

Aula 05 – Testes e Condições

# Algoritmos e Lógica de Programação

# Testes e Condições

Até agora nossos programas foram feitos de forma bem linear, ou seja, todas as instruções foram executadas na ordem que foram programadas.

Se fossemos criar um algoritmo para que uma pessoa se preparasse para seu dia de trabalho, poderíamos fazer o seguinte:

- Vestir camiseta
- Vestir casaco
- Vestir calças
- Calçar Sapatos

# Testes e Condições

No entanto isto não representaria a realidade. Geralmente decidimos o que vamos vestir baseados no clima.

O algoritmo anterior seria ideal se fosse inverno, mas se fosse verão não usaríamos casaco nenhum.

Qual seria a solução?

Se deixarmos como está seria ideal para o inverno.

Se tirarmos o “Vestir Casaco” ficaria ideal para o verão.

A melhor solução seria manter “Vestir Casaco” no programa, mas só executar no inverno.

# Testes e Condições

Veja como ficou:

- Vestir camiseta
- Se estiver frio, vestir casaco
- Vestir calças
- Calçar Sapatos

Agora o algoritmo ficou adaptado a qualquer situação de clima e estação do ano.

# Valores Lógicos – Verdadeiro ou Falso

Em C não existe um tipo de dado específico para armazenar um valor lógico.

Usamos um número inteiro para representar um valor lógico seguindo esta regra:

- 0 (zero) é falso
- Qualquer outro valor é verdadeiro.



# Operadores Relacionais

Operador	Nome	Exemplo	Significado
==	Igualdade	$a == b$	a é igual a b?
>	Maior que	$a > b$	a é maior que b?
>=	Maior ou igual	$a >= b$	a é maior ou igual a b?
<	Menor que	$a < b$	a é menor que b?
<=	Menor ou igual	$a <= b$	a é menor ou igual a b?
!=	Diferente	$a != b$	a é diferente de b?

# if-else

O primeiro comando de decisão que vamos estudar é o if-else:

```
if(condição)
    instrução;
else
    instrução;
```

Nota: na instrução if, o else é opcional.

# if-else

Exemplo:

```
#include <stdio.h>
main()
{
    int numero;
    printf("Digite um numero: ");
    scanf("%d", &numero);
    if(numero > 0)
        printf("numero positivo");
    else
        printf("numero negativo");
}
```



# Bloco de Instruções

O programa que vimos anteriormente mostra uma só instrução no if ou no else.

O que devemos fazer se precisarmos de mais de um comando?

Devemos utilizar { } para criar um bloco de código:

```
if(a>=b)
{
    printf("entre com um numero menor para a: ");
    scanf("%d", &a);
}
```

# Bloco de Instruções

O programa a seguir mostra dois números digitado em ordem crescente:

```
#include <stdio.h>

main()
{
    int n1, n2, aux;
    printf("Digite um numero: ");
    scanf("%d", &n1);
    printf("Digite outro numero: ");
    scanf("%d", &n2);
```

```
    if(n1 > n2)
    {
        aux = n1;
        n1 = n2;
        n2 = aux;
    }
    printf("%d -> %d", n1, n2);
}
```

# If-else encadeadas

```
#include <stdio.h>

main()
{
    float sal;
    printf("Qual o salario? ");
    scanf("%f", &sal);
    if(sal <= 0)
        printf("Salario invalido!");
```

```
    else
        if(sal > 1000)
            printf("imposto: %.2f", sal * 0.1);
        else
            printf("imposto: %.2f", sal * 0.05);
}
```

# Operadores Lógicos

Operador	Exemplo	Significado
&&	<code>a &gt; b &amp;&amp; a != 0</code>	E (AND) lógico
	<code>a &gt; b    a != 0</code>	OU (OR) lógico
!	<code>! a</code>	NÃO (NOT) lógico

# Exercícios

1. Faça um programa que leia um inteiro e diga se ele é maior que zero ou não.
2. Faça um programa que receba o mês e diga quantos dias ele possui (fevereiro = 28).
3. Faça um programa que verifica se uma data é válida ou não.
4. Faça um programa que verifica se um número é válido para a seguinte regra: ele deve ser maior que zero e menor que 100.
5. Faça um programa que receba um número e verifique se ele é par ou impar.

# Exercícios

6. Faça um programa que verifique se é possível calcular sua raiz quadrada.
7. Faça um programa que leia dois números, faça a soma deles e mostre o resultado somente se ele for maior que 10.
8. Faça um programa que leia dois números, faça a soma deles e implemente a seguinte regra: se a soma for maior que 20 adicione mais 8; se a soma for menor que 20 subtraia 5
9. Entrar com um número e informar se ele é divisível por 3 e por 7.
10. Faça um programa que indique se um número digitado está entre 20 e 90 ou não.