jyrçãs uyehvâxmge é yqe jyrçãs hi hsqírms viep, sy wine, uyepuyiv rúqivs viep tshi wiv s b i, e gehe rúqivs viep b, ewwsgmeqsw yq rúqivs he jsvqe... ". A segunda seção, intitulada "Cifra de Cesár descriptografado", contém o texto: "A função do 2º grau ou função quadrática é uma função de domínio real, ou seja, qualquer número real pode ser o x e, a cada número real x, associamos um número da forma ...".

Fonte: Autor (2023)

4.2 Pixel Arte

4.2.1 Materiais

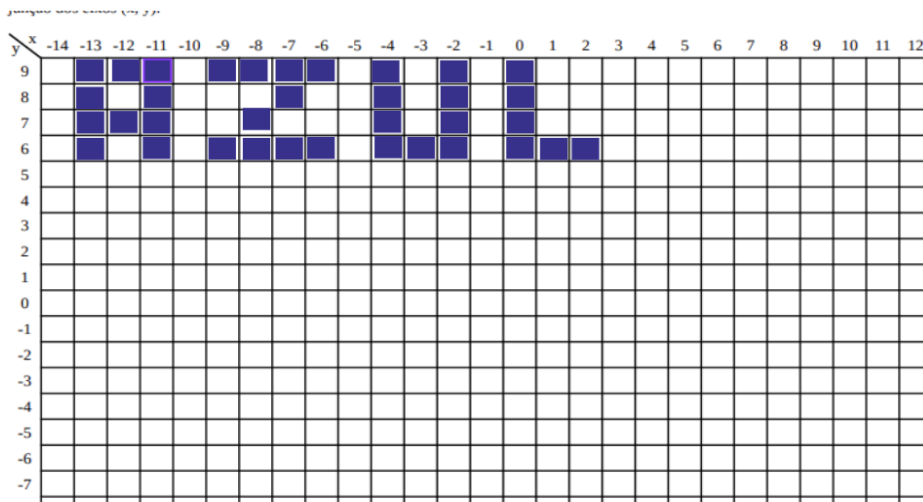
- Entrega da matriz
- Entrega dos códigos, dos eixos
- Papel e caneta

4.2.2 Objetivo

Trabalhar a habilidade de identificar padrões. O plano cartesiano (Figura 4) é formado por duas dessas retas: uma responsável pela coordenada horizontal e outra responsável pela coordenada vertical. É comum usar as letras x para a primeira e y para a segunda e os termos “coordenada x” e “coordenada y”.

- Com os alunos divididos em grupos, o primeiro grupo terá que identificar as primeiras coordenadas e em seguida repassar para o próximo grupo.
- A matriz utilizada é baseada no plano cartesiano, em que o aluno terá que encontrar o X e Y para desenvolver a atividade a partir das coordenada.

Figura 4 - Exemplo da matriz utilizada pelos alunos



Fonte: Autor (2023)

4.3 Labirinto

4.3.1 Materiais

- Entrega do mapa da rota
- Blocos de programação
- Pino de personagens

4.3.2 Objetivo

Apresentar o conceito de algoritmos, a linguagem de programação baseada em blocos funciona como um quebra-cabeça, no qual cada peça é um comando e quando montamos uma sequência de peças conseguimos escrever um programa.

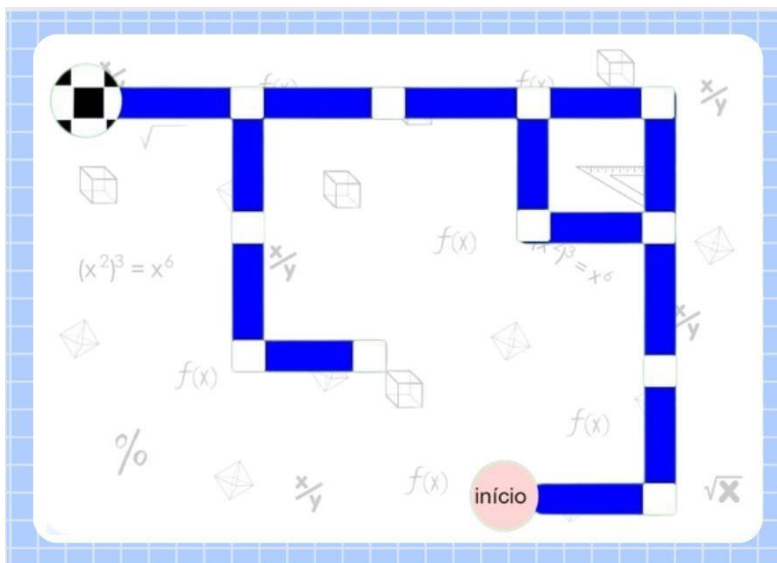
4.3.3 Instruções

Com os alunos já divididos em grupos, eles terão que decifrar a rota com utilização dos blocos de programação.

A utilização dos blocos está em (senão, enquanto, siga em frente, vire à esquerda). Os alunos terão que montar a rota utilizando o pensamento lógico, montando o algoritmo para que o personagem chegue ao seu destino.

No que concerne ao Labirinto, este jogo tem sua origem no Blockly Games, porém foi adaptado para o projeto de maneira a incorporar elementos tridimensionais utilizando blocos impressos em 3D. Essa adaptação (**Figura 5**) visa permitir que os alunos engajem-se na atividade educativa sem a necessidade de equipamentos computacionais.

Figura 5 - Modelo do labirinto apresentado aos alunos



Fonte: Autor (2024)

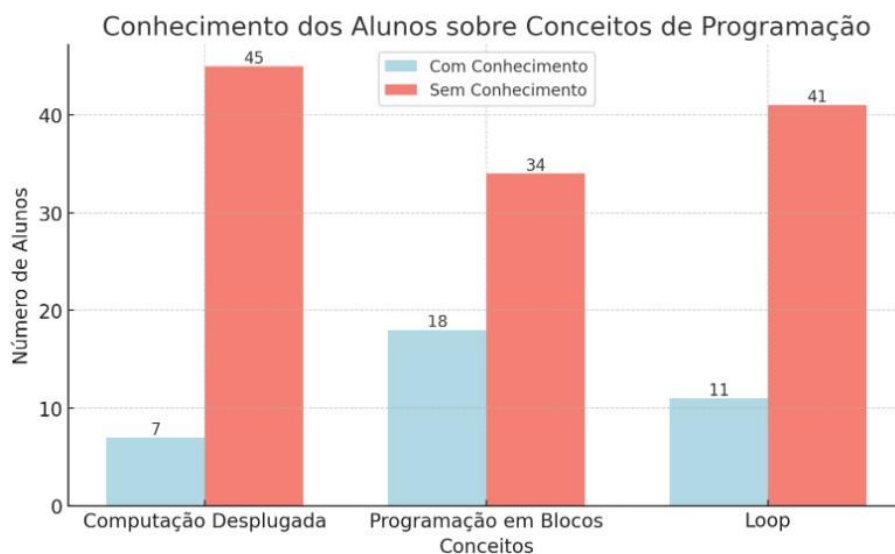
5 AVALIAÇÃO FINAL

No início da fase de coleta de dados, realizada com estudantes do 9º ano do Ensino Fundamental e do 2º ano do Ensino Médio, envolvendo um total de 52 alunos, foi proposta uma questão preliminar com o intuito de avaliar o nível de familiaridade dos discentes com o conceito de "computação desplugada". A indagação apresentada aos participantes foi: "Você já ouviu falar sobre computação desplugada?". A análise subsequente das respostas indicou que uma significativa maioria, correspondente a **86,54%** (45 alunos), não possuía conhecimento prévio acerca deste conceito, enquanto **13,46%** (7 alunos) já tinham algum conhecimento prévio.

Posteriormente, foi proposta uma nova questão: "Você possui familiaridade com a programação em blocos?". A partir das respostas obtidas, constatou-se que aproximadamente **65,38%** (34 alunos) dos participantes não tinham conhecimento prévio sobre programação em blocos, enquanto **19,23%** (10 alunos) relataram alguma experiência prévia nessa área. Uma questão adicional foi formulada para avaliar o entendimento dos alunos sobre o conceito de "loop" em programação. Essa pergunta foi estruturada da seguinte maneira: "Você sabe o que é um loop?". A análise das respostas indicou que uma maioria significativa, correspondente a **78,85%** (41 alunos), afirmou não possuir conhecimento sobre o tema, enquanto **21,15%** (11 alunos) declararam ter compreensão sobre o conceito de loop. Esses dados sugerem uma lacuna no conhecimento dos alunos a respeito de estruturas de repetição em programação.

O gráfico na **Figura 6** apresenta a distribuição do conhecimento dos alunos antes da aplicação dos jogos, em relação a três conceitos de programação: Computação Desplugada, Programação em Blocos e Estruturas de Loop.

Figura 6 - Gráfico representando as porcentagens das respostas dos alunos



Fonte: Autor (2024)

6 AVALIAÇÃO FINAL

Após a realização da atividade final, que envolveu um exercício de labirinto, foi conduzida uma investigação adicional para avaliar a compreensão dos estudantes em relação à programação em blocos. A questão específica apresentada foi: "Você conseguiu compreender a programação em blocos?". Os resultados indicaram que uma significativa maioria, de aproximadamente **88,46%** (46 alunos), afirmou ter adquirido uma compreensão efetiva, enquanto **11,54%** (6 alunos) expressaram não ter assimilado o conteúdo.

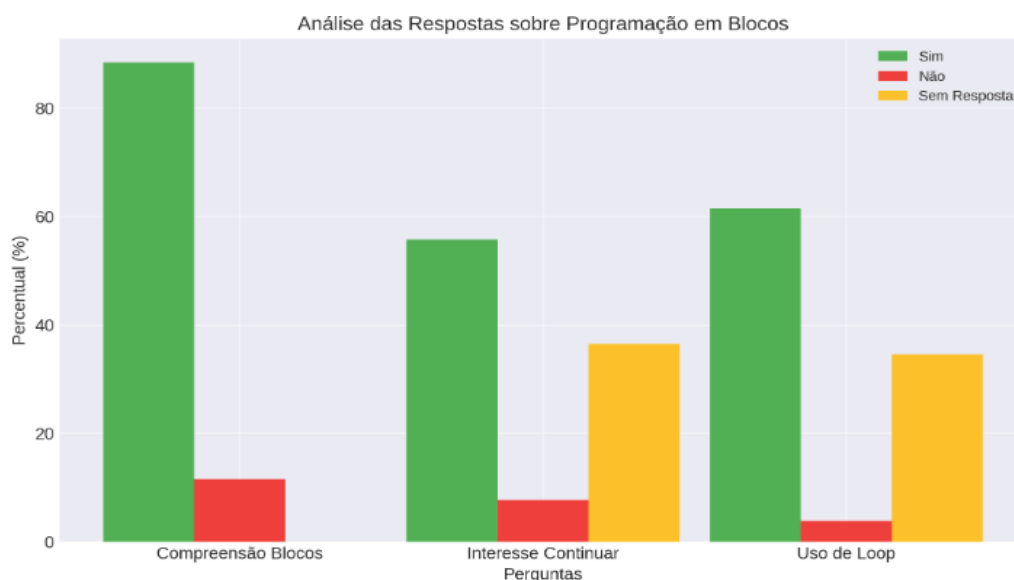
Em seguida, uma segunda pergunta foi feita aos participantes com o objetivo de investigar o interesse contínuo no aprendizado de programação em blocos. A pergunta era: "Você continuaria adquirindo mais conhecimentos sobre programação em blocos?". A análise das respostas mostrou que cerca de **55,77%** (29 alunos) manifestaram interesse em continuar o estudo dessa modalidade, **7,69%** (4 alunos) responderam negativamente, e **36,54%** (19 alunos) não responderam à questão.

No contexto do estudo sobre a aplicação prática dos conhecimentos adquiridos em programação, especialmente no que se refere ao uso de estruturas de loop, foi proposta aos alunos a seguinte questão: "Você utilizou loop em sua programação?". A análise das respostas revelou que uma maioria significativa dos alunos, correspondendo a **61,54%** (32 alunos), afirmou ter aplicado o conceito de loop em

suas práticas de programação. Por outro lado, uma parcela considerável, representando **34,62%** (18 alunos), optou por não responder à questão. Uma minoria de **3,85%** (2 alunos) indicou não ter utilizado loops em suas atividades de programação. Esses dados refletem a diversidade no nível de engajamento e aplicação dos conceitos de programação entre os alunos.

O gráfico na **Figura 7** apresenta a distribuição percentual das respostas dos alunos após a aplicação dos jogos, em relação à compreensão da programação em blocos, ao interesse em continuar aprendendo sobre o tema e à utilização de estruturas de loop em suas atividades de programação.

Figura 7 - Gráfico representando as porcentagens das respostas dos alunos



Fonte: Autor (2024)

7 CONCLUSÃO

O Pensamento Computacional, popularizado por Jeannette Wing em 2006, é uma habilidade essencial para a resolução de problemas complexos em diversas áreas do conhecimento, indo além da ciência da computação. Sua integração nas práticas educacionais, como mostrado através das atividades lúdicas e interativas realizadas com os alunos do Colégio Cívico-Militar José de Alencar, demonstra o impacto positivo na formação de habilidades cognitivas e de resolução de problemas. As atividades

baseadas em jogos, como Cifra de César, Pixel Arte e Labirinto, revelam que o uso de abordagens desplugadas e práticas pode despertar o interesse dos alunos e aprimorar sua compreensão sobre conceitos fundamentais, como algoritmos, decomposição e reconhecimento de padrões. A avaliação final dos estudantes sugere que, após a aplicação dessas metodologias, houve um aumento significativo no entendimento e no interesse pelos conceitos de programação em blocos, mostrando a importância de continuar desenvolvendo estratégias que incorporem o Pensamento Computacional no currículo escolar.

REFERÊNCIAS

- BEZERRA, A. S.; SILVEIRA, I. R. Pensamento computacional: conceitos e suas implicações. **Revista Brasileira de Informática na Educação**, v. 10, n. 2, p. 29-42, 2018.
- BRACKMANN, P. C. **Desenvolvimento do pensamento computacional através da robótica educacional**: um estudo de caso. Dissertação (Mestrado) – Universidade Federal de Santa Maria, 2016.
- BUNDY, A. **O pensamento computacional é penetrante**. Journal of Scientific and Industrial Research, v. 72, n. 10, p. 569-575, 2013.
- DARLING-HAMMOND, L.; BRANSFORD, J. (Eds.). **Preparando professores para um mundo em mudança**: O que os professores devem aprender e ser capazes de fazer. São Paulo: Editora Unesp, 2007.
- GROVER, S.; PEA, R. **Pensamento computacional em K–12**: Uma revisão do estado do campo. Pesquisador educacional, v. 3, n. 4, p. 1-16, 2013.
- KNUTH, D. E. **A Arte da Programação de Computadores**: Volume 1: Algoritmos Fundamentais. 3. ed. São Paulo: Editora Pearson, 2011.
- KURTZ, B. et al. **Ensino de pensamento computacional para crianças**. Communications of the ACM, v. 58, n. 9, p. 16-20, 2015.
- LIUKAS, L. **Olá Ruby**: aventuras em codificação. São Paulo: Editora Novatec, 2018.
- LISKOV, B. **Uma noção comportamental de subtipagem**. ACM Transactions on Programming Languages and Systems, v. 16, n. 2, p. 181-208, 1994.
- NUNES, M. C. **Computação no ensino básico**: uma proposta para o pensamento computacional. Revista Brasileira de Informática na Educação, v.5, n. 1, p. 57-64, 2017.
- PAPERT, S. **Mindstorms**: crianças, computadores e ideias poderosas. 2. ed. São Paulo: Editora Artesanato, 2013.
- PAPERT, S.; SOLOMON, C. **Vinte coisas para fazer com um computador**. Artificial Intelligence Laboratory Memo, n. 195, p. 1-24, 1985.
- SEDGEWICK, R.; WAYNE, K. **Introdução aos Algoritmos**. São Paulo: Editora Pearson, 2016.
- BELL, T.; WITTEN, I.; FELLOWS, M. **Ciência da Computação Unplugged**: Ensinando Ciência da Computação sem Computadores. São Paulo: Editora Novatec, 2016.

VIEIRA, A. P. C.; PASSOS, C. E.; BARRETO, L. O. **Computação desplugada**: um método alternativo de ensino. Revista Brasileira de Informática na Educação, v. 5, n. 2, p. 71-83, 2018.

WING, J. M. Pensamento computacional. Communications of the ACM, v. 49, n. 3, p. 33-35, 2006.