Digital Innovation One – bootcamp Become Remote



**TREINAMENTO 1: LÓGICA DE PROGRAMAÇÃO ESSENCIAL**

**Bloco 1: Introdução à lógica de programação**

**Entendendo a lógica**

- Lembrar que programar é resolver problemas!!

- “Abstração” é a habilidade de concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes

**Algoritmos e pseudocódigo**

- Algoritmo é uma sequência de passos que resolve um problema

Início-dia

Acordei

Levantei da cama

Troquei de roupa

Escovei os dentes

Fui à padaria

Tomei café

...

Deitei na cama

Dormi

Fim-dia

- O “pseudocódigo” é uma forma genérica de escrever um algoritmo, utilizando uma linguagem simples

**Fluxogramas e variáveis**

- Fluxograma é uma ferramenta utilizada para representar graficamente o algoritmo, isto é, a sequência lógica e coerente de dados

- Na programação, uma **variável** é um objeto (uma posição, frequentemente localizada na memória), capaz de reter e representar um valor ou expressão

- As **constantes** são valores imutáveis e não são alterados ao longo da vida do programa

**Tomadas de decisões e expressões**

- As **expressões literais** são expressões com constantes e/ou variáveis que tem como resultado valores literais; as usamos na atribuição de valor para uma variável ou constante

- As **expressões relacionais** são compostas por outras expressões ou variáveis numéricas com operadores relacionais (retornam valores lógicos verdadeiro/falso)

- Grande parte da programação é feita a partir das **tomadas de decisão**.... Geralmente, quando escrevemos um programa acontece a necessidade de decidir o que fazer dependendo de alguma condição encontrada durante a execução

**Concatenação**

- Este é um termo utilizado em computação para designar a operação de unir o conteúdo de duas strings

OU

- Agrupamento de duas ou mais células que, incluindo fórmulas, textos ou outras informações contidas no seu interior, dá origem a um **único resultado**

**Bloco 2: Introdução ao Portugol**

**Estrutura de repetição**

- O que é? É uma estrutura que permite **executar mais de uma vez** o mesmo comando ou conjunto de comandos, de acordo com uma condição ou um contador

**Linguagens de programação e Portugol**

- Linguagem de Programação é uma linguagem escrita e formal que especifica um conjunto de instruções e regras usadas para gerar programas (software)

- Um software pode ser desenvolvido para rodar em um computador, dispositivo móvel ou em qualquer equipamento que permita sua execução

- As linguagens servem como um meio de comunicação entre computadores e seres humanos

**Alto nível x Baixo nível**

- As linguagens de **alto nível** são aquelas que a sintaxe se aproxima mais da nossa linguagem e se distancia mais da linguagem de máquina, ex.: Python, C#, C++...

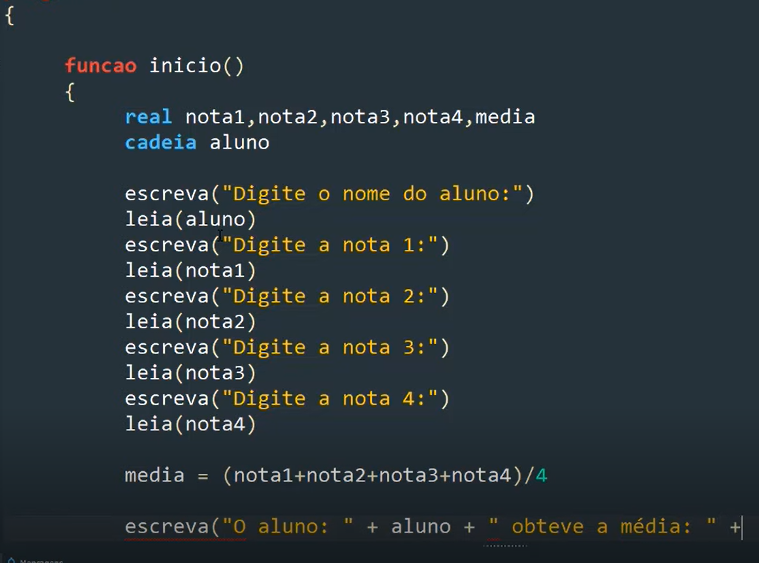
- Já as de **baixo nível** são as que a sintaxe se aproxima mais da linguagem de máquina (necessário ter o conhecimento direto da arquitetura do computador para fazer alguma coisa), ex.: Asembly

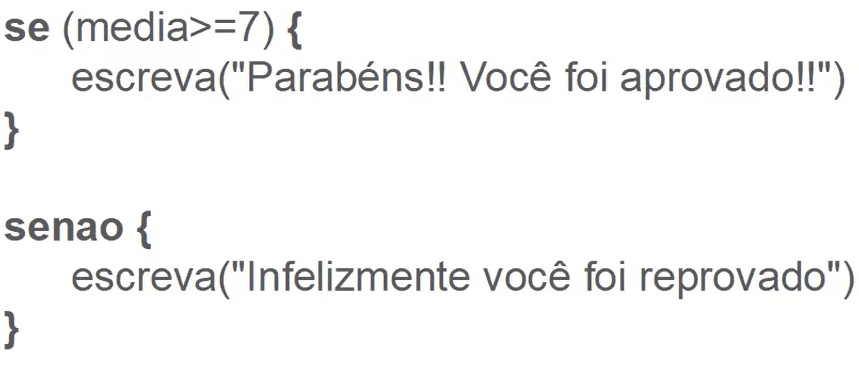
**Compiladas x Interpretadas**

- As **compiladas** são as linguagens em que o código fonte é executado diretamente pelo sistema operacional ou pelo processador, após ser traduzido por meio de um processo chamado compilação, ex.: C#...

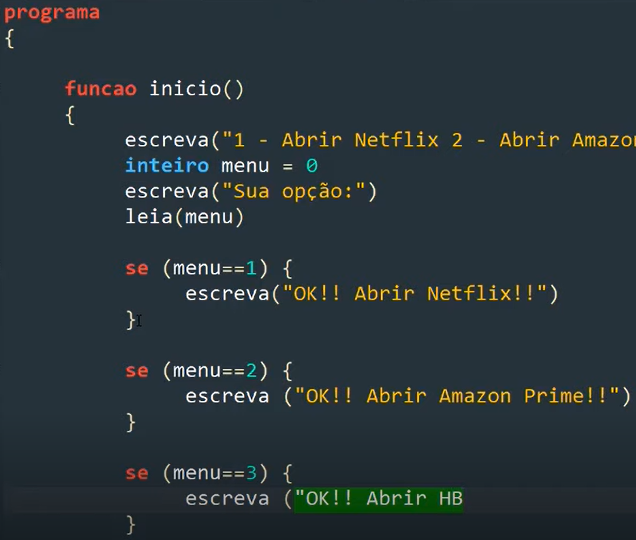
- As **interpretadas** são linguagens em que o código fonte é executado por um programa de computador chamado interpretador que em seguida é executado pelo sistema operacional, ex.: JavaScript, PHP, R, Python...

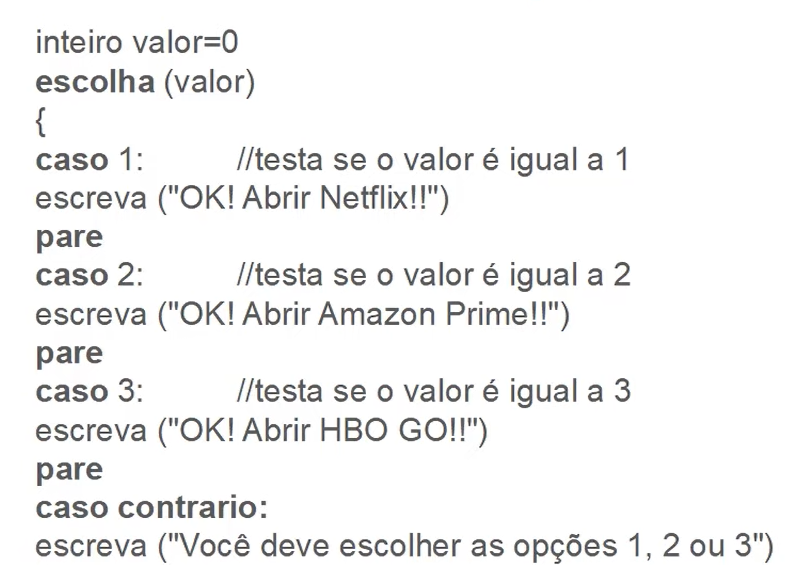
**Portugol...**





- No Portugol, comentário se inicia com //





**TREINAMENTO 2: ESTRUTURA DE DADOS E ALGORITMOS**

- O que é uma **estrutura de dados**?

É uma estrutura organizada de dados na memória do pc ou em qualquer dispositivo de armazenamento, de forma que os dados possam ser utilizados de forma correta

- As estruturas encontram muitas aplicações no desenvolvimento de sistemas, sendo que algumas muito especializadas e utilizadas em tarefas específicas (como mecanismo de busca e análise de bancos de dados)

- **Algoritmo** é um conjunto de instruções estruturadas e ordenadas, seu objetivo é realizar uma tarefa ou operação específica

- As **principais estruturas de dados** utilizadas em desenv. de sistemas são:

Vetores e matrizes (arrays)

Registro

Lista

Pilha

Fila

Árvore

Tabela Hash

Grafos

**Arrays**

- São estruturas de dados simples que podem auxiliar quando há muitas variáveis do mesmo tipo em um algoritmo

- O **vetor (array uni-dimensional)** é uma variável que armazena várias variáveis do mesmo tipo; é uma estutura indexada que pode armazenar uma determinada quant. de valores

- \t dá espaço, enquanto \n pula linha

- A **matriz (array multidimensional)** é um vetor de vetores, possuindo duas ou mais dimensões

**Registros**

- É uma estrutura que oferece um formato especializado para o armazenamento de dados

- Diferentemente dos arrays, o registro permite armazenar mais de um tipo de dado

- Toda estrutura de registro tem um nome (ex: livro) e seus campos podem ser acessados por meio do uso do operador ponto, ex.:

Para acessar o preço de um livro, poderíamos utilizar a declaração **livro.preco**

**Listas**

- Armazenam dados de um determinado tipo e numa ordem específica

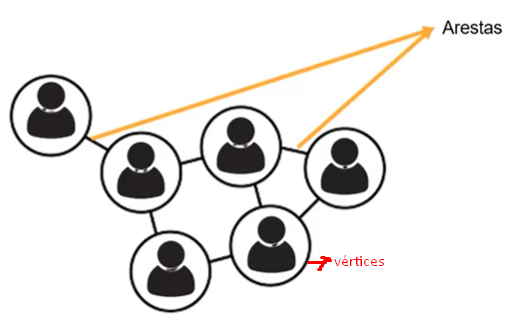
- Mas qual a diferença para os arrays? É que as listas possuem ajustável, enquanto os arrays possuem um tamanho fixo

- Existem dois tipos de listas: ligadas e duplamente ligadas

- Aqui não anotei PILHA, ÁRVORES E TABELAS HASH (tabela hash me lembra dicionários em pytho, porque associa uma chave a um valor)

Grafos

- Estruturas que permitem programar a relação entre objetos (objetos são vértices ou nós do grafo)



**TREINAMENTO 1: INTRO AO GIT E GITHUB**

- Controle de versão

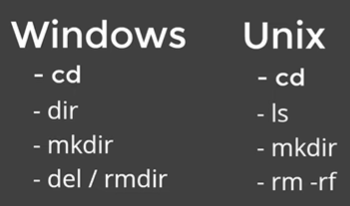
- Armazenamento em nuvem

- Melhoria do código

- Reconhecimento

**Comandos básicos para um bom desempenho no terminal**

- A maioria dos programas possuem interface gráfica, mas o GIT é um CLI e a forma de interagir com ele é por linha de comando



- No Windows, para limpar o terminal usamos o comando **cls** (clear screen)

“Silêncio é sucesso!”

**Entendendo como o GIT funciona**

- O SHA1 é um conjunto de funções hash criptografadas projetadas pela NSA -> essa encriptação gera conjunto de caracteres IDENTIFICADOR de 40 dígitos

- O GIT usa esse identificador sha1 para identificar através dessa sequência de caracteres cada versão de um arquivo que a gente salva

**Objetos internos do Git**

**Blobs**

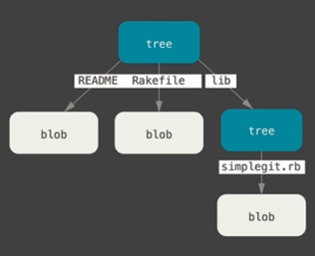
- O git organiza os arquivos dentro de um objeto do tipo blob que contém metadados (como o tamanho do arquivo)

-

**Trees**

- Também contém metadados e aponta para um blob (este, por sua vez, tem um sha1)

- As árvores, por sua vez, têm um sha1 para os metadados desta árvore! Ou seja, se mudar uma vírgula do arquivo que está ali, muda também o sha1 da árvore



**Commits**



- Os commits também possuem um sha1

- Eles devem receber um nome para explicar porque os sha’s ali dentro estão “alterados”