

INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE

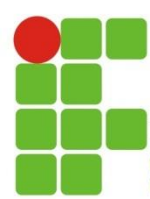
# Estruturas de Controle de Fluxo

## Estruturas de Seleção

**Givanaldo Rocha**

[givanaldo.rocha@ifrn.edu.br](mailto:givanaldo.rocha@ifrn.edu.br)

<http://docente.ifrn.edu.br/givanaldorochoa>



# Conceito

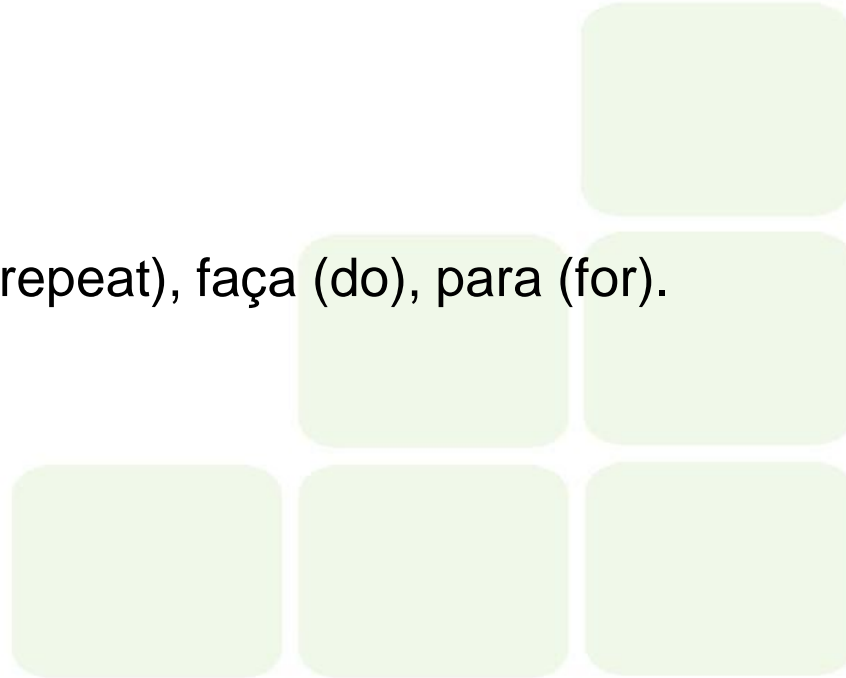
---

- Até agora, desenvolvemos algoritmos que nos permitem a entrada de dados, algum processamento desta informação e uma saída de dados.
- Porém, estes algoritmos ainda são incapazes de tomar decisões ou repetir instruções uma determinada quantidade de vezes.
- Imagine ter que imprimir uma sequência de 1000 números.
- Ou como fazer um algoritmo que me diga se estou aprovado ou não?

# Conceito

---

As estruturas de controle de fluxo permitirão desenvolver algoritmos capazes de testar expressões e, a partir delas, seguir pelas instruções de forma não linear.

- **Seleção:** se (if), escolha (switch).
  - **Repetição:** enquanto (while), repita (repeat), faça (do), para (for).
- 

## Comando **SE**

- Estrutura de controle de fluxo, que executa um ou vários comandos se a condição testada for verdadeira e, em alguns casos, executando um ou vários comandos se for falsa.
- Também conhecido como **desvio condicional**.

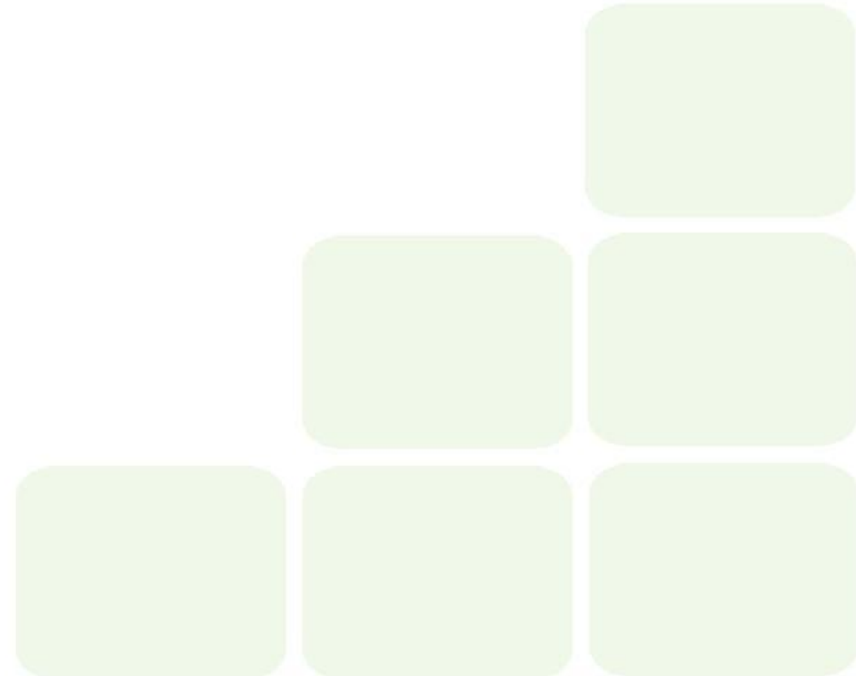
## Sintaxe

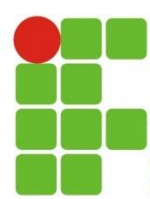
```
se <expressão-lógica> entao  
    <seqüência-de-comandos>  
fimse
```



# Comando SE

- A *<expressão-lógica>* é uma condição que será testada pela Unidade Lógica-Aritmética (ULA), devolvendo como resposta **Verdadeiro** ou **Falso**.
- Na *<expressão-lógica>* podemos utilizar os **operadores relacionais** e/ou os **operadores lógicos**.
- Ex:
  - ✓  $\text{num1} \geq 60$
  - ✓  $\text{num1} \leq 60$  E  $\text{num1} \geq 30$





# Comando SE

## Exemplo SE:

```
algoritmo "maior_de_idade"  
var idade: inteiro  
inicio  
    escreva("Informe sua idade: ")  
    leia(idade)  
    se idade >=18 entao  
        escreval("Você é maior de idade")  
    fimse  
fimalgoritmo
```



# Comando SE

- No exemplo anterior, o algoritmo apenas informará se o usuário é **maior de idade**. Se ele não for, seria bom que também recebesse uma mensagem informando.
- Neste caso, deve-se utilizar o comando SE com a sintaxe um pouco diferente. Ele possuirá o comando SENAO.
- Sintaxe:

```
se <expressão-lógica> entao  
    <seqüência-de-comandos-1>  
senao  
    <seqüência-de-comandos-2>  
fimse
```

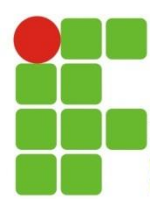
# Comando SE

---

- A *<seqüência-de-comandos-2>* será executada **SOMENTE** no caso de *<expressão-lógica>* ser FALSO.
- Desta forma, o algoritmo não executará a(s) instrução(ões) presentes em *<seqüência-de-comandos-1>*.







# Comando SE

## Exemplo SE ... SENAO:

algoritmo “Maior\_de\_idade”

var idade: inteiro;

inicio

    escreva(“Informe sua idade:”);

    leia(idade);

    se idade  $\geq$  18 entao

        escreval(“Você é maior de idade”);

    senao

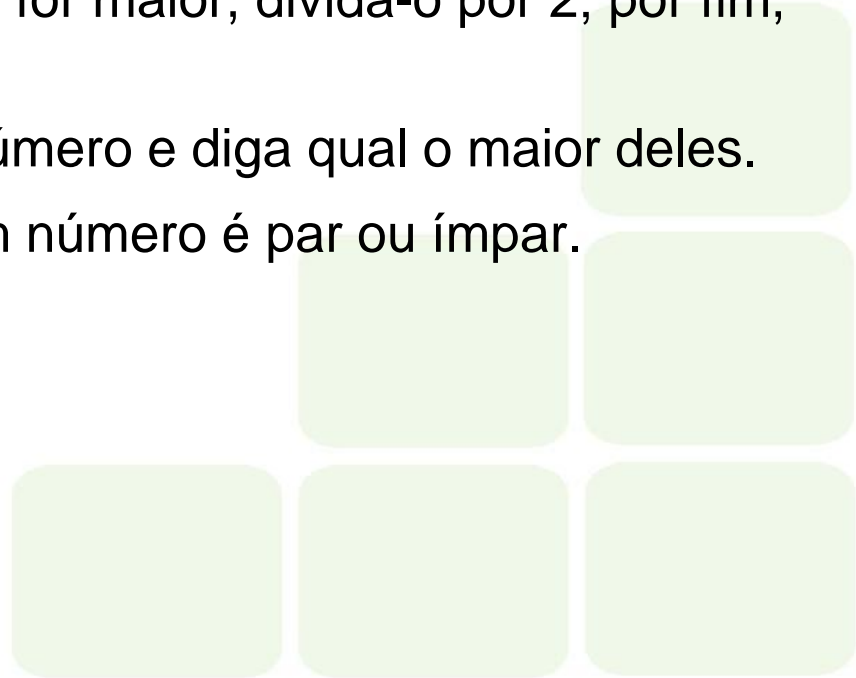
        escreval(“Você é menor de idade”);

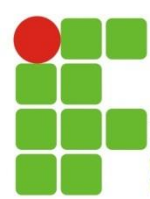
    fimse

fimalgoritmo

# Exercícios (SE)

---

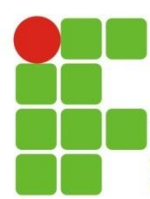
1. Escreva um algoritmo que leia um número e escreva-o caso seja maior ou igual a 10.
  2. Escreva um algoritmo que leia dois inteiros e efetue a adição; caso o resultado seja maior que 20, mostre este resultado
  3. Escreva um algoritmo que leia um número; caso este número seja menor que 10, multiplique-o por 2; se for maior, divida-o por 2; por fim, mostre o resultado.
  4. Escreva um algoritmo que leia três número e diga qual o maior deles.
  5. Escreva um algoritmo que diga se um número é par ou ímpar.
- 



# Comando ESCOLHA

- Testa sucessivamente o valor de uma expressão contra uma lista de constantes (inteiras ou de caracteres).
- Quando o valor coincide, os comandos associados àquela constante são executados.
- Sintaxe:

```
escolha <expressão-de-seleção>  
  caso <constante1>  
    <seqüência-de-comandos-1>  
  caso <constante2>  
    <seqüência-de-comandos-2>  
  ...  
  outrocaso  
    <seqüência-de-comandos-extra>  
fimescolha
```



# Exemplo (ESCOLHA)

```
algoritmo "menu"  
var  
    opc: caractere  
Inicio  
    escreval ("=====MENU=====")  
    escreval ("1 - para informática")  
    escreval ("2 - para eletrônica")  
    escreval ("3 - para comercio")  
    leia (opc)  
    escolha opc  
        caso "1"  
            escreva ("O Código do seu curso é 401!")  
        caso "2"  
            escreva ("O Código do seu curso é 206!")  
        caso "3"  
            escreva ("O Código do seu curso é 506!")  
    fimescolha  
fimalgoritmo
```

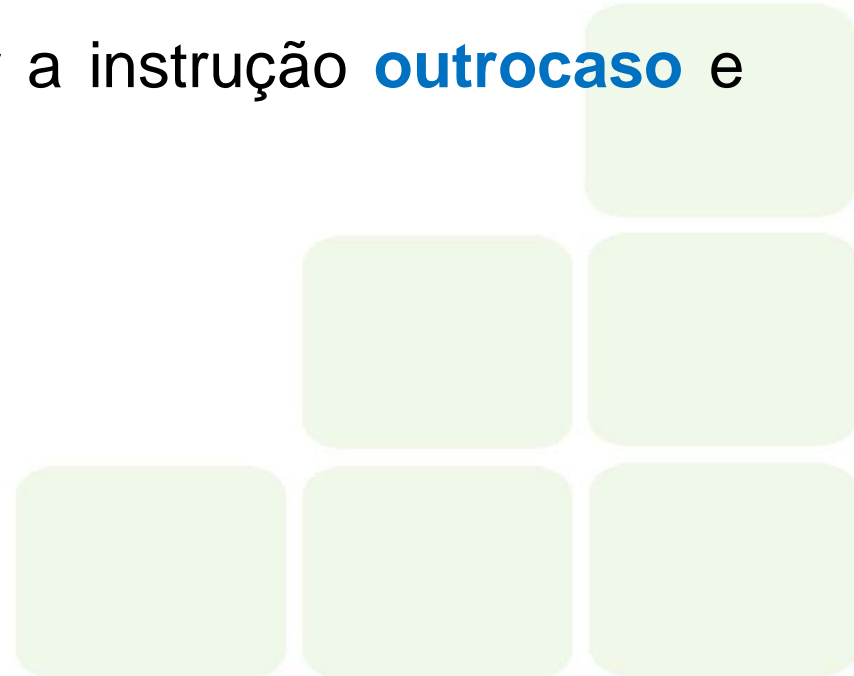
```
=====MENU=====  
1 - para informática  
2 - para eletrônica  
3 - para comercio  
|
```

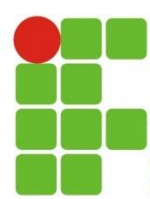


# Comando ESCOLHA

---

- Se o usuário inserir uma informação que o comando não está aguardando (em seus elementos “caso”), como por exemplo, se ele digitasse a opção 5 no exemplo anterior, o algoritmo não faria nada.
- Nestes casos, podemos usar a instrução **outrocaso** e definir uma ação padrão.





# Exemplo (ESCOLHA)

```
...
escreval ("=====MENU=====")
escreval ("1 - para informática")
escreval ("2 - para eletrônica")
escreval ("3 - para comercio")
leia (opc)
escolha opc
    caso "1"
        escreva ("O Código do seu curso é 401!")
    caso "2"
        escreva ("O Código do seu curso é 206!")
    caso "3"
        escreva ("O Código do seu curso é 506!")
    outrocaso
        escreva ("Opção Inválida!")
fimescolha
fimalgoritmo
```

```
=====MENU=====
1 - para informática
2 - para eletrônica
3 - para comercio
5
Opção inválida
```

# Exercício (ESCOLHA)

---

Escreva um algoritmo que receba 2 números do usuário e mostre na tela o seguinte menu:

(a)dição

(s)ubtração

(m)ultiplicação

(d)ivisão

Após ser informada a opção, o algoritmo deve operar as 2 variáveis (ex:  $x+y$  para “a” ou  $x/y$  para “d”) e mostrar o resultado para o usuário.

Caso o usuário digite uma opção inválida, deve ser exibida uma mensagem de erro.