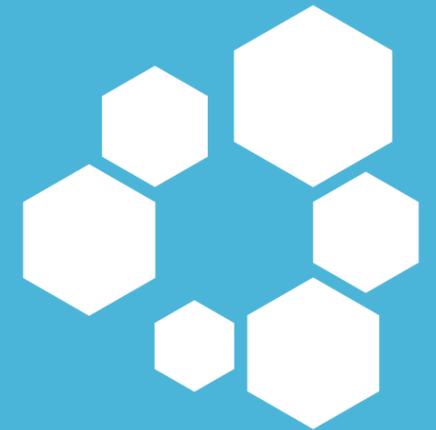


# Introdução à Computação Condicionais

Prof. Dr. Marcos Paulino Roriz Junior (marcosroriz@ufg.br)



**UFG**

UNIVERSIDADE  
FEDERAL DE GOIÁS



ENGENHARIA DE  
TRANSPORTES

FCT  
FACULDADE DE  
CIÊNCIAS E TECNOLOGIA



**UFG**  
UNIVERSIDADE  
FEDERAL DE GOIÁS

# Fluxo de execução

Sequência de passos que resolvem um problema específico

- Escritas no **nível de competência do agente** que irá executá-las
- Descrição finita, tempo finito, espaço finito

Exemplo: Algoritmo para “média de aluno”

1. Obtenha notas do aluno PP1, PT1, PP2, PT2
2. Calcule Média  $M$
3. Se  $M \geq 6$  então o aluno foi aprovado
4. Caso contrário, aluno foi reprovado



# Condição

- Condicionais em pseudocódigo: simples e composta.
- Simples:
  1. ...
  2. **SE (condição)**
  3. **ENTÃO INÍCIO**
  4.       passo 1;
  5.       passo 2;
  6.       passo 3;
  7.       **FIM**



# Condição

- Condicionais em pseudocódigo: simples e composta.
- Simples:

1. ...

2. **SE (condição)**

3. **ENTÃO INÍCIO**

4. passo 1;

5. passo 2;

6. passo 3;

7. **FIM**

```
passo 1;  
passo 2;  
passo 3;
```

= bloco de instruções

Se **condição** for verdadeiro execute o bloco de instruções  
Senão, não faça nada. (continua na linha 6)



```
1. INÍCIO_ALGORITMO
2. DECLARE pressaodesejada, pressaoatual, dif NUMÉRICO
3. LEIA pressaodesejada
4. LEIA pressaoatual
5. dif <- pressaodesejada - pressaoatual
6. SE dif > 0
7. ENTÃO INÍCIO
8.     ESCREVA "Aumenta a pressão em", dif
9.     FIM
10. SE dif < 0
11. ENTÃO INÍCIO
12.     ESCREVA "Diminua a pressão em", dif
13.     FIM
14. FIM_ALGORITMO
```

## Condição: Exemplo



# Condicionais: NA LINGUAGEM C

## ▪ Instrução: if

1. ...

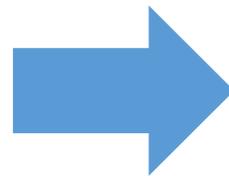
2. **SE (condição)**

3. **ENTÃO INÍCIO**

4. **ESCREVA x**

5. **ESCREVA y**

6. **FIM**



1. ...

2. **if (condição)**

3. **{**

4. `printf("%d", x);`

5. `printf("%f", y);`

6. **}**



# Condicionais: NA LINGUAGEM C

## ▪ Instrução: if

1. ...

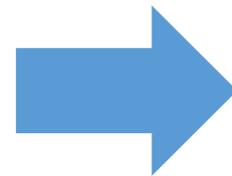
2. **SE (condição)**

3. **ENTÃO INÍCIO**

4. **ESCREVA x**

5. **ESCREVA y**

6. **FIM**



1. ...

```
2. if (condição) {
```

```
3.     printf("%d", x);
```

```
4.     printf("%f", y);
```

```
5. }
```



# Condição: Exemplo

1. INÍCIO\_ALGORITMO
2. DECLARE `pressaodesejada`, `pressaoatual`, `dif` NUMÉRICO
3. LEIA `pressaodesejada`
4. LEIA `pressaoatual`
5. `dif`  $\leftarrow$  `pressaodesejada` - `pressaoatual`
6. SE `dif` > 0
7. ENTÃO INÍCIO
8.     ESCREVA "Aumenta a pressão em", `dif`
9.     FIM
10. SE `dif` < 0
11. ENTÃO INÍCIO
12.     ESCREVA "Diminua a pressão em", `dif`
13.     FIM
14. FIM\_ALGORITMO

```
1  #include <stdio.h>
2
3  int main() {
4      int pd, pa, dif;
5      scanf("%i", &pd);
6      scanf("%i", &pa);
7      dif = pd - pa;
8
9      if (dif > 0)
10     {
11         printf("Aumenta a pressao em %i", dif);
12     }
13
14     if (dif < 0)
15     {
16         printf("Diminui a pressao em %i", dif);
17     }
18 }
```



## Exemplo: ?

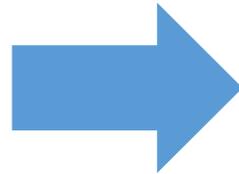
```
1. #include <stdio.h>
2. int main () {
3.     int a = 0;
4.     int b = a + 1;
5.
6.     if (a < b) {
7.         a = a + 1;
8.     }
9.     printf ("%d", a);
10.    return 0;
11. }
```



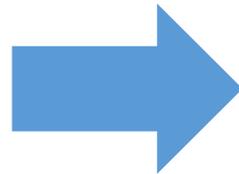
# Condicionais simples

- Instrução: **if**
- IF com única instrução tem bloco { } opcional

1. ...  
 2. SE (condição)  
 3. ENTÃO INÍCIO  
 4.       ESCREVA x  
 5.       ESCREVA y  
 6.       FIM



1. ...  
 2. **if** (condição) {  
 3.     printf(“%d”, x);  
 4. }



1. ...  
 2. **if** (condição)  
 3.     printf(“%d”, x);  
 4.



# Equivalentes

```
1. #include <stdio.h>
2. int main () {
3.     int a = 0;
4.     int b = a + 1;
5.
6.     if (a < b) {
7.         a = a + 1;
8.     }
9.     printf ("%d", a);
10.    return 0;
11.}
```

```
1. #include <stdio.h>
2. int main () {
3.     int a = 0;
4.     int b = a + 1;
5.
6.     if (a < b)
7.         a = a + 1;
8.     printf ("%d", a);
9.     return 0;
10.}
```

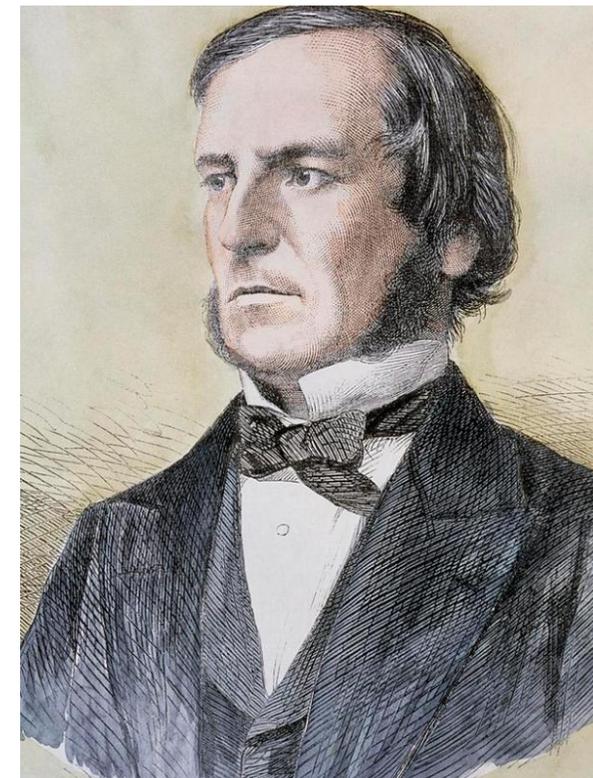


# Condição = Expressão booleana

Uma expressão booleana é uma expressão que, quando avaliada, resulta no valor falso ou verdadeiro.

C não tem um tipo de dado específico para armazenar valores booleanos:

- **FALSO** = 0
- **VERDADEIRO** = 1 (Não-zero), mas pode ser 1, -1, 10, 3.14, ...;



George Boole (fonte foto: Wikipedia)



# Operadores lógicos

- Uma expressão booleana é construída através da utilização de operadores relacionais.
  - maior que ( $>$ ),
  - menor que ( $<$ ),
  - maior ou igual a ( $\geq$ ),
  - menor ou igual a ( $\leq$ ),
  - diferente de ( $\neq$ ),
  - igual a ( $=$ ).
- Todos estes operadores comparam dois operandos, resultando no valor 0 (falso) ou 1 (verdadeiro).



# Exemplo

- `int x = 20;`
- `x < 30` → 1 (verdadeiro)
- `x > 30` → 0 (falso)
- `x >= 30` → 0 (falso)
- `x <= 30` → 1 (verdadeiro)
- `x >= 20` → 1 (verdadeiro)
- `x <= 20` → 1 (verdadeiro)



## Exemplo

- `int x = 20;`
- `x == 20` → 1 (verdadeiro)
- `x == 30` → 0 (falso)
- `x != 20` → 0 (falso)
- `x != 30` → 1 (verdadeiro)



# Exemplo

- `int x = 20;`
- `int y = 20;`
- `x < y`
- `x > y`
- `x >= y`
- `x <= y`
- `x == y`
- `x != y`



1. ALGORITMO
2. DECLARE p1, p2, p3, t, m NUMÉRICO
3. LEIA p1
4. LEIA p2
5. LEIA p3
6. LEIA t
7.  $m \leftarrow 0.7 (p1+p2+p3)/3 + 0.3*t$
8. SE  $m \geq 6$
9. ENTÃO INÍCIO
10.       ESCREVA "Aprovado", m
11.       FIM
12. FIM\_ALGORITMO



```
1  #include <stdio.h>
2
3  int main() {
4      int p1, p2, p3, t;
5      float m;
6      scanf("%i", &p1);
7      scanf("%i", &p2);
8      scanf("%i", &p3);
9      scanf("%i", &t);
10     m = (0.7 * (p1 + p2 + p3) / 3) + (0.3 * t);
11
12     if (m >= 6)
13     {
14         printf("Aprovado %f", m);
15     }
16 }
```



# Expressão Booleana

Expressões booleanas também podem ser formadas com operadores lógicos.

- conjunção (&&) E
- disjunção (||) OU
- negação (!) NÃO



# Conjunção (AND) &&

- Combinação (pseudocódigo) = **expr1 E expr2 E expr3**
- Combinação (C) = **expr1 && expr2 && expr3**
- Só é verdadeira se todas as expressões forem verdades
- `int x = 20;`
- `(x < 30) && (x != 20) → V E F = F`
- `(x < 30) && (x >= 20) → V E V = V`
- `(x > 30) && (x == 20) → F E V = F`
- `(x > 30) && (x != 20) → F E F = F`



# EXEMPLO: TODAS AS NOTAS MAIORES QUE 6

1. ALGORITMO
2. DECLARE p1, p2, p3 NUMÉRICO
3. LEIA p1
4. LEIA p2
5. LEIA p3
- 6.
7. SE  $p1 \geq 6$  E  $p2 \geq 6$  E  $p3 \geq 6$
8. ENTÃO INÍCIO
9.       ESCREVA "Aprovado"
10.       FIM
11. FIM\_ALGORITMO



# EXEMPLO: TODAS AS NOTAS MAIORES QUE 6

```
1  #include <stdio.h>
2
3  int main() {
4      int p1, p2, p3;
5      scanf("%d", &p1);
6      scanf("%d", &p2);
7      scanf("%d", &p3);
8
9      if (p1 >= 6 && p2 >= 6 && p3 >= 6) {
10         printf("Aprovado");
11     }
12 }
```



# CONDIÇÃO (OR) DISJUNÇÃO / OU

- Combinação (pseudocódigo) = **expr1 OU expr2 OU expr3**
- Combinação (C) = **expr1 || expr2 || expr3**
- É verdadeira se pelo menos uma das expressões forem verdadeiras

- `int x = 20;`
- `(x < 30) || (x != 20) → V OU F = V`
- `(x < 30) || (x >= 20) → V OU V = V`
- `(x > 30) || (x == 20) → F OU V = V`
- `(x > 30) || (x != 20) → F OU F = F`



# EXEMPLO: CINEMA

1. ALGORITMO
2. DECLARE idade, acompanhado NUMÉRICO
3. LEIA idade
4. LEIA acompanhado
- 5.
6. SE idade  $\geq 18$  OU (idade  $\geq 12$  E acompanhado == 1)
7. ENTÃO INÍCIO
8.       ESCREVA "Pode assistir o filme"
9.       FIM
10. FIM\_ALGORITMO



```
1  #include <stdio.h>
2  ▼ int main () {
3      int idade, acompanhado;
4      printf("Idade e Acompanhado:");
5      scanf("%d %d", &idade, &acompanhado);
6
7  ▼  if (idade >= 18 || (idade >= 12 && acompanhado)) {
8      printf("Pode assistir o filme");
9      }
10
11     return 0;
12 }
13
```



# NEGAÇÃO

- !expr Inverte o valor lógico da expr
- Em pseudocódigo = **não**
  
- `int x = 20;`
- `!(x < 30)`       $\rightarrow ! (1) = 0$  (falso)
- `!(x > 30)`       $\rightarrow ! (0) = 1$  (verdadeiro)
- `!(x == 20)`
- `!(x != 20)`



# NEGAÇÃO

- !expr Inverte o valor lógico da expr
- Em pseudocódigo = não
  
- float media = 6.5;
- if (!(media >= 6)) { → **!(1) = 0 (falso)**
- printf(“Não precisa fazer nota final”);
- }



# RESUMO

Uma expressão booleana é uma expressão que, quando avaliada, resulta no valor falso ou verdadeiro.

- Relações ( $>$ ,  $<$ ,  $\geq$ ,  $\leq$ ,  $\neq$ ,  $==$ )
- Operadores Lógicos: NÃO (!), E (&&), OU (||)
- **FALSO** = 0
- **VERDADEIRO** = 1 (Não-zero), mas pode ser 1, -1, 10, 3.14, ...;





# RESUMO

Os principais operadores lógicos são:  $\&$ ,  $||$  e  $!$ , que significam *e*, *ou*, *não* e são usados para conjunção, disjunção e negação, respectivamente.

TABELA E	TABELA OU	TABELA NÃO
$V e V = V$	$V ou V = V$	Não $V = F$
$V e F = F$	$V ou F = V$	Não $F = V$
$F e V = F$	$F ou V = V$	
$F e F = F$	$F ou F = F$	



**Questão 12.** (UFTM-2018 Engenheiro) Aponte, dentre as alternativas, os resultados da resolução da seguinte expressão lógica escrita na linguagem C, para os valores de A, B e C, definidos nos cenários I, II e III:

$(A \ \&\&B) \ \&\& \ ((C \ || \ A \ || \ B) \ || \ (!A \ \&\& \ C))$

I: A=true, B=true, C=false

II: A=false, B=true, C=true

III: A=false, B=true, C=false

(a) I: true, II: false, III: false.

(b) I: true, II: true, III: false.

(c) I: false, II: false, III: false.

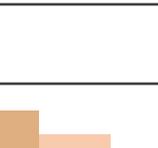
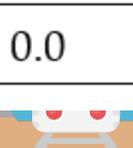
(d) I: false, II: true, III: false.



## Exercício

- Leia 2 valores com uma casa decimal (x e y), que devem representar as coordenadas de um ponto em um plano. A seguir, determine qual o quadrante ao qual pertence o ponto, ou se está sobre um dos eixos cartesianos ou na origem ( $x = y = 0$ ).
- Se o ponto estiver na origem, escreva a mensagem “Origem”. Se o ponto estiver sobre um dos eixos escreva “Eixo X” ou “Eixo Y”, conforme for a situação.
- Entrada A entrada contém as coordenadas de um ponto.
- Saída A saída deve apresentar o quadrante em que o ponto se encontra.

Entrada:	Saída:
4.5 -2.2	Q4
0.1 0.1	Q1
0.0 0.0	Origem



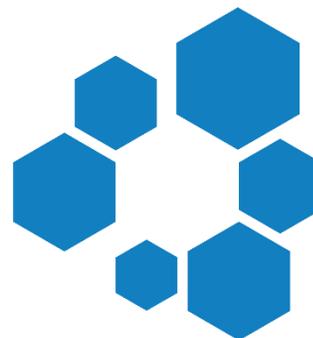
# Obrigado!

## Perguntas?

Marcos Roriz (marcosroriz@ufg.br)

**ENGENHARIA DE  
TRANSPORTES**

**FCT**  
FACULDADE DE  
CIÊNCIAS E TECNOLOGIA



**UFG**  
UNIVERSIDADE  
FEDERAL DE GOIÁS

Fonte dos ícones utilizados: flaticons.com



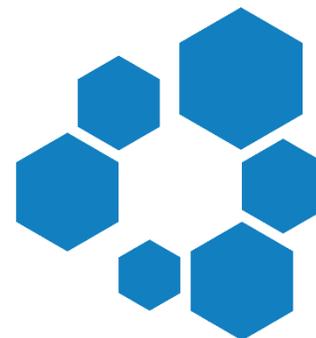
# Obrigado!

## Perguntas?

Marcos Roriz (marcosroriz@ufg.br)

**ENGENHARIA DE  
TRANSPORTES**

**FCT**  
FACULDADE DE  
CIÊNCIAS E TECNOLOGIA



**UFG**  
UNIVERSIDADE  
FEDERAL DE GOIÁS

