

Essa página diz  
Formulário está OK?

Campo obrigatório 1:

Campo obrigatório 2:

# JavaScript

Criação de scripts de automação de páginas WEB

# Unidade Curricular: JavaScript

## CONTEÚDO FORMATIVO

### 1. Criação de scripts de automação de páginas WEB

1. Características da linguagem
2. Introdução a lógica de programação
3. Tag Script
4. Sintaxe Básica
5. Operadores
6. Tipos de dados
7. Comparadores lógicos
8. Blocos condicionais
9. Array
10. Blocos de repetição
11. Funções
12. Console do navegador
13. Validação de formulário



#### Capacidades Técnicas

1. Automatizar documentos por meio da linguagem JavaScript
2. Integrar recursos dinâmicos da linguagem JavaScript à páginas HTML

#### Capacidades sociais, organizativas e metodológicas

1. Planejar tarefas no ambiente de trabalho
2. Demonstrar raciocínio lógico
3. Demonstrar iniciativa na busca por novas informações em diferentes meios

# Atividade para fixação do conteúdo



## Pergunta 1:

O JavaScript é uma linguagem:

1. Compilada
2. Interpretada

## Pergunta 2:

O instrução "var" serve para:

1. Indicar ao interpretador do JS que uma nova variável está sendo declarada.
2. Forçar o tipo da variável.
3. Atribuir valor a uma variável.

## Pergunta 3:

Selecione qual instrução é responsável por gerar uma saída no console do browser (ferramentas de desenvolvedor do browser)

1. `alert()`
2. `document.write()`
3. `console.log()`

# Atividade para fixação do conteúdo



## Pergunta 1:

O JavaScript é uma linguagem:

- 1.Compilada
- 2.Interpretada**

## Pergunta 2:

O instrução "var" serve para:

- 1.Indicar ao interpretador do JS que uma nova variável está sendo declarada.**
- 2.Forçar o tipo da variável.
- 3.Atribuir valor a uma variável.

## Pergunta 3:

Selecione qual instrução é responsável por gerar uma saída no console do browser (ferramentas de desenvolvedor do browser)

- 1.alert()
- 2.document.write()
- 3.console.log()**

# Atividade para fixação do conteúdo



## Pergunta 4:

O valor null atribuído a uma variável indica que:

- 1.A variável representa a ausência intencional de um valor.
- 2.A variável representa o valor undefined.
- 3.A variável não pode ter seu valor modificado.

## Pergunta 5:

Comentários servem, dentre outras coisas, para:

- 1.Permite capturar entrada de textos do usuário para uso dentro da lógica do script.
- 2.Exibir mensagens customizadas para o usuário.
- 3.Inclusão de textos descritivos que são ignorados pelo interpretador do JS e ajudam na identificação da lógica da programação facilitando futuras manutenções.

# Atividade para fixação do conteúdo



## Pergunta 4:

O valor null atribuído a uma variável indica que:

- 1.A variável representa a ausência intencional de um valor.**
- 2.A variável representa o valor undefined.
- 3.A variável não pode ter seu valor modificado.

## Pergunta 5:

Comentários servem, dentre outras coisas, para:

- 1.Permite capturar entrada de textos do usuário para uso dentro da lógica do script.
- 2.Exibir mensagens customizadas para o usuário.
- 3.Inclusão de textos descritivos que são ignorados pelo interpretador do JS e ajudam na identificação da lógica da programação facilitando futuras manutenções.**

## Operadores comparação

### Operadores de comparação/Relacionais (condicionais)

Igual ==

Maior igual >=

Diferente !=

Menor igual <=

Maior >

Estritamente ===

Menor <

Estritamente !==

- **Igual (==)** - verifica se os valores comparados são **iguais**.
- **Idêntico (===)** - verifica se os valores comparados são **iguais e do mesmo tipo**.
- **Diferente(!=)** - verifica se os valores comparados são **diferentes**.
- **Não Idêntico (!==)** - verifica se os valores comparados são **diferentes e de tipos diferentes e do mesmo tipo**.
- **Menor(<)** - verifica se valor da esquerda é **menor** que o valor da direita.
- **Maior(>)** - verifica se valor da esquerda é **maior** que o valor da direita.
- **Menor igual(<=)** - verifica se valor da esquerda é **menor ou igual** que o valor da direita.
- **Maior igual(>=)** - verifica se valor da esquerda é **maior ou igual** que o valor da direita.

## Operadores de comparação/Relacionais (condicionais)

- **Igual (==)** - <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Equality#syntax>
- **Idêntico (===)** - [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Strict\\_equality](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Strict_equality)
- **Diferente(!=)** - <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Inequality>
- **Não Idêntico (!==)** - [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Strict\\_inequality](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Strict_inequality)
- **Menor(<)** - [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Less\\_than](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Less_than)
- **Maior(>)** - [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Greater\\_than](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Greater_than)
- **Menor igual(<=)** - [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Less\\_than\\_or\\_equal](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Less_than_or_equal)
- **Maior igual(>=)** - [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Greater\\_than\\_or\\_equal](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Greater_than_or_equal)

Outra fonte: [https://www.w3schools.com/js/js\\_operators.asp](https://www.w3schools.com/js/js_operators.asp)



# Lógica de Programação

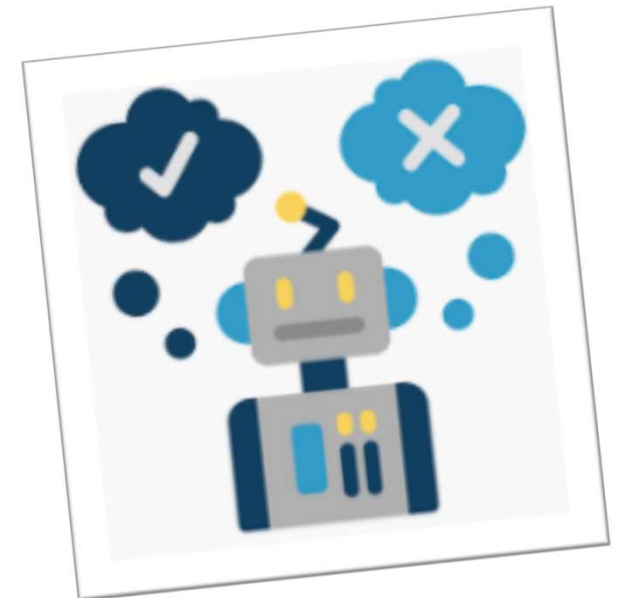
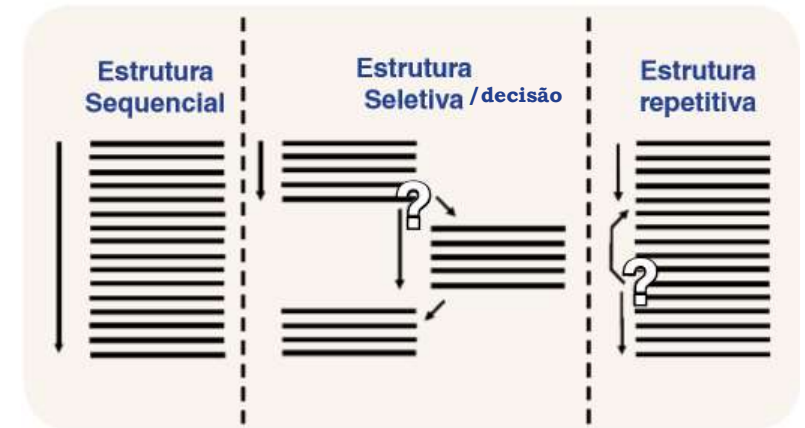
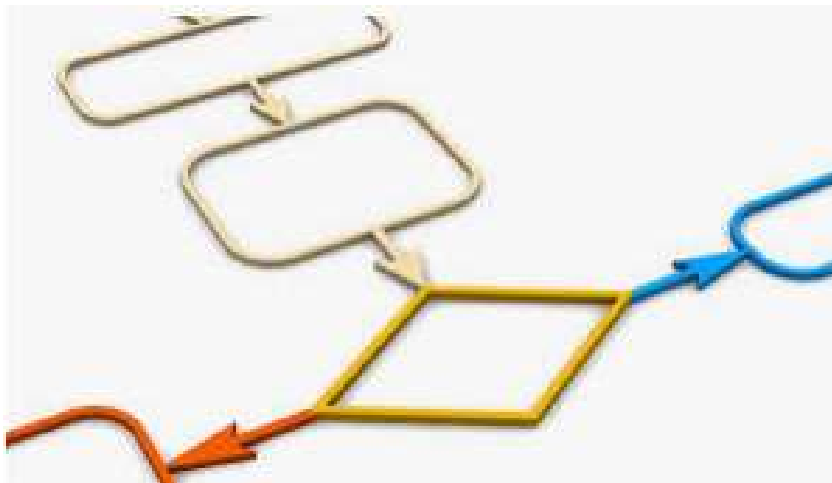
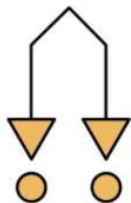
## ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

### Estrutura de Decisão/Seletiva

- O fluxo de instruções a ser seguido é escolhido em função do resultado da avaliação de uma ou mais condições;
- três tipos de estruturas de decisão:
  - Estrutura de Decisão Simples (**Se ... então**);
  - Estrutura de Decisão Composta (**Se ... então ... senão**);
  - Estrutura de Decisão Múltipla do Tipo Escolha (**Escolha ... Caso ... Senão**).



Condicionais



## Desenhista Web

### SINTAXES DAS CONDICIONAIS EM JAVASCRIPT

#### SINTAXE IF ... ELSE

```
if (condição) afirmação1 [else afirmação2]
```

#### SINTAXE IF ... ELSE ENCADEAD / ANINHADO

```
if (condição1)
    instrução1
else
    if (condição2)
        instrução2
    else
        if (condição3)
            ...
    ...
```



```
if (condição1)
    instrução1
else if (condição2)
    instrução2
else if (condição3)
    instrução3
...
else
    instruçãoN
```

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/if...else>

[https://www.w3schools.com/js/js\\_if\\_else.asp](https://www.w3schools.com/js/js_if_else.asp)

### SINTAXES DAS CONDICIONAIS EM JAVASCRIPT

#### SINTAXE SWITCH

```
switch (expressão) {  
  case valor1:  
    //Instruções executadas quando o resultado da expressão for igual á valor1  
    [break;]  
  case valor2:  
    //Instruções executadas quando o resultado da expressão for igual á valor2  
    [break;]  
  ...  
  case valueN:  
    //Instruções executadas quando o resultado da expressão for igual á valorN  
    [break;]  
  default:  
    //Instruções executadas quando o valor da expressão é diferente de todos os  
    cases  
    [break;]  
}
```

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/switch>

[https://www.w3schools.com/js/js\\_switch.asp](https://www.w3schools.com/js/js_switch.asp)

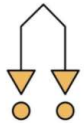
# Lógica de Programação

## ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

### Estrutura de Decisão/Seletiva

#### Estrutura de Decisão Simples (Se ... então)

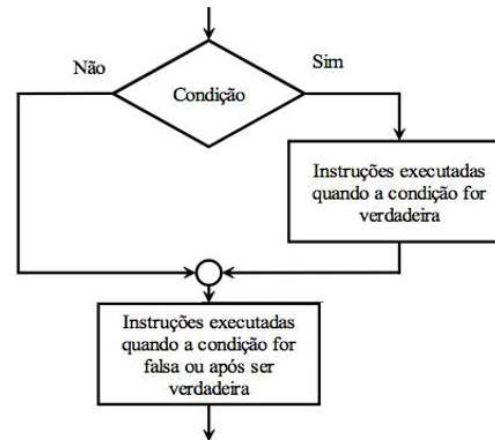
Condicionais



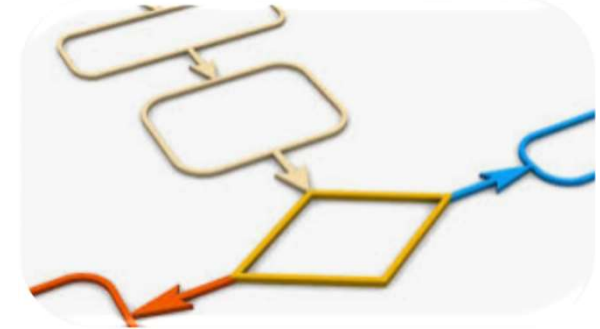
```
se (<<expressão lógica>>) entao
    //Bloco de código
fimse
```

```
se (num1 <> 0 E num2 <> 0) entao
    escreval("Ambos os números são diferentes de zero!")
fimse.
```

```
7 Var
8 valor : inteiro
9
10 Inicio
11 escreval("Informe um valor!")
12 leia(valor)
13 se (valor > 10) entao
14     escreval("O valor informado é maior que 10")
15 fimse
16
17 Fimalgoritmo
```



```
1 Algoritmo "GanhaDesconto"
2
3 Var
4     quantidade: inteiro
5     desconto: real
6
7 Inicio
8
9 desconto := 0
10 escreval ("Digite a quantidade de itens que pretende comprar")
11 leia (quantidade)
12
13 se (quantidade > 20) entao
14     desconto := 10
15 fimse
16
17 desconto := 5 + desconto
18 escreval ("Você terá", desconto, "% de desconto")
19
20 Fimalgoritmo
```



Console simulando o modo texto do MS-DOS

```
Digite a quantidade de itens que pretende comprar
15
Você terá 5% de desconto

>>> Fim da execução do programa !
```

Console simulando o modo texto do MS-DOS

```
Digite a quantidade de itens que pretende comprar
22

Você terá 15% de desconto

>>> Fim da execução do programa !
```

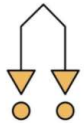
# Lógica de Programação

## ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

### Estrutura de Decisão/Seletiva

#### Estrutura de Decisão Simples (Se ... então)

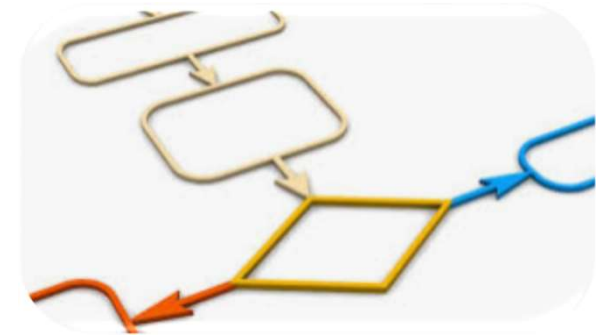
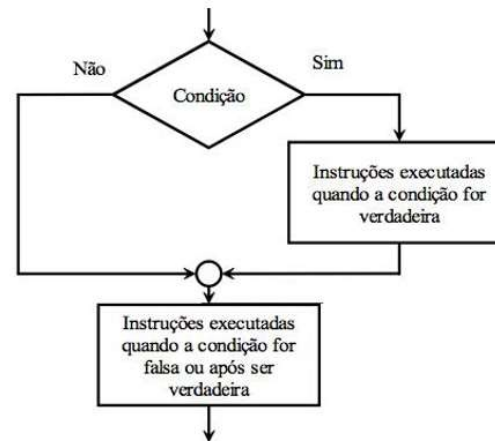
Condicionais



```
se (<<expressão lógica>>) entao
    //Bloco de código
fimse
```

```
se (num1 <> 0 E num2 <> 0) entao
    escreval("Ambos os números são diferentes de zero!")
fimse.
```

```
1  var valor = prompt ("Informe um valor:");
2
3  if (valor > 10) {
4      console.log("O valor informado é maior que 10");
5  }
```



```
1  var quantidade;
2  var desconto = 0;
3
4  quantidade = prompt("Digite a quantidade de itens que pretende comprar");
5  if (quantidade > 20){
6      desconto = 10;
7  }
8  desconto = 5;
9  console.log("Você terá", desconto, "% de desconto");
```



# Lógica de Programação

## ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

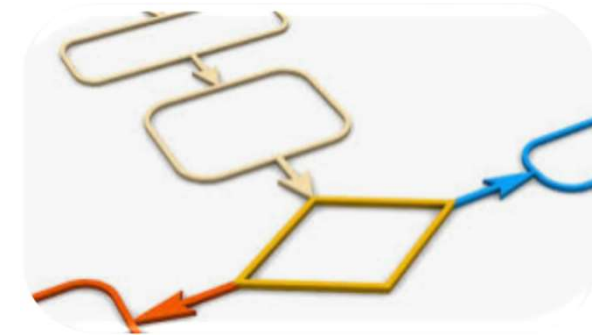
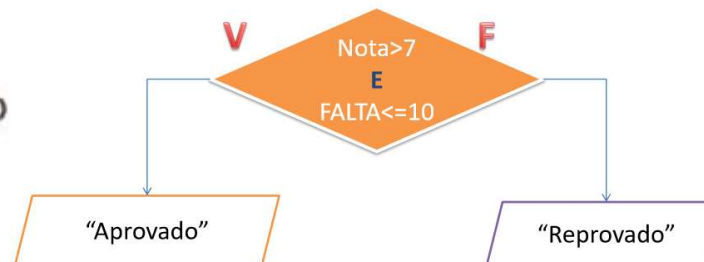
### Estrutura de Decisão/Seletiva

#### Estrutura de Decisão Composta (Se ... então ... senão

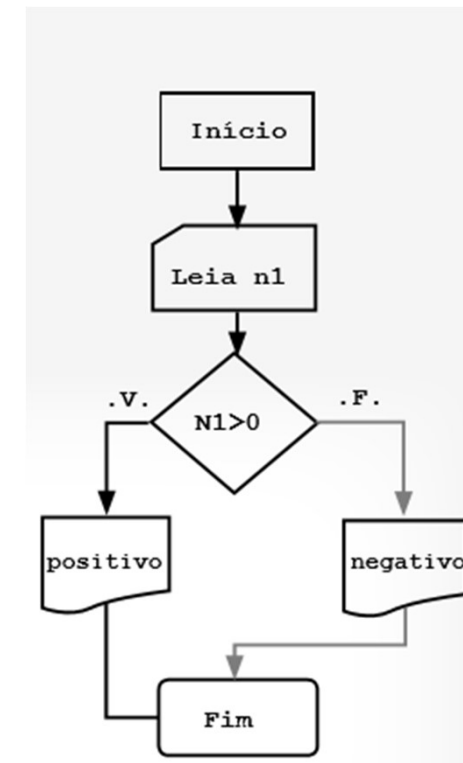
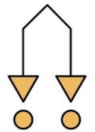
```
se (<<expressão lógica>>) entao
    //Bloco de código caso a expressão for verdadeira
senao
    //Bloco de código caso a expressão for falsa
Fimse
```

```
7 Var
8 valor : inteiro
9
10 Inicio
11 escreval("Informe um valor!")
12 leia(valor)
13 se (valor > 10) entao
14     escreval("O valor informado é maior que 10")
15 senao
16     escreval("O valor informado é menor ou igual a 10")
17 fimse
18
19 Finalgoritmo
```

```
Algoritmo "exemplo1"
Var n1:inteiro
Inicio
    Leia (n1)
    Se n1>=0 então
        Escreva ("O número é positivo.")
    Senão
        Escreva ("O número é negativo.")
    FimSe
Finalgoritmo
```



Condicionais



# Lógica de Programação

## ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

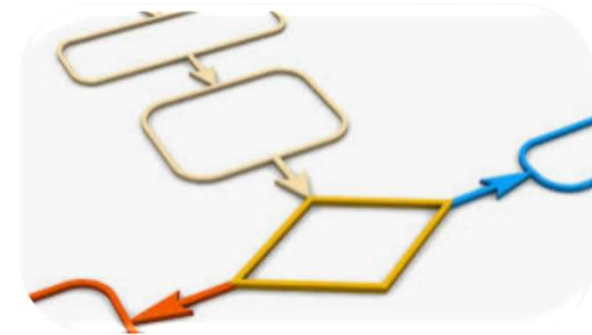
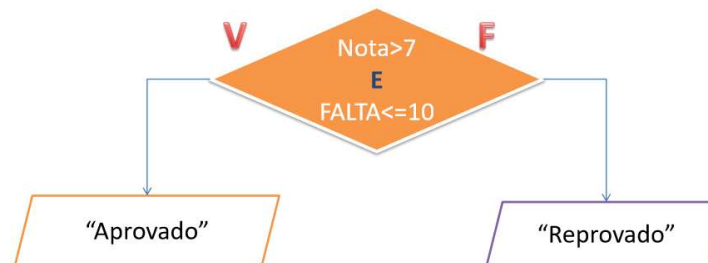
### Estrutura de Decisão/Seletiva

#### Estrutura de Decisão Composta (Se ... então ... senão

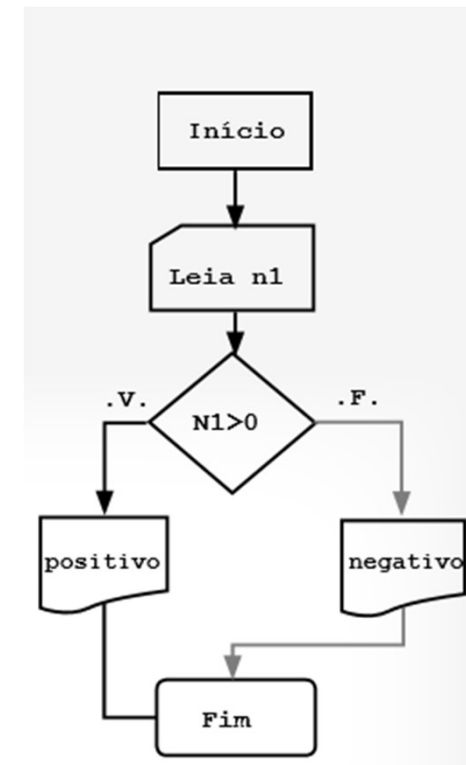
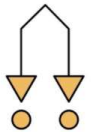
```
se (<<expressão lógica>>) entao
    //Bloco de código caso a expressão for verdadeira
senao
    //Bloco de código caso a expressão for falsa
Fimse
```

```
1  var valor = prompt("Informe um valor:");
2
3  if (valor > 10) {
4      console.log("O valor informado é maior que 10");
5  } else {
6      console.log("O valor informado é menor ou igual a 10");
7  }
```

```
Algoritmo "exemplo1"
Var n1:inteiro
Início
    Leia (n1)
    Se n1>=0 então
        Escreva ("O número é positivo.")
    Senão
        Escreva ("O número é negativo.")
    FimSe
FimAlgoritmo
```



Condicionais



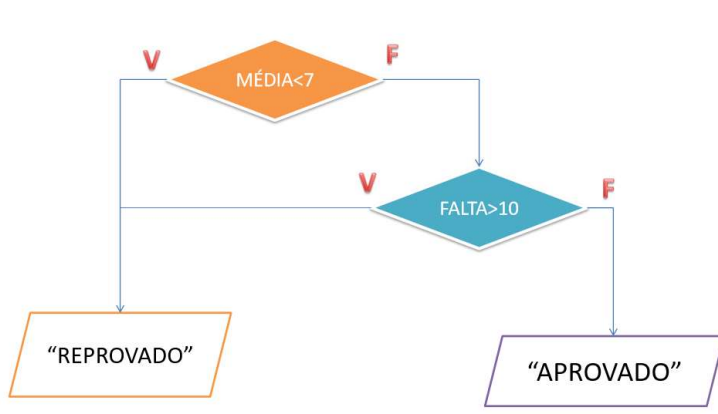
# Lógica de Programação

## ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

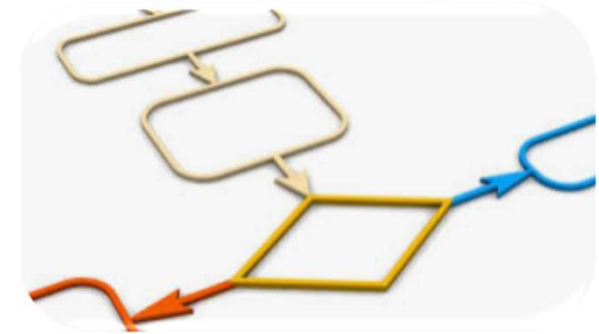
### Estrutura de Decisão/Seletiva

#### Estrutura de Decisão Composta ENCADEADA (Se ... então ... Senão)

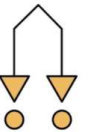
```
se (<<expressão lógica>>) entao
    //Executa este bloco de código.
senao
    se (<<expressão lógica>>) entao
        //Executa este bloco de código.
    senao
        //Executa este bloco de código.
    fimse
fimse
```



```
7 Var
8 valor : inteiro
9
10 Inicio
11 escreval("Informe um valor!")
12 leia(valor)
13 se (valor < 10) entao
14     escreval("O valor informado é menor que 10")
15 senao
16     se (valor < 20) entao
17         escreval("O valor é maior que 10 e menor que 20")
18     senao
19         se (valor < 30) entao
20             escreval("O valor é maior que 20 e menor que 30")
21         senao
22             escreval("O valor é maior ou igual a 30")
23         fimse
24     fimse
25 fimse
```



Condicionais





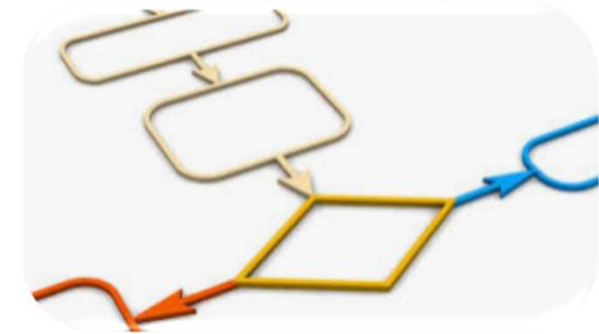
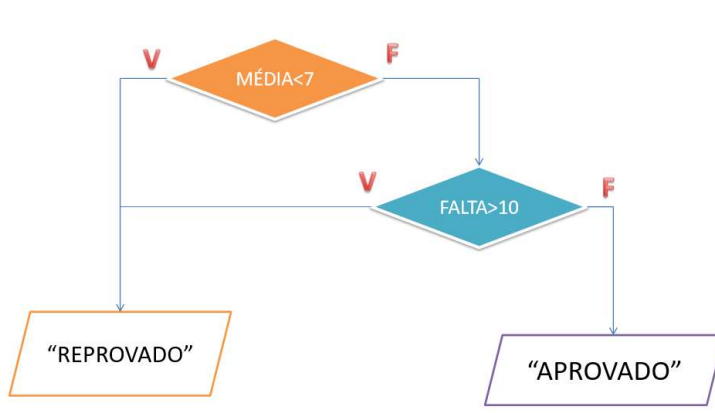
# Lógica de Programação

## ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

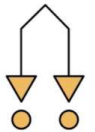
### Estrutura de Decisão/Seletiva

#### Estrutura de Decisão Composta ENCADEADA (Se ... então ... Senão)

```
se (<<expressão lógica>>) então
    //Executa este bloco de código.
senao
    se (<<expressão lógica>>) então
        //Executa este bloco de código.
    senao
        //Executa este bloco de código.
fimse
fimse
```



Condicionais



```
1  var valor = prompt("Informe um valor:");
2
3  if (valor < 10){
4      console.log("O valor informado é menor que 10");
5  } else if (valor < 20){
6      console.log ("O valor é maior ou igual a 10 e menor que 20");
7  } else if (valor < 30){
8      console.log("O valor é maior que 20 e menor que 30");
9  } else {
10     console.log("O valor é maior ou igual a 30")
11 }
```

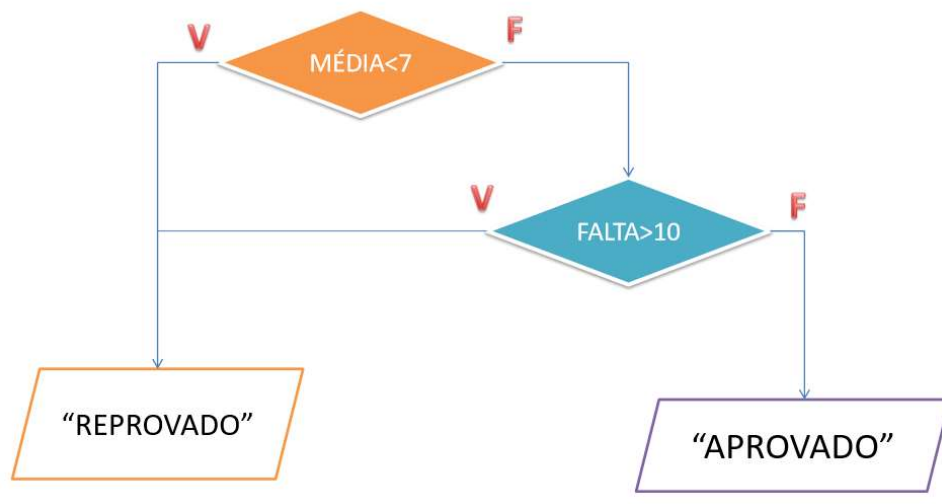
## Lógica de Programação

### ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

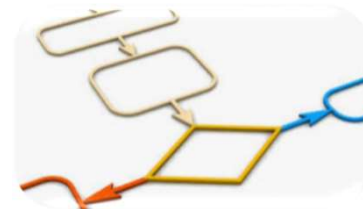
#### Estrutura de Decisão/Seletiva

(Se ... então ... Senão) *Vamos praticar?*

Construa um programa que leia **4 notas**, a quantidade de **faltas** e seguindo as regras do fluxograma apresente a situação do aluno (**Aprovado** ou **Reprovado**).



```
1  Algoritmo "Boletim"
2
3  Var
4  nota1, nota2, nota3, nota4, media: real
5  faltas: inteiro
6
7  Inicio
8  escreval("Digite as 4 notas")
9  leia( nota1, nota2, nota3, nota4)
10 escreval("Digite a quantidade de faltas do aluno")
11 leia(faltas)
12
13 media := (nota1+nota2+nota3+nota4)/4
14 escreval("A Média do aluno é:", media)
15
16 se (media<7) entao
17     escreval(">>>>>Reprovado")
18 senao
19     se (faltas>10) entao
20         escreval(">>>>>Reprovado")
21     senao
22         escreval(">>>>>Aprovado")
23     fimse
24 fimse
25 Fimalgoritmo
```



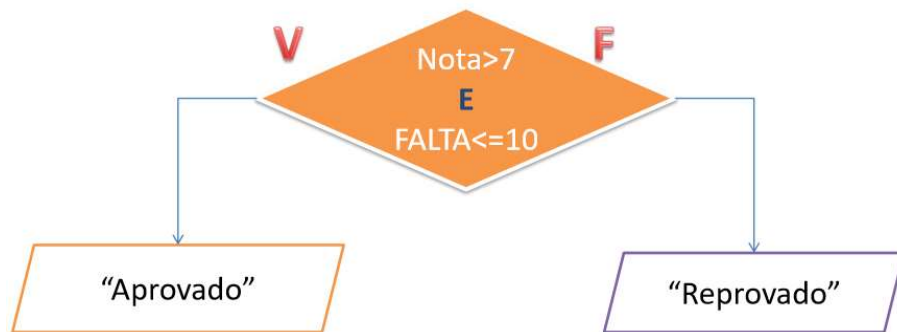
# Lógica de Programação

## ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

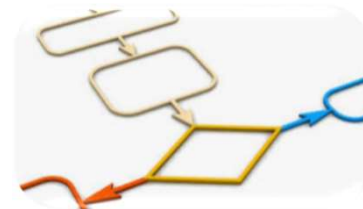
### Estrutura de Decisão/Seletiva

(Se ... então ... Senão) *Vamos praticar?*

Construa um programa que leia **4 notas**, a quantidade de **faltas** e seguindo as regras do fluxograma apresente a situação do aluno (**Aprovado** ou **Reprovado**).



```
1 Algoritmo "Boletim"
2
3 Var
4  nota1, nota2, nota3, nota4, media: real
5  faltas: inteiro
6
7 Inicio
8  escreval("Digite as 4 notas")
9  leia( nota1, nota2, nota3, nota4)
10 escreval("Digite a quantidade de faltas do aluno")
11 leia(faltas)
12
13 media := (nota1+nota2+nota3+nota4)/4
14 escreval("A Média do aluno é:", media)
15
16 se (media >= 7) E (faltas < 10) entao
17     escreval(">>>>Aprovado")
18 senao
19     escreval(">>>>Reprovado")
20 fimse
21 Fimalgoritmo
```

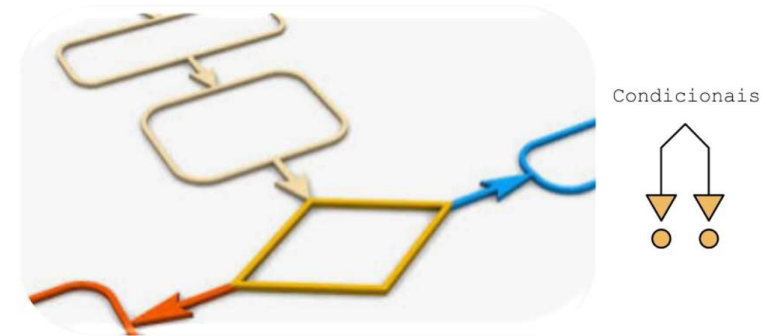


# Lógica de Programação

## ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

### Estrutura de Decisão/Seletiva <https://wiki.python.org.br/EstruturaDeDecisao>

(Se ... então ... Senão) *Vamos praticar?*

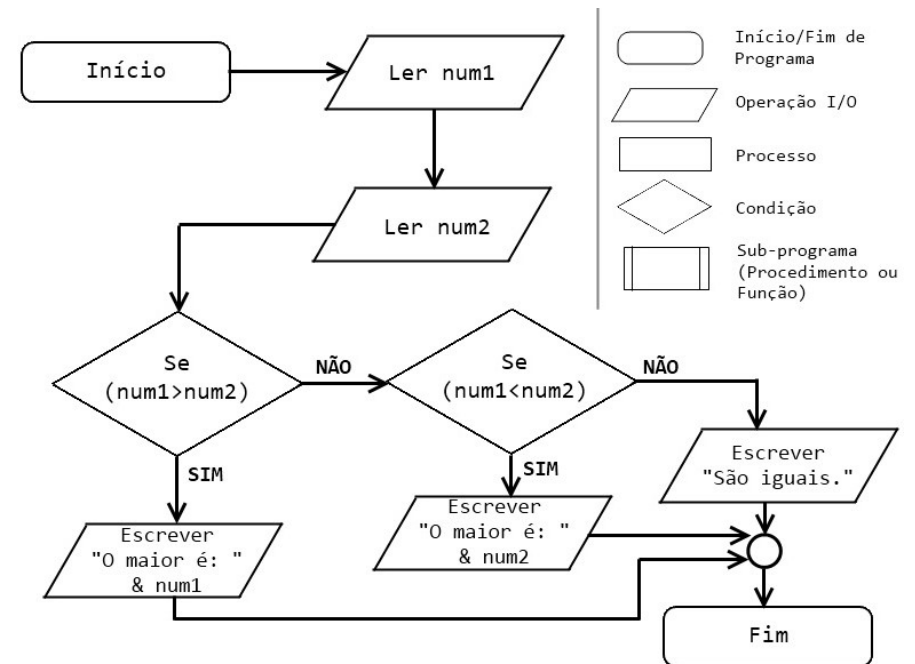


1- Monte um Programa que leia um valor e escrever se é positivo, negativo ou zero.

2- Ler o ano atual e o ano de nascimento de uma pessoa. Escrever uma mensagem que diga se ela poderá ou não votar este ano (não é necessário considerar o mês em que a pessoa nasceu).

3- Ler 3 valores (A, B e C) representando as medidas dos lados de um triângulo e escrever se formam ou não um triângulo. OBS: para formar um triângulo, o valor de cada lado deve ser menor que a soma dos outros 2 lados.

4- Monte um Programa com base no fluxograma que receba dois números e mostre qual é o maior numero ou se são iguais.



## Lógica de Programação

### ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

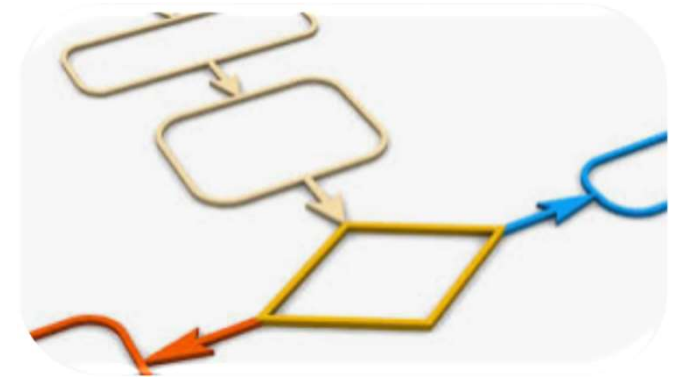
#### Estrutura de Decisão/Seletiva

(Se ... então ... Senão)

*Vamos praticar?*

1- Monte um Programa que leia um valor e escrever se é positivo, negativo ou zero.

```
1 Algoritmo "PositivoNegativo"  
2  
3 Var  
4   umValor: inteiro  
5  
6 Inicio  
7   escreval("Digite um número para saber se ele é positivo ou negativo")  
8   leia(umValor)  
9  
10  se (umValor<0) entao  
11    escreval(">>>>>É Negativo")  
12  senao  
13    escreval(">>>>>É Positivo")  
14  fimse  
15 Fimalgoritmo
```



## Lógica de Programação

### ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

#### Estrutura de Decisão/Seletiva

(Se ... então ... Senão)

*Vamos praticar?*

2- Ler o ano atual e o ano de nascimento de uma pessoa. Escrever uma mensagem que diga se ela poderá ou não votar este ano (não é necessário considerar o mês em que a pessoa nasceu).

```
1 Algoritmo "Voto"
2
3 Var
4   anoAtual, anoNascimento, idade: inteiro
5
6 Inicio
7   escreval("Digite o ano em que estamos.")
8   leia(anoAtual)
9   escreval("Digite o ano em que nasceu.")
10  leia(anoNascimento)
11
12  idade := anoAtual-anoNascimento
13  escreval("Idade:",idade)
14
15  se (idade>=18) entao
16    escreval(">>>>>Você DEVE votar.")
17  senao
18    se (idade>15) E (idade<18) entao
19      escreval(">>>>>Seu voto é FACULTATIVO.")
20    senao
21      escreval(">>>>>Você NÃO PODE VOTAR ainda.")
22    fimse
23  fimse
24 Fimalgoritmo
```

## Lógica de Programação

### ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

#### Estrutura de Decisão/Seletiva

(Se ... então ... Senão)

*Vamos praticar?*

3- Ler 3 valores (A, B e C) representando as medidas dos lados de um triângulo e escrever se formam ou não um triângulo. OBS: para formar um triângulo, o valor de cada lado deve ser menor que a soma dos outros 2 lados.

```
1 Algoritmo "TrianguloOuNao"
2
3 Var
4   ladoA, ladoB, ladoC: real
5
6 Inicio
7   escreval("Digite o valor dos 3 lados:")
8   leia(ladoA, ladoB, ladoC)
9
10  se (ladoA < ladoB + ladoC) E (ladoB < ladoA + ladoC) E (ladoC < ladoA + ladoB) entao
11    escreval(">>>>>É um triangulo")
12  senao
13    escreval(">>>>>NÃO é um triangulo")
14  fimse
15 Fimalgoritmo
```



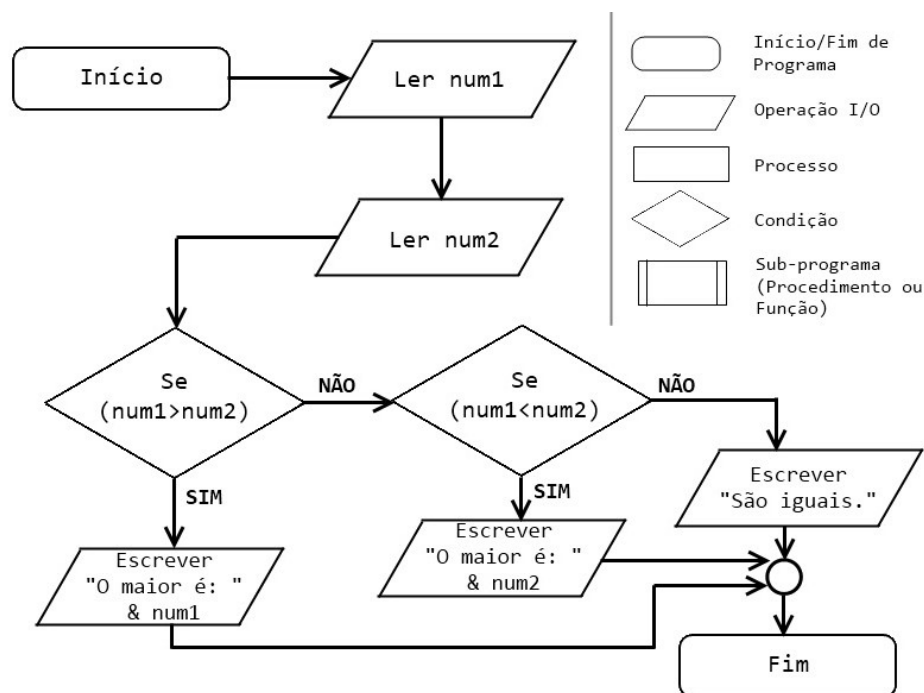
# Lógica de Programação

## ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

### Estrutura de Decisão/Seletiva

(Se ... então ... Senão) *Vamos praticar?*

4- Monte um Programa com base no fluxograma que receba dois números e mostre qual é o maior numero ou se são iguais.



```
1 Algoritmo "MaiorOuIguais"
2
3 Var
4 num1, num2 : inteiro
5
6 Inicio
7   escreva("Digite num1: ")
8   leia(num1)
9   escreva("Digite num2: ")
10  leia(num2)
11
12  se(num1>num2) entao
13    escreval("o maior é: ", num1)
14  senao
15    se(num1<num2) entao
16      escreval("o maior é: ", num2)
17    senao
18      escreval("São iguais")
19    fimse
20  fimse
21
22 Fimalgoritmo
```

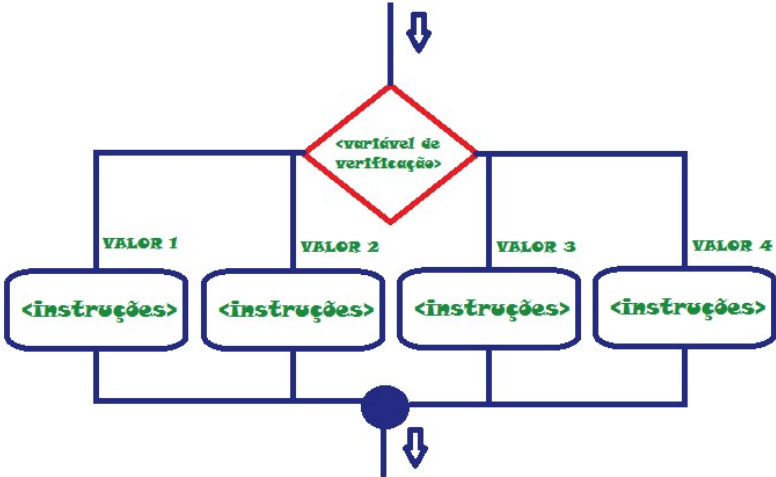


# Lógica de Programação

## ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

### Estrutura de Decisão/Seletiva

Estrutura de Decisão Múltipla do Tipo Escolha  
(Escolha ... Caso ... Senão).



escolha (<<variável>>)

caso 1

//Comandos caso o valor da variável for 1.

caso 2

//Comandos caso o valor da variável for 1.

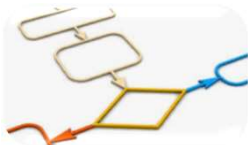
caso 3

//Comandos caso o valor da variável for 1.

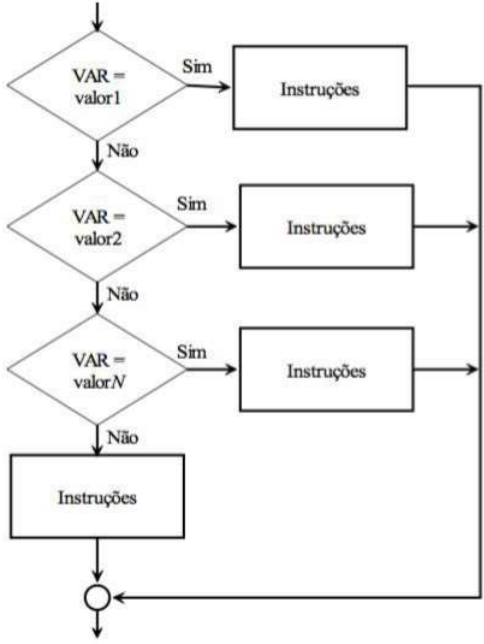
outrocaso

//Comandos caso nenhuma das opções acima correspondam //ao valor da variável.

Fimescolha



```
7 Var
8 valor : inteiro
9
10 Inicio
11 escreval("Informe um valor!")
12 leia(valor)
13 escolha (valor)
14 caso 1
15     escreval("Um")
16 caso 2
17     escreval("Dois")
18 caso 3
19     escreval("Três")
20 caso 4
21     escreval("Quatro")
22 outrocaso
23     escreval("Valor inválido!")
24 fimescolha
25
26 Fimalgoritmo
```

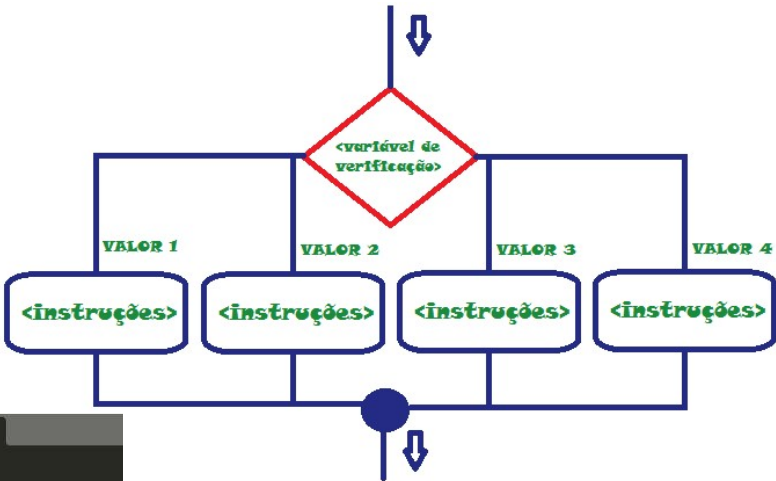


# Lógica de Programação

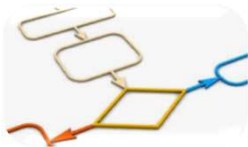
## ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

### Estrutura de Decisão/Seletiva

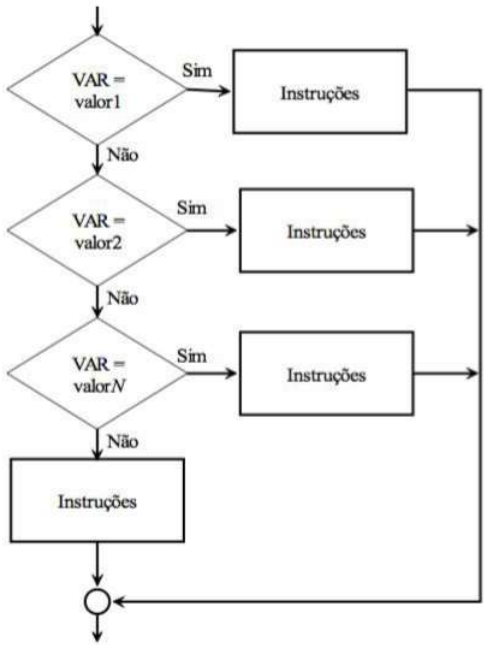
Estrutura de Decisão Múltipla do Tipo Escolha  
(Escolha ... Caso ... Senão).



```
escolha (<<variável>>)  
caso 1  
    //Comandos caso o valor da variável for 1.  
caso 2  
    //Comandos caso o valor da variável for 1.  
caso 3  
    //Comandos caso o valor da variável for 1.  
outrocaso  
    //Comandos caso nenhuma das opções acima  
    correspondam //ao valor da variável.  
Fimescolha
```



```
1 var valor = 2;  
2  
3 switch (valor){  
4     case 1:  
5         console.log("Um");  
6         break;  
7  
8     case 2:  
9         console.log("Dois");  
10        break;  
11  
12    case 3:  
13        console.log("Três");  
14        break;  
15  
16    case 4:  
17        console.log("Quatro");  
18        break;  
19  
20    default:  
21        console.log("Outro caso");  
22 }
```



## Lógica de Programação

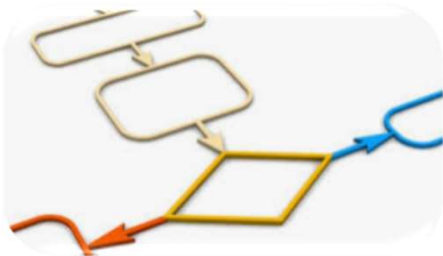
### ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

#### Estrutura de Decisão/Seletiva

(Escolha ... Caso ... Senão) *Vamos praticar?*

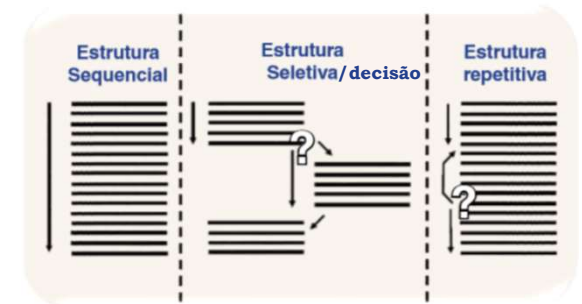
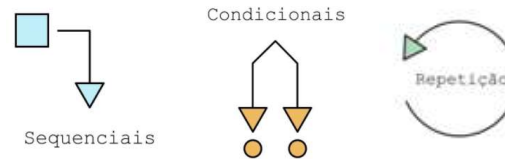
Construa um programa que leia o **salário** e o **cargo** do funcionário e aplique o **reajuste salarial** com base na tabela:

Cargo	Reajuste
Técnico	50%
Gerente	30%
Estagiário	20%



```
1 algoritmo "ReajusteSalarial"
2 var
3     salario : REAL
4     cargo   : CARACTERE
5
6 inicio
7
8     ESCREVAL ("----Digite o salário:---- ")
9     LEIA (salario)
10    ESCREVAL ("----Digite o cargo: ----")
11    ESCREVAL ("Digite 1 para Técnico")
12    ESCREVAL ("          2 para Gerente")
13    ESCREVAL ("          3 para Estagiário")
14    LEIA (cargo)
15
16    ESCOLHA cargo
17        CASO "1"
18            salario := salario *1.5
19        CASO "2"
20            salario := salario *1.3
21        CASO "3"
22            salario := salario *1.2
23        OUTROCASO
24            ESCREVAL (" ")
25            ESCREVAL ("Cargo não reconhecido")
26    FIMESCOLHA
27
28    ESCREVAL (" ")
29    ESCREVA ("Salário atualizado: ", salario)
30
31 fimalgoritmo
```

# ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO



**Comandos Compostos** → conjunto de zero ou mais comandos (ou instruções) **simples**, como atribuições e **instruções primitivas** de entrada ou saída de dados.

## Estrutura de Decisão/Seletiva

- Simples (**Se ... então**);
- Composta (**Se ... então ... senão**);
- Múltipla do Tipo Escolha (**Escolha ... Caso ... Senão**).

```
repita
    //Bloco de código a ser repetido
ate (<<expressão lógica>>)
```

```
enquanto (<<expressão lógica>>) faça
    //Bloco de código a ser repetido
Fimenquanto
```

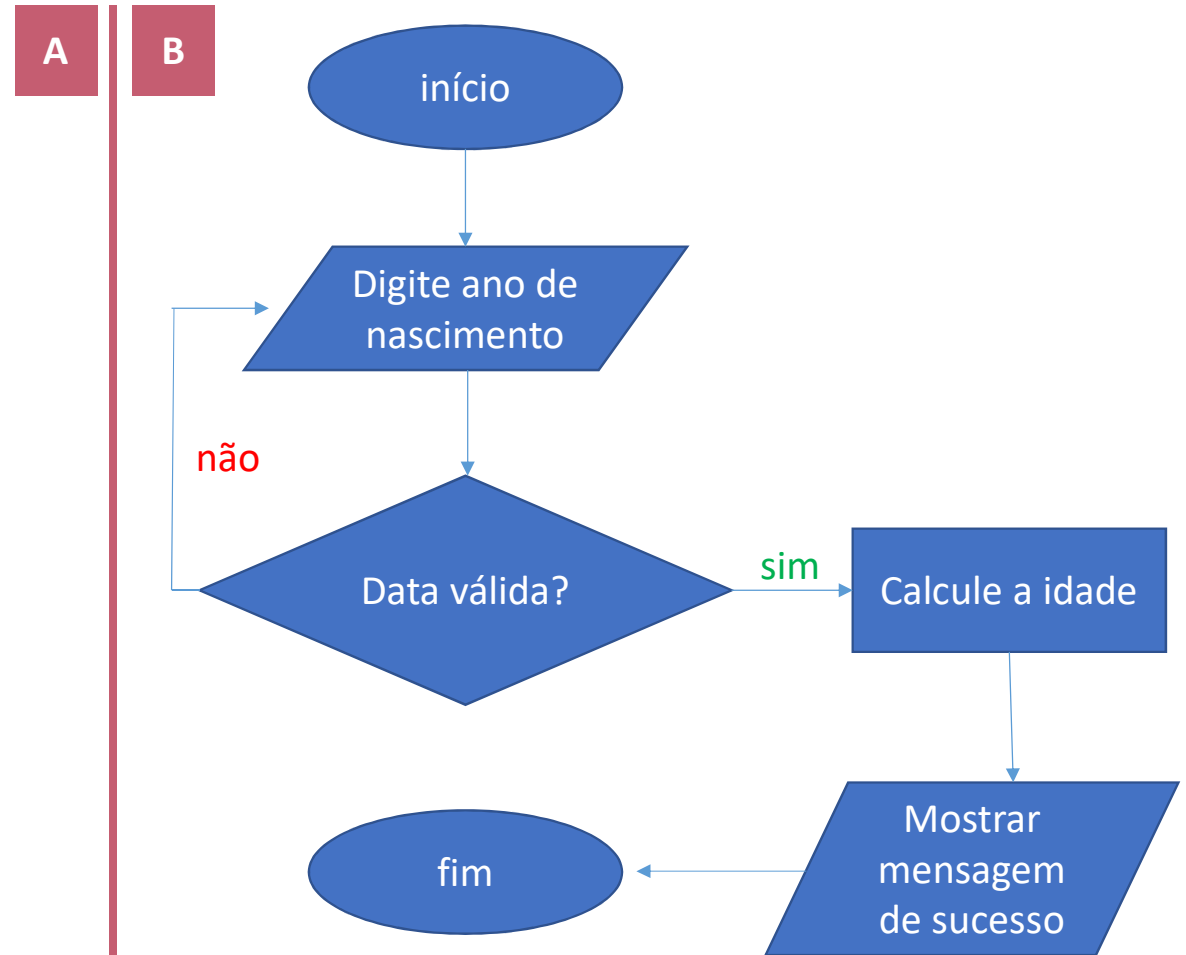
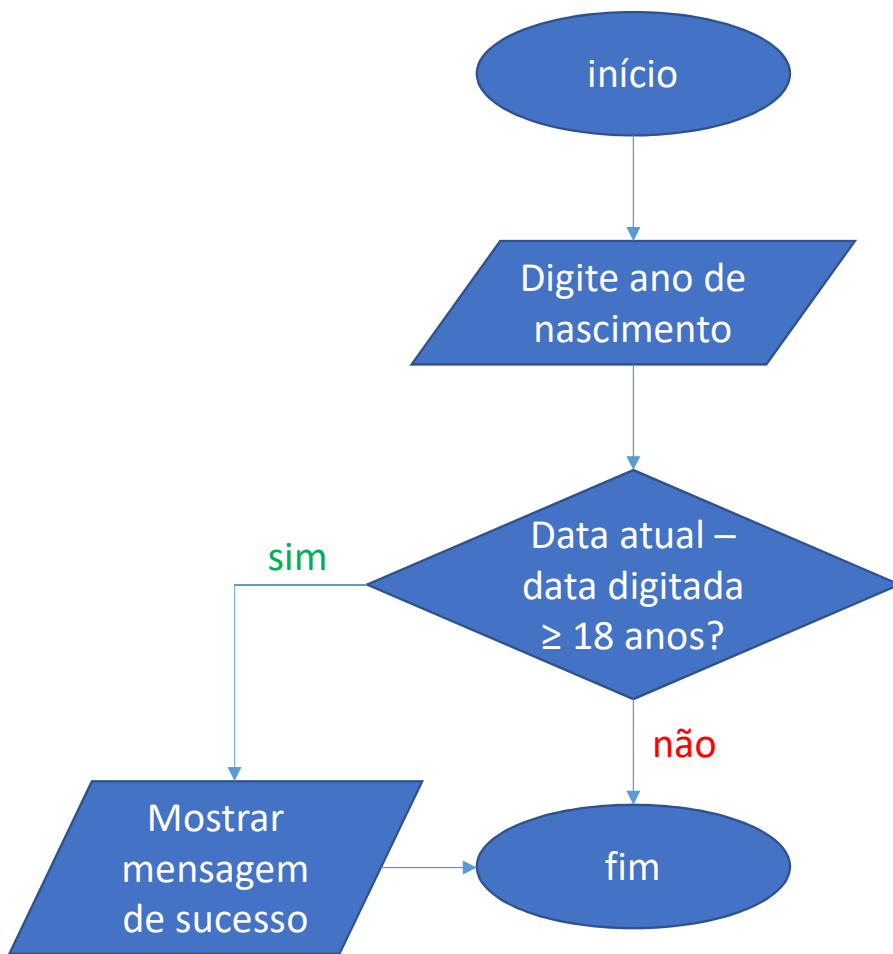
```
escolha (<<variável>>)
caso 1
    //Comandos caso o valor da variável for 1.
caso 2
    //Comandos caso o valor da variável for 1.
caso 3
    //Comandos caso o valor da variável for 1.
outrocaso
    //Comandos caso nenhuma das opções acima
    correspondam //ao valor da variável.
Fimescolha
```

```
se (<<expressão lógica>>) entao
    //Bloco de código caso a expressão for verdadeira
senao
    //Bloco de código caso a expressão for falsa
Fimse
```

tipo inteiro "i" e "j"

```
para <<variável de controle>> de <<valor de início>> até <<valor de término>>
passo <<valor de incremento ou decremento>> faça
    <<bloco a ser repetido>>
Fimpara
```

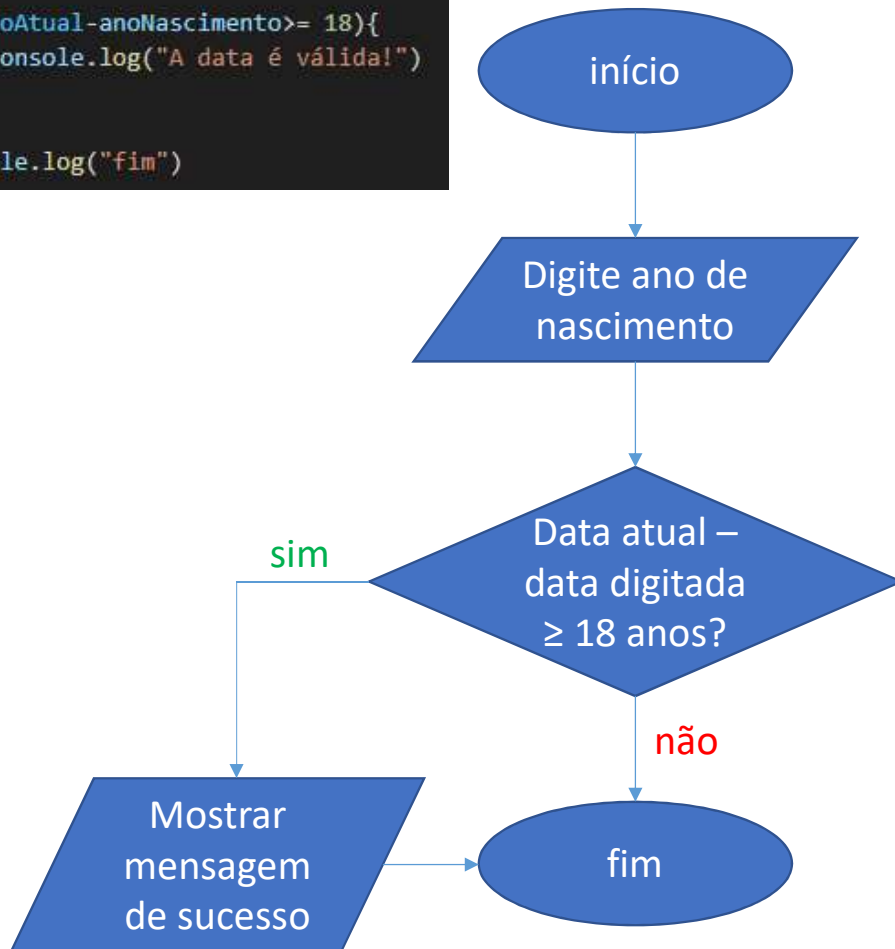




```
//Fluxograma A
var anoNascimento = 2015
const anoAtual = 2021

if(anoAtual-anoNascimento>= 18){
  console.log("A data é válida!")
}

console.log("fim")
```



A

B

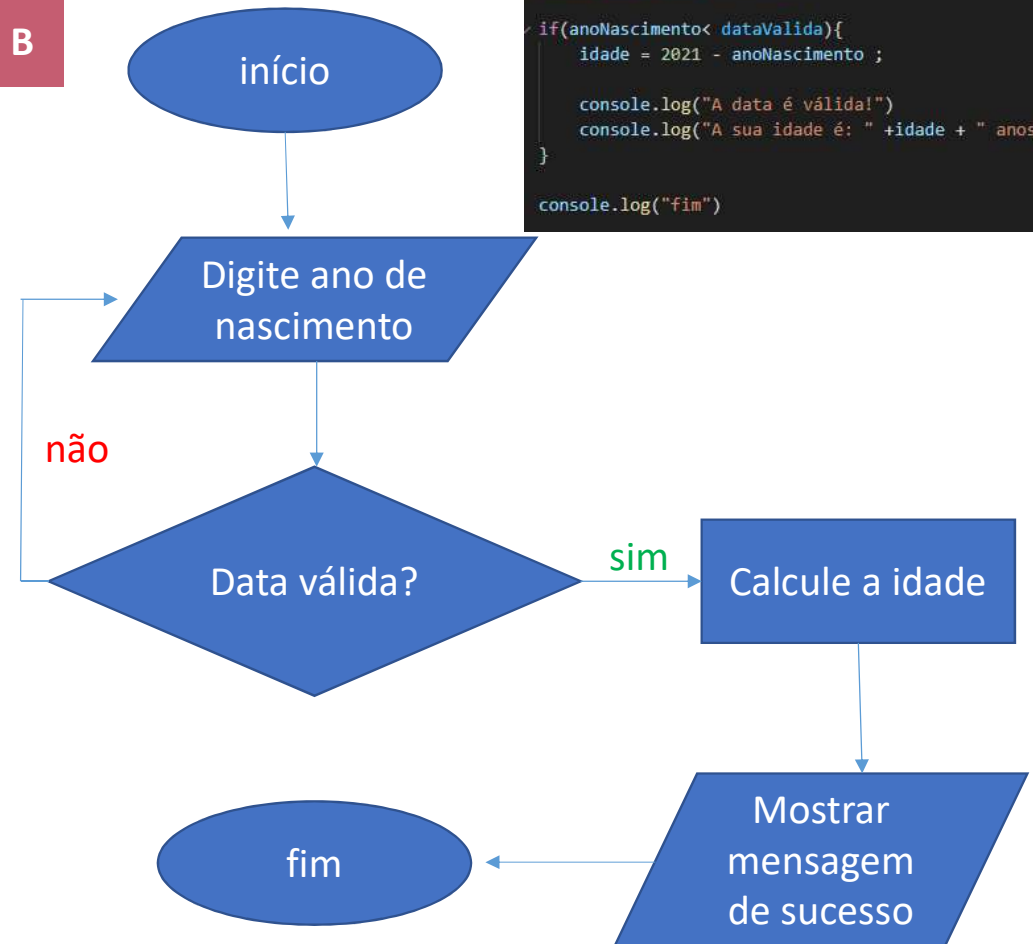
```
//Fluxograma B
var anoNascimento = 1986

//2021-18 = 2003
const dataValida = 2003

if(anoNascimento< dataValida){
  idade = 2021 - anoNascimento ;

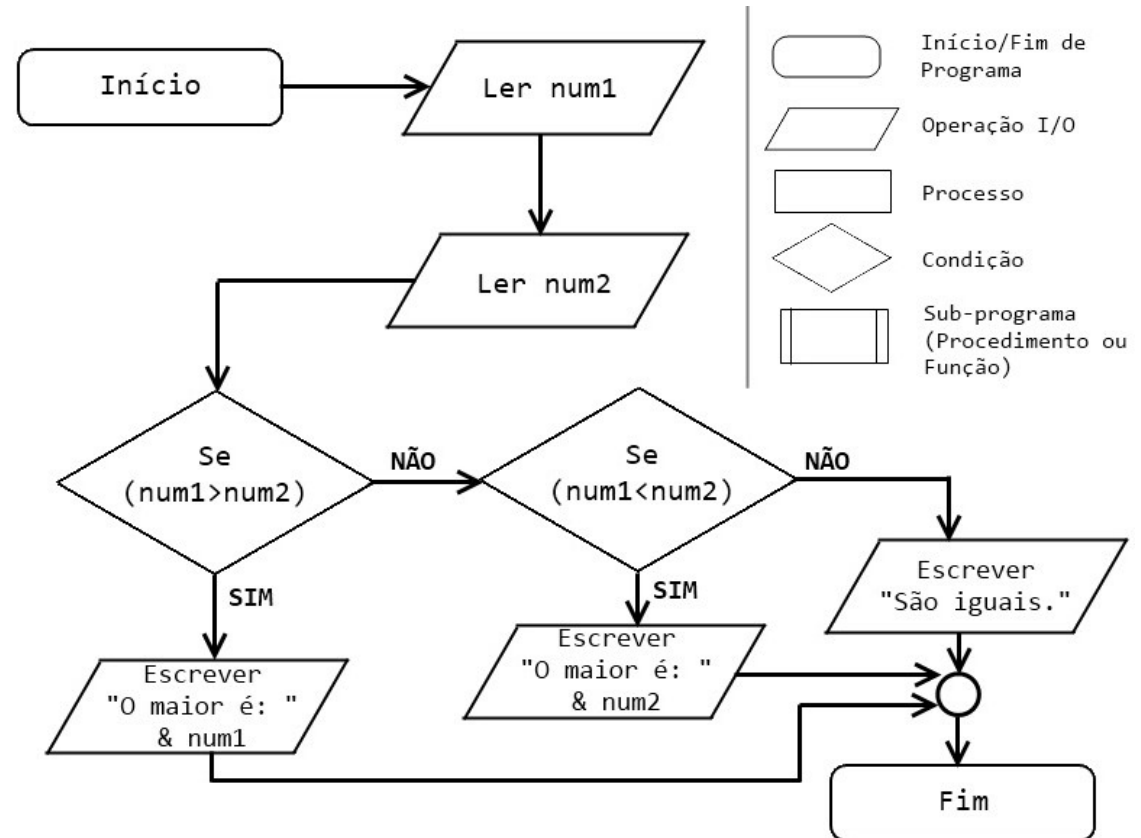
  console.log("A data é válida!")
  console.log("A sua idade é: " +idade + " anos.")
}

console.log("fim")
```



Monte um Programa que receba dois números e mostre qual é o maior numero ou se são iguais.

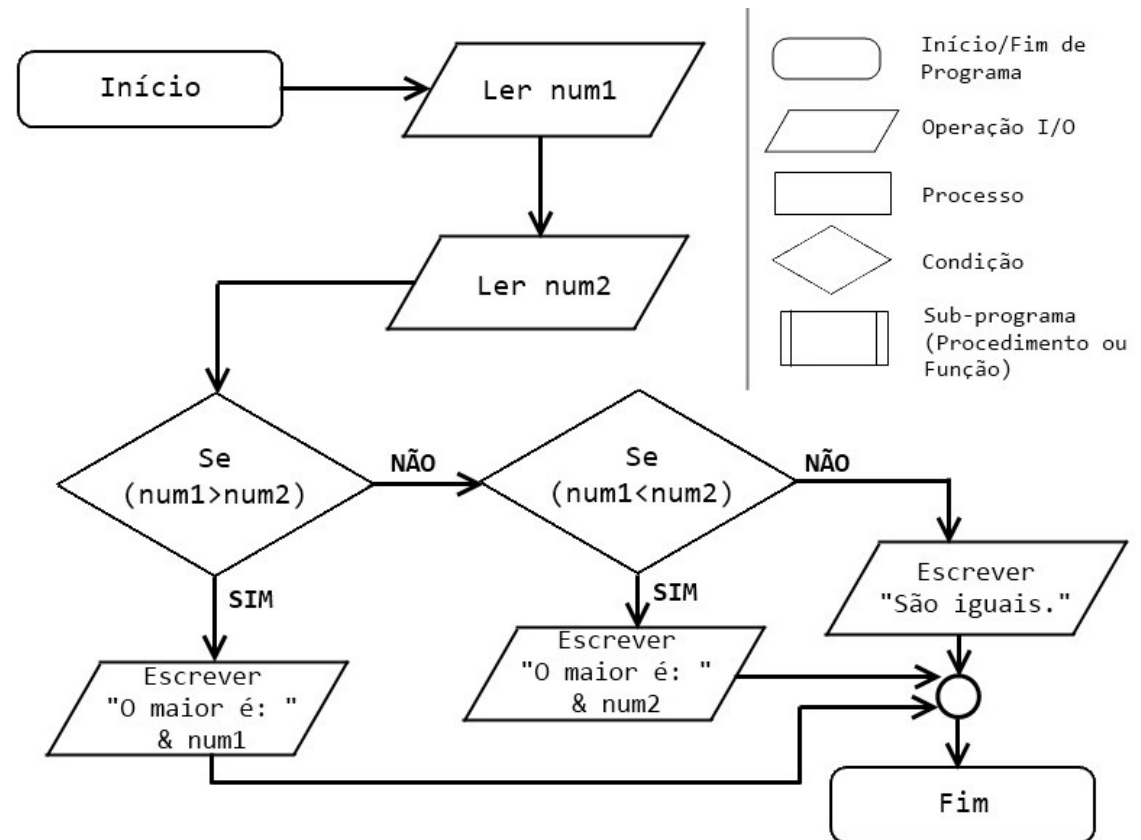
```
1 Algoritmo "MaiorOuIguais"
2
3 Var
4 num1, num2 : inteiro
5
6 Inicio
7   escreva("Digite num1: ")
8   leia(num1)
9   escreva("Digite num2: ")
10  leia(num2)
11
12  se (num1 > num2) entao
13    escreval("o maior é: ", num1)
14  senao
15    se (num1 < num2) entao
16      escreval("o maior é: ", num2)
17    senao
18      escreval("São iguais")
19  fimse
20  fimse
21
22 Fimalgoritmo
```



Monte um Programa que receba dois números e mostre qual é o maior numero ou se são iguais.

```
var num1 = 10
var num2 = 10

if(num1>num2){
  console.log("O maior é: " +num1)
}else{
  if(num1<num2){
    console.log("O maior é: " +num2)
  }else{
    console.log("São iguais.")
  }
}
```





## Atividade - Codificação de um sistema de contagem e classificação.


A empresa Savinis, focada no desenvolvimento de software de alta performance, contratou você para realizar a programação de um sistema de controle de alunos de uma escola. O sistema deverá funcionar da seguinte forma:

Considere a quantidade de alunos presentes na sala, percorra do zero até o número total de alunos e retornar os seguintes resultados:

- se o número for par, escreva 'par' e o número correspondente.
- se o número for ímpar, escreva 'ímpar' e o número correspondente.
- se o número for zero, escreva 'zero' e o número correspondente.

Para criação desse sistema de cadastro, você, enquanto programador(a) responsável, deverá solucionar os seguintes desafios:

- Aplicar técnicas de programação e ferramentas para desenvolvimento do código.
- Aplicar linguagem de programação JavaScript para implementação do sistema.

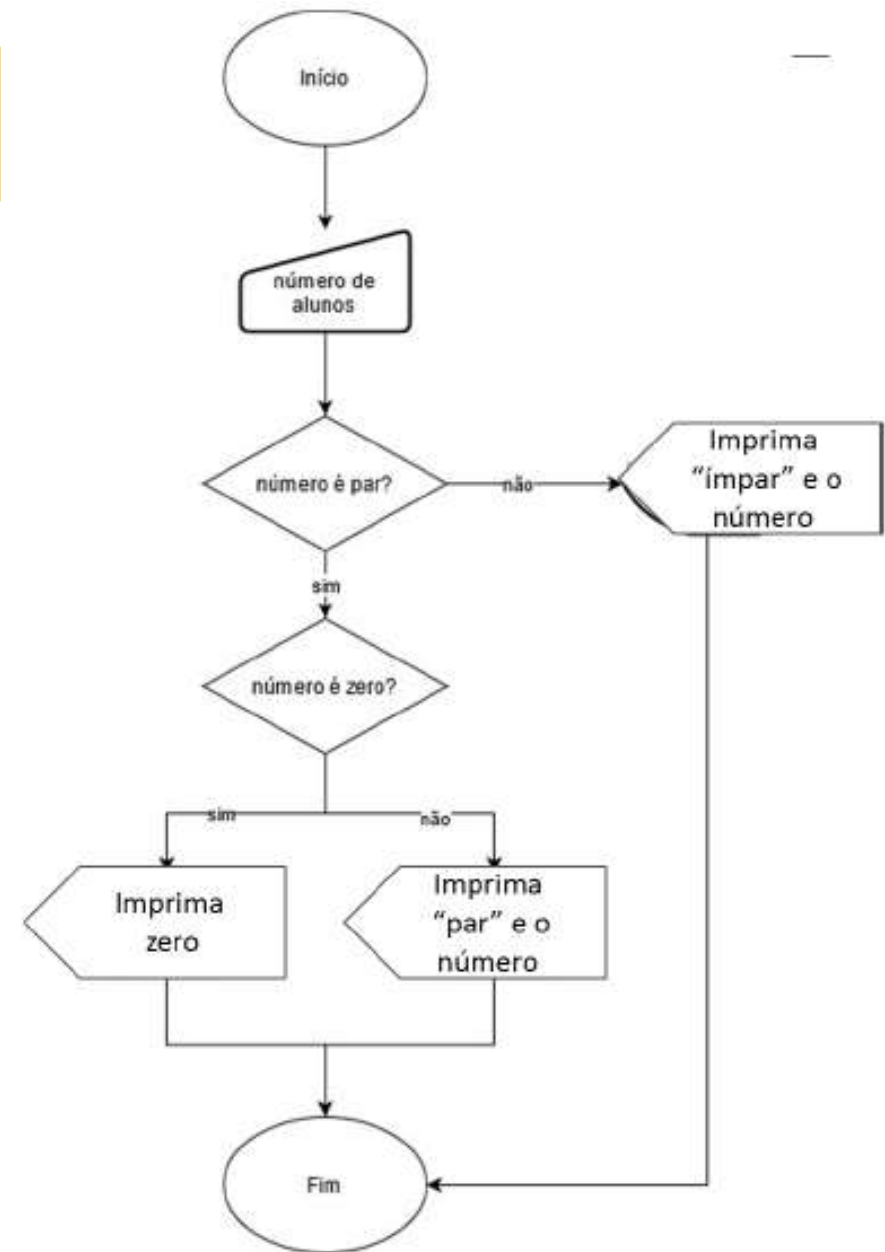


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
⚡ Administrador@DESKTOP-H4GJH1M C:\senai\
> node .\estrutura-condicionais.js
o número é zero
o número é ímpar1
o número é par2
o número é ímpar3
o número é par4
o número é ímpar5
o número é par6
o número é ímpar7
o número é par8
o número é ímpar9
o número é par10
o número é ímpar11
o número é par12
o número é ímpar13
o número é par14
```

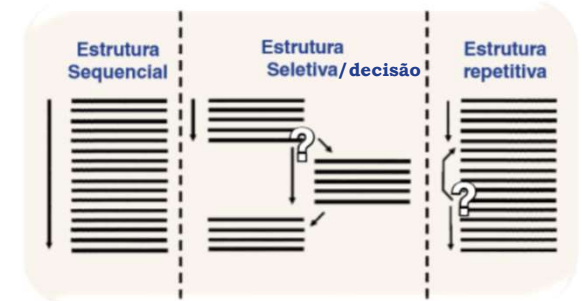
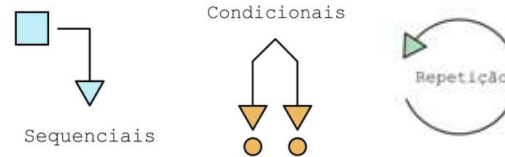
## Atividade - Codificação de um sistema de contagem e classificação.

Segue exemplo de fluxograma do algoritmo :

Falta o looping nesse fluxograma →



# Revisão



## ESTRUTURA DE CONTROLE E FLUXO DE EXECUÇÃO

**Comandos Compostos** → conjunto de zero ou mais comandos (ou instruções) **simples**, como atribuições e **instruções primitivas** de entrada ou saída de dados.

### Estrutura de Decisão/Seletiva

- Simples (**Se ... então**);
- Composta (**Se ... então ... senão**);
- Múltipla do Tipo Escolha (**Escolha ... Caso ... Senão**).

```
repita
    //Bloco de código a ser repetido
ate (<<expressão lógica>>)
```

```
enquanto (<<expressão lógica>>) faça
    //Bloco de código a ser repetido
Fimenquanto
```

```
escolha (<<variável>>)
caso 1
    //Comandos caso o valor da variável for 1.
caso 2
    //Comandos caso o valor da variável for 1.
caso 3
    //Comandos caso o valor da variável for 1.
outrocaso
    //Comandos caso nenhuma das opções acima
    correspondam //ao valor da variável.
Fimescolha
```

```
se (<<expressão lógica>>) entao
    //Bloco de código caso a expressão for verdadeira
senao
    //Bloco de código caso a expressão for falsa
Fimse
```

```
tipo inteiro "i" e "j"
para <<variável de controle>> de <<valor de início>> até <<valor de término>>
passo <<valor de incremento ou decremento>> faça
    <<bloco a ser repetido>>
Fimpara
```