



Introdução à Engenharia

ENG1000

Aula 08 – Estruturas de Repetição

2016.1



Prof. Augusto Baffa
<abaffa@inf.puc-rio.br>



Estruturas de Repetição

- Diversos problemas somente podem ser resolvidos numericamente por um computador se o resultado de pequenas computações forem acumulados.
 - **Exemplo:** calcular o fatorial de um número.
- Precisamos de mecanismos que nos permitam requisitar que um conjunto de instruções seja repetidamente executado, até que uma determinada condição seja alcançada.
- **Repetições são programadas através da construção de laços (ou ciclos).**

Estruturas de Repetição (WHILE)

- **Estruturas de repetição** são utilizadas para indicar que um determinado conjunto de instruções deve ser executado um número definido ou indefinido de vezes, ou enquanto uma condição não for satisfeita.
- Em Lua, uma das formas de se trabalhar com repetições é através do comando **while**:

```
...  
while expressão_lógica do  
    -- Bloco de comandos  
end  
...
```

Enquanto a “**expressão_lógica**” for verdadeira, o “bloco de comandos” é executado!

Depois, a execução procede nos comandos subsequentes ao bloco while.

Estruturas de Repetição – Exemplo 1

- **Exemplo 1:**

“Crie um programa que escreva na tela todos os números entre 0 e 100”

```
local x = 0

while x <= 100 do
  io.write(x, "\n")
  x = x + 1
end
```

Estruturas de Repetição – Exemplo 2

- **Exemplo 2:**

“Fatorial de um número não-negativo”

$$n! = \prod_{i=1}^n i = n \times (n - 1) \times (n - 2) \times \cdots \times 3 \times 2 \times 1$$

```
function fatorial(n)
    local f = 1

    while n > 1 do
        f = f * n
        n = n - 1
    end

    return f
end
```

Estruturas de Repetição – Exemplo 3

- **Exemplo 3:**

“Verificar se um Número é Primo”

Sabe-se que: Um número é dito primo se ele for divisível apenas pelo número 1 e pelo próprio número, sendo que 1 não é primo (2 é o primeiro número primo).

```
function primo(n)

    local i
    if n < 2 then
        return 0
    end
    i = 2
    while i < n do

        if n % i == 0 then
            return 0
            i = i + 1
        end
    end
    return 1
end
-- retorna 0 se n não for primo, 1 se for
```

Estruturas de Repetição (FOR)

- Outra forma de se trabalhar com repetições é através do comando **for** – que é equivalente ao comando **while** com uma sintaxe mais compacta:

```
...  
for var = valor_inicial, valor_final, incremento do  
  -- Bloco de comandos  
end  
...
```

- O bloco de comandos será executado para cada valor de var partindo de valor_inicial e indo até valor_final, usando o incremento para incrementar o valor de var

Estruturas de Repetição – Exemplo 1

- **Exemplo 1:** “Escrever na tela os números entre 0 e 100”

```
local x = 0

while x <= 100 do
  io.write(x, "\n")
  x = x + 1
end
```



```
for x = 0, 100, 1 do
  io.write(x, "\n")
end
```

- **Importante:**
 - Na estrutura for a variável de controle x é uma variável local;
 - Nunca modifique o valor da variável de controle dentro da estrutura for;

Estruturas de Repetição – Exemplo 2

- **Exemplo 2:**

“Fatorial de um número não-negativo”

$$n! = \prod_{i=1}^n i = n \times (n-1) \times (n-2) \times \cdots \times 3 \times 2 \times 1$$

```
function fatorial(n)

    local f = 1

    for i = n, 1, -1 do
        f = f * i
    end

    return f
end
```

Estruturas de Repetição – Exemplo 3

- **Exemplo 3:**

“Verificar se um Número é Primo”

```
function primo(n)
  local i
  if n < 2 then
    return 0
  end
  for i = 2, n - 1, 1 do
    if n % i == 0 then
      return 0
    end
  end
  return 1
end
```

Estruturas de Repetição (repeat)

- A estrutura **while** avalia a expressão booleana que controla a execução do bloco de comandos no **início do laço**.
- A linguagem Lua oferece uma terceira construção de laços através do comando **repeat**:
 - A expressão booleana é avaliada no final do laço.
 - Isso significa que o bloco de comandos é executado pelo menos uma vez.

```
...  
repeat  
    -- Bloco de comandos  
until expressão_lógica  
...
```

Exercícios

Lista de Exercícios 04 - Estruturas de Repetição

<http://www.inf.puc-rio.br/~abaffa/eng1000/>