

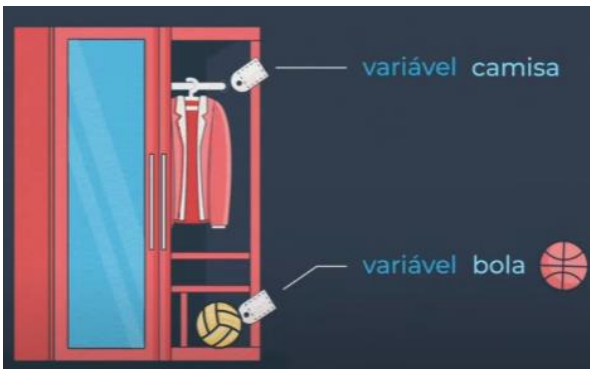
DESENVOLVIMENTO DE SISTEMAS I:

Agenda 02:

Variável - é um lugar na memória RAM que guarda algum tipo de dado (valor). Os valores de uma variável hora assumem um valor, hora assumem outro, é **relativo e dinâmico**, ou seja, é **variável**. A principal função de uma variável é fazer **reutilizar** um valor que foi armazenado.

A variável precisa ser identificada (nome) e possuir um único tipo de dado. Ela armazena um único valor por vez.

Por que armazenar dados? O computador precisa “memorizar” um número para que possa somar com outro, caso contrário quando receber o novo ele já esqueceu o antigo.



Tal qual a um armário que tem lugares específicos para guardar objetos específicos, uma memória tem slots específicos para guardar diferentes tipos de dados e diferentes tamanhos.

Toda vez que eu quiser guardar um novo dado “bola vermelha” talvez seja necessário remover um dado antigo “bola amarela” para ganhar o slot

Constante – Local da memória que armazena dados inalteráveis.

Regras para identificadores:

1. Deve começar com uma **letra**;
2. Os próximos caracteres podem ser **letras** ou **números**;
3. **Não** pode ter **símbolo**, apenas – ou _;
4. **Não** pode ter **espaço em branco**;
5. **Não** pode ter **letras acentuadas**;
6. **Não** pode ser **palavra reservada** (pela própria linguagem – console.log);

Dica: Evite abreviações, gírias internas e escolha nomes intuitivos (DolarNaoConvertido).

Exemplo prático: Login de usuário. Usuario = formulário -> nome; Telefone = formulário -> telefone; As variáveis Usuario e Telefone = (recebem) os dados das respectivas entradas.

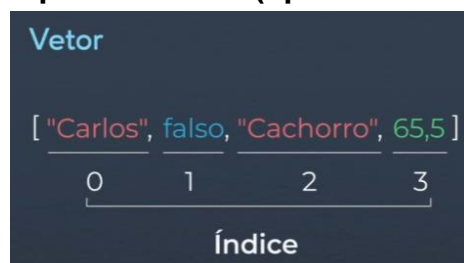
Concatenação: “Olá [Usuario], seu número de telefone foi cadastrado, ligaremos para [Telefone].”

- ➔ <https://www.youtube.com/watch?v=SEQ57illdy4> (Cataline – Lógica de programação 04 – Variáveis).
- ➔ <https://www.youtube.com/watch?v=EIPdLkvAb0w> (Cataline – Lógica de programação 05 – Tipos de dados).

Tipos de dados (tipos primitivos):

- **Inteiro (Integer):** Variável que armazena n° inteiros. {1, 3, -7, 198, 0};
- **Real (Float):** {1.5, 5, 9.8, -77.3, 15.9};
- **Caractere (String):** É uma *string* - cadeia de caractere entre aspas "Tudo que está aqui dentro";
- **Lógico (Boolean):** Verdadeiro ou Falso

Tipos de dados (tipos Vetor / Array):



Tipo de dado composto. Na maioria das linguagens definida por []. Um vetor pode acomodar vários dados de tipo primitivo dentro dele e além disso enumera esses dados por meio de um **índice** da ordem em que aparecem começando do zero.

Operadores aritméticos (suportados pela maioria das linguagens): São operadores que realizam operações **aritméticas** igual na matemática.

- **Adição:** +
- **Subtração:** -
- **Multiplicação:** *
- **Divisão:** /
- **Módulo (Resto da divisão):** % ex: 5 % 2 = 1
- **Potenciação:** **

→ <https://www.youtube.com/watch?v=BQHcwIPxZ9E> (Cataline – Lógica de programação 06 – Operadores matemáticos).

Operadores de comparação: Comparam valores e geram resultados lógicos (booleans).

Você é um sapato? = **falso**
 Você é um humano? = **verdadeiro**

resultados lógicos (verdadeiro ou falso).

- > Maior que
- < Menor que
- >= Maior ou igual a
- <= Menor ou igual a
- == Igual
- === Igual restrito
- != Diferente
- !== Desigual restrito

maior que	5 > 2	verdadeiro
menor que	5 < 2	falso
maior ou igual	5 >= 2	verdadeiro
menor ou igual	5 <= 2	falso
igual	5 == 2	falso
diferente	5 != 2	verdadeiro

→ <https://www.youtube.com/watch?v=s12-dd6PxU0> (Cataline – Lógica de programação 07 – Operadores de comparação).

Operadores Lógicos: Comparam valores lógicos para retornar resultados lógicos.

- **E (Conjunção - Maximalista):** Só é verdadeiro quando todas as proposições forem verdadeiras. (&& em Java).
- **OU (Minimalista):** Só é falso quando todas as proposições forem falsas. (|| em Java).
- **NÃO:** É a inversão de valores lógicos. (!)

Tabelas verdade:

Paula e Quéssia estão felizes.

p	q	p E q
V	V	V
V	F	F
F	V	F
F	F	F

Paula ou Quésia estão felizes.

p	q	p OU q
V	V	V
V	F	V
F	V	V
F	F	F

p	q	p SE, ENTÃO q
V	V	V
V	F	F
F	V	V
F	F	V

p	q	p OU, OU q
V	V	V
V	F	F
F	V	F
F	F	V

→ https://www.youtube.com/watch?v=2l1Hz3U_yx0 (Cataline – Lógica de programação 08 – Operadores lógicos).

→ <https://www.youtube.com/watch?v=ntCQmyfhA30> (Youtube – Informática módulo 1 agenda 11 operadores aritméticos, relacionais e lógicos – GEEaD).



Por quê entender lógica de programação?

Lógica de programação é o processo de pensar na mesma **sequência** em que o computador executa as tarefas.

Algoritmo – sequência finita de passos que levam a execução de uma tarefa. É a planta da casa (ideia, projeto). Inicialmente

Programa de computador – são algoritmos escritos em uma linguagem de programação.

Ordenação do pensamento para estruturação da lógica.

IMPORTANTE (PARA DESENVOLVEDOR): Toda vez que pensar em **algoritmo** ou pensar em **programação** tem que se pensar em **Entrada** (variáveis, constantes, Arrays) > **Processamento** (operadores aritméticos, relacionais, atribuição e lógicos) > **Saída**.

Como começar a planejar um algoritmo – Narrativa (descrição) > Fluxograma (representação gráfica) > Pseudocódigo.

→ <https://www.youtube.com/watch?v=WTQiZ4zylJw> (Youtube – Agenda 02 DS- Módulo 1 – GEEaD).

→ <https://www.youtube.com/watch?v=aYYpwb7zAsU> (Youtube – Informática – módulo 1 - agenda 10 – GEEaD).

→ <https://www.youtube.com/watch?v=lg4QZNpVZYs> (Youtube – Operadores lógicos e relacionais curso algoritmo #4 – Gustavo Guanabara).

→ https://www.youtube.com/watch?v=sTX0UEpIF54&list=PLHz_AreHm4dkl2ZdjTwZA4mPMxWTfNSpR (Playlist Curso de Java para Iniciantes – Gustavo Guanabara)