

# EAJ0365-Programação de computadores

**Prof. Dr. Josenalde Barbosa de Oliveira**

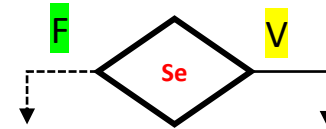
josenalde@eaj.ufrn.br

# Estrutura de controle de fluxo: *condicional - decisão*

- ▶ Desenvolver a solução algorítmica de determinado problema consiste em organizar os passos em estruturas de controle de fluxo, a saber, sequencial, **condicional (Se...então...senão)** e de repetição.
- ▶ A estrutura condicional refere-se à **tomada de decisão**, aos caminhos que o fluxo de execução pode seguir
- ▶ Pode ser simples ou aninhada/composta (*nested if*)
- ▶ No pseudocódigo: se...então...senão
- ▶ Em JavaScript: SIMPLES

```
▶ if <condicaoLogica> {  
    <blocoComandosSeVerdadeiro>  
}  
[else {  
    <blocoComandosSeFalso>  
}]
```

Avaliado como V ou F



# Estrutura de controle de fluxo: *condicional - decisão*

► SIMPLES: com apenas um comando/instrução não é obrigatório o uso de chaves para delimitar bloco

```
var x = 5.2, y = 3.4;
```

```
if (y < x) {
```

```
    y++;
```

```
} else {
```

```
    y--;
```

```
}
```

```
console.log(y); // FIM
```

```
if (y < x) {
```

```
    y++; x--;
```

```
} else {
```

```
    y--; x++
```

```
} // FIM
```

```
console.log("x e y", x, y);
```



```
var x = 5.2, y = 3.4;
```

```
if (y < x) y++;
```

```
else y--;
```

```
console.log(y); // FIM
```

Erro de interpretação. Como não há bloco, espera o **else** logo após o primeiro comando.

```
if (y < x) y++; x--;
```

```
else y--;
```

```
console.log("x e y", x, y);
```

```
if (y < x) y++; x--;
```

```
console.log("x e y", x, y);
```

Para semântica correta {} obrigatório

Já neste caso, x- não pertence ao IF, e é executado independente da comparação.  
X = 4.2, y = 4.4;

# Estrutura de controle de fluxo: *condicional* - *decisão*

- ▶ SIMPLES: operador ternário ? Opção ao uso do IF. A sintaxe é:

<condição>?<comandosSeVerdadeiro>:<comandosSeFalso>

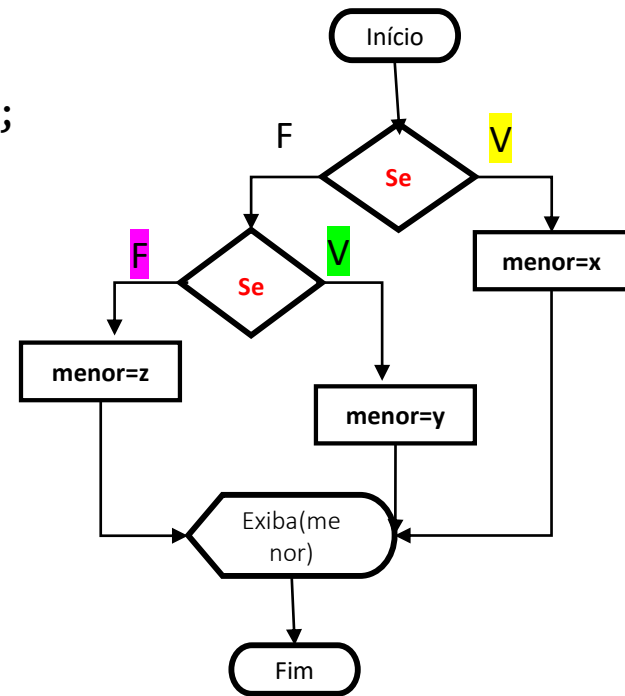
```
var letra = scanf('%c');  
(letra>='a' && letra<='z')?console.log('m'):console.log('M');  
//exemplo 2  
var z = (letra=='x')? 10: 20;  
console.log(z);
```

# Estrutura de controle de fluxo: *condicional - decisão*

## ➡ ANINHADO

```
var x = 5.2, y = 3.4, z = 1.4, menor;  
if (x < y && x < z) menor = x;  
else if (y < x && y < z) menor = y;  
else menor = z;  
console.log(menor.toFixed(1));
```

- Na teoria não existem limites para os níveis
- Porém, para escolha de itens etc. outro comando é mais recomendável: **switch**



# Estrutura de controle de fluxo: *condicional* - *decisão*

► SWITCH: avalia expressão (constante, ou seja, numérica ou character) e encaminha o fluxo para um determinado caso (case), o qual deve ser terminado com um comando **break**, para sair do **switch** e não necessitar testar as demais condições.

```
var opCalc;
var a, b;
[a opCalc b] = scanf('%d %c %d');
switch(opCalc) {
    case '+': return a+b; break;
    case '-': return a-b; break;
    case '*': return a*b; break;
    case '/': if (b!=0) return a/b; break;
    default: console.log("operação inválida");
}
```

```
case '+': {
    return a+b;
    break;
}

case '+': {
    let c;
    c = a+b;
    return c;
    break;
}
```

Pode haver bloco delimitado, opcional, mas se houver declaração de variável, o bloco é obrigatório.

Caso nenhuma opção prevista seja digitada pelo usuário, equivale ao **else final** de um IF ANINHADO.

# Estrutura de controle de fluxo: *condicional* - *decisão*

---

► **Proposta de exercícios:** algoritmo e código em JavaScript para ler 03 números reais e apresentar menu ao usuário para escolher opção: 1 – maior, 2 – menor, 3 – media aritmética, 4 – media geométrica, 5 – media ponderada com pesos 4, 5 e 6 (usar **switch**). Exibir o resultado do cálculo escolhido pelo usuário.

► Escrever o algoritmo e código em JavaScript, submetendo no Sistema da OBI, para o problema **Conta de Água** (obi 2010, fase 1): <https://olimpiada.ic.unicamp.br/pratique/p1/2010/f1/conta/>

► DICA: em problemas de competições de programação, normalmente existem restrições na faixa de valores para as variáveis de entrada. Por exemplo, a restrição

$$1 \leq A \leq 1000$$

representa o interval fechado entre 1 e 1000, inclusive, ou seja,  $A \in [1,1000]$ . Isto pode ser escrito em C++ como:  $1 \leq A \text{ and } A \leq 1000$  ou  $A \geq 1 \text{ and } A \leq 1000$