INTRODUÇÃO AO PENSAMENTO COMPUTACIONAL

O pensamento computacional deve ser sistemático e eficiente. E deve ser capaz da formulação e resolução de problemas.

OS 4 PILARES DO PENSAMENTO COMPUTACIONAL:

1. **Decomposição:** consiste em pegar um problema complexo e segmentá-lo em problemas menores e atacar um ponto por vez. (Modularização). Exemplo do bolo: massa, recheio e cobertura. E você faz ambos separados. E só tem uma ordem na hora de montar a estrutura.
2. **Reconhecimento de padrões:** identificar similaridades e tendências.
3. **Abstração:** extrapolar o conceito do problema para uma forma generalista.
4. **Design de algoritmo** define o passo a passo da solução do problema.

LEMBRE-SE:

A RESOLUÇÃO DE UM PROBLEMA é um processo contínuo. Onde você sempre vai estar se perguntando:

- Será que eu achei a melhor solução?
- Testar nova solução.
- Aperfeiçoamento da solução encontrada.

Portanto, será sempre um ***Ciclo Virtuoso***: onde você precisa: Testar ➡ Analisar ➡ Refinar
O Pensamento Computacional nos permite utilizar as habilidades humanas e os recursos computacionais. Pois, é o ser humano que ensina o computador como ele deve resolver determinado problema.

Veja as funções de cada:

Habilidades Humanas: Abstração (resolução de problemas) e Automatização (expressão de solução).
Recursos Computacionais: Análise (execução da solução e avaliação (refinamentos).

Competências que vem através do utilização do pensamento computacional:

- ➔ Pensamento sistemático (analítico);
- ➔ Colaboração dentro da equipe;
- ➔ Criatividade e design;
- ➔ Facilitador.

HABILIDADES COMPLEMENTARES

As habilidades necessárias para desenvolver o pensamento computacional são: Raciocínio Lógico e Aperfeiçoamento.

- RACIOCÍNIO LÓGICO: é uma forma de pensamento estruturado, ou raciocínio, que permite encontrar a conclusão ou determinar a resolução de um problema.

OBS: o raciocínio lógico precisa ser treinado para ficar mais fácil.

CLASSIFICAÇÃO DO RACIOCÍNIO LÓGICO:

- INDUÇÃO : Relacionada com leis e teorias, pois você induz que algo acontece a partir de uma observação. Ex. Ciências experimentais.

- DEDUÇÃO : a partir de leis e teorias você prova que A + B chega em determinado resultado. Você precisa provar. Ex ciências exatas .

- ABDUÇÃO: você parte de uma premissa de observação e consegue supor um fato. Ex "a grama está molhada, logo deve ter chovido".
OBS: Premissa de que choveu, mas nesse exemplo, a grama pode estar molhada por outro motivo. Muito utilizada na área dos detetives (Peralta).

DECOMPOSIÇÃO OU MODULARIZAÇÃO

Dado um problema complexo, devemos quebrá-lo em problemas menores. Portanto, problemas mais fáceis e gerenciáveis.

A decomposição precisa ser treinada. Pois, é necessário treinar maneiras distintas de decompor o mesmo problema.

Exemplo 1: Cozinhar

I. Identifique os ingredientes.
II. Determina as etapas.
III. Executar cada etapa.
IV. Adicione os ingredientes para finalizar.

Exemplo 2: Criar um app

I. Finalidade: para que vai servir o app;
II. Interface;
III. Funcionalidades: o que o usuário vai ter de funções disponíveis nesse app.
IV. Pré requisitos.

PADRÕES

Reconhecimento de padrões é necessário:

- Um modelo base (a base vai ser sempre a mesma, ex: Uma cadeira, ela sempre vai ter quatro pernas e pés, o que vai mudar é a aparência dela)

- Estrutura Invariante.

- Repetição.

Exemplo: redes sociais que comprimem as fotos e as disponibilizam no perfil das pessoas que as

postam.

Por que devemos determinar padrões?

Porque você consegue generalizar, com o objetivo de obter resolução para problemas diferentes.

Como os padrões são determinados?

Através de classes e categorias.

Exemplo: os seres humanos sabem que determinado ser vivo é um inseto, baseado nas características dele.

Como um computador reconhece padrões?

Ele reconhece por comparação. Ele precisa ter a informação para determinar a que categoria se refere o objeto.
O PC armazena os dados para consultas posteriores.

Resumindo:

No reconhecimento de padrões temos uma extração de características a fim de classificar os nossos dados.

ABSTRAÇÃO

Abstrair, um ou mais elementos avaliando características e propriedades em separado. - Abstração é um processo intelectual de isolamento de um objeto da realidade. - Generalizar é tornar algo geral, mais amplo.

- Abstrair = Generalizar.

Como classificar os dados?

Através de características, pontos essenciais e generalizar. Pois, para classificar os dados é necessário excluir os detalhes irrelevantes e só deixar os pontos essenciais.

ALGORITMOS

COMO É O DESENVOLVIMENTO DE UM PROGRAMA?

1. ANÁLISE: onde você vai estudar e entender o que você precisa inserir no programa. E se precisa ter valor de saída.
2. ALGORITMO: Descreve o problema por meio de ferramentas narrativas ou fluxograma.
3. CODIFICAÇÃO: onde o algoritmo é codificado de acordo com a linguagem de programação escolhida.

Portanto, teremos:

- Sequência de passos com objetivo definido.
- Execução de tarefas específicas.

- Conjunto de operações que resultam em uma sucessão de finitas ações.

Exemplo simples e prático de algoritmo que envolve um passo a passo: preparar um sanduíche ou trocar uma lâmpada.

ALGORITMO são instruções executadas passo a passo para concluir a tarefa.

COMO CONSTRUIR UM ALGORITMO?

1. Compreender o problema
2. Definir dados de entrada
3. Definir quais operações serão realizadas dentro do algoritmo (cálculos e restrições)
   Definir dados de saída
4. Utilizar um método de construção e refinamento de algoritmo.

CONSTRUÇÃO DE ALGORITMOS ENVOLVE:

★ Narrativa;
★ Fluxograma;
★ Pseudocódigo.

O QUE É A LÓGICA?

Ela vem para ajudar a solucionar um problema. É uma forma de raciocínio. É uma forma de aprender a pensar de maneira objetiva.

TÉCNICAS DE LÓGICA DE PROGRAMAÇÃO

Segue a lógica e determinam as instruções.

TÉCNICA LINEAR: é uma técnica que pressupõe a relação linear entre as características do problema, buscando a solução ótima para o problema estudado. Essas características do problema são representadas e relacionadas por meio de uma série de equações lineares.

Exemplos do dia a dia: acordar, caminhar até a cozinha, preparar o café e tomar.

Portanto, uma sequência de ações a serem executadas de maneira ordenada e que possuem dependência entre si.

TÉCNICA ESTRUTURADA: é uma técnica de programação, independente da linguagem de programação, que tem como objetivo construir programas claros, legíveis, eficientes e de fácil manutenção.

As partes da técnica estruturada são basicamente: Escrita ➡ Entendimento ➡ Validação ➡ Manutenção.

OBS: A escrita é feita no programa escolhido e a manutenção busca facilitar.

A parte de estruturação não é linear, portanto ela pode ter mais de uma opção., como no exemplo do dia

a dia:
Acordar, caminhar até a cozinha, fazer um café OU fazer um suco, tomar o café da manhã.

TÉCNICA MODULAR: consiste em uma técnica de design de softwares em que as soluções digitais são formadas por módulos independentes e intercambiáveis. Dentro dessa estrutura, cada módulo funciona como um bloco de instruções à parte dispondo de tudo que necessita para cumprir com a sua própria função.

O seu modelo padrão é composto por: Dados de Entrada ➡ Processo de transformação ➡ Dados de saída.

Com a técnica modular conseguimos a simplificação do algoritmo e da solução do problema. Pois, conseguimos decompor o problema em subproblemas menores. E fazer a verificação diretamente no módulo específico.

Exemplo dia a dia: Módulo preparar para acordar, Módulo preparar bebida e Módulo tomar café. Cada

módulo terá suas regras definidas.