Aqui daremos uma base sobre o mundo do TI e como entrar, é um curso para iniciante.

**Pensamento computacional** processo de pensamento envolvido na expressão de soluções em passos computacionais ou algoritmos que podem ser implementados no computador. Baseado em 4 pilares:

**decomposição** separa um problema complexo em problemas curtos, passos, que devem ser seguidos para resolver o problema ao todo.

**reconhecimento de padroes** Identificação de padrões e tendências, similariedades e diferenças.

**Abstração** Extrapolar o conceito do problema para uma forma generalista, utilizar e outro conhecimento a fim de resolver problemas similares.

**design de algoritmo** automatização da solução do problema, define o passo a passo que será seguido para a resolução.

Refinamento 🡪 testagem 🡪 analise

--

Habilidades complementares são especialmente o **raciocínio lógico** e **aperfeiçoamento**.

**Raciocinio lógico** se refere a uma forma de pensamento estruturado para determinar a solução de um problema, não é algo simplesmente ensinado, mas **treinado** pensa na memória muscular. Vem através da **indução**, proveniente da observação e estudo, **dedução**, apartir de teorias se compõem explicações par um fenômeno e a **abdução**, onde apartir de um conclusão se supõe outra informação como por exemplo dizer que a grama esta molhada, e apartir dessa informação assumir que choveu.

**Aperfeiçoamento**, encontrar a maneira mais eficiente e precisa para otimizar um processo, no nosso caso, simplificar o código, limpar o código, melhorar o uso de recursos, etc.

--

Como dito anteriormente, o primeiro passo para a resolução de um problema computacional é a **decomposição**. A estratégia aqui, é quebrar e determinar o problema em espaços menores, através de uma **análise**, um exame detalhado do que deve ser feito, após feito essa análise, faz-se uma **síntese** para combinar esses elementos e os encaixarem de forma coerênte e dando o sentido à solução. A ordem das tarefas também pode ser feito aqui, o que deve ser feito primeiro, se pode ser feito de maneira pararela, etc.

Dentro da composição temos as variáveis presentes nesses problemas pequenos, as segmentações, etc.. o mesmo problema pode ter soluções diferentes, sempre havendo uma maneira mais eficiente de fazer isso.

**Identificar e coletar dados 🡪 agregar os dados 🡪 funcionalidade**

--

**Padrões** são as formas comuns de resolver problemas que podem ser utilizados para diversas questões. A compreensão de dados também pode ser vista através dos padrões, utilizando de similariedades para chegar a conclusões claras. Determinamos padrões naturalmente, fazemos isso para generalizar problemas para facilmente achar soluções através de problemas anteriores. Diferente do humano, que possui sua subjetividade para procurar padrões, a maquina precisa de dados precisos e objetivos para procurar padrões, feito através da linguagem. Padrões são muito usados em **redes neurais**, **machine learning**, **ciencia de dados** e **inteligência artificial**.

--

**Abstração** é observar um ou mais elementos, avaliando suas caracteristicas e propriedades de forma separada, um processo intelectual de isolamento do objeto da realidade, para generaliza-lo e avalia-lo.

Na lógica, a generalização nos permite ver as classes e subclasses de determinado objeto, para abri-lo e entende-lo. Na abstração a gente se debate com uma questão importante: o **generalizar x detalhar**. Nem todas as informações vão ser relevantes para o problema, então retiramos aquilo que não é importante para o trabalho e generalizamos o resto. As proprias linguagens de programação são exemplos de abstrações, utilizando de diversas informações generalizadas para solucionar o problema.

--

**Algoritmos** são os programas, softwares que armazenam e utilizando as instruções do que o computador vai fazer para solucionar o problema. O algoritmo faz sentido tanto para **a pessoa quanto para a maquina**. A construção envolve as seguintes questões:

A **narrativa** é a linguagem natural que vai ser usada para a escrever a solução.

O **fluxograma** são os simbolos pré-definidos com significados específicos para a solução do problema.

O **pseudocódigo**, o Portugol, a pseudolinguagem de aprendizado utilizado para fazer código.

--

Estudos de caso foram feitos, 3. Alguns dos próximos tópicos trazem coisas bem repetidas de outros cursos, vale a pena assistir, mas não quero anotar as mesmas coisas com outras palavras :^3

--

A **lógica** para nós é a **organização e planejamento de instruções, assertivas em um algoritmo, a fim de viabilizar a implementação de um programa**. É a sequencia de ações necessárias para alcançar um objetivo.

--

**Técnicas de lógica** são as diversas técnicas e processos usados para formar a lógica do logaritmo, seja de forma **estruturada**, **direta** ou **modular**.

--

**Qual a função do computador**? Processar as informações que passamos para ele, os dados e instruções que geram a infomação. Tais dados são tratados e processados, podendo ser **numéricos**, **caracteres** e **código**.

**Variavel** é um tipo de estrutura mutavel inconstante, podendo receber determinado valor de acordo com as especificações feitas pelo programador, podendo ser uma **string**, **booleano**, **caracteres**, e . A variavel pode ser de **controle** ou de **ação**.

--

Instruções primitivas são os **operadores**, instruções particulares e fechadas que só podem ser utilizados de forma específica, como + ou -.

--

**Estrutura condicional** é uma estrutura que só faz determinada instrução se satisfeita, por exemplo, se um numero é maior que outro, então a instrução sera realizada, se não, não haverá instrução. Também é possível ter uma instrução que faz ou uma ou outro instrução, essas são o if e else que conhecemos.

--

**Estruturas de repetição** é os chamados **loops**, eles geram uma redução de linhas e mantem o código mais limpo, geralmente funcionam como **while**, **até** e **de a até**.

--

Existe algumas coisas que não estão escritas aqui, relacionadas a vetores e matrizes, funções e instruções de saida/entrada, escolhi não escrever nada, mas recomendo ver o video caso em duvida

--

Há bastante historia do funcionamento de hardwares e softwares no início dessa parte.

--

um programa é caracterizado pela sua **legibilidade**, a compreensão do código, **regibilidade**, a facilidade de escrita do código, **confiabilidade**, se o código o faz o que propõe fazer, **custo**, a utilização de recursos do código e sua compatibilidade com diversos hardware, além desses, **atualizações**, **IA**, **disponibilidade de ferramentas**, **atividade de comunidade** e a adoção para o mercado.

--

O programa faz a leitura do código utilizando a análise sintática de cada linguagem de programação.

--

Paradigma estruturado ajuda a auxiliar os problemas simples e a aprender programação.

A **programação orientada para o objeto** é um grande foco da TI, para o POO (OOP) tudo é um **objeto**, algo descrito por caracteristicas específicas, comportamentos e estados. Existem também as chamadas classes mães ou superclasses, classes cujas caracteristicas são herdadas para outras classes, chamadas de classes filhas. A POO foca em reuso de código, utilizando de varias classes que se complementam através de herança.

--

**Portugol** Conhecido já conhece na verdade, MAS agora tem-se na web o Portugol! Por que não soube disso antes? T>T <https://portugol-webstudio.cubos.io/ide>