

Estrutura de Dados e Lógica de Programação

Prof. Ms. José Antonio Gallo Junior

gallojunior@gmail.com

Aula 02 - Algoritmos

Um algoritmo é uma sequência de instruções definidas para serem executadas mecanicamente. Todos os dias realizamos algoritmos e não percebemos. Basicamente todas as manhãs nós nos levantamos da cama, tomamos o café da manhã, nos arrumamos e vamos ao trabalho. Entre cada uma dessas ações descritas, realizamos várias outras ações. Nosso próprio corpo realiza algoritmos o tempo inteiro realizando decisões. Se nos machucamos, sentimos dor. Se o ambiente externo tem temperatura baixa, o corpo treme para tentar aumentar a temperatura novamente.

Aula 02 - Algoritmos

Todo algoritmo segue uma sequência lógica. Geralmente um algoritmo é lido de cima pra baixo e da esquerda pra direita. Cada linha de código é lida e, caso não haja nenhum erro, executada e logo depois é lida a próxima linha e assim sucessivamente até terminarem as linhas do algoritmo.

Como exemplo de algoritmo podemos pensar em como fritar um ovo. Observe:

Aula 02 - Algoritmos

1. Pegar o ovo
2. Pegar a frigideira
3. Acender o fogo e colocar a frigideira sobre ele
4. Colocar óleo na frigideira
5. Quebrar o ovo na frigideira
6. Adicionar sal
7. Aguardar a fritura do ovo, observando para não queimar.
8. Pegar um prato.
9. Desligar o fogo
10. Mover o ovo frito da frigideira para o prato.
11. Comer

Aula 02 - Algoritmos

Outro exemplo de algoritmo seria fazer um cálculo simples como a soma ou multiplicação de dois números como no exemplo abaixo:

1. Ler primeiro número
2. Ler segundo número
3. Somar os dois números lidos
4. Exibir o resultado

Aula 02 - Programas

Um programa é um ou mais algoritmos escritos numa linguagem de programação. Existem diversas linguagens de programação atualmente. A grosso modo as linguagens mais fáceis são chamadas de “linguagens de alto nível” e as mais difíceis são chamadas de linguagens de baixo nível”. As Linguagens de Baixo nível são assim chamadas por que você precisa trabalhar mandando comandos e recebendo respostas direto para o hardware. As Linguagens de Alto nível são assim chamadas por que elas “escondem” as tarefas difíceis como trabalhar direto com hardware, o que facilita o desenvolvimento do código e aumenta a velocidade de programação.

Aula 02 – Etapas para Desenvolver um Programa



ANÁLISE

ALGORITMO

CODIFICAÇÃO

Aula 02 - Análise

Primeiramente é necessário **estudar o problema**, definindo-se bem **quais são os dados a ser informados na entrada, como esses dados serão processados e quais são os dados de saída esperados**.

Essas definições são importantes para que o **objetivo do programa seja bem definido**.



Aula 02 - Algoritmos

Após a análise e a identificação dos dados de entrada e saída, escreve-se o **algoritmo, que, basicamente, é o **passo a passo da resolução do problema em questão**.**

Na construção do algoritmo, utilizamos a **lógica**.

Aula 02

Javascript

O que é necessário para programar em Javascript?

Um Editor de Texto:

- Bloco de Notas
- Notepad ++
- Visual Studio Code
- GVim

Um Navegador:

- Google Chrome
- Edge
- OperaGX
- Brave

Javascript - Comentários

Comentário de uma única linha

// Lorem ipsum dolor sit amet.

Comentário de várias linhas

/* Lorem ipsum dolor sit amet, consectetur adipiscing elit. In massa arcu, facilisis ut sapien et, iaculis bibendum enim. Aliquam massa velit, fringilla sit amet sodales eu, lobortis mattis erat. Nulla facilisi. ***/**

Variáveis

Variáveis são espaços em memória onde podemos alocar algum valor temporariamente. Como o nome diz, o valor de uma variável pode ser alterado. Você pode declarar uma variável de três formas:

- Com a palavra chave **let**. Por exemplo, **let y = 13**. Essa sintaxe pode ser usada para declarar uma variável local de escopo de bloco. Veja escopo de variável abaixo.
- Com a palavra-chave **var**. Por exemplo, **var x = 42**. Esta sintaxe pode ser usada para declarar tanto variáveis locais como variáveis globais.
- Por simples adição de valor. Por exemplo, **x = 42**. Isso declara uma variável global. Essa declaração gera um aviso de advertência no Javascript. **Você não deve usar essa variante.**

Constantes

Você pode criar uma constante apenas de leitura por meio da palavra-chave **const**. Uma constante não pode alterar seu valor por meio de uma atribuição ou ser declarada novamente enquanto o script está em execução. Deve ser inicializada com um valor. As regras de escopo para as constantes são as mesmas para as variáveis let de escopo de bloco.

const PI = 3.14

Variáveis e Constantes: Escopo

Quando você declara uma variável ou constante fora de qualquer função, ela é chamada de variável **global**, porque está disponível para qualquer outro código no documento atual. Quando você declara uma variável dentro de uma função, é chamada de variável **local**, pois ela está disponível somente dentro dessa função.

1997

var

2015

2023

let e const



Javascript

Estrutura de Decisão

Estrutura de Decisão

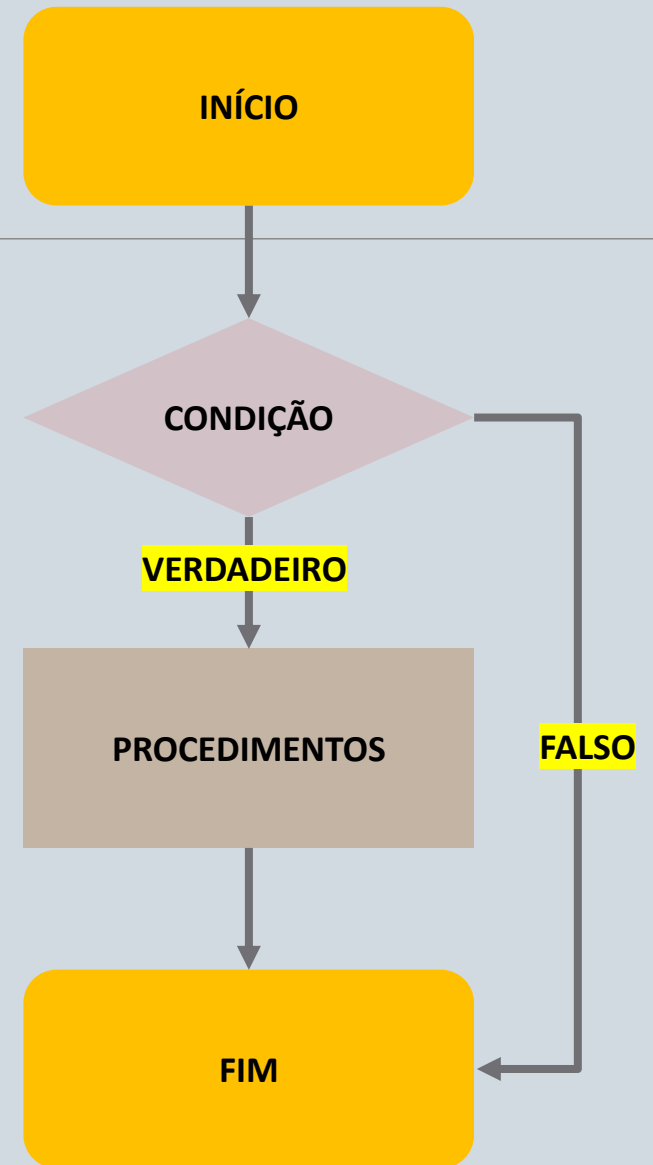
As estruturas de decisão (também conhecidas como Estruturas Condicionais) exigem que o programador especifique uma ou mais condições a serem testadas pelo programa. Ela se resume a efetuar um teste, se a condição for atendida, o interpretador Javascript entra em um bloco de código, se não for, ela passa para o próximo bloco, e se não houver nenhum outro, ela prossegue com o código normalmente.

Condição SE (IF)

A condicional **if** é uma estrutura condicional que executa a afirmação, dentro do bloco, se determinada condição for verdadeira. Se for falsa, executa as afirmações dentro de **else**.

```
if (condição) {  
    instrução1  
} else {  
    instrução2  
}
```

Condição SE (IF)



Múltiplas Condições

Múltiplas condicionais **if ... else** podem ser aninhados quando necessário. Observe que não existe **else if** (em uma palavra). O correto é a instrução com espaços (**else if**), conforme ao lado:

```
if (condição1)
    Instrução1
else if (condição2)
    instrução2
else if (condição3)
    instrução3
...
else
    instruçãoN
```

Exemplo IF

```
let semaforo = "amarelo";
```

```
if (semaforo == "verde") {  
    mensagem = "Prossiga";  
} else if (semaforo == "vermelho") {  
    mensagem = "Pare";  
} else {  
    mensagem = "Atenção";  
}
```

Exemplo IF

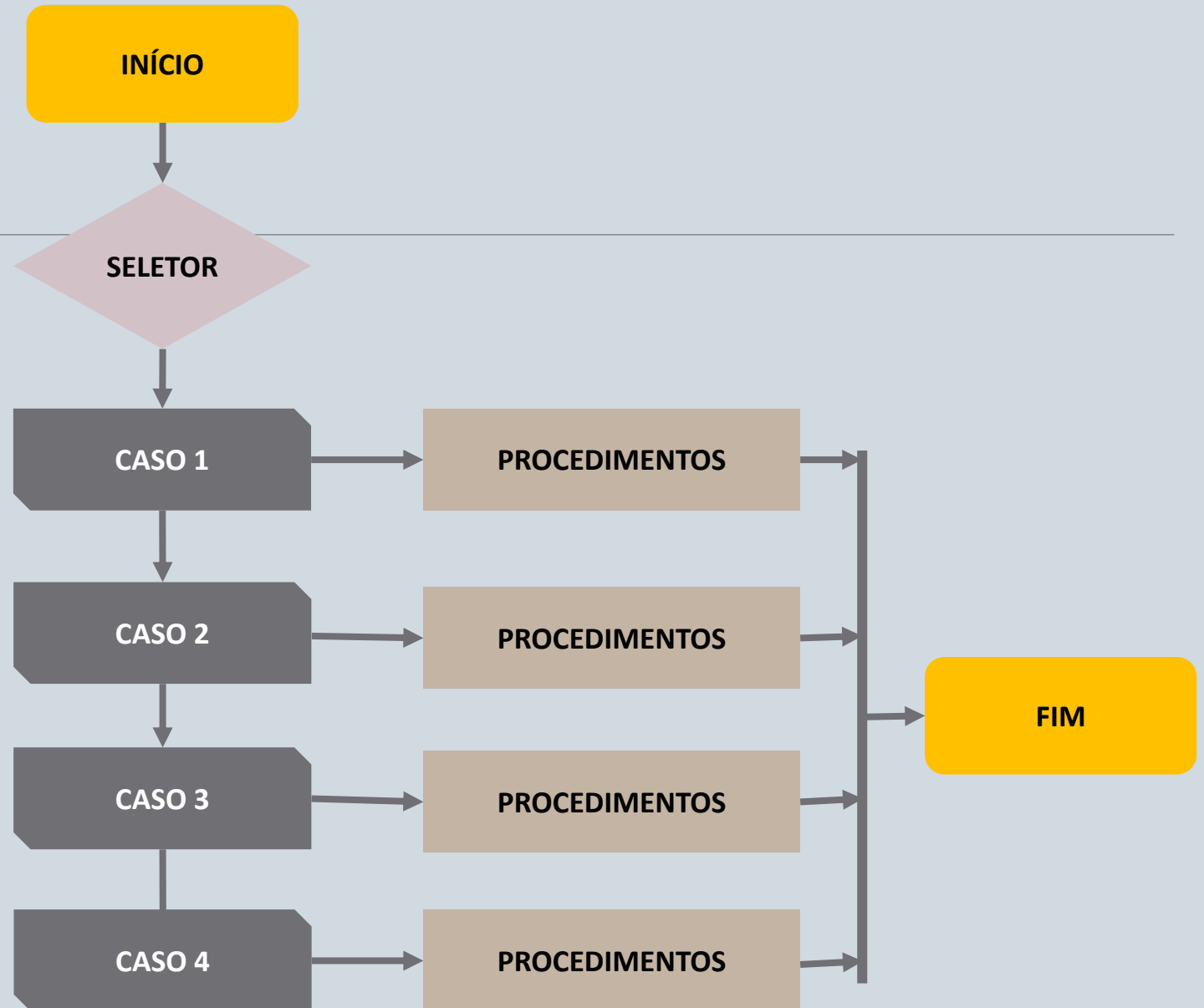
```
let peso = 26.3;  
let status = "";
```

```
if (peso < 20.7) {  
    status = "Abaixo do peso";  
} else if (peso >= 20.7 && peso < 26.4) {  
    status = "Peso normal";  
} else if (peso >= 26.4 && peso < 27.8) {  
    status = "Pouco acima do peso";  
} else if (peso >= 27.8 && peso < 31.1) {  
    status = "Acima do peso";  
} else {  
    status = "Obeso";  
}
```

Condição SWITCH

Se a condição for correspondida, o programa executa as instruções associadas. Se múltiplos casos corresponderem o valor, o primeiro caso que corresponder é selecionado, mesmo se os casos não forem iguais entre si.

Condição SWITCH



Exemplo SWITCH

```
switch (expr) {  
  case "Laranjas":  
    console.log("As laranjas custam $0.59 o quilo.");  
    break;  
  case "Maçãs":  
    console.log("Maçãs custam $0.32 o quilo.");  
    break;  
  case "Bananas":  
    console.log("Bananas custam $0.48 o quilo.");  
    break;  
  case "Cerejas":  
    console.log("Cerejas custam $3.00 o quilo.");  
    break;  
  default:  
    console.log("Desculpe, estamos sem nenhuma " + expr + ".");  
}
```