



USO DO PENSAMENTO COMPUTACIONAL EM SALA DE AULA

Silvana Aparecida Ataíde do Nascimento
Claudio Zarate Sanavria



PROFEPT
PROGRAMA DE PÓS-GRADUAÇÃO EM
EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA



**INSTITUTO
FEDERAL**
Mato Grosso do Sul

N244u Nascimento, Silvana Aparecida Ataíde do
O uso do pensamento computacional em sala de aula / Silvana
Aparecida Ataíde do Nascimento [Recurso eletrônico]. – Campo Grande-
MS, 2024.

1 recurso online (31 p.): il. Color.

Publicação digital (e-book) no formato PDF.
ISBN 978-65-01-12931-0

1. Computação. 2. Ensino. 3. Didática. 4. Metodologia. I. Sanavria,
Claudio Zarate. II. Instituto Federal de Mato Grosso do Sul. Programa de
Pós-Graduação em Educação Profissional e Tecnológica. III. Título.

CDD 23. ed. 371.334

Ficha catalográfica elaborada pela Biblioteca do Campus Campo Grande - IFMS
Bibliotecário: Marli Selini CRB1 – n. 2570

USO DO PENSAMENTO COMPUTACIONAL EM SALA DE AULA

Autores



Silvana Nascimento

Formada em Tecnologia em Processamento de Dados pela UNIDERP em 1999, especialista lato sensu em Informática na Educação pela UNIDERP. Mestre em Educação Profissional e Tecnológica(IFMS). Experiência de 22 anos na área de TI e Educação: Análise e desenvolvimento de sistemas, Coordenação de equipe e Gerenciamento de Projetos; Docência para curso superior em Tecnologia em Análise de Sistemas, na área de modelagem e desenvolvimento de projetos; Tutoria de curso de pós-graduação em Educação a Distância, Governança em TI e Uso de Tecnologias na Educação; Certificada ITIL V3; Certificada Scrum Foundation Professional; Realização de treinamentos diversos (programação em PHP, e banco de dados MySql, Modelagem de Dados, UML, CMM, CMMI, Gerenciamento de Projetos com PMBOK, e melhores práticas de Governança: COBIT e ITIL, Qualidade no Atendimento)

USO DO PENSAMENTO COMPUTACIONAL EM SALA DE AULA

Autores



Claudio Sanavria

É bacharel em Análise de Sistemas e especialista em Engenharia de Websites pela Universidade Federal de Mato Grosso do Sul (UFMS), pedagogo pelo Centro Universitário da Grande Dourados (UNIGRAN), especialista em Gestão Escolar pela Universidade de São Paulo (USP), mestre em Educação pela Universidade Católica Dom Bosco (UCDB) e doutor em Educação pela Universidade Estadual Paulista Júlio de Mesquita Filho (UNESP). É professor, pesquisador e, atualmente, Diretor-Geral do Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso do Sul (IFMS) - Campus Nova Andradina, onde também coordena o NIPETI (Núcleo Interdisciplinar de Pesquisa, Estudo e Desenvolvimento em Tecnologia da Informação). Atua como secretário regional adjunto da Sociedade Brasileira de Computação (SBC), Regional MS. É docente do IFMS no Programa de Mestrado Profissional em Educação Profissional e Tecnológica (ProfEPT) e tem experiência na área de Educação e Ciência da Computação, abordando principalmente os temas: Educação Profissional e Tecnológica, Formação de Professores, Tecnologias Educacionais e Engenharia de Software.

APRESENTAÇÃO

Bem-vindo ao nosso e-book sobre Pensamento Computacional na Sala de Aula! Este guia foi elaborado como guia de Referência da Formação Continuada para Docentes de Informática do Ensino Médio Integrado do IFMS, a qual fez parte da Dissertação de Mestrado desenvolvida pelos autores com o título: O pensamento computacional como prática educativa no ensino médio integrado: o entendimento de professores de informática a partir de uma formação continuada. Além disso esse guia tem como objetivo apresentar os conceitos do Pensamento Computacional a educadores que desejam incorporar o pensamento computacional no ambiente escolar de maneira prática e envolvente. Se você é um professor, coordenador pedagógico ou um entusiasta da educação, este e-book é para você.

O pensamento computacional não é apenas uma habilidade essencial para futuros profissionais de tecnologia; é uma competência fundamental para todos os estudantes, independentemente da sua área de interesse. Desenvolvido pela renomada pesquisadora Jeannette Wing, o pensamento computacional envolve quatro pilares principais: Decomposição, Reconhecimento de Padrões, Abstração e Algoritmos. Esses conceitos são fundamentais para resolver problemas de maneira eficiente e estruturada e são aplicáveis a uma ampla gama de disciplinas e contextos.

Neste e-book, vamos explorar cada um desses pilares em detalhes, oferecendo exemplos práticos e ideias de atividades que você pode implementar diretamente em sua sala de aula. Nossa abordagem é focada em tornar o pensamento computacional acessível e relevante para os alunos, promovendo um aprendizado significativo e envolvente.

O que você encontrará neste e-book:

- **Exemplos Práticos de Aplicação:** Cada pilar do pensamento computacional será ilustrado com exemplos concretos e aplicações no mundo real. Veremos como a decomposição pode ajudar na resolução de problemas complexos, como reconhecer padrões pode facilitar a aprendizagem de novos conceitos e como a abstração e algoritmos podem ser usados para criar soluções inovadoras.

- **Ideias de Atividades a serem aplicadas:** Forneceremos sugestões para atividades que você pode realizar com seus alunos para praticar e internalizar os pilares do pensamento computacional. Essas atividades foram projetadas para serem envolventes e adaptáveis a diferentes níveis de ensino e disciplinas.

O pensamento computacional oferece ferramentas valiosas para desenvolver habilidades de resolução de problemas, pensamento crítico e criatividade. Esperamos que este e-book inspire e capacite você a implementar essas práticas em sua sala de aula, proporcionando aos seus alunos uma experiência de aprendizado enriquecedora e relevante.

Vamos juntos explorar e aplicar o pensamento computacional para transformar a maneira como ensinamos e aprendemos!

SUMÁRIO

Introdução	7
Capítulo 1 - Pilares do Pensamento Computacional.....	9
Capítulo 2 - Desenvolvendo Competências e Habilidades com Pensamento Computacional	14
Capítulo 3 - Computação Desplugada.....	17
Capítulo 4 - Introdução à computação	19
Anexos - Usando Algotcard para trabalhar os conceitos do PC	22
Anexos - Links para consulta e outras atividades.....	29
Referências	30

Introdução

No artigo intitulado "Computational Thinking", Wing argumentou que o Pensamento Computacional deveria ser uma habilidade fundamental, assim como ler, escrever e fazer aritmética. Ela delineou as características principais do Pensamento Computacional, incluindo abstração, algoritmos, decomposição e reconhecimento de padrões.

O Pensamento Computacional (PC) é uma habilidade fundamental no mundo moderno, permeando diversas áreas da vida e sendo essencial para a resolução de problemas complexos. Ele não está restrito apenas aos programadores de computadores, mas é uma abordagem mental que pode ser aplicada em várias disciplinas na resolução de problemas. No material a seguir vamos explorar os Pilares do Pensamento Computacional segundo Jeannette Wing (2006):

1. **Decomposição:** A capacidade de dividir um problema complexo em partes menores e mais manejáveis. Isso facilita a resolução de problemas ao lidar com cada componente separadamente.
2. **Reconhecimento de Padrões:** Identificar semelhanças ou padrões comuns dentro de um problema. Isso ajuda a simplificar a resolução de problemas, permitindo que soluções semelhantes sejam aplicadas a problemas similares.
3. **Abstração:** Focar nos aspectos mais relevantes de um problema e ignorar os detalhes desnecessários. A abstração permite a criação de modelos que representam o problema de forma simplificada e compreensível.
4. **Algoritmos:** Desenvolver uma sequência de passos lógicos e finitos para resolver um problema ou realizar uma tarefa. Algoritmos são essenciais para a implementação de soluções e são a base para a programação e automação.

Esses pilares são fundamentais para abordar problemas de forma estruturada e eficiente e são aplicáveis em uma ampla gama de disciplinas além da computação.

O Pensamento Computacional não se limita a programadores ou cientistas da computação; é uma habilidade valiosa em diversas profissões e contextos. Através da abordagem do PC, as pessoas podem melhorar sua capacidade de resolver problemas complexos, tomar decisões informadas e compreender melhor o mundo cada vez mais digital em que vivemos.

Ao longo do e-book serão explorados cada um dos pilares do Pensamento Computacional (PC) e atividades que podem ser trabalhadas em sala de aula para o desenvolvimento do PC em sala de aula.

Capítulo 1 – Pilares do Pensamento Computacional



Fonte: Freepik

O Pensamento Computacional envolve quatro pilares fundamentais, conforme proposto por Jeannette Wing. A seguir a conceituação de cada um dos pilares bem como exemplo de sua aplicação:

1. **Decomposição (Decomposition):**

Definição: Dividir um problema complexo em partes menores e mais gerenciáveis.

Exemplo: Ao abordar o problema "criar um jogo de computador", a decomposição envolveria dividir o problema em partes menores, como design de personagens, programação de movimento, implementação de regras do jogo, entre outros.

2. **Reconhecimento de Padrões (Pattern Recognition):**

Definição: Identificar padrões ou tendências nos dados.

Exemplo: Analisar uma série numérica e reconhecer um padrão de crescimento exponencial. Isso também pode ser aplicado na identificação de padrões em algoritmos ou na otimização de código.

3. Abstração (Abstraction):

Definição: Focar nos detalhes mais importantes e ignorar os não essenciais.

Exemplo: Ao criar um programa para um jogo, abstrair envolveria concentrar-se nas funcionalidades essenciais do jogo, como a lógica de movimento dos personagens, sem se preocupar detalhadamente com a implementação de gráficos ou efeitos sonoros.

4. Algoritmos (Algorithmic Thinking):

Definição: Desenvolver uma solução passo a passo para um problema ou tarefa.

Exemplo: Para o problema de ordenar uma lista de números, o algoritmo de ordenação por seleção pode ser usado, envolvendo a seleção iterativa do menor elemento e sua troca com o primeiro elemento não classificado.

Esses pilares não estão isolados, e muitas vezes eles se entrelaçam durante a resolução de problemas. Ao aplicar esses princípios, indivíduos desenvolvem uma mentalidade que os capacita a abordar desafios complexos de maneira mais eficaz, independentemente do campo ou da situação.

Vamos usar um exemplo prático para ilustrar como os pilares do pensamento computacional se inter-relacionam na resolução de um problema. Suponha que o problema seja **“organizar uma festa de aniversário”**. Aqui está como os pilares do pensamento computacional podem ser aplicados:

1. Decomposição

Primeiro, você decompõe o problema principal (organizar a festa) em tarefas menores e mais manejáveis:

- **Planejamento:** Definir a data, hora e local da festa.
- **Convidados:** Fazer a lista de convidados e enviar convites.
- **Decoração:** Escolher e preparar a decoração.
- **Comida e Bebida:** Planejar o menu e providenciar a comida e bebida.
- **Entretenimento:** Organizar atividades e entretenimento para os convidados.
- **Logística:** Coordenação de transporte e outros aspectos logísticos.

Cada uma dessas tarefas é um subtarefa que pode ser tratada individualmente, tornando o problema geral mais fácil de abordar.

2. Reconhecimento de Padrões

Em seguida, você pode identificar padrões e semelhanças entre as tarefas para simplificar o processo:

- **Convidados:** A tarefa de fazer a lista de convidados é semelhante à tarefa de enviar convites. Ambas envolvem comunicação e coordenação com as pessoas.
- **Decoração e Comida e Bebida:** Ambos envolvem aspectos de planejamento e aquisição de itens. Talvez você possa aplicar uma abordagem semelhante para escolher fornecedores ou definir orçamentos.
- **Entretenimento:** Pode haver padrões em tipos de atividades que você já usou em festas anteriores ou em sugestões que você já ouviu.

Reconhecer esses padrões ajuda a evitar a reinvenção da roda e simplifica o planejamento.

3. Abstração

A abstração envolve focar apenas nos aspectos relevantes de cada tarefa e ignorar detalhes desnecessários:

- **Convidados:** Em vez de se preocupar com detalhes como o tipo específico de convite, você abstrai para apenas garantir que todos os convidados recebam o convite com informações claras.
- **Decoração:** Em vez de escolher todos os detalhes da decoração, você pode se concentrar em um tema geral que unifique os elementos de decoração.
- **Comida e Bebida:** Em vez de planejar cada prato individualmente, você define categorias principais de comida e bebida (entrada, prato principal, sobremesa) e lida com detalhes conforme necessário.

A abstração ajuda a manter o foco no que é essencial e evita sobrecarregar o processo com informações irrelevantes.

4. Algoritmos

Finalmente, você cria algoritmos ou passos lógicos para realizar cada tarefa:

- **Planejamento da Festa:**
 - a. Escolher uma data e hora.
 - b. Selecionar e reservar o local.
- **Lista de Convidados:**
 - c. Fazer uma lista preliminar de convidados.
 - d. Confirmar a lista final.
 - e. Enviar convites.
- **Decoração:**
 - f. Escolher um tema.
 - g. Comprar ou fazer decorações.
 - h. Montar a decoração no local da festa.
- **Comida e Bebida:**
 - i. Escolher um menu.
 - j. Contratar ou preparar a comida.
 - k. Organizar bebidas e utensílios necessários.
- **Entretenimento:**
 - l. Selecionar atividades ou jogos.
 - m. Preparar ou contratar entretenimento.
- **Logística:**
 - n. Organizar transporte para os itens e pessoas, se necessário.

o. Garantir que tudo esteja em ordem no dia da festa.

Os algoritmos são passos concretos e sequenciais que garantem que todas as partes do plano sejam executadas de forma organizada e eficiente.

Resumo

No exemplo citado, a decomposição divide o problema em tarefas menores, o reconhecimento de padrões ajuda a aplicar soluções semelhantes a tarefas relacionadas, a abstração permite focar nos aspectos mais importantes de cada tarefa e algoritmos fornecem passos claros para executar cada tarefa. Esses pilares trabalham juntos para tornar o planejamento e a execução de uma festa complexa mais gerenciáveis e sistemáticos.

Capítulo 2 – Desenvolvendo Competências e Habilidades com Pensamento Computacional



Fonte: Freepik

O Pensamento Computacional, conforme proposto por Jeannette Wing, é uma abordagem que visa desenvolver habilidades cognitivas fundamentais que são essenciais para resolver problemas complexos e compreender os fundamentos da ciência da computação. Aqui estão os principais conceitos do Pensamento Computacional, de acordo com Wing, e como desenvolver competências e habilidades relacionadas:

1. Decomposição:

- **Definição:** Quebrar um problema ou sistema em partes menores.
- **Desenvolvimento de Habilidade:** Pratique a habilidade de dividir problemas complexos em partes menores. Isso pode ser feito através da resolução de problemas de programação ou projetos que envolvam a divisão de tarefas em etapas gerenciáveis.

2. Reconhecimento de Padrões:

- **Definição:** Identificar tendências ou regularidades em dados.
- **Desenvolvimento de Habilidade:** Participe de atividades que envolvam a análise de dados. Isso pode incluir a resolução de problemas algorítmicos, a manipulação de conjuntos de dados e a aplicação de técnicas de aprendizado de máquina simples.

3. Abstração:

- **Definição:** Focar nos detalhes mais importantes, ignorando os não essenciais.
- **Desenvolvimento de Habilidade:** Pratique a abstração ao desenvolver algoritmos. Concentre-se nos aspectos essenciais de um problema e projete soluções que possam ser aplicadas a diferentes contextos.

4. Algoritmos:

- **Definição:** Uma sequência ordenada de passos para realizar uma tarefa ou resolver um problema.
- **Desenvolvimento de Habilidade:** Resolva problemas algorítmicos, participe de competições de programação, e desenvolva projetos que exijam a aplicação de algoritmos eficientes.

Outras habilidades que podem ser desenvolvidas:

- Pensamento Lateral:
 - **Definição:** Abordagem não convencional para encontrar soluções criativas.
 - **Desenvolvimento de Habilidade:** Engaje-se em atividades que promovam a criatividade e a resolução de problemas de maneiras não tradicionais. Isso pode incluir projetos de programação criativa ou desafios de resolução de problemas.
- Resolução de Problemas:
 - **Definição:** Encontrar soluções eficazes para desafios.
 - **Desenvolvimento de Habilidade:** Pratique a resolução de problemas através de projetos práticos. Envolve-se em competições de programação, participe de hackathons e desafie-se com problemas do dia a dia.

- Pensamento Crítico:
 - **Definição:** Avaliar informações de maneira crítica e fazer julgamentos informados.
 - **Desenvolvimento de Habilidade:** Ao desenvolver software, questione constantemente a eficácia e a segurança de suas soluções. Esteja disposto a rever e melhorar constantemente seu trabalho.

Ao integrar esses conceitos em sua prática diária, seja através de projetos, estudos autodidatas, participação em comunidades de programação ou cursos formais, você estará fortalecendo suas competências e habilidades de Pensamento Computacional de acordo com a abordagem de Jeannette Wing.

Capítulo 3 – Computação Desplugada



Fonte: Freepik

Trabalhar a computação desplugada, ou seja, sem o uso direto de computadores, é uma ótima maneira de desenvolver as habilidades do Pensamento Computacional.

Em 2015, Tim Bell, um dos principais defensores da computação desplugada, ofereceu uma definição clara e abrangente desse conceito. A computação desplugada refere-se a atividades de ensino e aprendizagem que abordam conceitos de ciência da computação sem a necessidade de computadores ou tecnologia digital. A ideia central é ensinar princípios fundamentais da computação usando apenas materiais físicos e atividades práticas.

Aqui estão algumas atividades e estratégias que você pode incorporar para promover o Pensamento Computacional sem a necessidade de dispositivos eletrônicos:

Jogos de Abstração:

Atividade: Peça aos participantes para representarem objetos ou situações complexas com símbolos simples. Isso pode ser feito através de desenhos, símbolos matemáticos, ou mesmo por meio de jogos de palavras.

Simulações de Algoritmos:

Atividade: Introduza problemas que podem ser resolvidos com algoritmos simples. Por exemplo, simule a ordenação de objetos em uma sala ou a busca por um item específico em uma sequência.

Jogos de Pensamento Lateral:

Atividade: Apresente quebra-cabeças e desafios que estimulem a resolução de problemas de maneiras não convencionais. Esses jogos podem incluir problemas de lógica, enigmas e quebra-cabeças.

Teatro de Algoritmos:

Atividade: Peça aos participantes para representarem algoritmos por meio de encenações. Por exemplo, eles podem demonstrar como um algoritmo de classificação funciona, cada pessoa representando uma etapa do processo.

Quebra-Cabeças de Resolução de Problemas:

Atividade: Forneça quebra-cabeças que exigem resolução de problemas. Isso pode incluir enigmas matemáticos, quebra-cabeças lógicos ou desafios de pensamento crítico.

Atividades Colaborativas:

Atividade: Encoraje a colaboração em atividades que envolvem a resolução de problemas. Isso pode incluir jogos de equipe onde os participantes precisam trabalhar juntos para atingir um objetivo comum.

Ao realizar essas atividades, é importante enfatizar os princípios do Pensamento Computacional, como a decomposição, o reconhecimento de padrões, a abstração e a criação de algoritmos. Essas atividades práticas ajudarão a desenvolver as habilidades essenciais, preparando os participantes para aplicar o Pensamento Computacional em contextos mais amplos, incluindo o uso de tecnologia digital quando apropriado.

Capítulo 4 – Introdução à Programação



Fonte: Freepik

Introduzir a programação por meio do Pensamento Computacional é uma abordagem eficaz para ajudar os aprendizes a desenvolverem uma base sólida na resolução de problemas e na compreensão dos conceitos fundamentais da ciência da computação. Aqui estão algumas estratégias que podem ser utilizadas:

1. Decomposição:

- **Atividade Prática:** Apresente um problema e peça aos alunos para identificar as partes essenciais. Em seguida, peça que eles criem uma lista de tarefas ou algoritmo para resolver cada parte. Por exemplo, ao criar um jogo, descomponha-o em etapas como design, movimentação de personagens e lógica do jogo.

2. Padronização:

- **Atividade Prática:** Introduza padrões e tendências em conjuntos de dados simples. Use problemas práticos ou jogos para que os alunos identifiquem padrões. Isso pode ser feito com sequências numéricas, padrões de movimento ou até mesmo padrões em textos.

3. Abstração:

- **Atividade Prática:** Mostre como abstrair detalhes não essenciais ao resolver problemas. Por exemplo, ao criar um programa, concentre-se inicialmente na lógica central antes de se aprofundar em detalhes de implementação, como a interface gráfica.

4. Algoritmos:

- **Atividade Prática:** Introduza conceitos algorítmicos por meio de problemas do dia a dia. Peça aos alunos para criar algoritmos passo a passo para tarefas simples. Isso pode ser feito com atividades de ordenação, busca ou manipulação de dados.

5. Projetos Práticos:

- **Projeto de Grupo:** Divida os alunos em grupos e atribua um projeto prático que envolva a aplicação dos princípios do Pensamento Computacional. Isso poderia ser a criação de um jogo simples, um aplicativo interativo ou até mesmo uma solução para um problema real, a ser formal no futuro.

Jogos e Quebra-Cabeças:

- **Desafios de Lógica:** Use jogos de lógica e quebra-cabeças para desenvolver a capacidade de resolver problemas. Isso pode envolver jogos de tabuleiro, desafios de código ou problemas de pensamento crítico.

Programação sem Computador:

- **Exercícios Manuais:** Introduza conceitos de programação sem o uso de computadores. Use papel, cartões ou objetos físicos para representar códigos simples. Isso ajuda a compreender a estrutura lógica sem se preocupar com a sintaxe de uma linguagem de programação específica.

Historinhas Interativas:

- **Narrativas de Decisão:** Crie histórias interativas em que os alunos tomem decisões que afetam o curso da história. Isso ajuda a desenvolver habilidades de lógica condicional.

Teatro de Algoritmos:

- Encenação Algorítmica: Peça aos alunos para representarem algoritmos por meio de encenações. Isso torna os conceitos abstratos mais tangíveis e envolventes.

Feedback Construtivo:

- Revisão de Código Simbólico: Peça aos alunos para revisarem o "código" de seus colegas, que pode ser uma representação simbólica de um algoritmo ou lógica de programação.

É importante adaptar as atividades ao nível de habilidade dos alunos e incentivá-los a trabalhar em conjunto para resolver problemas. Ao integrar o Pensamento Computacional nas atividades práticas e projetos, os alunos desenvolverão uma base sólida para aprender programação de forma.

ANEXO – Usando Algotcard para trabalhar os conceitos do PC

BRACKMANN, C. AlgotCards. Disponível em: <www.computacional.com.br>. 2019.

AlgotCards

Algo Movimento

COMPUTACIONAL
Educação em Computação
www.computacional.com.br

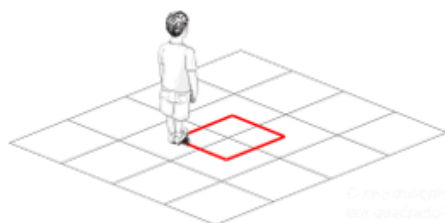
1. MATERIAIS NECESSÁRIOS

- a) Entregar um baralho AlgotCards para uma dupla de estudantes. Para esta atividade serão utilizadas todas as cartas. A função de cada carta será explicada posteriormente. Mais informações sobre o baralho AlgotCards estão disponíveis no site www.computacional.com.br.



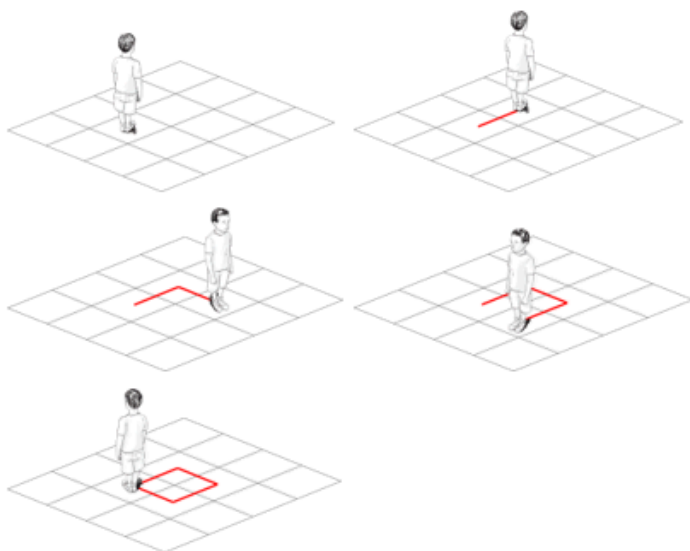
As cartas acima, exceto a última, não podem ser giradas, ou seja, a faixa branca deve estar sempre na parte inferior (conforme ilustração acima). As cartas AlgotCards ECO possuem uma faixa preta na parte inferior.

- b) Recomenda-se que esta atividade ocorra em uma sala, pátio ou outro espaço onde o piso tenha delimitações claras, como pisos cerâmicos, porcelanato, calçadas, placas de concreto etc.
- c) Podem ser usados objetos adicionais que se deseje utilizar na atividade para completar alguma tarefa ou atingir uma meta.
- ### 2. ATIVIDADE: DESENHAR UM QUADRADO
- a) Dividir o grupo de jogadores em duplas;
- b) A dupla deve decidir qual dos dois jogadores será o participante que executará as ações (robô) e quem criará a sequência de cartas (programador);
- c) Como etapa final de preparação, o professor explica quais são as funções das cartas;
- d) O professor então lança o desafio: "desenhar" um quadrado com os movimentos do robô, conforme a ilustração a seguir.



Com o diagrama, tem qual ideia?

- e) Os alunos então devem desenvolver uma sequência de comandos para que o robô percorra todo o trajeto, “desenhando” um quadrado no chão:



Solução:



- f) Quando o robô terminar de desenhar o quadrado, as duplas podem trocar suas funções de robô e programador. O desafio pode ser novamente lançado com uma pequena alteração na sequência, utilizando apenas as cartas “Gire à Esquerda” e “Para Frente” (invertendo o sentido).

Solução:



- g) Até o momento, desenhamos apenas um quadrado tamanho 2x2. Quais alterações seriam necessárias se formos desenhar quadrados maiores de 4x4? Solução:



- h) Que tamanho seria esse quadrado abaixo?



Solução: 9x9

- i) Se quisermos um quadrado muito grande, temos uma alternativa para não usar tantas cartas, conforme as imagens abaixo:



- j) Desafie os alunos para que utilizem as cartas "Repita" e "Parênteses", e assim reduzir a quantidade de cartas da sequência abaixo:








Solução:



- k) Desafie-os a desenhar um retângulo.
- l) Quais outros formatos conseguimos desenhar com as cartas? Ex: sinal de adição (+)


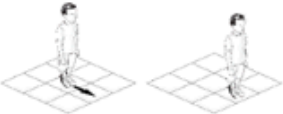

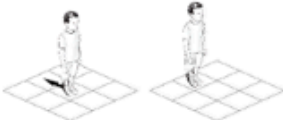





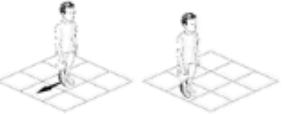

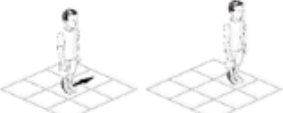
3. OUTRAS SUGESTÕES DE ATIVIDADES


Você pode utilizar as ~~AlgoCards~~ para uma diversidade de atividades, listamos aqui apenas alguns pequenos exemplos:

- a. Fazer a volta em uma mesa; 
- b. Realizar um trajeto em uma sala para ligar uma lâmpada; 
- c. Criar um itinerário no pátio da escola; 
- d. Encontrar um "tesouro" em um quintal; 
- e. Simular o funcionamento de um robô. 





Quais outras atividades você pode utilizar o ~~AlgoCards~~?

4. FUNÇÕES/EFEITOS DAS CARTAS

CARTA	ILUSTRAÇÃO	DESCRIÇÃO
		O estudante dá um passo ou casa para frente na direção em que está apontando.
		O estudante dá um passo ou anda uma casa para trás na direção contrária à que está apontando.
		O estudante muda a direção em seu eixo para a direita (90° à direita).
		O estudante muda a direção em seu eixo para a esquerda (90° à esquerda).
		O estudante dá um passo lateral ou anda uma casa à direita, sem alterar o sentido que está apontando.
		O estudante dá um passo lateral ou anda uma casa à esquerda, sem alterar o sentido que está apontando.

		<p>O estudante muda a direção, em seu eixo, para a direção oposta girando no sentido horário (180° à direita).</p>
		<p>Pode ser utilizado de maneira versátil, pois exerce funções variadas. A ação pode variar conforme a atividade que está sendo trabalhada. Na figura ao lado foram exemplificadas as situações de saltar e bater palmas, porém compreende outras ações como: abaixar, bater com um pé no chão, dar um grito, entre outras.</p>

5. FUNCIONAMENTO DAS CARTAS ESPECIAIS

 	<p>A carta "Repita": a instrução posicionada após o "Repita" será executada a quantidade de vezes que é informada no sinal de multiplicação ("X"). Exemplo:</p>  <p>Esta sequência equivale aos seguintes movimentos:</p>  <ol style="list-style-type: none"> 1. Um passo para frente; 2. Um passo para frente; 3. Um passo para frente; <p>Carta "Parênteses": deve ser utilizada em conjunto com a carta "Repita" e tem a finalidade de agrupar um conjunto de cartas. O</p>
---	---

agrupamento deve sempre ser sinalizado com um “(” no início e “)” no final. Exemplo:



Esta sequência equivale aos seguintes movimentos:



1. Um passo para frente;
2. Um giro à direita;
3. Um passo para frente;
4. Um giro à direita;
5. Um passo para frente;
6. Um giro à direita;
7. Um passo para frente;
8. Um giro à direita.

Atenção! Não esqueça dos parênteses para representar um conjunto de instruções. Veja o exemplo a seguir, onde os parênteses foram esquecidos:



Esta sequência equivale aos seguintes movimentos:



1. Um passo para frente;
2. Um passo para frente;
3. Um passo para frente;
4. Um giro à direita.

ANEXO – Links para consulta (outras atividades)

Cartas Algotcard

<https://www.computacional.com.br/#AlgoCards>

Atividades (Praticando o Pensamento Computacional)

<https://www.computacional.com.br/#atividades>

Sites e softwares poderosos na educação do Século XXI

https://docs.google.com/spreadsheets/d/1kmyggNreluDsJOfVYGgYF_8i01wXrcN_j0f1TqyKCNM/view#gid=1127926244

21 atividades computação desplugada:

<http://desplugada.ime.unicamp.br/>

Projeto Computação Fundamental

<https://porvir.org/cartas-permitem-trabalhar-pensamento-computacional-de-maneira-desplugada/>

REFERÊNCIAS

Almeida, M. E. & Prado, M. E. (2008). **Desafios e possibilidades da integração de tecnologias ao currículo.**

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular: Educação é a Base – Ensino Médio.** Brasília, 2018. Disponível em: http://basenacionalcomum.mec.gov.br/wp-content/uploads/2018/06/BNCC_EnsinoMedio_embaixa_site_110518.pdf

FUNDAÇÃO LEMANN; FUNDAÇÃO TELEFÔNICA. **Aprenda a ensinar programação com o Programaê!** Disponível em: <https://pt.coursera.org/learn/programae>. Acesso em out. 2023.

SICA, C. **Pensamento Computacional: Um relato de práticas pedagógicas para o ensino de computação em Escolas Públicas.** Disponível em: <http://blogs.odiarior.com/carlossica/2011/10/07/ciencia-da-computacao-no-ensinomedio/> Acesso em out. 2023>.

SILVA, V.; SILVA, K.; FRANÇA, R. S. de. **Pensamento computacional na formação de professores: experiências e desafios encontrados no ensino de computação nas escolas.** VI Congresso Brasileiro de Informática na Educação (CBIE 2017); Anais do XXIII Workshop de Informática na Escola (WIE 2017). Disponível em:<<http://www.br-ie.org/pub/index.php/wie/article/viewFile/7299/5097>> Acesso em out. 2023

Tecnologias na educação: ensinando e aprendendo com as TIC. Brasília: Ministério da Educação. Disponível em: <http://portaldoprofessor.mec.gov.br/storage/materiais/0000011621.pdf>

WING, J. **Computational thinking. Communications of the ACM, v. 49, n. 3, p. 33, 2006.** Disponível em: <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>. Acesso em: 23 jul. 2022.