**Pensamento Computacional –** Juliana Mascarenhas

**Modulo 1: Visão geral do Pensamento Computacional**

**Aula 01: Visão Geral**

**-** Pensamento computacional: Refere-se ao processo de pensamento envolvido na expressão de soluções em passos computacionais ou algoritmos que podem ser implementados no computador.

- Pensamento computacional: Formulação e resolução de problemas / Sistemático e Eficiente / Humanos e maquinas devem ser capazes de resolver.

- Pensamento computacional é uma habilidade generalista de forma escrita, leitura, matemática

- Baseado em 4 Pilares:

Decomposição: Dividir um problema complexo em subproblemas

Reconhecimento de padrões: Identificar padrões ou tendencias (Similaridades e ou diferenças entre os problemas)

Abstração: Extrapolar o conceito do problema para uma forma generalista

Design de algoritmos: Automatizar – definir passo a passo a solução do problema (In put – Operator – Output)

- É necessário o processo contínuo ou seja: Definir uma solução – Testar a solução – Aperfeiçoamento da solução encontrada.

Refinamento – Teste – Analise

- Habilidades Humanas: Abstração (Resolução de Problemas); Automatização (Expressão de solução)

- Recursos Computacionais: Analise (Execução da solução e Avaliação (refinameno))

- Além dos 4 pilares também é necessário o Raciocínio Logico e o Refinamento (eficiência da solução)

- Competências que trazem este estudo: Pensamento Sistemático; Colaboração dentro da equipe; Criatividade e design; Facilitador.

Aula 02: Habilidades Complementares

!º Habilidade - Raciocínio Logico é uma forma de pensamento estruturado, ou raciocínio, que permite encontrar a conclusão ou determinar a resolução de um problema.

nos permite encontrar soluções para problemas que parecem ser difíceis, mas são simples. Habilidade adquirida através de treinamento.

- Classificação do Raciocínio Logico: Indução: Leis e teorias (Fenômeno Observado/ tentativo e erro) – Ciências Experimentais

Dedução: Leis e Teorias – (Previsões e explicações/ provar uma lei que já foi formulada) Ciencias exatas.

Abdução: Premissa – (Processo investigativo, Diagnostico) Conclusão

- Essas Classificação pode ser Sintética (Construção) que é a Indução e a Abdução; ou pode ser Analítica que é a Dedução.

2º Habilidade – Aperfeiçoamento: a partir de uma solução determinar pontos de melhora e refinamento

- Ato de aperfeiçoar: Encontrar soluções eficiente e Otimizar processos – Melhor uso de recursos.

Simplificar linhas de códigos e Funções bem definidas – Melhorar códigos e algoritmos

- Aperfeiçoamento é um processo continuo, presente em todas as etapas.

Vantagens são: Maior compreensão; Mais eficiência; Menos tempo; Menos armazenamento

Desenvolver solução – Aperfeiçoar.

**Modulo 2: Pilares do Pensamento Computacional**

**Aula 01: Decomposição**

- Decomposição primeiro passo da resolução de problemas dentro do conceito de pensamento computacional.

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

- Estrategia para Decomposição:

1º fase: Analise: Processo de quebrar e determinar partes menores e gerenciáveis. (Estudar, explorar, realizar exame detalhado, decompor em elementos constituintes)

2º fase: Síntese: Combinar os elementos recompondo o problema original (Processo de reconstrução, Fundir os elementos de maneira coerente, Consiste em reunir elementes distintos em um único elemento)

- Ordem de execução de tarefas menores dentro da estratégia:

Sequencial: dependência entre tarefas; executadas em fila

Paralelo: tarefas podem ser executadas concomitantemente. (mais eficiência e menos tempo)

- Decomposição da as seguintes ferramentas: variáveis, pequenas problemas e segmentação.

- É necessário desenvolver a decomposição “by yourself” (Maneiras distintas de decompor o mesmo problema)

Como Decompor? Entenda o problema (Componentes) Identifique (Características e dependências) – Deixe o Problema Principal em 2º plano

- Identificar ou coletar dados – Agregar os dados – Funcionalidade: Decomposição

- Para a decomposição em sistemas é necessário entender o funcionamento do sistema: Identificar os componentes – Papel de cada componente – Interdependência de cada peça

- Decomposição para criar um APP: Definição de componentes e etapas: Finalidade – Interface – Funcionalidade – Pré-requisitos (Desenvolvimento mais eficiente)

**Aula 02: Reconhecimento de Padrões**

- Reconhecimento de padrões: Modelo base; Estrutura Invariante; Repetição

- Reconhecimento de padrões determina similaridades e diferenças

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

- Como um computador reconhece padrões?

Os homens reconhecem pelo Grau de Similaridade; Grupos conhecidos x objetos desconhecidos.

A maquina utiliza: representação de atributos; aprendizado (conceito associado ao objeto); armazenar dados; regras de decisão

- Abordagem utilizada: extração de características e classificação de dados. Diferentes métodos e aplicações.

- A aplicação de reconhecimento de padrões se dá: Classificação de dados; Reconhecimento de imagem; Reconhecimento de fala; Analise de cenas; Classificação de documentos. E também muito utilizado em: Machine Learning; Redes Neurais; Inteligência Artificial; Ciência de dados

**Aula 03: Abstração**

**-** Abstrair: Observar, um ou mais elementos, avaliando características e propriedades em separado.

- Abstração: Processo intelectual de isolamento de um objeto da realidade.

- Generalizar: Tornar-se, mais amplo extensão

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

- Como classificar os dados: Características; Pontos essências; Generalizar x detalhar. = Representação de dados.

- Conceitos baseados em abstração: Algoritmos; Estrutura de dados; Maquina de Estado Finito; Linguagem de programação; Comunicação; Arquiteturas.

**Aula 04: Algoritmos**

- Uma máquina tem energia, é trabalhadora, é eficiência e tem rapidez, contudo não opera sozinho, precisa de instruções detalhadas.

- O computador recebe (instruções), manipula e armazena dados.

- Algoritmo (Instruções): Processo de resolução de problemas ‘step by step’ utilizando instruções.

- As instruções determinam o que precisa ser feito e qual a ordem de execução. De maneira a ser entendido por um humano e por uma máquina.

Desenvolvimento de Programa:

1º- Análise: Estudo e definição dos dados de entrada e saída (Quais sãos os dados que irão compor o problema, quais os dados de saída que espero receber)

2º- Algoritmo: Descreve o problema por meio de ferramentas narrativas, fluxograma ou pseudocódigo (instruções detalhadas para serem utilizados pelos programas)

3º- Codificação: O algoritmo é codificado de acordo com a linguagem de programação escolhida.

- Algoritmo é uma sequencia de passos com objetivo definido

- Algoritmo é uma execução de tarefas especificas (o computador não entende subjetividade)

- Conjunto de operações que resultam em uma sucessão finita de ações

- Como construir um algoritmo?

Compreensão do problema (Pontos mais importantes)

Definição dados de entrada (Dados fornecidos e Cenário)

Definir processamento (Cálculos e Restrições)

Definir dados de saída (Após processamento)

Utilizar um método de construção (Construção e refinamento do algoritmo) Ainda não é um programa, precisa ser codificado.

Teste e Diagnostico (Construção e refinamento do algoritmo)

- Construção de algoritmos:

Narrativa: Utilização da linguagem Natural (Ponto Negativo: Diversas interpretações possíveis; Ponto Positivo: Sem conceitos novos)

Fluxograma: Utilização de símbolos pré-definidos (Ponto Negativo: Conhecimento Prévio da Estrutura e Símbolos; Ponto positivo: Simples Entendimento)

Pseudocódigo: Portugol (Ponto positivos: Regras definidas e Passos a Serem Seguidos)

-