# Introdução ao pensamento computacional

## O que é pensamento computacional?

-Pensamento computacional é uma habilidade generalista, raciocínio lógico, a capacidade de formular e resolver problemas pensando na mesma lógica que uma máquina “pensaria”.

-Entrada: informação importante para resolver um problema

-Processamento: tomadas de decisões que transformam a entrada em saídas

-Saída: resultado do processamento das entradas

-Pensamento computacional: Capacidade de desenvolver algoritmos capazes de resolver problemas de modo que um humano e um computador sejam capazes de entender a solução

-Algoritmos: analogia a uma receita de bolo, um passo a passo estruturado com começo meio e fim organizado logicamente para alcançar um resultado.

## Baseia-se em 4 pilares:

### Decomposição:

Dividir um problema complexo em subproblemas.

Executar os problemas menores em ordem sequencial o em paralelo para então reconstruí-los.

### Reconhecimento de padrões:

Identificar padrões ou tendências, similaridades e diferenças entre os problemas.

### Abstração:

Extrapola o conceito dos problemas para uma forma generalista. análise elementos que têm relevância, diferenciando-os daqueles que podem ser deixados de lado

Desenho animado para crianças

Descrição gerada automaticamente com confiança média

Conceitos de merge sort, clusteting, busca binaria.

### Design de algoritmos:

Automatizar, definir passo a passo a solução do problema. Entendimento por humanos e máquinas.

São processos contínuos onde definimos a solução, testamos, e aperfeiçoamos a mesma. Teste 🡪 análise 🡪 Refinamento.

## O que é raciocínio lógico:

É uma forma de pensamento estruturado que permite encontrar a conclusão ou determinar a resolução de um problema.

### Classificado em 3 tipos:

-Indução (leis e teorias, empirismo, ciências experimentais), fenômeno observado.

-Dedução (a partir de leis e teorias faz previsões e explica, ciências exatas)

-Abdução (premissas, diagnóstico, processo investigativo)

inferência pode ser:

-Sintética: abdução ou indução

-Analítica: dedução

## Algoritmos:

São sequências de passos com objetivo definido, execução de tarefas especificas, conjunto de operações que resultam em uma sucessão finita de ações.

Compreender o problema (pontos importantes)

Dados de entrada (dados fornecidos e cenário)

Processamento (cálculos e restrições)

Dados de saída (após o processamento)

Métodos de construção (construção e refinamento do algoritmo)

-Narrativa, linguagem natural;

-Fluxograma, estrutura gráfica;

-Pseudocódigo, exemplo Portugol.

## Lógica:

técnica linear: modelo tradicional, não tem vínculo, estrutura hierárquica, matemática, execução sequenciada, ordenação de elementos por uma única propriedade, recursos limitados, única dimensão. Resolução em linha.

técnica estruturada: organização e disposição dos elementos essenciais que compõem um corpo (concreto ou abstrato), processamento de dados com objetivo (escrita, entendimento, validação, manutenção). Resolução em paralelo ou por escolha.

técnica modular: partes independentes controlada por um conjunto de regras, metas (simplificação, decompor o problema, verificação do módulo), regras, módulos.

## Conceitos de programação:

função do computador: processar informações, compostas por dados e instruções. Dados são tratados e processados, dados numéricos (inteiros, reais), caracteres e lógicos (0 e 1, true e false)

variável é um tipo de estrutura mutável que pode variar seu valor, inconstante, contudo, ela fica restrita ao seu tipo. (pense numa caixinha dentro de um armário)

papel da variável: ação (modificação de estado), controle (vigiada, controle)

Constante, é uma invariável, não muda, como exemplo uma variável que recebe o valor de PI, é sempre o mesmo

instruções determinam as ações a serem executadas, geralmente cálculos matemáticos, operadores unários ou binários

instruções primitivas, operadores, manutenção de sinal +, inversão de sinal -, exponenciação, divisão, multiplicação, adição e subtração

Estrutura condicional: suposição ou hipótese, satisfeita resulta em operação, inverdade resulta em exceção

ela pode ser simples (apenas verifica se foi satisfeita) composta (possui exceção) ou encadeada (possui várias verificações)

operadores lógicos (AND, OR, NOT), resposta simplificada de verdadeiro ou falso

-AND = interseção

-OR = união

-NOT diferença

estrutura de repetição, laços, controle de fluxo, malha de repetição, loop

precisa de uma condição de parada, condição a ser satisfeita ou número de repetições pré-definidas

enquanto...faça, repita...até, para...de...até...faça

estruturas condicionais: simples (se então) composta (se então senão) encadeada (se então senão se então senão)

vetores e matrizes: um vetor é caracterizado por uma variável dimensionada com tamanho pré-fixado), matriz é uma tabela organizada em linhas e colunas, matriz é uma coleção de vetores

pilha, lista, arvore, grafo, matriz.........

funções: blocos de instruções que realizam tarefas especificas, decomposição de algoritmos, reaproveitamento de código, modularização do problema

nomes e parâmetros

tipos de saída de dados: saída programada, saída por interrupção, bem-sucedida, erros de sintaxe ou outro, erros de programação, problemas com a interface

## Curiosidade e definições.

- Instrumento computacional é diferente de computador

Um programa é um amontoado de palavras, existe um processo de tradução ou compilação, uma linguagem de alto nível (programa fonte) passa por um compilador (executa análise do programa), então linguagem de máquina/Assembly (programa objeto)

tradução: geração do programa objeto, execução do programa objeto

interpretação: programa fonte executado diretamente

Python é uma linguagem interpretada e temos o interpretador puro, por isso que ele é mais lento

boas práticas em programação: legibilidade( fácil leitura, compreensão, ortonalidade, definição adequada de estruturas) , regidibilidade ( ortogonalidade, simplicidade da escrita, suporte a abstração, reuso do código, expressividade), confiabilidade( verificação de tipos, trata exceções, uso de ponteiros, compatibilidade entre compiladores), baixo custo(treinamento, codificação, compilação, execução, infraestrutura, análise de impacto), atualizações, uso para IA, disponibilidade de ferramentas, comunidade ativa, adoção pelo mercado

análises de código: análise léxica, análise de sintaxe, análise de semântica

análise léxica: particionar (identificar elementos e agrupá-los), classificar, eliminar (aquilo que não é necessário, exemplo: espaço branco)

análise sintática: forma que ele define qualquer estrutura relacionada a codificação, depende da linguagem de programação

análise semântica: significado, palavras sinais e símbolos

Paradigma: forma de resolução de problemas com diretrizes e limitações especificas de cada paradigma utilizando linguagem de programação...

orientada a objetos, procedural (chamadas sucessivas e procedimentos separados (ideia de sequência)), funcional (instruções baseadas em funções), estruturado (estrutura de blocos aninhados), computação distribuída (funções executadas de forma independente), lógico

estruturado: sequência, decisão, iteração... linguagem C

POO: objeto tem características, comportamento e estado (o que tenho, sou capaz de fazer, como faço), Classe(classificação) em MODELO (atributos métodos e estados)

pilares da POO: herança, encapsulamento, poliformismo, abstração