**Pensamento Computacional**

Desenvolver a habilidade de resolver problemas de maneiras mais eficientes.

**Aula 1: Visão Geral**

Pensamento computacional é muito atribuído a essência da computação. Mas não é exclusivo da computação, pode ser aplicado em outras áreas.

É um processo de pensamento estruturado que envolve a expressão de um problema e um passo a passo de como solucionar esse problema. Esse passo a passo deve ser lido por um computador.

É formulação e resolução de problemas. É uma habilidade generalista e não uma disciplina acadêmica.

A resolução de problemas deve ser lida por pessoas e por computadores, deve ser entendida por ambos.

**Baseado em 4 pilares:**

1. Decomposição – Segmentar o problema de maneira que ele se torne mais fácil de ser resolvido. Dividir um problema em subproblemas;
2. Reconhecimento de padrões – Identificar padrões ou tendências dentro de determinados problemas ou problemas distintos. Similaridades e diferenças entre os problemas;
3. Abstração – Explorar o conceito do problema para uma forma generalista. Pegar uma particularidade e extrapolar de tal forma que se torne generalista. Tirar uma ideia do concreto e levar pro abstrato. Conseguir pegar algo particular e generalizar;
4. Design de algoritmos – Automatizar. Definir passo a passo a solução do problema. Há uma “entrada” e uma “saída”, entre elas há operações para que encontre a saída;

É um processo contínuo: define-se a solução, testa e aperfeiçoa. Quanto mais eficiente for o algoritmo, mais rápido encontra a solução e há menos custos. Processo de teste > análise > refinamento.

Há uma troca entre habilidades humanas e recursos computacionais.

Competências desenvolvidas:

* Pensamento sistemático;
* Colaboração dentro da equipe;
* Criatividade e Design;
* Facilitador (facilita o desenvolvimento);

Além desses 4 pilares, é necessário o raciocínio lógico e o refinamento.

**Raciocínio Lógico:**

É uma forma de pensamento estruturado, ou raciocínio, que permite encontrar a conclusão ou determinar a resolução de um problema. Encontrar soluções para problemas que parecem ser difíceis, mas são simples. Ela é desenvolvida através de treinamento.

É classificado em três tipos:

Indução – Dado o fenômeno que estou observado, vou extrapolar ele pro campo das ideias e elaborar leis e teorias (baseado no que é experimentado). Usado muito por ciências experimentais.

Dedução – É inversa a indução. Dado uma lei ou teoria, vou realizar deduções, previsões e explicações baseadas na lei. Usado em ciências exatas.

Abdução – Utiliza a conclusão para determinar a premissa. “A grama esta molhada, logo deve ter chovido.” Pode acarretar em erros, já que pode estar molhada por outros motivos. Usado em diagnósticos, áreas mais investigativas.

A inferência pode ser sintética ou analítica. Na área sintética, temos abdução e indução. Na parte analítica, fica a dedução.

**Aperfeiçoamento:**

Consiste em melhoramento, aprimoramento, refinamento. A partir de uma solução, identificar pontos de melhora e refinamento. Encontrar umasolução eficiente, otimizar processos, simplificar linhas de código e funções bem definidas.

É um processo contínuo, presente em todas as etapas.

**Aula 2**

**Decomposição:**

“Se você tem um problema que não consegue resolver, deve existir um problema mais fácil que você consiga: encontre ele.”

É o primeiro passo em pensamento computacional. Dado um problema complexo, devemos quebra-lo em problemas menores. Portanto, problemas fáceis e gerenciáveis.

Primeira fase é chamada de análise. Estudar e explorar o problema de forma que consiga decompor os elementos e realizar uma análise detalhada.

Segunda fase é a síntese. Combinar os elementos recompondo o problema original. Processo de reconstrução, reunir elementos de maneira que faça sentido.

A estratégia na ordem de execução de tarefas menores pode ser sequencial ou paralela.

A decomposição nos da a possibilidade de detectar variáveis e transformar em pequenos problemas. É necessário aprender a desenvolver a decomposição por si mesmo.

Como decompor? Deixa o problema principal no segundo plano e foca nos componentes. Quais os aspectos fundamentais do problema para que eu consiga resolve-lo? Há dependência entre eles?

Para criar um app, por exemplo, a decomposição começa em definição de componentes e etapas. Qual a finalidade do app? Qual a interface? Quais as funcionalidades e os pré-requisitos necessários para que o app funcione?

**Reconhecimento de padrões:**

Há um modelo base, uma estrutura invariante e uma repetição dessa estrutura invariante. Através de similaridades e diferenças se define qual esse modelo.

Através do reconhecimento dos padrões, há a possibilidade de fazer a compressão de dados.

O reconhecimento de padrões é algo inerente ao ser humano, estamos sempre reconhecendo padrões das coisas que já conhecemos.

Por que determinar padrões? Generalizar, com o objetivo de obter resolução para problemas diferentes.

Realizamos a classificação dos objetos a partir de classes e categorias conhecidas.

O ser humano reconhece padrões por grau de similaridade e grupos conhecidos ou desconhecidos. Isso se da por comparação.

Como o computador reconhece padrões? É necessário a representação de atributos. Ele precisa entender o que significa, qual o conceito associado ao objeto. Armazenar os dados e a partir disso determinar regras de decisão (baseado em tudo o que foi aprendido). Essas regras são determinadas pelo programador.

O reconhecimento de padrões parte de uma abordagem onde é extraído as características de um elemento, classifica os dados e aplica em contextos diferentes.

*Aplicações:*

Classificação de dados, reconhecimento de imagens, reconhecimento de fala, análise de cenas, classificação de documentos.

No geral, classificação, reconhecimento e análise.

Muito utilizado em Machine Learning, Redes Neurais, Inteligência Artificial e Ciência de dados. Tem possibilidade de aplicação bem ampla.

**Abstração:**

Abstração é o ato de abstrair/generalizar. Observar diferentes elementos e avaliar as características individuais. É um processo intelectual de isolamento de um objeto da realidade. Tornar algo geral, amplo, sendo possível de ser utilizado em várias situações.

Generalização, na lógica, é uma operação intelectual que consiste em reunir numa classe geral, um conjunto de seres ou fenômenos similares.

Como classificar os dados?

Características e pontos essenciais. Cada objeto tem sua peculiaridade. Deixar de lado os detalhes menos importantes e focar em pontos mais importantes.

Assim conseguimos uma representação do objeto.

Na representação de dados é preciso focar nos pontos essenciais. Elimina os detalhes desnecessários.

Existem diversos conceitos computacionais baseados em abstrações:

Algoritmos, estrutura de dados, máquinas de estado, linguagens de programação.

**Algoritmos**:

Principal pilar dentro do pensamento computacional.

O principal objetivo do computador é receber, manipular e armazenar dados. Ele realiza essas atividades através de programas que recebem instruções e as passam para o computador. Ele precisa que essas instruções sejam passadas para ele.

O algoritmo vem de um processo de resolução de problemas “step by step” utilizando instruções. O que precisa ser feito e qual a ordem de execução? Ele precisa ser entendido por um humano e por uma máquina.

Dentro do desenvolvimento de um programa existem 3 etapas:

* Análise – estudo e definição dos dados de entrada e saída.
* Algoritmo – descreve o problema por meio de ferramentas narrativas, fluxograma ou pseudocódigo.
* Codificação – O algoritmo é codificado de acordo com a linguagem de programação escolhida.

É uma sequência de passos com um objetivo definido. A execução de tarefas específicas e o conjunto de operações que resultam em uma sucessão finita de ações. Não há subjetividade com o computador. É necessário instruções executadas passo a passo para concluir uma tarefa.

Como construir um algoritmo?

Compreender o problema, deixar de lado detalhes desnecessários que não influenciam no resultado. Definir os dados de entrada, definir o processamento (quais cálculos, operações serão realizadas), definir os dados de saída (caso espere algum dado), utilizar um método de construção e realizar testes e diagnósticos. O método ainda não é a codificação. Depois há o refinamento do algoritmo.

*Métodos utilizados:*

* Narrativa – utiliza a linguagem natural com a escrita. Não carrega nenhum conceito novo. Ele pode acarretar em diversas interpretações e esse é o problema desse método.
* Fluxograma – utiliza símbolos já pré-definidos. É de simples entendimento mas precisa de conhecimento prévio da estrutura e dos símbolos.
* Pseudocódigo – Portugol. Parece linguagem de programação, mas usa a linguagem natural. Há regras definidas e passos a serem seguidos.

**Aula 3: Exemplificação**