UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Estruturas de repetição

QXD0001 - Fundamentos de Programação

Prof. Bruno Góis Mateus (brunomateus@ufc.br)

Agenda

- Introdução
- Laços contados
- Laços condicionais
- Erros mais comuns

Introdução

Introdução

- Escreva um programa para imprimir todos os números de 1 até 5
- Escreva um programa para imprimir todos os números de 1 até 10
- Escreva um programa para imprimir todos os números de 1 até 100


```
fmt.Println(1)
fmt.Println(2)
fmt.Println(3)
fmt.Println(4)
fmt.Println(5)
fmt.Println(6)
fmt.Println(7)
fmt.Println(8)
fmt.Println(9)
fmt.Println(10)
…
fmt.Println(100)
```

Introdução

O Conceito de Iteração

- O que é:
  - A execução repetida de um bloco de código até que uma condição de parada seja atingida
- A analogia do Atleta:
  - Um corredor em uma pista de atletismo
  - Ele corre uma volta (iteração), verifica se completou o treino (condição) e, se não, corre a próxima
- Por que usar:
  - Eficiência. O esforço para o programador escrever um laço que roda 10 vezes ou 10 milhões de vezes é exatamente o mesmo

Introdução

Laços

- São muito comuns situações em que se deseja repetir um determinado trecho de código várias vezes
- Para isso usamos estruturas de repetição
  - Muitas vezes chamados de laços ou loops
  - O corpo dessa estrutura são os comandos executados repetidas vezes
- Podemos ter dois tipos de laços: contados e condicionais

I will lock my computer when I leave my desk.

```
#include <stdio.h>

int main()
{

   for(int x=0; x < 500; x++) {
   printf("Never more will not to be use programming language for my home work!");
   }

   return 0;
}
```

Laços contados

Laços contados

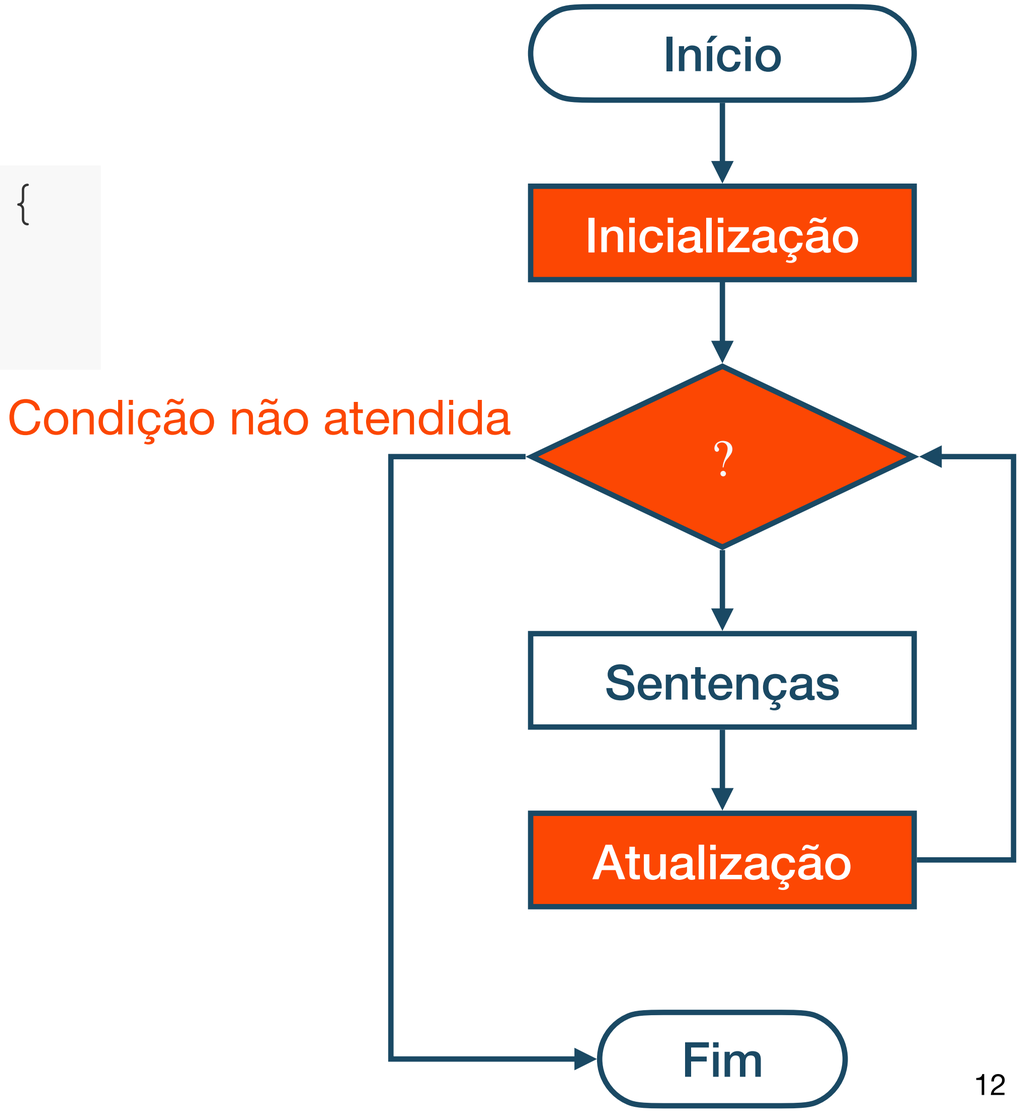
O que são?

- Também conhecidos como laços definidos
- Temos conhecimento prévio de quantas vezes o corpo deve ser executado
  - Lista de tarefas (quantidade)
  - Conjunto finito

Faça isso N vezes

Laços contados

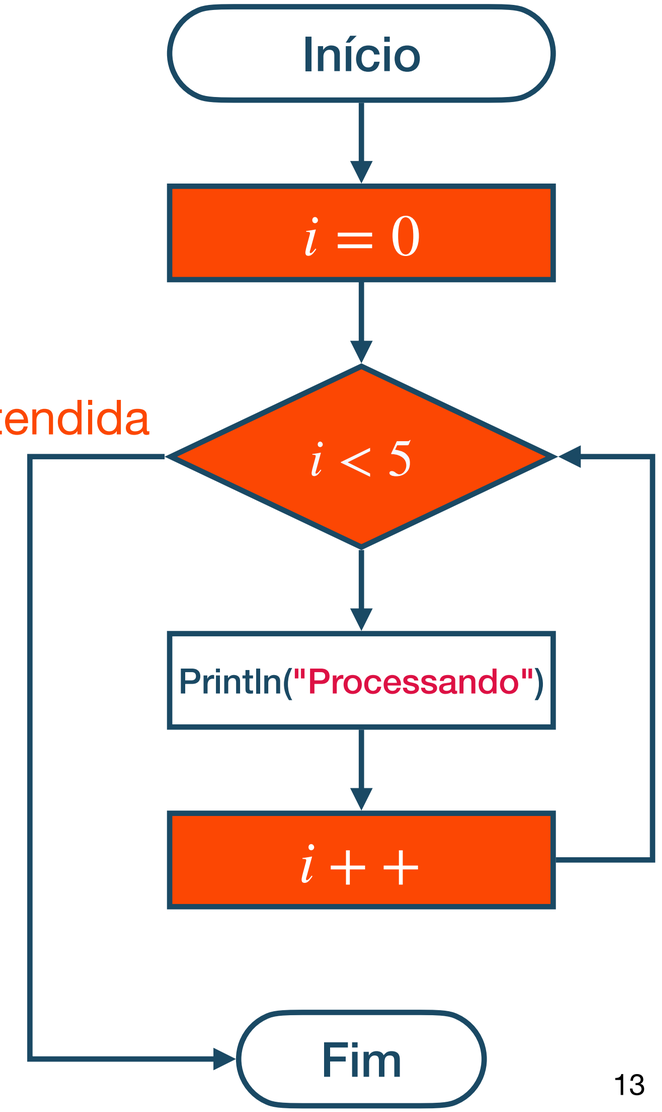
```
for inicialização; condição; incremento {
    // Sentenças
}
```



Laços contados

```
for i := 0; i < 5; i++ {
    fmt.Println("Processando item", i)
}
```

- Os três componentes:
- Inicialização: i := 0 (cria o contador)
- Condição: i < 5 (enquanto for verdade, o laço continua)
- Pós-execução: i++ (incrementa o contador a cada volta)



Laços condicionais

Laços condicionais

O que são?

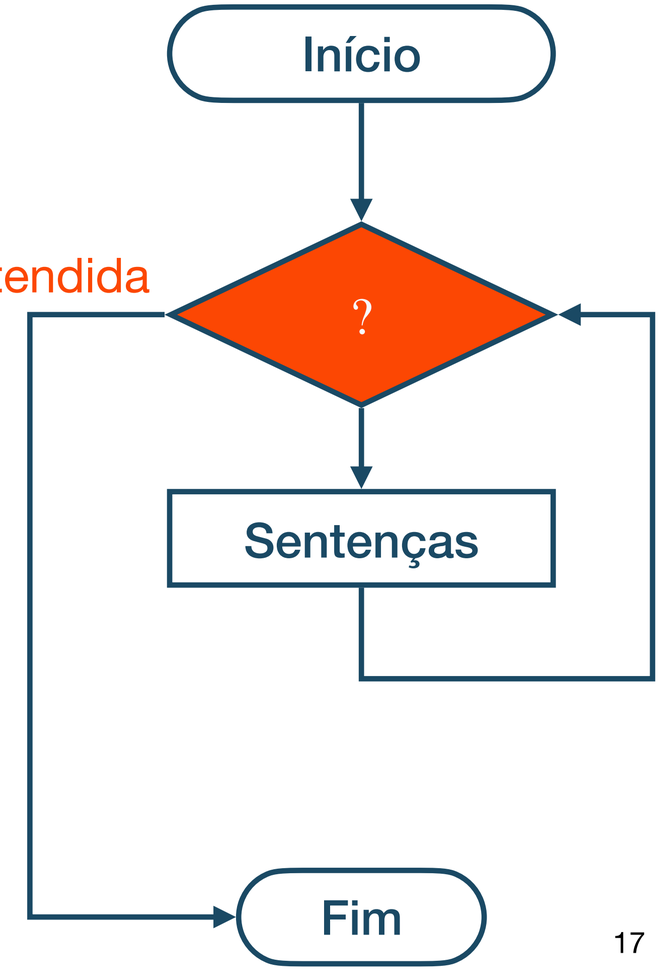
- Também conhecidos com laços indefinidos
- Não se conhece previamente o número necessário de repetições
- A repetição é realizada enquanto uma condição for satisfeita

Enquanto a condição for verdade, faça

Laços contados

```
for condição {
    // Sentenças
}
```

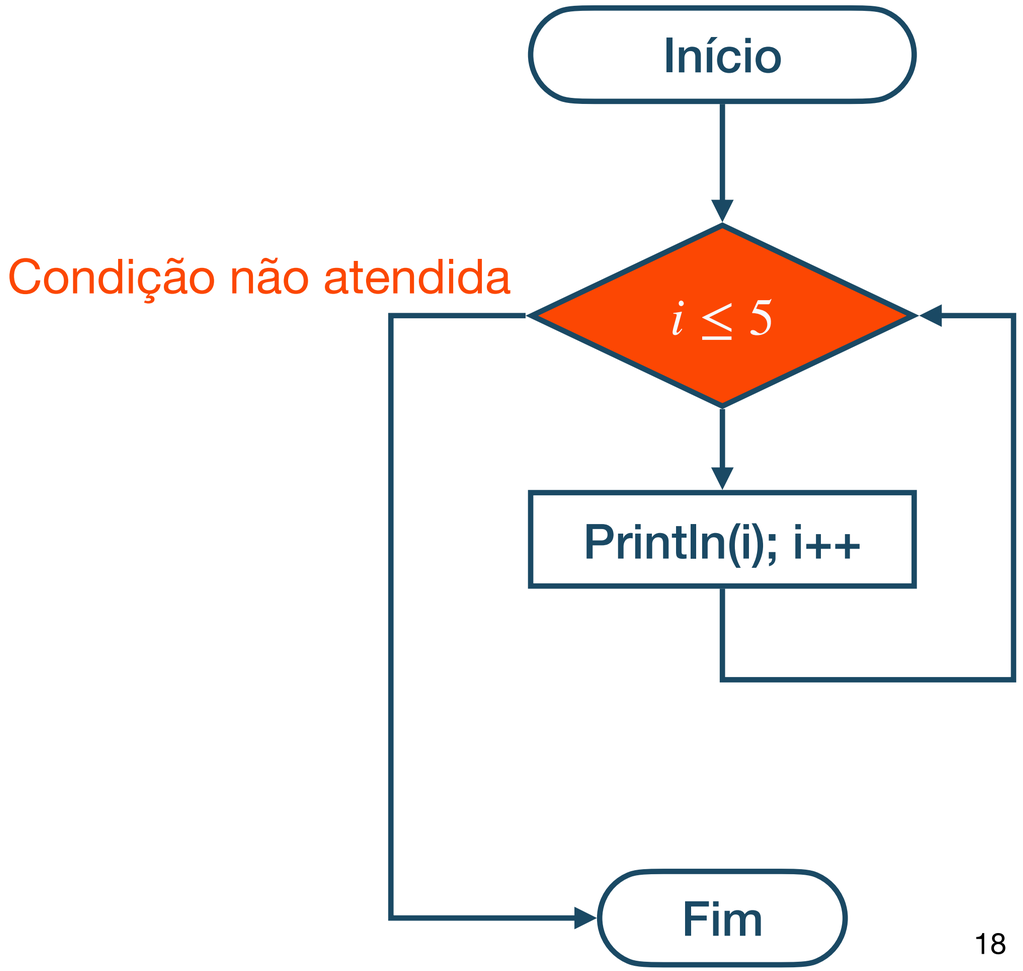
- Condição é verificada antes do bloco
- Executa os comandos enquanto a condição é verdadeira



Laços contados

```
i := 1

for i <= 5 {
    fmt.Println(i)
    i++
}
```



Padrões

Padrões

- Algumas situações costumam aparecer bastante quando usamos laços
  - Contagem
  - Acumulação
  - Contagem com condição

Padrões

Contagem

```
count := 0

for i := 5; i <= 10; i++ {
    fmt.Println(i)
    count++
}
```

Conta o número de vezes que o laço foi executado

Padrões

Acumulador

```
soma := 0

for i := 5; i <= 10; i++ {
    soma += i
}
```

Acumula a soma. Neste caso soma dos valores de i

Padrões

Contagem com condição

```
pares := 0

for i := 1; i <= 10; i++ {
    if i % 2 == 0 {
        pares++
    }
}
```

Conta somente quando os números são pares

Perceba que em todos os cenários, a variável de foi declarada fora do laço

Erros comuns

Erros comuns

❌ Erro 1 — Loop infinito 1 - esquecer o incremento

```
i := 1

for i <= 5 {
    fmt.Println(i)
}
```

Erros comuns

❌ Erro 2— Loop infinito 2 - condição sempre verdadeira

```
for i := 1; i <= 5; i-- {
    fmt.Println(i)
}
```

Erros comuns

❌ Erro 3— Loop não executado - Condição nunca é atendida

```
for i := 1; i >= 5; i++ {
    fmt.Println(i)
}
```

Erros comuns

❌ Erro 4 — Por um

```
for i := 1; i < 5; i++ {
    fmt.Println(i)
}
```

Não imprime o número 5

Referências

- Go 101