Linguagem de Programação
Revisão: Comandos condicionais e Repetitivos

Alexandre Mello

Fatec Campinas

2023

Roteiro

Comandos condicionais

```
if (cond1) {
  if (cond2)
    comando1;
} else
  comando2;
```

Quando o **comando2** é executado?

Comandos condicionais

```
if (cond1) {
  if (cond2)
    comando1;
} else
  comando2;
```

Quando o **comando2** é executado?
Resposta: quando cond1 for falsa.

Comandos condicionais

```
if (cond1){
  if (cond2)
    comando1;
  else
    comando2;
}else{
  if(cond3)
     comando3;
  else
     comando4;
}
```

Quando o **comando4** é executado?

Comandos condicionais

```
if (cond1){
  if (cond2)
    comando1;
  else
    comando2;
}else{
  if(cond3)
     comando3;
  else
     comando4;
}
```

Quando o **comando4** é executado?
Resposta: quando a **cond1** for falsa e **cond3** for falsa.

Comandos condicionais

Use chaves e indentação para deixar claro a qual comando condicional um outro comando pertence!!

```
if (cond1)
if (cond2)
   comando1;
else
   comando2;
```

Quando o **comando2** é executado?

Comandos condicionais

Use chaves e indentação para deixar claro a qual comando condicional um outro comando pertence!!

```
if (cond1)
if (cond2)
    comando1;
else
    comando2;
```

Quando o **comando2** é executado?
Resposta: O comando **if-else** é um único comando, portanto ele está dentro do primeiro **if**. Logo comando2 é executado quando cond1 for verdadeira e cond2 falsa.

Comandos condicionais

Usando chaves e indentação para deixar mais claro:

```
if (cond1){
  if (cond2)
    comando1;
  else
    comando2;
}
```

Comandos condicionais

```
int main(){
    int a = 5;

    if(a > 3){
        if(a < 7)
            printf("a");
    }else{
        if(a>-10)
            printf("b");
        else
            printf("c");
    }
}
```

O que será impresso?

Comandos condicionais

```
int main(){
    int a;
    a = -12;
    if(a > 3){
        if(a < 7)
            printf("a");
    }else{
        if(a>-10)
          printf("b");
        else
          printf("c");
    }
}
```

O que será impresso?

Comandos condicionais

```
int main(){
    int a;
    a = 9;
    if(a > 3){
        if(a < 7)
            printf("a");
    }else{
        if(a>-10)
          printf("b");
        else
          printf("c");
    }
}
```

O que será impresso?

Comandos condicionais

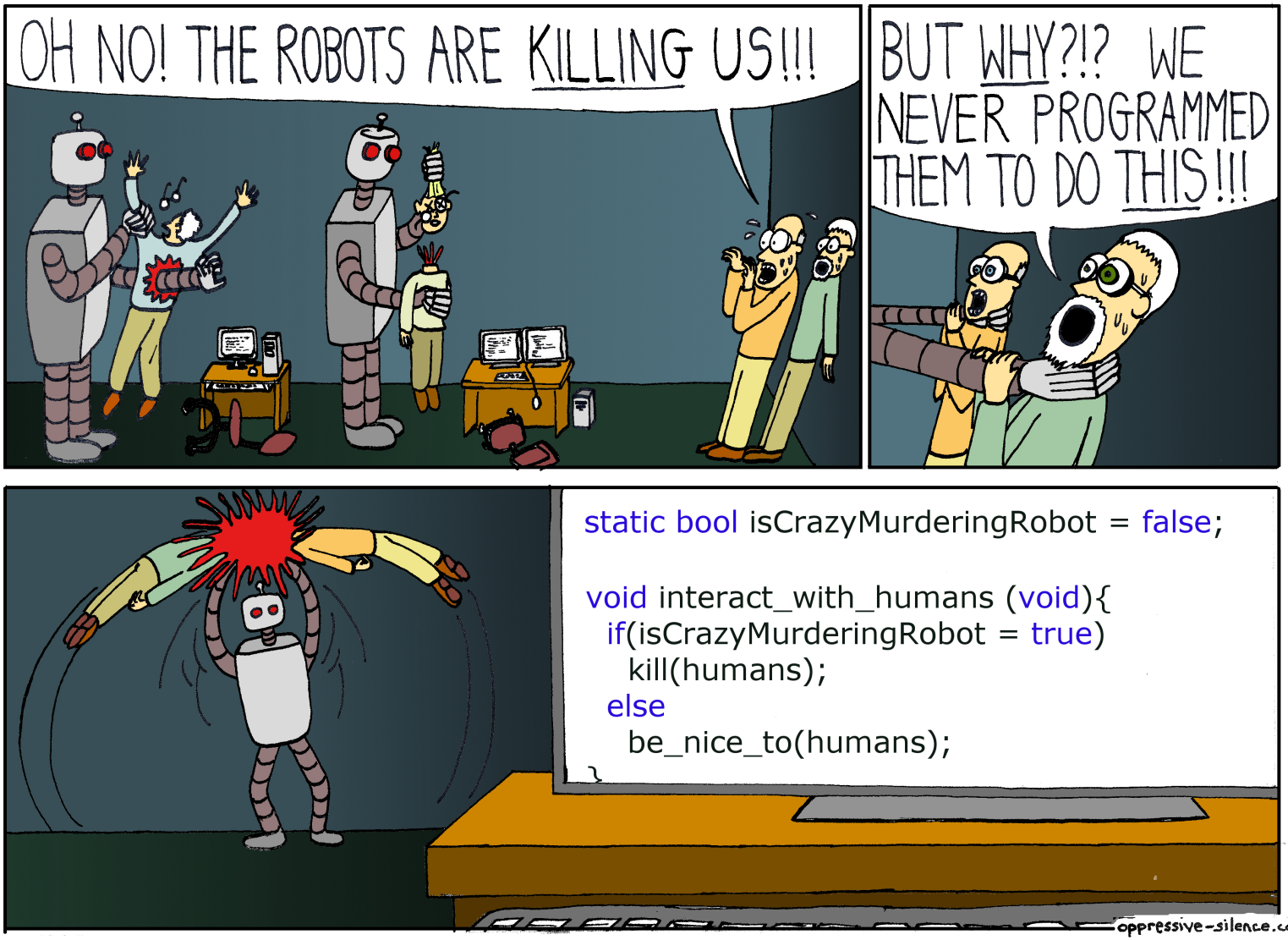
- Em C o comando de atribuição é = enquanto que o teste de igualdade é ==.
- Não os confunda, pois isto pode gerar erros!

```
#include <stdio.h>

int main(){
  int a = 2;
  if(a = 3){
   printf("fazer algo se a for 3\n");
  }else{
   printf("fazer algo se a não for 3\n");
  }
}
```

O programa acima imprime "fazer algo se a for 3", pois na expressão relacional dentro do comando **if**, temos uma atribuição, que sempre é verdadeiro.

Comandos condicionais



Exercícios

A solução abaixo está correta para classificar um número como par e menor que 100, ou par e maior ou igual a 100, etc, como no exemplo visto anteriormente?

```
#include <stdio.h>

int main(){
  int a;
  printf("Digite um número inteiro:");
  scanf("%d", &a);
  if( ( a % 2 == 0) && (a<100) )
      printf("O número é par e menor que 100\n");
  else if( a>=100 )
      printf("O número é par e maior ou igual a 100\n");

  if( ( a % 2 != 0) && (a<100) )
      printf("O número é ímpar e menor que 100\n");
  else if (a>=100)
      printf("O número é ímpar e maior que 100\n");
}
```

Exercícios

- Escreva um programa que lê um número inteiro do teclado e imprime "SIM"se o número for par e maior do que 10, ou for ímpar e menor do que 50. Caso contrário o programa deve imprimir "NAO".

Exercícios

- Escreva um programa lê três números e imprime o maior deles.

Exercícios

- Escreva um programa lê três números e os imprime em ordem (ordem crescente).

Comando **while**

- Estrutura:

```
while ( condição )
    comando;
```

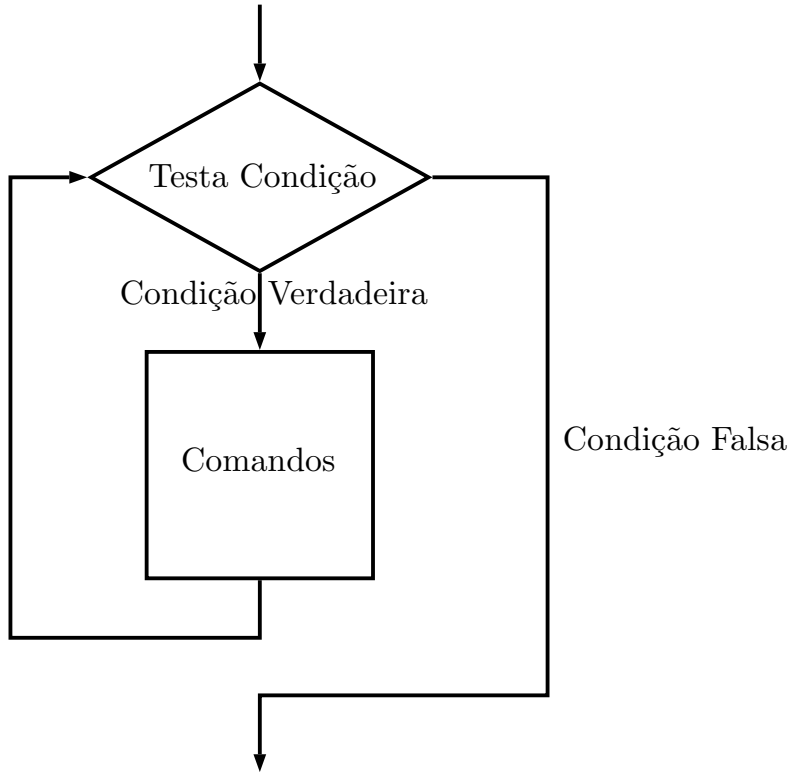
- Ou:

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

- Enquanto a condição for verdadeira (!=0), ele executa o(s) comando(s).

Comando **while**

- Passo 1: Testa a condição. Se a condição for verdadeira vai para o Passo 2.
- Passo 2.1: Executa os comandos.
- Passo 2.2: Volta para o Passo 1.



Comando **while**

Imprimindo os 100 primeiros números inteiros:

```
int i=1;
while (i<=100)
{
   printf("%d ",i);
   i++;
}
```

Comando **while**

Imprimindo os *n* primeiros números inteiros:

```
int i=1,n;
scanf("%d",&n);
while (i<=n)
{
   printf("%d ",i);
   i++;
}
```

Comando **while**

- 1. O que acontece se a condição for falsa na primeira vez?

```
while (a!=a)
    a=a+1;
```

- 2. O que acontece se a condição for sempre verdadeira?

```
while (a == a)
    a=a+1;
```

Comando **while**

- 1. O que acontece se a condição for falsa na primeira vez?

```
while (a!=a)
    a=a+1;
```

Resposta: Ele nunca entra na repetição (no laço).

- 2. O que acontece se a condição for sempre verdadeira?

```
while (a == a)
    a=a+1;
```

Resposta: Ele entra na repetição e nunca sai (laço infinito).

Comando **do-while**

- Estrutura:

```
do
 comando;
while ( condição );
```

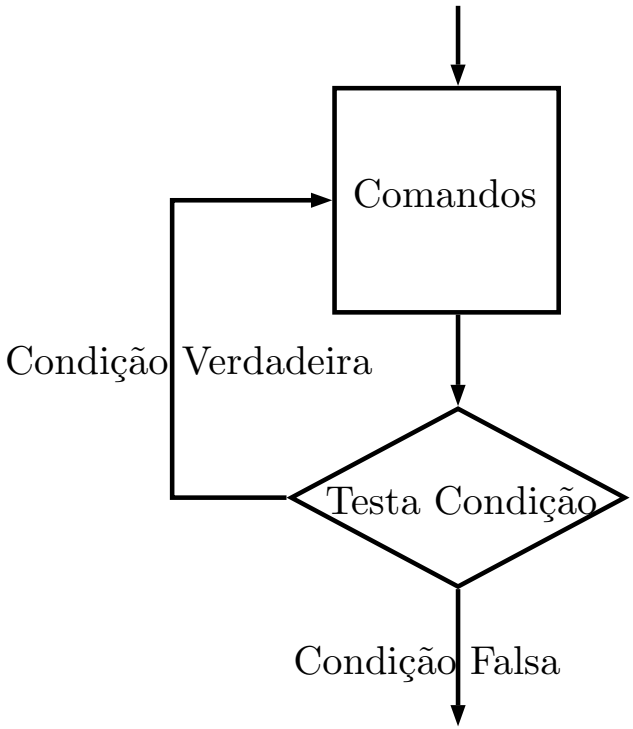
- Ou:

```
do{
 comandos;
}while ( condição );
```

- Diferença do **while**: sempre executa comandos na primeira vez. Teste condicional é feito por último.

Comando **do-while**

- Passo 1: Executa comandos.
- Passo 2: Testa condição. Se for verdadeira vai para Passo 1.



Comando **do-while**

Imprimindo os 100 primeiros números inteiros:

```
int i;
i=1;
do{
  printf("\n %d",i);
  i = i+1;
}while(i<= 100);
```

Comando **do-while**

Imprimindo os *n* primeiros números inteiros:

```
int i, n;
i=1;
scanf("%d",&n);
do{
  printf("\n %d",i);
  i++;
}while(i<=n);
```

- O que acontece se o usuário digitar 0?
- O que acontece se usarmos o comando **while**?

O comando **for**

- Estrutura:

```
for (início ; condição ; passo)
      comando do bloco;
```

- Ou:

```
for (início ; condição ; passo) {
   comandos do bloco;
 }
```

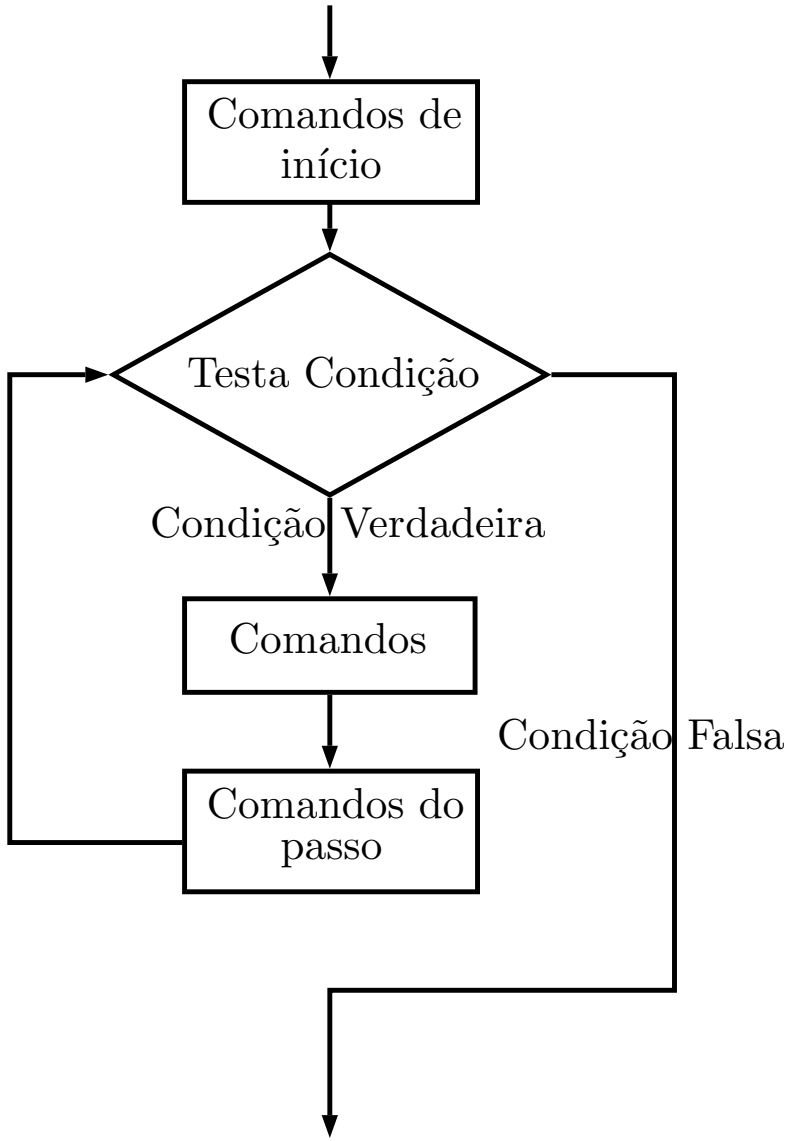
- **Início**: Uma ou mais atribuições, separadas por “,”.
- **Condição**: Comandos são executados enquanto a condição for verdadeira.
- **Passo**: Um ou mais comandos separados por “,”. Os comandos do passo são executados após os comandos do bloco.

O comando **for**

```
for (início ; condição ; passo) {
   comandos do bloco;
 }
```

- Passo 1: Executa comandos em "início".
- Passo 2: Testa condição. Se for verdadeira vai para passo 3.
- Passo 3.1: Executa comandos do bloco.
- Passo 3.2: Executa comandos em "passo".
- Passo 3.2: Volta ao Passo 2.

O comando **for**



O Comando **for**

O **for** é equivalente a seguinte construção utilizando o **while**:

```
início;
while(condição){
        comandos;
        passo;
}
```

O Comando **for**

Imprimindo os 100 primeiros números inteiros:

```
int i;
for(i=1; i<= 100; i=i+1){
  printf("\n %d",i);
}
```

O Comando **for**

Imprimindo os *n* primeiros números inteiros:

```
int i, n;
scanf("%d",&n);
for(i=1; i<=n; i++){
  printf("\n %d",i);
}
```

Exercício

- Faça um programa que imprima um menu de 4 pratos na tela e uma quinta opção para sair do programa. O programa deve imprimir o prato solicitado. O programa deve terminar quando for escolhido a quinta opção.

Exercício

- Faça um programa que lê dois números inteiros positivos a e b. Utilizando laços, o seu programa deve calcular e imprimir o valor a^b.

Exercício

- Faça um programa que lê um número n e que computa e imprima o valor

$$\sum_{i=1}^{n} i.$$

  OBS: Não use fórmulas como a da soma de uma P.A.

Exercício

- Faça um programa que lê um número n e imprima os valores entre 2 e n que são divisores de n.

Exercício

- Faça um programa que lê um número n e imprima os valores

$$\sum_{i=1}^{j} i$$

para j de 1 até n, um valor por linha.