# Bootcamp DIO

## Pensamento Computacional

Decomposição:

Segmentação do problema a problemas menores e resolvíveis, ou seja, trabalhar com um problema em partes, tornando a questão mais simples e eficiente.

Reconhecimento de padrões:

Compreensão de ideias dentro de qualquer coisa ou situação. Acontece bastante dentro de dados.

Abstração:

Trazer uma ideia do mundo real para um mundo lógico ou das ideias. Como por exemplo pegar um círculo e gerar a partir dele a fórmula geral de cálculo da área de um círculo, podendo aplicá-la em qualquer círculo.

Design de Algoritmos:

Operações executas a partir de uma entrada que geram um resultado, ou seja, é um programa que gera alguma saída, que pode ser esperada ou não.

Formas de treiná-lo: análise -> refinamento -> teste -> análise -> refinamento ....

Raciocínio lógico:

Forma de pensamento estruturado ou raciocínio, que auxiliam na resolução de um problema.

Indução: observação leva a uma lei geral

Dedução: a partir de uma lei, vc prova que algo é verdade com comparação.

Abdução: de uma conclusão nasce uma premissa.

Aperfeiçoamento:

A partir de uma solução, você encontra pontos de melhora para refinar a sua resposta.

Encontrar soluções eficiente, otimizar processos, simplificar linhas de códigos, funções bem definidas.~

## Pilares:

Decomposição:

Dado um problema complexo, devemos quebrá-lo em problemas menores. Portanto, problemas fáceis de resolver.

Estratégia:

Análise do problema e olhar qual o objetivo.

Síntese/combinação do elementos para verificar se a solução do problema faz sentido.

Como decompor?

Junção e análise dados, agregação dos dados e funcionalidade.

Padrões:

* Modelo base
* Estrutura invariante
* Repetição

Como reconhecer padrões: através de similaridades e diferenças.

O computador reconhece padrões por comparação

Abstrair: Observar um ou mais elementos avaliando características e propriedades em separado

Abstração: processo intelectual de isolamento de um objeto da realidade

Generalizar: torna-se geral, mais amplo, extenso. Operação intelectual em reunir numa classe geral, um conjuntos de seres e fenômenos.

Como classificá-los:

* Características:

Exemplos - nome, idade etc.

Nem tudo será relevante para classificação.

* Pontos essenciais:
* Generalizar x detalhe:

Algoritmos

Instruções detalhadas sobre uma rotina para que um computador possa realizar tarefas.

Instruções step by step.

Construção:

Compreensão do problema

Definição dos dados de entrada

Definir o processamento

Definir dados de saída

Utilizar um método de construção

Realizar testes e diagnósticos

Lógica:

Organização e planejamento das instruções, assertivas em um algoritmo, a fim de viabilizar a implantação de um programa.

Técnicas

Linear:

execução sequenciada de uma série de operações

Recursos limitados, uma única dimensão.

Ordenação de elementos por uma única propriedade.

Estruturada:

Organização, disposição e ordem dos elementos essenciais que compõem m corpo (concreto ou abstrato).

Modular:

Regras estabelecidas, para um objeto, para que algo aconteça.

Simplificação

Decompor o problema

Verificação do módulo

## Fundamentos de algoritmos

Tipologia e variáveis:

As informações são compostas por dados e instruções. Esse dados podem ser números, caracteres e lógico (booleano – verdade/false).

Variável: Um tipo de estrutura, que pode variar e é inconstante. No final ela recebera um dados ou valor, mas não tem certeza do seu valor.

# Git & GitHub

Git:

Serve para criar versões de código, auxiliando no monitoramento do mesmo e facilitando a vida dos desenvolvedores. (software de versionamento de código).

GitHub:

Repositório online (para guardar códigos online).

Benefícios de ambos:

- Controle de versão

- Armazenamento em nuvem

- Trabalho em equipe

- Melhorar seu código

- Reconhecimento

### CONCEITOS FUNDAMENTAIS DO GIT

* SHA1:

Algoritmo de hash seguro (conjunto de criptografia) para garantir a segurança de códigos.

Gera um conjunto de caracteres de 40 dígitos.

* Objetos fundamentais:

- Blobs:

São metadados.

O objetivo blob possui o tipo de um objeto, o tamanho e o conteúdo do arquivo.

Contém o sha1 do arquivo.

- Trees:

Armazenam blobs, podendo apontar para vários tipos de blobs ou para outras árvores.

Também possui metadados.

A árvore guarda o blobs, o sha1 (próprio) e o nome do arquivo.

- Commits:

Aponta para uma árvore, parente (o último commit realizado antes dele), autor, mensagem.

A mensagem serve para dar sentido no momento de dados dentro dele.

Também possui um sha1 de seus metadados.

Alterar o conteúdo do blobs, muda o sha do blobs, o sha da árvore (e qualquer caminho de árvores que ela apontar) e o sha do commit.

- Sistema distribuído/Segurança:

Garante que todas as versões do código (sejam na nuvem ou na máquina de pessoas que estão trabalhando no código em uma máquina remota) sejam confiáveis.

Chaves SSH e Tokens

* Chaves SSH: Forma de estabelecer uma conexão segura e confiável com uma máquina.

Sempre vai existir uma chave pública e privada. É necessário passar essa chave para o git.

* Token: utiliza como uma senha pessoal.

Comandos GIT:

- git init:

Tracked: Unmodified – não foi modificado

Modified – foi modificado, isso é verificado a partir da comparação do sha. Depois da alteração ele vai para o Stage.

Staged – onde ficam os arquivos que irão fazer parte de um outro agrupamento.

Untracked: arquivo que não existia para o git, podendo ser um arquivo que foi excluído.

- git add: joga o arquivo para o Stage (git add nome do arquivo, git add \*)

- git commit: joga do Stage para dentro do git e viram commit, colocando ele no unmodified.

Repositório: o git commit faz o arquivo sair do working space e movido para o repositório local, a partir disso ele pode ir para o repositório remoto.

### Github