

# Linguagem C - Estruturas de Seleção

## PARTE 1

### Lógica de Programação

Professor: Vinícius T. Guimarães

[viniciusguimaraes@ifsul.edu.br](mailto:viniciusguimaraes@ifsul.edu.br)



Tecnólogo em Sistema para Internet  
Campus Charqueadas

# Estruturas de Seleção

- Até agora, os programas desenvolvidos seguiam o mesmo padrão: efetuava-se a leitura de dados, estes dados eram processados e algum resultado era mostrado na tela.

**Os programas pareciam uma calculadora!!**



# Estruturas de Seleção

- **O que iremos aprender agora?**

Vamos aprender os comandos na linguagem de programação C que nos permitem colocar condições em nossos programas.

Dessa forma, o nosso código poderá seguir por um caminho ou por outro, dependendo das condições que estabelecermos nele.

# Exemplo



- **Observe a imagem abaixo**

O que vai determinar se o homem vai pelo caminho **A** ou pelo caminho **B**?

**A** condição que ele estabelecer. Por exemplo, ele pode definir se vai por A ou B, dependendo da distância. Ele poderia utilizar também o tipo de calçamento de cada caminho. Enfim, ele vai estabelecer um critério e tomar uma decisão.

**Vamos aprender a como colocar decisões condicionais nos nossos códigos.** 😊



# Ponto de partida

- Para iniciarmos, vamos pensar no seguinte algoritmo de exemplo:

Elabore um programa que solicite ao usuário um número qualquer e imprima se o número informado é maior ou menor do que 100.

*Com o que aprendemos até o momento vamos iniciar fazendo o código que lê um número qualquer e mostra ele na tela.*

# Ponto de partida

- Abaixo temos a primeira parte do código que, simplesmente, lê o número e imprime ele na tela.

```
1  #include <stdio.h>
2
3  int main(void) {
4      int numero;
5
6      printf("Informe um número\n");
7      scanf("%i", &numero);
8      printf("Número = %i\n", numero);
9
10     return 0;
11 }
```

**E agora, como eu  
faço para saber se  
ele é maior ou  
menor que 100?**



<https://replit.com/@vicoguim/exemplo-selecao-1-1>

# Comando if ... else

- Vamos utilizar o comando **if...else** (se ... senão)

Trata-se de uma estrutura de controle de fluxo, executando um ou vários comandos se a condição testada for **verdadeira** e, **em alguns casos**, executando um ou vários comandos se for **falsa**.

```
if(condição){  
    comandos;  
}  
else {  
    comandos;  
}
```

# Detalhando if ... else



```
if(condição){  
    comandos;  
}  
else {  
    comandos;  
}
```

1

A palavra **if** é um comando da linguagem (chamamos de palavra reservada) e vem seguido de ( ) (parênteses). Dentro dos parênteses, vamos escrever a **condição** que desejamos testar.



# Detalhando if ... else

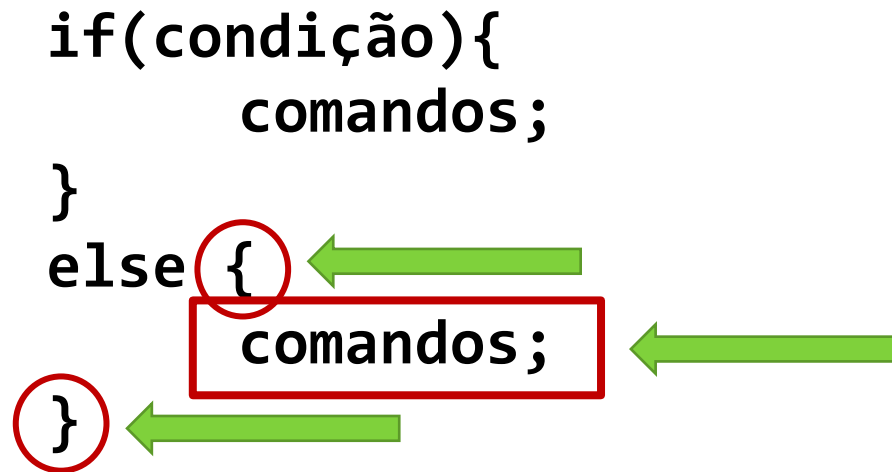
```
if(condição){  
    comandos;  
}  
else {  
    comandos;  
}
```

2

Se a condição colocada no **if** resultar em um valor **VERDADEIRO**, será executado o bloco de comandos delimitado entre as **{ }** (chaves). Observem que as chaves servem para delimitar o bloco de comandos, ou seja, definir o início e o fim do bloco de comandos.

# Detalhando if ... else

```
if(condição){  
    comandos;  
}  
else {  
    comandos;  
}
```



3

Se a condição colocada no **if** resultar em um valor **FALSO**, será executado o bloco de comandos do **else** que também está delimitado entre as **{ }** (chaves).

# Detalhando if ... else



```
if(condição){  
    comandos;  
}  
else {  
    comandos;  
}
```

**APENAS** o bloco de comandos associado ao **if**  
**OU** o código associado ao **else** será executado,  
**NUNCA** ambos.

# Detalhando if ... else



O comando **else** é **OPCIONAL**, ou seja, podemos ter o comando **if** sem o comando **else**, conforme o exemplo abaixo.

```
if(condição){  
    comandos;  
}
```



# Detalhando if ... else



O comando **else** sempre precisa de um **if**, ou seja, não podemos ter um **else** órfão (solto) no código.

```
else{  
    comandos;  
}
```



# Voltando ao nosso problema

E agora, como eu faço para saber se ele é maior ou menor que 100?



```
1  #include <stdio.h>
2
3  int main(void) {
4      int numero;
5
6      printf("Informe um número\n");
7      scanf("%i", &numero);
8
9      if(numero > 100){
10         printf("Número maior que 100!\n");
11     }
12     else{
13         printf("Número menor que 100!\n");
14     }
15
16     return 0;
17 }
```

## Solução 1

<https://replit.com/@vicoguim/exemplo-selecao-1-2>

Lógica de Programação

# Voltando ao nosso problema

E agora, como eu faço para saber se ele é maior ou menor que 100?



```
1  #include <stdio.h>
2
3  int main(void) {
4      int numero;
5
6      printf("Informe um número\n");
7      scanf("%i", &numero);
8
9      if(numero > 100){
10         printf("Número maior que 100!\n");
11     }
12     else{
13         printf("Número menor que 100!\n");
14     }
15
16     return 0;
17 }
```

Observem que colocamos a expressão relacional com o operador **>** (maior) para estabelecer nossa condição, ou seja, **verifica se o valor contido na variável `numero` é maior que 100.**

<https://replit.com/@vicoguim/exemplo-selecao-1-2>

# Voltando ao nosso problema



E agora, como eu faço para saber se ele é maior ou menor que 100?

```
1  #include <stdio.h>
2
3  int main(void) {
4      int numero;
5
6      printf("Informe um número\n");
7      scanf("%i", &numero);
8
9      if(numero > 100){
10         printf("Número maior que 100!\n");
11     }
12     else{
13         printf("Número menor que 100!\n");
14     }
15
16     return 0;
17 }
```

Caso o resultado do teste seja **VERDADEIRO**, será executado o bloco associado ao comando **if**, ou seja:

**printf("Número maior que 100!\n");**

<https://replit.com/@vicoguim/exemplo-selecao-1-2>



# Voltando ao nosso problema



E agora, como eu faço para saber se ele é maior ou menor que 100?

```
1  #include <stdio.h>
2
3  int main(void) {
4      int numero;
5
6      printf("Informe um número\n");
7      scanf("%i", &numero);
8
9      if(numero > 100){
10         printf("Número maior que 100!\n");
11     }
12     else{
13         printf("Número menor que 100!\n");
14     }
15
16     return 0;
17 }
```

Caso **CONTRÁRIO**, ou seja, o resultado do teste seja **FALSO** será executado o bloco associado ao comando **eles**, ou seja:

**printf("Número menor que 100!\n");**

<https://replit.com/@vicoguim/exemplo-selecao-1-2>

# Voltando ao nosso problema

## Uma outra solução (Solução 2) ...

```
1  #include <stdio.h>
2
3  int main(void) {
4      int numero;
5
6      printf("Informe um número\n");
7      scanf("%i", &numero);
8
9      if(numero > 100){
10         printf("Número maior que 100!\n");
11     }
12     if(numero < 100){
13         printf("Número menor que 100!\n");
14     }
15
16     return 0;
17 }
```

<https://replit.com/@vicoguim/exemplo-selecao-1-3>

# Voltando ao nosso problema

## Uma outra solução (Solução 2) ...

```
1  #include <stdio.h>
2
3  int main(void) {
4      int numero;
5
6      printf("Informe um número\n");
7      scanf("%i", &numero);
8
9      if(numero > 100){
10         printf("Número maior que 100!\n");
11     }
12     if(numero < 100){
13         printf("Número menor que 100!\n");
14     }
15
16     return 0;
17 }
```

Observem que, nessa solução, utilizamos dois comandos **if**, um para cada teste, e **NÃO** utilizamos o **else**.

<https://replit.com/@vicoguim/exemplo-selecao-1-3>

# Voltando ao nosso problema

## Uma outra solução (Solução 2) ...

```
1  #include <stdio.h>
2
3  int main(void) {
4      int numero;
5
6      printf("Informe um número\n");
7      scanf("%i", &numero);
8
9      if(numero > 100){
10         printf("Número maior que 100!\n");
11     }
12     if(numero < 100){
13         printf("Número menor que 100!\n");
14     }
15
16     return 0;
17 }
```



Pergunta para responder no Fórum do Moodle:

**Você consegue identificar alguma vantagem da Solução 2 em relação a Solução 1?**

<https://replit.com/@vicoguim/exemplo-selecao-1-3>

# Operadores Relacionais

- Na solução do nosso algoritmo, tivemos que montar um teste condicional e, para isso, utilizamos o **operador relacional >**.



Sempre que executamos um teste condicional temos dois resultados possíveis:

**VERDADEIRO ou FALSO.**

# Operadores Relacionais

- A partir de agora, iremos utilizar com muita frequência os operadores relacionais para elaborar as estruturas condicionais em nossos códigos. **Vamos ver quais são esses operadores em Linguagem C?**

BORALÁ

# Operadores Relacionais

- Tabela de operadores relacionais

Operador	Ação	Precedência
!	Negação (ou NOT)	Maior
>, >=, <, <=	Maior que, Maior ou igual que, Menor que, Menor ou igual que	↓
==, !=	Igual, Diferente	
&&	Condição “E” (ou AND)	
	Condição “OU” (ou OR)	Menor



- O = (um igual) é a **atribuição**. Para verificar a **igualdade** entre dois valores, vamos utilizar == (dois iguais).
- O >= verifica se um valor é **maior ou igual**, ou seja, vai retornar **VERDADEIRO** tanto se o valor for **maior** quanto se o valor for **igual**. O mesmo vale para o operador <=, com a diferença que vai verificar se o valor é **menor ou igual**.

# Estruturas de Seleção

- Em C qualquer valor diferente de 0 (zero) é tido com verdadeiro. O valor 0 (zero) é considerado falso. As expressões que utilizam operadores relacionais ou lógicos retornam 0 para falso e 1 para verdadeiro.

```
1  #include <stdio.h>
2
3  int main(void) {
4      int a;
5      printf("Informe o valor para a:\n");
6      scanf("%i", &a);
7      if(a){
8          printf("Entrou no IF!\n");
9      }
10     else{
11         printf("Entrou no ELSE!\n");
12     }
13     return 0;
14 }
```

<https://replit.com/@vicoguim/exemplo-selecao-1-4>



# Estruturas de Seleção

- Observe o código abaixo. Onde estão os delimitadores de bloco?

```
1  #include <stdio.h>
2
3  int main(void) {
4
5      int numero;
6
7      printf("Digite um numero: ");
8      scanf("%i", &numero);
9
10     if(numero > 20)
11         printf("Número maior que 20\n");
12
13     return 0;
14 }
```

**Está correto  
assim?**



**SIM!!**



<https://replit.com/@vicoguim/exemplo-selecao-1-5>

# Estruturas de Seleção

- O resultado é o mesmo?

```
1  #include <stdio.h>
2
3  int main(void) {
4
5  int  numero;
6
7  printf("Digite um numero: ");
8  scanf("%i", &numero);
9
10 if(numero > 20)
11 |   printf("Número maior que 20\n");
12 |
13 return 0;
14 }
```



```
1  #include <stdio.h>
2
3  int main(void) {
4
5  int  numero;
6
7  printf("Digite um numero: ");
8  scanf("%i", &numero);
9
10 if(numero > 20){
11 |   printf("Número maior que 20\n");
12 | }
13
14 return 0;
15 }
```

<https://replit.com/@vicoquim/exemplo-selecao-1-5>

<https://replit.com/@vicoquim/exemplo-selecao-1-6>

# Estruturas de Seleção



## A regra é clara!

**Se tivermos apenas um comando a ser executado no bloco do *if* ou do *else*, o uso dos delimitadores de bloco é facultativo.**

# Estruturas de Seleção

- O resultado é o mesmo?

```
1  #include <stdio.h>
2
3  int main(void) {
4
5      int numero;
6
7      printf("Digite um numero: ");
8      scanf("%i", &numero);
9
10     if(numero > 20)
11     |   printf("Número maior que 20\n");
12     |   printf("Outro comando\n");
13     |
14     return 0;
15 }
```



<https://replit.com/@vicoguim/exemplo-selecao-1-7>

```
1  #include <stdio.h>
2
3  int main(void) {
4
5      int numero;
6
7      printf("Digite um numero: ");
8      scanf("%i", &numero);
9
10     if(numero > 20){
11     |   printf("Número maior que 20\n");
12     |   printf("Outro comando\n");
13     |   }
14
15     return 0;
16 }
```

<https://replit.com/@vicoguim/exemplo-selecao-1-8>

# Estruturas de Seleção

- Faça o teste:

```
1  #include <stdio.h>
2
3  int main(void) {
4
5      int  numero;
6
7      printf("Digite um numero: ");
8      scanf("%i", &numero);
9
10     if(numero > 20)
11         printf("Número maior que 20\n");
12         printf("Outro comando\n");
13
14     return 0;
15 }
```

<https://replit.com/@vicoguim/exemplo-selecao-1-7>

Observe que, independente do valor informado para a variável **numero**, o **printf** da linha 12 sempre será executado. Isso acontece pois não foram colocados delimitadores de bloco para o comando **if**. Assim, o compilador entende que apenas o **printf** da linha 11 deve ser executado quando o teste do **if** resultar em um valor verdadeiro.

# Estruturas de Seleção

- Faça o teste:

```
1  #include <stdio.h>
2
3  int main(void) {
4
5  int numero;
6
7  printf("Digite um numero: ");
8  scanf("%i", &numero);
9
10 if(numero > 20){
11     printf("Número maior que 20\n");
12     printf("Outro comando\n");
13 }
14
15 return 0;
16 }
```

<https://replit.com/@vicoguim/exemplo-selecao-1-6>

Observe que a presença dos delimitadores de bloco altera o comportamento do código. Agora, o **printf** da linha 12 só será executado quando o teste do **if** resultar em um valor verdadeiro. Isso acontece pois os delimitadores deixam explícito o bloco de comando que pertence ao comando **if**.

# Estruturas de Seleção



**Dica do He-man**

**SEMPRE UTILIZAR OS DELIMITADORES DE  
BLOCO, EVITANDO COMPORTAMENTOS  
INDESEJADOS!!!**