



PENSAMENTO COMPUTACIONAL

⌚ Created	@April 11, 2022 10:32 AM
➤ Nível de dificuldade	Fácil
☰ Professores	Juliana Mascarenhas

▼ DEFINIÇÃO

- ° Trazer soluções de problemas em passos computacionais (algoritmos).
- ° Habilidade;
- ° Sistemático e eficiente;
- ° Formação e resolução de problemas (tanto homens quanto máquinas entendem).

▼ PILARES

- **Decomposição**: dividir um problema em “probleminhas”;
- **Reconhecimento de padrões**: identificar padrões/tendências (similares ou diferentes);

- **Abstração**: leva um problema p/ o generalista (ex.: a equação da circunferência serve para todas as circunferências);
- **Design de algoritmos**: automatizar (passo a passo).

| **Processo contínuo**: Definir soluções → Testar → Melhorá

▼ **COMPETÊNCIAS**

- Pensamento sistemático
- Colaboração dentro da equipe
- Criatividade e design
- Facilitador

HABILIDADES COMPLEMENTARES

▼ **RACIOCÍNIO LÓGICO**

- ° Pensamento estruturado que ajuda a encontrar a solução de algo.
- ° Treino.
- **Indução**: observar e criar leis/teorias;
- **Dedução**: pegar as leis/teorias e tentar explicá-las;
- **Abdução**: gerar premissas (conclusões preliminares).

▼ **INFERÊNCIA**

Sintética: Abdução e Indução

Analítica: Dedução

▼ **APERFEIÇOAMENTO**

- Melhorar, aprimorar;
- Pontos de melhoria.
- *Encontrar soluções eficientes;*
- *Otimizar processos;*
- *Simplificar linhas de códigos;*
- *Funções bem definidas.*

- Processo contínuo e presente em toda as etapas!

PILARES DO PENSAMENTO COMPUTACIONAL

▼ **DECOMPOSIÇÃO**

- **Análise**: estudar, examinar detalhadamente. Quebrar o problema em partes menores.
- **Síntese**: reconstrução dessas partes de maneira coerente.
- **Ordem de execução de tarefas**:
 - Sequencial: Dependência entre tarefas. Uma “fila”.
 - Paralelo: execução concomitante das tarefas (ganho de eficiência e tempo).

Ex.: [Cozinhar](#)

- ° Identificar ingredientes;
- ° Determinar as etapas (sequencial ou paralelo);
- ° Execução;
- ° Agregar p/ finalizar (com coerência).

OBS.: Um problema pode ser resolvido de várias maneiras!

▼ **RECONHECIMENTO DE PADRÕES**

- ° Similaridades/diferenças
- ° *Padrão*: modelo base, invariante

Ex.: [Compressão de dados](#)

####** → 3# e 2*

Importância: generalizar p/ obter resoluções de problemas variados!

- **Como as máquinas reconhecem padrões?**

→ **Comparações**

- **Aplicações:**

1. Reconhecimento de imagem/fala;
2. Classificação de dados;
3. Análise de cenas;
4. I.A e M.L

▼ **ABSTRAÇÃO**

Abstrair: observar elementos individualmente.

Abstração: transformar algo concreto em abstrato e usá-lo em “n” situações.

Generalizar: tornar amplo.

- **Como classificar dados?** → Abstraia os detalhes

- Características
- Pontos essenciais
- Generalizar x Detalhar

- **Aplicações:**

1. Algoritmos
2. Linguagens de programação
3. Comunicação (Redes, Processadores de computadores)
4. Cliente/Servidor, Ponto-a-Ponto e etc.

▼ **ALGORITMOS**

Processamento de dados: PC recebe, manipula e armazena dados. Os programas tem instruções.

- O que fazer? → instruções simples
- Qual a ordem? → instruções simples

- **Desenvolvimento de um programa:**

- **Análise:** dados (entrada/saída) definidos;

- **Algoritmo:** descrição do problema por ferramentas narrativas, fluxogramas ou pseudocódigo;
 - **codificação:** linguagem específica.
- **Como construir algoritmos?**
 - Compreenda o problema
 - Definir dados de entrada
 - Definir processamento
 - Definir dados de saída
 - Método de construção
 - Teste e diagnóstico