



INSTITUTO FEDERAL
Santa Catarina
Câmpus Jaraguá do Sul – Rau

Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
INSTITUTO FEDERAL DE SANTA CATARINA

Pensamento Computacional: Conceitos e Atividades Desplugadas

Profº Bruno Crestani Calegaro
Profª Raquel Nitsche dos Santos

Instituto Federal de Santa Catarina – Câmpus Jaraguá do Sul - Rau
Rua: dos Imigrantes, 445 | Rau | Jaraguá do Sul /SC | CEP: 89.254-430
Fone: (47) 3276-9600 | www.ifsc.edu.br | CNPJ 11.402.887/0019-90



INSTITUTO FEDERAL
Santa Catarina
Câmpus Jaraguá do Sul – Rau

Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
INSTITUTO FEDERAL DE SANTA CATARINA

2025 Dos autores



Esta obra é disponibilizada nos termos da Licença Creative Commons Atribuição–Não Comercial 4.0 Internacional. É permitida a reprodução parcial ou total desta obra, desde que citada a fonte.

Coordenação Gráfica

Bruno Crestani Calegaro

Capa, Projeto gráfico e Diagramação

Bruno Crestani Calegaro

Revisão Textual

Bruno Crestani Calegaro

Raquel Nitsche dos Santos

Rodrigo de Matos Vieira



SUMÁRIO

1. Prefácio	4
2. Introdução	5
3. Pensamento Computacional	6
3.1 Os Quatro Pilares do Pensamento Computacional	6
1. Decomposição: Dividir para conquistar	6
Tabela 1: Quadro comparativo	8
2. Reconhecimento de Padrões: Encontrando o que se repete	8
3. Abstração: Focando no que importa	9
4. Algoritmos: Criando um passo a passo	9
3.2 Como os pilares se conectam na prática	10
4. Atividades Propostas	12
4.1 Robôs na Fábrica	12
4.2 Desenho em 2D	14
4.3 Cifra de César e o Bingo da Computação	17
5. Conclusão	19
Anexos	20
Anexo 1 - Grade 8x8	20
Anexo 2 - Grade 20x20	20
Anexo 3 - Cifra de Cesar +3	22

1. Prefácio

Este material, intitulado “Pensamento Computacional: Conceitos e Atividades Desplugadas”, foi elaborado como um Recurso Educacional Aberto (REA), resultado das atividades de extensão do Instituto Federal de Santa Catarina (IFSC), sob coordenação do(a) Prof. Bruno Crestani Calegaro e com a colaboração de discentes do Curso Técnico em Desenvolvimento de Sistemas.

O Pensamento Computacional é apresentado nesta apostila de forma gradual. Nas próximas seções, o leitor encontrará uma introdução aos seus conceitos fundamentais, sua aplicação prática na resolução de problemas e a criação de soluções digitais.

A proposta combina fundamentação teórica com a sugestão de algumas atividades desplugadas, voltadas ao ensino de conceitos computacionais sem o uso do computador. Algumas atividades foram planejadas para momentos rápidos de engajamento, enquanto outras podem ser aplicadas em tempos mais longos, incentivando a análise crítica, o trabalho em grupo e a aprendizagem colaborativa.

As atividades foram testadas em turmas do Curso Técnico em Desenvolvimento de Sistemas e em oficinas realizadas em eventos como SEI (Semana da Informática) e SNCT (Semana Nacional da Ciência e Tecnologia), ambos organizados pelo IFSC Campus - RAU. Os resultados indicaram que a metodologia adotada estimula o raciocínio lógico e a criatividade, além de facilitar a compreensão de conceitos complexos de forma lúdica e acessível.

Por se tratar de um REA, esta apostila pode ser livremente utilizada, adaptada e redistribuída por outros professores e instituições de ensino, ampliando seu alcance e possibilitando diferentes aplicações. Embora tenha sido originalmente desenvolvida no contexto do IFSC, sua linguagem acessível e flexibilidade permitem adaptações a distintas realidades educacionais, promovendo inclusão digital e ampliando o acesso ao conhecimento.

Espera-se que este material inspire práticas pedagógicas inovadoras, fortaleça o raciocínio lógico dos estudantes e estimule sua criatividade na resolução de problemas, contribuindo para momentos de interação e aprendizagem significativa entre professores e alunos.

2. Introdução

Os computadores estão em toda parte: no nosso bolso, na nossa casa, no trabalho e até nos carros que dirigimos. Eles transformaram a forma como vivemos, nos comunicamos e resolvemos problemas. Mas você já parou para pensar como essas máquinas funcionam? Ou de que maneira elas conseguem “pensar” e nos ajudar a encontrar soluções cada vez mais rápidas e inteligentes?

É aí que entra o Pensamento Computacional: uma maneira de raciocinar inspirada na lógica dos computadores, que nos ajuda a compreender problemas e criar soluções de forma estruturada, criativa e eficiente.

Esta apostila foi preparada para apresentar o Pensamento Computacional de modo simples e acessível, por meio de atividades desplugadas — ou seja, sem depender do uso do computador. A proposta é mostrar que muitos dos conceitos da Computação podem ser explorados de forma lúdica e colaborativa, usando apenas papel, lápis, objetos do dia a dia e trabalho em equipe.

Você não precisa ser especialista em tecnologia para acompanhar este material. Pelo contrário, aqui vamos mostrar que os fundamentos da Ciência da Computação podem ser aprendidos em qualquer contexto educacional, do ensino médio a cursos técnicos, ou mesmo em oficinas e projetos de extensão.

As atividades propostas estimulam a criatividade, a comunicação, a resolução de problemas e o raciocínio lógico — habilidades essenciais para qualquer área de atuação em uma sociedade cada vez mais tecnológica. Ao longo das próximas páginas, você encontrará explicações introdutórias sobre o que é Pensamento Computacional e uma sequência de atividades práticas para aplicar em sala de aula ou em outros ambientes de aprendizagem.

Cada atividade foi pensada para favorecer a troca de ideias, a colaboração e a construção coletiva do conhecimento, mostrando que aprender sobre algoritmos, padrões, abstração e lógica computacional pode ser divertido, significativo e acessível para todos.

3. Pensamento Computacional

O Pensamento Computacional é uma forma de raciocinar que nos ajuda a resolver problemas de maneira organizada e eficiente, inspirada na forma como os computadores funcionam. Não se trata apenas de saber programar, mas de desenvolver habilidades que permitem analisar situações, encontrar soluções e criar estratégias claras e bem estruturadas.

Quando pensamos de forma computacional, estamos treinando nossa mente para raciocinar logicamente, organizar informações e buscar soluções criativas. É como montar um quebra-cabeça: cada peça precisa ser observada, organizada e encaixada até formar a figura completa.

O Pensamento Computacional não é útil apenas para quem deseja trabalhar com programação ou tecnologia, ele é uma habilidade para a vida, porque nos ensina a lidar com desafios de forma prática e eficiente. Veja alguns motivos:

- **Organiza o raciocínio:** ajuda a dividir problemas complexos em partes menores e mais fáceis de resolver.
- **Promove eficiência:** permite encontrar soluções que economizam tempo, esforço e recursos.
- **Prepara para a programação:** desenvolve habilidades como lógica, sequenciamento e depuração (descobrir e corrigir erros), que são essenciais na criação de programas de computador.

Em resumo, o Pensamento Computacional nos ajuda a pensar como cientistas, agir como engenheiros e criar como inventores.

3.1 Os Quatro Pilares do Pensamento Computacional

O Pensamento Computacional nos ajuda a enfrentar desafios complexos — sejam eles digitais ou do nosso dia a dia — usando quatro ideias principais: **Decomposição**, **Reconhecimento de Padrões**, **Abstração** e **Algoritmos**. Vamos entender cada uma delas com exemplos simples e práticos.

1. Decomposição: Dividir para conquistar

Decomposição significa quebrar um problema grande ou complexo em partes menores. Em vez de tentar lidar com tudo de uma vez (o que pode ser confuso e desorganizado), se divide em partes menores, que são mais fáceis de entender e trabalhar.



Por que usar a decomposição?

- **Facilita a compreensão:** problemas grandes parecem menos assustadores quando quebrados em etapas.
- **Organiza o raciocínio:** cada parte pode ser resolvida separadamente.
- **Favorece a colaboração:** diferentes pessoas podem trabalhar em partes diferentes do problema.
- **Reduz erros:** quando as partes são pequenas, fica mais fácil identificar onde algo não deu certo.
- **Cria soluções reutilizáveis:** às vezes, uma parte resolvida pode ser aplicada em outros problemas parecidos.

Exemplo do dia a dia:

Preparar uma festa de aniversário.

Podemos decompor esse problema em algumas partes:

- Fazer a lista de convidados.
- Escolher o local.
- Comprar comida e bebida.
- Preparar música e decoração.
- Confirmar os detalhes.

Assim, cada tarefa fica mais simples quando tratada separadamente.

Observe a **Tabela 1** mostrando um quadro comparativo entre resolver o problema sem e com decomposição.

Como treinar:

Pegue um problema (ex: organizar uma viagem) e divida em mini-tarefas.



Tabela 1: Quadro comparativo

Sem decomposição	Com decomposição
O problema é visto como um todo e parece confuso.	O problema é dividido em partes menores e claras.
“Preciso organizar uma festa inteira... não sei por onde começar!”	Etapas: 1. Fazer a lista de convidados. 2. Escolher o local. 3. Comprar comida e bebida. 4. Preparar música e decoração. 5. Confirmar os detalhes.
Grande chance de esquecer algo importante.	Cada parte pode ser resolvida separadamente, reduzindo erros.
Difícil pedir ajuda, porque tudo está misturado.	É fácil dividir as tarefas com amigos ou colegas.

2. Reconhecimento de Padrões: Encontrando o que se repete

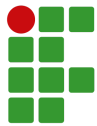
O Reconhecimento de Padrões nos ajuda a encontrar semelhanças entre problemas diferentes. Quando percebemos que uma nova situação tem algo em comum com outra que já resolvemos, podemos reaproveitar a mesma estratégia ou solução em vez de começar do zero.

Por que usar reconhecimento de padrões?

- **Economiza tempo:** não precisamos inventar uma solução nova sempre que o problema aparece.
- **Aumenta a eficiência:** aplicamos métodos já testados e que funcionaram.
- **Facilita a aprendizagem:** quando vemos padrões, conseguimos entender conceitos mais rápido.
- **Ajuda na previsão:** se sabemos como algo funcionou antes, conseguimos prever o que pode acontecer em situações parecidas.

Exemplo do dia a dia:

Quando você aprende a andar de bicicleta, depois fica mais fácil aprender a andar de moto, porque percebe padrões parecidos (equilíbrio, guidão, movimento das rodas).



Como treinar:

Observe padrões em músicas, sequências de cores, ou até na arrumação de objetos em casa.

3. Abstração: Focando no que importa

Abstrair é ignorar detalhes desnecessários e concentrar-se apenas nas informações relevantes para resolver o problema. A Abstração nos permite focar apenas no que é realmente importante para resolver um problema, deixando de lado os detalhes que não fazem diferença, isso evita distrações e torna o raciocínio mais claro e objetivo.

Por que usamos abstração?

- **Reduz a complexidade:** não precisamos lidar com todas as informações ao mesmo tempo.
- **Facilita a solução:** olhando só para o que importa, encontramos respostas mais rápido.
- **Evita sobrecarga de informação:** elimina dados desnecessários.
- **Favorece a generalização:** podemos aplicar a mesma lógica em diferentes situações sem nos perder em detalhes específicos.

Exemplo do dia a dia:

Um mapa do metrô é uma abstração: ele não mostra cada rua ou prédio, apenas as estações e linhas — exatamente o que você precisa para se locomover.

Como treinar:

Descreva um objeto (ex.: uma cadeira) usando apenas 3 características essenciais.

4. Algoritmos: Criando um passo a passo

Um algoritmo é uma sequência lógica de passos para resolver um problema. Algoritmos são como instruções com uma sequência lógica que garantem que um problema será resolvido de forma clara, ordenada e repetível. Sem um algoritmo, a solução pode ficar confusa, incompleta ou diferente a cada tentativa.

Por que usamos algoritmos?

- **Organizam o raciocínio:** colocam as ações em uma ordem lógica.
- **Garantem resultados consistentes:** se seguimos os passos corretamente,



sempre chegamos ao mesmo resultado.

- **Facilitam a comunicação:** qualquer pessoa pode repetir o processo se tiver o algoritmo.
- **Evitam esquecimentos:** não pulamos etapas importantes.
- **Podem ser automatizados:** o computador precisa de algoritmos para executar qualquer tarefa

Exemplo do dia a dia:

Uma receita de bolo é um algoritmo: seguir os passos na ordem correta garante que o bolo ficará pronto do jeito esperado.

Como treinar:

Escreva os passos para trocar uma lâmpada ou outra situação do cotidiano.

3.2 Como os pilares se conectam na prática

Vamos explorar como os pilares do Pensamento Computacional trabalham juntos para resolver problemas de forma eficiente. Usaremos um exemplo simples: organizar uma coleção de livros. Observe como cada pilar contribui para a solução:

1. Decomposição: Dividindo o Problema

Primeiro, quebramos a tarefa complexa de organizar livros em partes menores:

- Separar por gênero literário (fantasia, ficção, didáticos).
- Dentro de cada gênero, agrupar por série ou autor.
- Decidir como dispor fisicamente nas prateleiras.

Assim, cada subtarefa torna-se gerenciável.

2. Reconhecimento de Padrões: Identificando Repetições

Agora, observamos características comuns:

- Livros de uma mesma série geralmente têm cores ou designs similares nas capas.
- Obras de um mesmo autor podem ter logos ou elementos visuais repetidos.
- Livros didáticos geralmente são mais pesados e grossos.

Reconhecer padrões acelera a organização e ajuda a criar critérios consistentes.

3. Abstração: Mantendo o Foco no Essencial

Ignoramos detalhes irrelevantes para este contexto:

- Não importa o número de páginas ou a editora.
- O ano de publicação só é relevante se for um critério de organização (ex.: ordem cronológica).

Focamos apenas nas características-chave: gênero, cor da capa e série.

4. Algoritmo: Criando um Passo a Passo

Com base nas etapas anteriores, elaboramos instruções claras.

Algoritmo para organizar livros:

- Separar todos os livros por gênero principal (fantasia, ficção, didáticos).
- Dentro de cada grupo, ordenar por cor da capa (ex.: do mais claro para o mais escuro).
- Empilhar ou arrumar nas prateleiras seguindo a ordem estabelecida.
- Verificar se livros de uma mesma série estão juntos.

Este exemplo ilustra como os quatro pilares do Pensamento Computacional se integram de forma prática e lógica para resolver um problema cotidiano. Agora que você compreendeu a estrutura, que tal aplicar esse mesmo raciocínio a outro contexto?

- Como você usaria o Pensamento Computacional para planejar uma viagem?
 - Em quais etapas você dividiria o planejamento (decomposição)?
 - Que experiências de viagens anteriores poderiam ajudar (padrões)?
 - Quais detalhes são essenciais para o planejamento (abstração)?
 - Qual seria a sequência lógica de tarefas (algoritmo)?



4. Atividades Propostas

Dentro desta seção serão elencadas atividades para aplicar os conceitos do pensamento computacional, formar um pensamento crítico e levantar discussões sobre diversas temáticas do mundo da computação.

4.1 Robôs na Fábrica

Objetivo:

Os alunos deverão planejar uma sequência de instruções para movimentar um "robô" dentro de uma fábrica fictícia, utilizando uma grade 8x8, para coletar peças e entregá-las no destino correto.

Material Necessário:

- Folha impressa com a grade 8x8 (Disponível no **Anexo 2**).
- Lápis ou caneta.

Instruções:

1. Cada aluno receberá uma folha com a grade 8x8.
2. Na grade, ele deve marcar:
 - a. Um ponto de início (onde o robô começa o percurso).
 - b. Um ponto de destino (onde o robô deve chegar).
 - c. Algumas células representando peças que o robô precisa coletar ao longo do caminho.
 - d. Posicionar obstáculos na grade para que o robô precise pensar em rotas alternativas.
3. Os alunos devem escrever uma sequência de comandos para movimentar o robô até o destino, seguindo as seguintes instruções:
 - a. **F (Frente)**: Move o robô uma casa para frente.
 - b. **E (Esquerda)**: Gira o robô 90° para a esquerda.
 - c. **D (Direita)**: Gira o robô 90° para a direita.
 - d. **P (Pegar Peça)**: Coleta uma peça na posição atual.
4. O percurso precisa ser eficiente, ou seja, o aluno deve tentar utilizar o menor número de comandos possível.

Habilidades Trabalhadas:

- Raciocínio lógico;



- Noção de algoritmos e sequência de instruções;
- Pensamento computacional e resolução de problemas;
- Otimização de soluções.

Considerações Finais:

Essa atividade vai além de apenas “brincar de robô”: ela coloca o aluno no papel de programador, mostrando que qualquer tarefa pode ser resolvida com uma sequência organizada de passos — um algoritmo. Ao criar e testar instruções, o estudante exercita a decomposição (quebrando o desafio em movimentos simples), a abstração (focando apenas nos comandos necessários), o reconhecimento de padrões (percebendo estratégias que podem ser reutilizadas em outros percursos) e a criação de algoritmos (sequência de instruções claras e eficientes).

Estimule os alunos a:

- Comparar suas soluções e discutir qual foi mais eficiente;
- Experimentar diferentes rotas, entendendo que muitas vezes há mais de uma resposta correta;
- Refletir sobre como esse mesmo raciocínio pode ser aplicado em problemas do dia a dia, dentro e fora da computação.
- Assim, a atividade ajuda a desenvolver não apenas habilidades técnicas, mas também criatividade, colaboração e pensamento crítico, elementos essenciais do Pensamento Computacional.

Resultados da aplicação da atividade:

As Figuras 1 e 2 apresentam os resultados da aplicação da atividade em uma das oficinas de pensamento computacional.

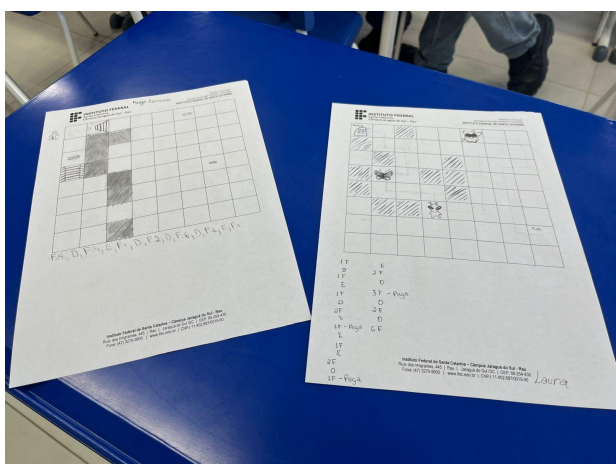


Figura 1. Solução A

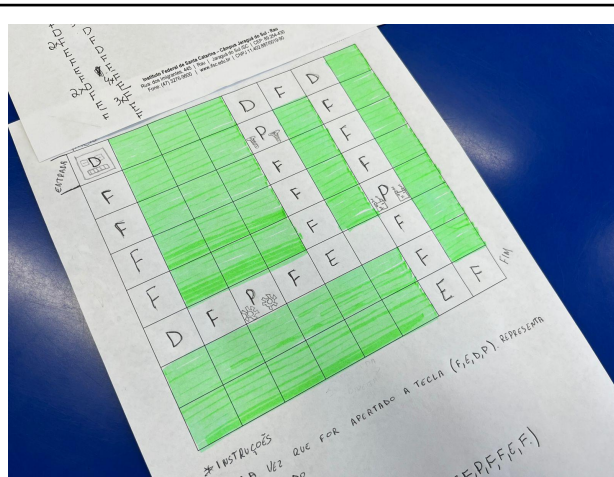


Figura 2. Solução B

4.2 Desenho em 2D

Objetivo:

Ensinar os princípios de precisão algorítmica e *debugging* por meio da execução de instruções para pintar uma grade 20x20. O exercício simula erros de programação em um contexto visual, permitindo comparar resultados corretos e incorretos.

Materiais Necessários:

- Folha impressa com grade 20x20 (Disponível no **Anexo 2**)
 - Linhas e colunas numeradas de 1 a 20.
- Lápis preto ou caneta.
- Lista de instruções de pintura.

Instruções:

Cada aluno receberá uma grade de 20x20, onde cada célula funciona como um “pixel”. As linhas são numeradas de cima para baixo (1 a 20) e as colunas da esquerda para a direita (1 a 20).

Seguindo as instruções fornecidas, o aluno deverá preencher os quadrados indicados para revelar uma imagem oculta (uma caneca soltando fumaça).

- Exemplo de coordenada:
 - “Linha 5, Colunas 8 a 13” → pintar da coluna 8 até a 13 na linha 5.

Habilidades Trabalhadas:

- Precisão na leitura e execução de instruções.
- Noção de algoritmos como sequência de comandos.
- *Debugging* (identificação e correção de erros).
- Pensamento lógico e atenção aos detalhes.

Passos da Atividade

1. Contorno da Caneca

a. Boca da caneca:

- i. Linha 7, Colunas 3 a 16 → Pintar tudo.

b. Laterais:

- i. Linhas 8 a 13, Colunas 3 e 16 → Pintar apenas essas duas



colunas em cada linha.

c. Base:

- i. Linha 15, Colunas 5 a 14 → Pintar um retângulo largo.
- ii. Linha 14, Colunas 4 e 15 → Unir a base às laterais.

2. Alça

- a. Linha 8, Colunas 17 a 19
- b. Linha 12, Colunas 17 e 18
- c. Linhas 9 e 11, Coluna 19
- d. Dica: a alça está do lado direito da caneca, formando uma curva.

3. Fumaça

- a. Linha 5, Coluna 6
- b. Linhas 3 e 4, Coluna 7
- c. Repetir o mesmo padrão começando pelas posições Linha 5, Colunas 10 e 14.
- d. Dica: a fumaça sobe em três “tracinhos” acima da caneca.

Considerações Finais:

Essa atividade mostra que, assim como em programação, pequenas imprecisões nas instruções podem gerar resultados completamente diferentes do esperado. Ao seguir o passo a passo e depois comparar o desenho com o modelo correto, o aluno exercita o debugging, identificando onde errou e como corrigir.

Além disso, o exercício reforça a importância da precisão algorítmica — cada coordenada é um comando que precisa ser executado de forma exata para que a imagem final faça sentido. A prática também ajuda a desenvolver paciência, atenção e raciocínio lógico.

Essa atividade pode ser enriquecida com a entrega de quatro versões diferentes das instruções: em uma delas a caneca sai perfeita, mas nas outras aparecem erros — como a base fora do lugar, a alça mal encaixada ou partes faltando. Organize os alunos em grupos de quatro e entregue versões distintas, sem revelar quem tem a correta.

O objetivo é estimular a reflexão: será que segui as instruções corretamente? Se o resultado não corresponde ao esperado, o erro está na execução ou nas próprias instruções? Esse momento de dúvida gera um espaço importante para discutir a importância da precisão algorítmica e, principalmente, para exercitar o pensamento crítico.

Depois de identificar os erros, desafie os alunos a pensar em soluções:

- Corrigir o desenho seguindo a versão correta;
- Criar novas instruções para “consertar” o problema;
- Adaptar o resultado para que ainda faça sentido, mesmo diferente do esperado.

Dessa forma, os estudantes percebem que muitas vezes não existe apenas um

caminho para resolver um problema. O mais importante é analisar a situação, propor alternativas e testar soluções criativas — exatamente como acontece na programação e em desafios do dia a dia.

Resultados da aplicação da atividade:

As Figuras 3 e 4 representam os resultados da atividade em uma Oficina de Pensamento Computacional.



Figura 3. Oficina de Pensamento Computacional

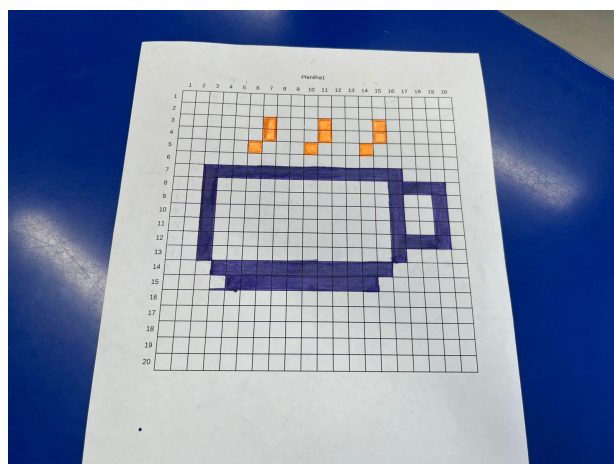


Figura 4. Resultado original

4.3 Cifra de César e o Bingo da Computação

Objetivo:

Desenvolver raciocínio lógico, atenção a padrões e abstração através da decodificação de mensagens usando cifras simples, simulando algoritmos de criptografia básica.

Material Necessário:

- Folhas com tabelas da Cifra de César (+3) e cartelas para o jogo do Bingo da Computação (Disponível no **Anexo 3**)
- Cartões com mensagens codificadas (níveis variados).
- Lápis e papel para anotações.

Descrição da Cifra de César:

A Cifra de César foi usada por Júlio César para enviar mensagens secretas ao exército romano. Ela 'empurra' cada letra do alfabeto um número fixo de posições para frente.

Como Funciona (Exemplo com Rotação +3):

Letra Original	A	B	C	D	...	Z
Letra Codificada	D	E	F	G	...	C

Demonstração Prática:

1. Use como exemplo a palavra: "OVO".
2. Aplique a rotação +3:
 - a. $O \rightarrow R$
 - b. $V \rightarrow Y$
 - c. $O \rightarrow R$
3. Resultado: "RYR".

Habilidades Trabalhadas:

- Pensamento lógico e analítico: compreender a lógica por trás da criptografia e da substituição de caracteres.
- Resolução de problemas: aplicar estratégias para decodificar mensagens e



identificar padrões.

- Reconhecimento de padrões: perceber relações entre letras deslocadas e reconstruir o texto original.
- Trabalho em equipe e comunicação: os alunos precisam discutir hipóteses e validar respostas em grupo.

Considerações Finais:

Esta atividade permite que o aluno explore, de maneira prática e lúdica, um conceito clássico da Computação: a criptografia. A Cifra de César é um exemplo simples de como a informação pode ser transformada seguindo uma regra lógica — o que está diretamente relacionado à forma como os computadores processam e protegem dados.

A aplicação prática da atividade pode ser incrementada com a participação num jogo como o Bingo da Computação. Os alunos são organizados em grupos, onde cada grupo recebe uma cartela com palavras comuns, enquanto o professor sorteia palavras codificadas. Os alunos precisam decifrar cada código usando o deslocamento da Cifra de César e verificar se a palavra decodificada aparece em sua cartela. O primeiro grupo que completar a cartela grita “Bingo” e recebe um brinde, como uma caixa de bombom.

A dinâmica transforma o conteúdo em um jogo de raciocínio: o desafio não é apenas descobrir o código, mas fazê-lo mais rápido e com precisão, aplicando estratégias como testar deslocamentos, reconhecer padrões e compartilhar hipóteses dentro do grupo.

O momento de socialização após o jogo é ideal para discutir questões como:

- Qual foi a estratégia mais eficiente para decifrar as mensagens?
- Houve tentativas erradas? Como o grupo percebeu e corrigiu o erro?
- Em que situações do mundo real a criptografia é usada?

Essa reflexão ajuda os alunos a entenderem que o pensamento computacional não se limita ao ambiente da programação — ele está presente em processos de comunicação, segurança digital e na própria forma como solucionamos problemas no cotidiano.

5. Conclusão

Ao longo desta apostila, exploramos o Pensamento Computacional como uma forma de raciocínio que nos ajuda a compreender e resolver problemas de maneira estruturada, lógica e criativa. Vimos que ele não é apenas sobre programar computadores, mas sobre pensar como um cientista da computação, identificando padrões, decompondo desafios, abstraindo informações e criando soluções passo a passo.

As atividades propostas — Robôs na Fábrica, Desenho em 2D e Cifra de César — foram cuidadosamente selecionadas para ilustrar, de forma lúdica e colaborativa, como é possível desenvolver o raciocínio computacional sem a necessidade de dispositivos eletrônicos.

Essas experiências reforçam que pensar computacionalmente é mais do que saber programar — é desenvolver uma forma de raciocínio aplicável a qualquer área do conhecimento. Seja organizando processos em uma fábrica, criando imagens a partir de instruções ou decifrando mensagens secretas, os alunos exercitam habilidades essenciais para o século XXI: lógica, colaboração, pensamento crítico e autonomia intelectual.

Cada uma das atividades propostas convida o aluno a experimentar os quatro pilares do Pensamento Computacional de maneira concreta:

- **Decomposição:** ao dividir tarefas e planejar etapas;
- **Reconhecimento de padrões:** ao identificar semelhanças e repetições;
- **Abstração:** ao focar apenas nas informações realmente importantes;
- **Algoritmos:** ao criar sequências claras de ações para alcançar um objetivo.

Por fim, este material busca demonstrar que a computação pode — e deve — ser aprendida de forma desplugada, com papel, lápis e muita imaginação. Afinal, quando o computador é desligado, o que permanece aceso é o raciocínio. É ele que torna o aprendizado verdadeiramente significativo, duradouro e transformador.

Orientações ao Professor

Para o professor, este material pode servir como ponto de partida para explorar novas metodologias ativas de ensino. As atividades podem ser adaptadas para diferentes faixas etárias, níveis de ensino e contextos educacionais — desde o ensino médio e técnico até oficinas de extensão e projetos interdisciplinares.

Incentive os alunos a refletirem sobre o processo de resolução, não apenas sobre o resultado final. Promova momentos de discussão em grupo, permitindo que eles expliquem suas estratégias, comparem soluções e proponham melhorias. Esse diálogo favorece o pensamento crítico, a autonomia e o trabalho colaborativo.

Além disso, o professor pode expandir as propostas aqui apresentadas com novos desafios, variações nas regras, ou atividades que conectem o Pensamento Computacional com outras áreas, como matemática, física, artes e até ciências humanas.



INSTITUTO FEDERAL
Santa Catarina
Câmpus Jaraguá do Sul – Rau

Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
INSTITUTO FEDERAL DE SANTA CATARINA

Anexos

Anexo 1 - Grade 8x8



INSTITUTO FEDERAL
Santa Catarina
Câmpus Jaraguá do Sul – Rau

Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
INSTITUTO FEDERAL DE SANTA CATARINA

Anexo 2 - Grade 20x20

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1																				
2																				
3																				
4																				
5																				
6																				
7																				
8																				
9																				
10																				
11																				
12																				
13																				
14																				
15																				
16																				
17																				
18																				
19																				
20																				



Anexo 3 - Cifra de César +3

CIFRA DE CÉSAR ROTAÇÃO +3																									
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Exemplo de Cartela

Bingo da Computação - Cartela 1

PANELA	CASA	JANELA
LIVRO	PRATO	BANANA
CADEIRA	QUADRO	ESCOLA